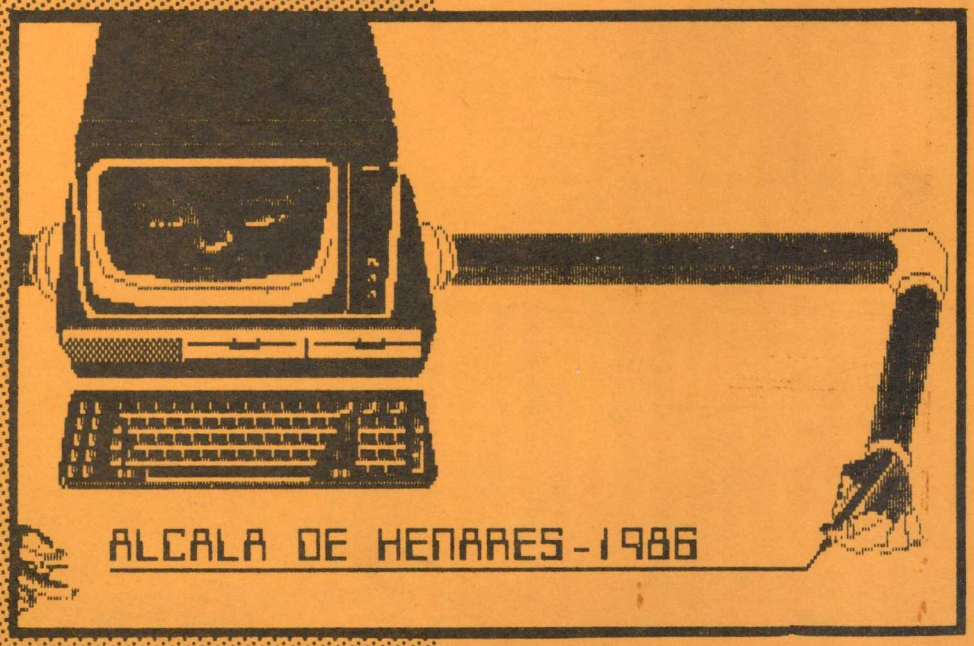
ite

instituto de tecnicas educativas

PROYECTO ATENEA

INICIACION A LA INFORMATICA PARA DOCENTES



ALCALA DE HENARES-1986

Ministerio de Educacion y Ciencia
Subdireccion General de
Perfeccionamiento del Profesorado

56459

56459

ite

instituto de tecnicas educativas

PROYECTO ATENEA

INICIACION A LA INFORMATICA PARA DOCENTES



ALCALA DE HENARES-1986

R-105800

BIBLIOMEC 056381

Ministerio de Educacion y Ciencia
Subdireccion General de
Perfeccionamiento del Profesorado

MINISTERIO DE EDUCACION Y CIENCIA - BIBLIOTECA

REDACCION Y ADMINISTRACION:
Ministerio de Educación y Ciencia
Dirección General de Enseñanzas Medias — Subdireción General de C. E. I.
Instituto de Técnicas Educativas
(Director: Esteban Orive Castro)
Alcalá de Henares (Madrid)
Teléfonos: 889 18 50 y 889 18 54

IMPRIME:
T. P. Artes Gráficas.
Cra. de Meco, Km. 5 - Alcalá de Henares
Depósito legal: M-16274/1986
© I. T. E. — Autores

AUTORES:

D. Victor Alonso Domínguez
D. Fernando Arias Fernández-Perez
D. Adolfo Blond Arredondo
D.ª M.ª Pilar Bueno Maroto
D. Esteban Cueva Alvarez
D. Manuel-Santiago Fernández Prieto
D. Antonio Enrique Gallego Palomero
D. José Luis Ganuza Fernández
D. Francisco Herrero Ruiz
D. Javier Ibáñez de Opacua Molina
D.ª Concepción López Sutil
D.ª Piedad Lozoya Ripoll
D. Juan Madrigal Muga
D. Francisco Martín Casalderrey
D.ª Dolores Martín Catalán
D. José Martín-Maestro Pérez
D. José Maximino García González
D. Daniel Meziat Luna
D.ª M.ª José Montero Corominas
D. Esteban Orive Castro
D. Luis Rodríguez-Roselló Martínez
D. Pablo Sanz García
D.ª Elena Veiguela Martínez

COORDINADORES:

D. Luis Rodríguez-Roselló Martínez
D.ª Concepción López Sutil

INDICE

ININ	37	Curso de iniciación a la informática para docentes. (6 páginas).
ININ	10	Objetivos pedagógicos de la informática en la educación. (34 páginas)
ININ	36	Introducción a los ordenadores. (21 páginas)
ININ	12	Posibilidades de la informática en la enseñanza. (7 páginas)
ININ	27	Enseñanza asistida por ordenador. (19 páginas)
EAO	1	Revisión y evaluación de programas de ordenador para la enseñanza. Formato de revisión y evaluación. (11 páginas)
EAO	2	Ejemplos de programas de E.A.O. Documentación anexa al disco. (19 páginas)
IA	1	Inteligencia artificial. (7 páginas)
IA	2	Bibliografía de I.A. (1 páginas)
SOP	30	Sistema operativo MS-DOS. (45 páginas)
SOP	32	Prácticas de sistema operativo MS-DOS. (4 páginas)
SOP	33	Anexo:Sistema operativo MS-DOS. (6 páginas)
PTEX	18	El procesador de textos. (13 páginas)
PTEX	44	Introducción al procesador de textos del Open Access. (2 páginas)
PTEX	45	Comandos del procesador de textos del Open Access. (2 páginas)
PTEX	46	Guia de operación. (13 páginas)
PTEX	47	Apéndices. (12 páginas)

PTEX 19 Posibilidades educativas de los procesadores de textos. (10 páginas)

PTEX 31 Modelo de aplicación: Redacción. (4 páginas)

PTEX 24 Aplicación didáctica. Lengua Española: Textos en prosa. (8 páginas)

PTEX 35 Aplicación didáctica. Lengua Española: Poesía. (10 páginas)

PTEX 54 Prácticas de procesador de textos 1. (1 página)

PTEX 55 Prácticas procesadores de textos 2. (2 páginas)

PTEX 56 Prácticas procesadores de textos 3. (1 página)

PTEX 57 Prácticas procesadores de textos 4. (1 página)

PTEX 58 Prácticas procesadores de textos 5. (2 páginas

b/PTEX 33 Bibliografía de procesadores de textos. (1 páginas)

BDAT 36 Hacia una sociedad de la información. (5 páginas)

BDAT 35 Terminología básica y estructuras de las bases de datos. (8 páginas)

BDAT 31 Introducción a la base de datos del Open Access. (3 páginas)

BDAT 32 Comandos de la base de datos del Open Access. (3 páginas)

BDAT 30 Guía de operación. (20 páginas)

BDAT 33 Apéndices. (6 páginas)

BDAT 34 Prácticas de la base de datos del Open Access. (3 páginas)

BDAT 25 Las bases de datos en las aulas. (15 páginas)

BDAT 23 Aplicación didáctica: Geografía de la población. (38 páginas)

BDAT 27 Bases de datos y gráficos en el Open Access. (15 páginas)

BDAT 38 Aplicación de base de datos: Tabla de alimentos. (9 páginas)

BDAT 39 Ejercicios. (1 páginas)

BDAT 37 Bibliografía de bases de datos. (1 páginas)

ININ 31 Los lenguajes de programación en la enseñanza. (3 páginas)

ININ 13 Análisis comparativo entre los diferentes lenguajes de programación desde un punto de vista educativo. (5 páginas)

LOGO 127 Filosofía educativa del lenguaje LOGO. (11 páginas)

LOGO 125 Caracteristicas del lenguaje LOGO. (11 páginas)

LOGO 122 Aplicaciones didácticas del lenguaje LOGO. (20 páginas)

LOGO 26 Primitivas y procedimientos. (31 páginas)

LOGO 73 Consideraciones prácticas: Pantalla, editor, área de trabajo. (7 páginas)

LOGO 35 Prácticas. (5 páginas)

LOGO 25 Soluciones a las prácticas. (5 páginas)

LOGO 27 Geometría de la tortuga. (6 páginas)

LOGO 39 Modularización. (4 páginas)

LOGO 51 Prácticas modularización. (4 páginas)

LOGO 53 Soluciones a las prácticas. (6 páginas)

LOGO 44bis Funciones. (8 páginas)

LOGO 130 Estructuras de control. (16 páginas)

LOGO 52 Prácticas estructuras de control. (2 páginas)

LOGO 54 Soluciones a las prácticas. (2 páginas)

LOGO 55 Prácticas recursividad. (4 páginas)

LOGO 56 Soluciones a las Prácticas. (4 páginas)

LOGO 124 Palabras y listas. (23 páginas)

LOGO 131 Prácticas palabras y listas. (4 páginas)

LOGO 132 Soluciones a las prácticas. (3 páginas)

LOGO 126 La tortuga tridimensional. (11 páginas)

LOGO 128 Prácticas tortuga tridimensional. (1 páginas)

LOGO 129 Editor de formas en LOGO. (15 páginas)

LOGO 117 Aplicación: Dibujo de una flor. (6 páginas)

LOGO 18 Programación de un reloj. (5 páginas)

LOGO 118 El cuento interminable. (5 páginas)

LOGO 121 Apredizaje del concepto de variable. (11 páginas)

LOGO 120 Aprendizaje del concepto de función. (11 páginas)

LOGO 23 La tortuga sale del laberinto. (7 páginas)

LOGO 20 Polígonos anidados. (18 páginas)

LOGO 123 Lista de procedimientos primitivos LOGO en castellano. (21 páginas)

LOGO 119 Bibliografía de LOGO. (1 páginas)

HOE 13 Hoja electrónica del Open Access. Introducción. (8 páginas)

HOE 14 Comandos de la hoja de cálculo del Open Access. (8 páginas)

HOE 15 Guia de operación. (12 páginas)

HOE 19 Creación de un modelo. (5 páginas)

HOE 17 Apéndices. (2 páginas)

HOE 16 Prácticas de hoja de cálculo. (1 páginas)

HOE 20 Hoja de cálculo: Aplicación didáctica. Cuadro de amortización de un préstamo. (10 páginas)

HOE 21 Prácticas. (1 páginas)

HOE 22 Ejemplo de hoja de cálculo: Indice de precios al consumo. (7 páginas)

HOE 23 Prácticas. (2 páginas)

INSTITUTO DE TECNICAS EDUCATIVAS (PROYECTO ATENEA)

CURSO DE INICIACION A LA INFORMATICA PARA DOCENTES

INTRODUCCION

Dentro del marco general expuesto en la descripción del Plan de Formación, este curso debe servir como un primer contacto del profesor con el mundo informático en general y de sus aplicaciones educativas.

Si bien una parte de los profesores pudieran poseer conocimientos previos sobre Informática, con este curso se trata de conseguir una asunción del Plan por todos los profesores implicados, estableciendo criterios uniformes respecto a metodologías, integración curricular; en resumen, se trata de aportar una visión pedagógica integral de la Informática en la educación.

Este curso, en sus líneas generales y en todos los aspectos metodológicos de contenidos, evaluación, etc., es elaborado por el I.T.E., aportando la documentación y material preciso.

Las orientaciones necesarias a los coordinadores de este curso se realizan en la segunda/tercera fase del Plan, en la que los monitores unificarán criterios y asumirán los objetivos del mismo.

Este curso comenzará con los cinco profesores de cada Centro Piloto seleccionado en 1985.

OBJETIVOS

- Valorar las repercusiones de la tecnología informática en la Sociedad y en la Escuela.

- Desmitificar la Informática.

- Proporcionar criterios al profesorado para el uso de los ordenadores en la escuela.

- Desarrollar la capacidad del profesorado de control de ordenador.

- Conocer a un nivel básico la constitución y funcionamiento de los ordenadores.

- Adquirir conocimientos básicos sobre los distintos usos de los ordenadores.

- Presentar las distintas variedades de uso de los ordenadores en la enseñanza.

- Conocer las características de distintos lenguajes de programación y programas de usuario y sus posibilidades educativas.

- Introducirse en la metodología de programación de lenguajes de alto nivel.

- Iniciarse en la programación con un lenguaje de alto nivel.

- Introducirse en la metodología del lenguaje Logo.

- Aplicar el lenguaje Logo como instrumento para resolución de problemas.

- Adquirir conocimientos básicos para la selección de equipos informáticos y programas de ordenador para la enseñanza.

- Capacitar para el uso de aplicaciones didácticas.

CONTENIDOS

1. **N.T.I. y Educación.**

   - N.T.I. y sociedad.
   - La escuela ante las nuevas tecnologías de la información.
   - Objetivos pedagógicos de la informática.

2. **Introduccion a los ordenadores**

3. **Posibilidades de los ordenadores en la enseñanza**

   - Introducción histórica.
   - Enseñanza programada.
   - Programas tutoriales.
   - Simulaciones.
   - Juegos.
   - Revisión y evaluación de programas de E.A.O.
   - Resolución de problemas.

NUCLEO: ININ
UTI: 0037
PAG: 3

- Inteligencia artificial.
- Programas de usuario.

4. **Iniciación a los Sistemas Operativos. Comandos básicos del MS DOS**

5. **Procesador de textos.**

- Características y uso.
- Estudio de un procesador de textos.
- Evaluación de procesador de textos.
- Posibilidades educativas.
- Modelo de aplicación: redacción.

6. **Bases de datos**

- Las bases de datos en la sociedad de la Información.
- Terminología básica y estructuras de las bases de datos.
- Base de datos documental: uso y aplicaciones.
- Las Bases de Datos en las aulas.
- Aplicación didáctica: Geografía de población.

7. **Lenguajes de programación.**

- Clasificación.
- Aplicaciones a la enseñanza.

8. **Lenguaje LOGO**

- Filosofía educativa.
- Características del lenguaje.
- Consideraciones prácticas: pantalla, editor, área de trabajo.
- Primitivas y procedimientos.
- Geometría de la tortuga.
- Modularización.
- Funciones, números.
- Estructura de control.
- Palabras y listas.
- Aplicaciones: gráficos y listas, gráficos 3D, tortugas múltiples.
- Aplicaciones didácticas.
- Metodología

METODOLOGIA

Las actividades de este curso se desarrollarán utilizando la documentación escrita junto a otros documentos auxiliares; artículos,

NUCLEO: ININ
UTI: 0037
PAG: 4

aplicaciones, etc. Las sesiones de trabajo iniciales se desarrollarán con el apoyo de material visual (diapositivas) y programas de ordenador ilustrativos de las posibilidades de los ordenadores en la enseñanza.
Las sesiones cuyo objetivo principal sea la adquisición de destrezas específicas de tipo instrumental se realizarán en microordenadores (comandos básicos del sistema operativo, estudio de un procesador de textos, base de datos documental, editores de pantalla, área de trabajo, etc,)

Las aplicaciones didácticas serán revisadas por los asisitentes para lo cual las pasarán como alumnos (redacción con procesadores de textos, uso de las bases de datos en Geografía de población, etc.)

En cuanto al lenguaje Logo, el primer objetivo a lograr en la formación de profesores es que éstos asuman, hagan suyos sus fines educativos puesto que el aprendizaje del lenguaje de programación como tal, es una tarea que sólo requiere tiempo y ejercitación, y una metodología adecuada, de forma que ese aprendizaje no representará un problema diferente del que se presente al enseñar otra disciplina cualquiera. Lo dicho no debe interpretarse tampoco como que la formación debe centrarse sólo en los aspectos de una filosofía pedagógica, sino en el sentido de que éste es el fin más dificil de lograr, y que, por tanto, ha de ser el más cuidado.

Hay que mostrar LOGO como es, en su doble vertiente, como un lenguaje de ordenador poderoso en sus aspectos de cálculo y de manipulación de símbolos, y, paralelamente, como una igualmente poderosa; es decir, sin olvidar que van a trabajar con un lenguaje de programación que se presta mejor que otros a la consecución de unos fines educativos.

Para conseguir en el proceso de formación los objetivos anteriores, una posible estrategia podría ser la que se describe a continuación.

Una primera fase consistiría en el análisis del papel de los lenguajes de programación en la enseñanza, el papel específico de Logo y su filosofía asociada, un análisis de las características de Logo como lenguaje y un aprendizaje de las órdenes básicas para realización de gráficos de tortuga, manipulación de listas y de las estructuras de control básicas.

En esta primera fase aún no podemos pretender que el profesor capte el papel real de Logo pero, sin duda, servirá ya de referencia para enmarcar el proceso de formación en su doble vertiente del papel de los lenguajes en la educación y de las raíces de Logo.

El aprendizaje de las órdenes del lenguaje no suele representar problemas para un adulto. Aquí Logo y su conocida tortuga, generan entornos de aprendizaje significativos que favorecen la concreción de ideas abstractas como la recursión, variables, el condicional, los

NUCLEO: ININ
UTI: 0037
PAG.: 5

procedimientos, etc.

Desde el primer momento, se tratarán de eliminar progresivamente al máximo las dificultades instrumentales de uso del ordenador: teclado, editor, etc., para evitar desde el principio que el medio instrumento entorpezca el proceso de aprendizaje. El método ha de ser el de "aprender haciendo", pero dejándole cada vez el camino expedito de dificultades, tanto de las instrumentales ya dichas, como de las expresivas o del lenguaje, y poniendo especial énfasis en las técnicias de resolución de problemas favoreciendo su modularización.

Se les animará a realizar pequeños proyectos como un trabajo de resolución de problemas y a que establezcan los propios grupos un método de análisis conjunto del problema y una división de los distintos procedimientos entre los componentes del grupo.

Con ese modo de trabajo adquirirán de forma natural y no dirigida un doble conocimiento: práctica en el lenguaje y comprensión de su dimensión educativa.

Naturalmente, estos trabajos irán precedidos de unas orientaciones generales y unas reflexiones sobre la filosofía de Logo y la metodología deseable para el diseño de aplicaciones y su desarrollo. La exposición de los distintos trabajos servirá de marco para la evaluación del proceso seguido por cada grupo y comprobar tanto el grado de asunción de la filosofía del lenguaje por parte de los profesores, como el nivel de sus conocimientos técnicos instrumentales.

En el proceso de formación deberán incluirse siempre desarrollos que impliquen aspectos didácticos, y la realización de estos pequeños proyectos deberá servir de marco para reflexiones sobre el papel del lenguaje en el aula. Para favorecer la asimilación de este papel por el profesor, se deberán proporcionar modelos de aplicaciones construídas, micromundos poderosos que incluyan reflexiones sobre los objetivos de los mismos y su metodología de uso que servirán de elementos motivadores respecto a las posibilidades del lenguaje.

El proceso de aprendizaje descrito evitará una reducción del lenguaje a la tortuga, situándola en su justo término de un micromundo de gran poder educativo, pero no el único posible.

DOCUMENTACION

Se entregará una documentación escrita, que será una documentación de apoyo y no una guía exacta de actividades a realizar, que, en lo posible, se ajustarán a la metodología descrita. Se adjunta bibliografía de cada tema, formatos de evaluación y aplicaciónes didácticas.

NUCLEO: ININ
UTI: 0037
PAG.: 6

Se acompaña igualmente un material visual (diapositivas) y programa de ordenador con distintos ejemplos de aplicación.

COORDINACION

La coordinación del curso la llevará a cabo el Centro de Profesores, con el apoyo y seguimiento del Instituto de Técnicas Educativas.

ASISTENTES

Cinco profesores por cada Centro Experimental del Proyecto "Atenea"

OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN LA EDUCACION

1. INTRODUCCION. OBJETIVOS: PEDAGOGIA Y EDUCACION

2. PREGUNTAS PARA UN ANALISIS SOCIAL.

3. ANALISIS DE LOS CAMBIOS: TRANSFORMACIONES SOCIALES, ECONOMICAS Y EDUCACION.

4. CAMBIA EL CONCEPTO DEL SABER.

5. LA ESCUELA SE ADAPTA MUY LENTAMENTE A LOS CAMBIOS SOCIALES .

6. UNA ESCUELA ACTUALIZADA, CRITICA Y EMANCIPADORA.

7. EL ORDENADOR EN LA ESCUELA.

8. OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN LA EDUCACION.

9. FORMULACION DE OBJETIVOS.

10. CONCLUSION.

11. REFERENCIAS BIBLIOGRAFICAS.

****

NUCLEO: ININ
UTI: 0010
PAG.: 2

OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN LA EDUCACION

Esteban Orive Castro
Director del Instituto de Técnicas Educativas

CONFERENCIA PARA LAS JORNADAS SOBRE "INFORMATICA Y EDUCACION EN LA ENSEÑANZA BASICA Y MEDIA"

1. INTRODUCCION. OBJETIVOS: PEDAGOGIA Y EDUCACION.

Dada la polisemia de los términos que estructuran el enunciado de esta conferencia, parece oportuno abordar, como cuestión previa, una elemental delimitación léxica, con el ánimo de convenir con ustedes unos presupuestos comunicativos mínimos de cara a la comprensión de lo que deseo sea un mensaje inicial para un posterior debate que, aunque en este lugar tiene que ser breve, después habrá de afrontarse en profundidad porque la proximidad, urgencia y trascendentalidad del tema así lo requieren.

El término "objetivos" refiere, en este contexto preciso, los fines o metas de la Educación, de antemano fijados, para conducir intencionalmente hacia ellos al sujeto en proceso de formación.

No es finalidad de esta reflexión presentar una taxonomía de objetivos que describan aprendizajes concretos. Aquí se trata de considerar la oportunidad de la enseñanza de la Informática en nuestro sistema educativo. Intentamos describir en función de qué fines educativos - objetivos pedagógicos - sería deseable para nuestra Escuela la inserción de esta materia en los currícula; y, finalmente, reflexionar sobre las posibles incidencias y los previsibles resultados que en los procesos de aprendizaje de las diferentes áreas, la Informática puede brindar a la tarea educativa general.

Y debo declarar aquí la particular concepción pedagógica que me ha servido de arranque para la formulación de estos objetivos: Me interesa destacar de la Pedagogía su naturaleza dinámica; la que conduce a denotar un concepto de Educación en desarrollo diacrónico paralelo al contexto social en el que se estructura. Una Educación en devenir relacional interactivo con el medio social en el que proyecta sus virtualidades y su acción. Desde esta actitud inicial he optado por un concepto crítico y emancipador de la educación y formación de la persona humana, que pueda facilitar la creación y aplicación de paradigmas que respondan adecuadamente a un análisis de nuestra etapa histórica en toda su complejidad: social, cultural, política, económica. Este análisis debe proyectarse también a una razonable anticipación de futuro para procurar un aprendizaje innovador (1). Una educación así concebida es liberadora para el sujeto y crítica con respecto a la sociedad.

2. PREGUNTAS PARA UN ANALISIS SOCIAL

Para realizar este análisis de las pecualiaridades y exigencias actuales de la sociedad nos preguntamos:

¿Qué está aconteciendo en nuestra era actual? ¿Qué transformaciones profundas tienen lugar en nuestro mundo, y de qué manera y en qué medida influyen y alteran el concepto de educación? ¿Cómo lograr que la Educación pueda llegar a ser el instrumento necesario de ayuda al alumno, en cuanto persona humana, para que entienda críticamente, se adapte libremente y sea capaz de mantenerse activamente en la estructura social para poder dominarla?

Desde estas cuestiones se encamina nuestra reflexión sobre el momento social - en un sentido muy amplio - por el que atraviesa nuestra civilización, hasta la propuesta generalizada de los objetivos pedagógicos de la Informática en la Educación, tal como han sido delimitados conceptualmente al comienzo; pasando consecuentemente por una sucinta consideración sobre la decadencia del modelo educativo tradicional y la sugerencia de la necesidad de nuevos paradigmas educativos convenientes y conformes con las nuevas demandas para la educación y la formación.

3. ANALISIS DE LOS CAMBIOS: TRANSFORMACIONES SOCIALES, ECONOMICAS Y EDUCACION.

Proliferan en la última década los análisis, informes, trabajos y ensayos - incluso novelados - (2) que desde todas las instancias sociales - economía,derecho, cultura, educación, administración ... - nacionales e internacionales se vienen realizando con referencia a la profunda transformación que tiene lugar en los sistemas sociales. Es de tales características y signos irreversibles que lleva a los expertos de cada sector a elevar este cambio a la categoría de nueva era. Evolucionan las estructuras económicas, determinando profundas modificaciones sociales, consecuentemente culturales, y son desbordados paradigmas firmemente establecidos en el siglo. El desarrollo ha superado ya las cotas puramente económicas y materiales para forzar una sobredimensión cultural que urge a la cooperación reflexionante de todos los pueblos, desde sus propios valores, para la creación de una conciencia común del problema y para hallar las soluciones del futuro.

La sociedad postindustrial asiste a una profunda revisiõn de con - ceptos básicos y fundamentales tales como saber, libertad y poder, que afectan, no sólo a paradigmas sociales, culturales y educativos, sino al propio modelo ontológico de persona humana. Y esto afecta a las referencias más elementales para una cosmovisión o interpretación del mundo. A este fenómeno han contribuído, de manera decisiva, y seguirán contribuyendo con magnitudes de crecimiento exponencial, el uso y explotación de las nuevas tecnologías de la información microelectrónicas y, más en concreto, la Informática.

La sociedad moderna ha pasado aceleradísimamente desde una explotación de recursos naturales energéticos - hoy drásticamente escasos - a una industrialización del trabajo intelectual, mediante los recursos de la alta tecnología de la información. No puedo proceder a un análisis

de la evolución necesaria de cada uno de los conceptos enunciados por condicionantes temporales evidentes. Me detendré únicamente en aquellos aspectos que afectan directamente a nuestra Escuela, al sistema y al paradigma educativo, en tanto son requeridos por una sociedad en cambio para crear nuevas oportunidades para la educación en relación con las nuevas circunstancias sociales.

4. CAMBIA EL CONCEPTO DEL SABER

La asociación de tecnologías de comunicación e informáticas nos están forzando a esta reconsideración del concepto de **"saber"** porque planteado éste, tradicionalmente, como válido "per se" en cuanto a la formación última del ser humano, viene hoy a reforzarse principalmente en su aspecto de valor útil. No estoy tratando de restar valores al saber culto, más necesario hoy que nunca en Educación, con objeto de evitar una suplantación de la Realidad por un aspecto puramente utilitario de la misma. Siempre el saber fue contacto con todo tipo de realidad y requirió estructurar y comprender los datos discriminados de ella que el sujeto recuerda, reestructura y reinterpreta. Esta utilidad persiste pero se modifica profundamente el saber aplicar técnico incluso para la propia adquisición y reelaboración contínua del saber científico.

Las Nuevas Tecnologías de la Información poseen una facilidad casi ilimitada para el almacenamiento, procesamiento y disposición de datos, que sería imposible manejar sin los medios actuales, con lo cual se aminoran las necesidades humanas de ciertas capacidades intelectuales, pero se exige el desarrollo de otras: Saber no es sólo ni principalmente retener sino conocer el modo más eficaz de disponer de la información.

Un hombre tiene conciencia de una cierta formación cuando se sabe en poder de un discurso científico. Está pasando ya el momento en que esta situación se consideraba dada con la tenencia y asimilación de saberes circunscritos a relatos o meta-relatos científicos. Hoy, el proceso de adquisición y consolidación de tal discurso, viene condicionado por la capacidad para procesar y tratar información a partir de lenguajes formalizados.

Para que un hombre se considere suficientemente formado en el futuro, ten-

drá necesidad de conocer, en alguna medida, estos lenguajes de las nuevas tecnologías de la información y deberá saber usarlos para alcanzar las metas educativas propuestas, con lo cual podrá seguir estudiando y remodelando los conocimientos adquiridos en un constante aprendizaje a lo largo de su vida: En suma, deberá adquirir y perfeccionar la capacidad para procesar, tratar y distribuir información.

5. LA ESCUELA SE ADAPTA MUY LENTAMENTE A LOS CAMBIOS SOCIALES

La información y la comunicación interactúan con la enseñanza a través de los medios sociales, durante tanto o más tiempo que la Escuela. La casi totalidad de las técnicas de producción de bienes se han modificado profundamente bajo el efecto del progreso técnico, pero los servicios educativos apenas han evolucionado ni en sus espacios ni en sus técnicas, fracasando por anacrónicos (4).

El problema de la modernización de la enseñanza no es nuevo. La eficacia de la escuela tradicional ha venido descendiendo progresivamente a causa de diversos factores, entre los que destacan, en la última década (5).

1) La expansión de los sistemas educativos, a causa de la demanda de puestos escolares y la prolongación de ofertas educativas para adultos. La Escuela no estaba preparada para el paso de una formación de élite a la educación masiva.

2) Desnivel cualitativo entre países desarrollados y países en despegue o claramente subdesarrollados.

3) Demanda de mejora de calidad educativa en relación con la mejora de otras técnicas de producción de bienes.

4) Inadaptación entre los niveles culturales y tecnológicos de la educación formal y las necesidades sociales y empresariales.

5) Competencia con otras fuentes de información y con otros sistemas de formación (TV, Vídeo, etc.).

De las investigaciones internacionales más recientes y, en particular, las españolas (6) se infiere que atravesamos por un período de reajuste y creación teórica y metodológica que arranca del análisis y valoración de paradigmas inmediatamente pasados, hoy vigentes pero caducos. Se constata que nos hallamos en una fase "preparadigmática", revelada fundamentalmente por un cambio de perspectiva curricular y por el abandono de modelos conductistas y de variables neopositivistas, condicionadas por una consideración de la educación que estaba cercana a la de un bien comercial ordinario.

6. UNA ESCUELA ACTUALIZADA, CRITICA Y EMANCIPADORA.

Llegados, pues, a una situación en la que la experiencia cultural parece ser que apunta a un saber formal basado en lenguajes, y que el ámbito escolar se desplaza a la calle, a la población en general, es necesario ya un cambio radical en las ecologías educativas que adapte las tecnologías del mundo circundante al quehacer escolar y recoja consecuentemente el reto de transformar los procesos educativos y los roles de los profesores. Si la Escuela no se adapta al ritmo tecnológico actual, tanto en la adopción de sus recursos como en el análisis y puesta a punto de las repercusiones que estos medios producirán inevitablemente en la cultura, en la educación y en el aprendizaje y sus métodos, quedará marginada del resto de los sectores sociales, carente de interés y culpabilizada por una voluntaria ignorancia de su entorno. No se podría entender hoy una Escuela que, poseyendo conciencia "crítica" de la negatividad de las nuevas tecnologías de la información, pretenda formar emancipadamente a los alumnos y no los eduquen para la comprensión y autodefensión frente a estas tecnologías que irrumpen en el mundo.

Corresponde a la Escuela una reconceptualización de la enseñanza, que necesariamente va a repercutir en los espacios, en los proyectos curriculares y en las funciones del docente.

Este principio y su práctica requeriría una propuesta detallada que evidentemente no poseo, ni es posible .legar a ella sin un debate previo profundo y laborioso entre todos los extremos sociales y personales a quienes directamente afecta. Pero no es aventurado sugerir que debe desarrollarse en fases sucesivas y coherentes, la primera de las cuales arranca de una sensibilización profesional y política del problema que debe plasmarse en realizaciones inmediatas:

1) Una conciencia progresiva y clara de que la Escuela comparte la cultura con la comunidad, en la que el saber popular, cada vez más, va creando centros de interés vital para los alumnos. La Escuela debe ser espacio de síntesis: tiene la responsabilidad de integrar esos saberes de una manera organizada y crítica. La Escuela debe utilizar los medios que motivan al alumno fuera de ella, no tanto para evitar el ser odiada por inadecuada, (7) cuanto por la responsabilidad que tiene de hacer devenir las experiencias de la vida en cultura (8).

2) Un plan serio y amplio de formación y perfeccionamiento del profesorado, sin el cual todo esfuerzo de reforma es baldío. Su rol cobra hoy una importancia realmente trascendental en la medida que se ve precisado a ser creador cultural de integración. No debo entrar en este terreno que tendrá su tratamiento adecuado esta tarde en este lugar.

3) Un cambio curricular inicial, que pasa por la introducción de la Informática, adecuando sus contenidos y métodos a cada una de las fases educativas - niveles - formalmente establecidas, no sin antes haber reflexionado sobre la oportunidad y fines educativos que se persiguen con tal decisión, materializado todo ello en una definición de objetivos pedagógicos.

7. EL ORDENADOR EN LA ESCUELA

En las conclusiones del análisis precedente se podría advertir un marcado carácter prospectivo, pero lo que constituye un fenómeno real y pre-

sente son las demandas actuales de la sociedad para el conocimiento y uso del ordenador; manifestación real, expresa y publicitada, hecho que determina al Sistema Educativo para analizar si es conveniente adecuarse ya al concierto tecnológico de la actualidad, bajo dos imperativos:

a) La incorporación a sus concepciones pedagógicas de aquellos principios educativos acordes con los fines de la formación integral del alumno como persona y como ciudadano libre.

b) La adecuación de la Escuela en sus métodos de enseñanza a los principios anteriores, poniéndolos al compás de los medios utilizados por el resto de los sectores sociales: Seleccionar los recursos más idóneos con el fin de perfeccionar el desarrollo de los aprendizajes educativos.

Teniendo en cuenta este doble mandato, que nace del análisis de la sociedad actual y de una razonable previsión de futuro - propia y esencial de la educación - se proponen a continuación dos grandes grupos de objetivos que requieren y justifican la inclusión de la Informática en los distintos grados o niveles de la enseñanza no universitaria, tanto de la educación formal convencional como de las otras concepciones o servicios educativos: formación permanente y educación compensatoria.

8. OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN LA EDUCACION

Por seguir una terminología convencional y conocida (9) los hemos agrupado bajo la denominación de "Objetivos generales", aquellos considerados como los fines que suponen metas de la Educación; y "Objetivos intermedios", los que definen grandes categorías de comportamiento, respectivamente. Dejo para ulteriores trabajos y especificaciones los objetivos que describen el saber hacer más preciso y concretado en materias de estudio o en estrategias de aprendizaje en relación con las mismas.

En ambas agrupaciones, que no pretenden seguir el orden riguroso de una supuesta jerarquía de valores, se han utilizado criterios de organización relativos a valores socioculturales, individuales y curriculares.

Son considerados, asimismo, los posibles campos de aplicación de la informática en las aulas:

a) Enseñanza de la Informática.

b) Uso del ordenador dentro de los procesos de aprendizaje como instrumento de desarrollo del pensamiento (E.A.O., simulación de procesos, instrumento de apoyo...)

9. FORMULACION DE OBJETIVOS

1) OBJETIVOS GENERALES

1.1. Contribuir a una educación emancipadora que permita a la persona, mediante una comprensión racionalmente crítica de los códigos y lenguajes formalizados de las Nuevas Tecnologías de la Información, entender el mundo en que vive, adaptarse activamente a la sociedad de cuya comunidad forma parte y defenderse de un consumo indiscriminado de mensajes que pueda recibir de su entorno inmediato.

Este objetivo viene exigido por la necesidad de no permitir que la persona sea sorprendida inerme ante una sociedad moderna que le va a someter a condiciones difíciles tales como hacer frente a una información superabundante, previsiblemente con-

tradictoria y de progresiva complejidad y cambio. La capacidad de enfrentarse a este problema y seleccionar el flujo informativo supone un alto nivel de conocimiento, una gran adaptabilidad y un desarrollado espíritu crítico, so pena de integrarse pasivamente como mero usuario o receptor acrítico de mensajes. No debe olvidarse que estos mensajes constituyen centros de interés para el alumno porque conoce lenguajes en los entornos no escolares (cultura popular) que la Escuela debe hacerle reelaborar en síntesis organizativa. Es esta una finalidad a la que debe tenderse desde el primer contacto del alumno con el sistema educativo; extenderse progresivamente por todos los niveles de enseñanza y ser cultivada mediante cualquier uso del ordenador - y otros medios - en el aula, bajo una cuidada orientación humanística del profesor.

1.2. Colaborar al desarrollo de un sistema educativo moderno, que garantice en la práctica el derecho de todos a la educación en pie de igualdad, en el proceso de educación formal y en los servicios de educación permanente y compensatoria.

Es probado que el ordenador, combinado con otras tecnologías audiovisuales, facilita no sólo las tareas del aprendizaje ordinario, sino también, las de recuperación de alumnos rezagados. Por otro lado, la creación de bancos de datos y la difusión de programas didácticos, permite que las zonas más deprimidas culturalmente, alejadas o diseminadas, se puedan lucrar de estos recursos de información más avanzados.

1.3. Favorecer la integración del alumno con dificultades para el aprendizaje (discapacitado), mediante el apoyo individual extraordinario que el ordenador le puede brindar.

Dado el cambio de concepción educativa que representa la integración en la escuela común y en el sistema de educación formal de los niños llamados "especiales", con profesores de apoyo, creemos que ahora es el momento oportuno para utilizar el ordenador para desarrollar este objetivo. (10)

1.4. Capacitar psicológicamente al individuo para que pueda autorregular y reequilibrar sus "umbrales perceptivos" con respecto a la ansiedad e inseguridad en el cambio.

Es preciso desarrollar la capacidad psicológica para tratar la inseguridad, como un tipo de destreza básica organizativa, así como para afrontar la ambigüedad, porque tanto en el aspecto cognitivo como en el afectivo el alumno necesita confianza y seguridad interna, no solamente a partir del dominio de un corpus de conocimientos específicos y destrezas fijas, sino también, de la posesión flexible y polivalente de distintas técnicas de aprendizaje y de acceso y tratamiento de la información, que el ordenador facilita, dentro de una situación en la que el cambio es, y previsiblemente va a seguir siendo, una constante sociovital. Por ello, este objetivo debe plantearse ya desde los niveles básicos y elementales de la Educación.

1.5. Proporcionar al alumno nuevos elementos de motivacion para el aprendizaje.

El error debe incorporarse como elemento constructivo de un proceso en permanente depuración, con lo cual se convierte en elemento motivador. Deja así de ser algo frustrante para el alumno.

La autosatisfacción que produce en el alumno el logro de metas, sean finales o intermedias, es asimismo motivadora. Esto se puede lograr tanto si se realizan programas y se les hace funcionar correctamente, como en aquellas situaciones en las que el ordenador es instrumento de ayuda para el aprendizaje de otras materias.

1.6. Conferir un rol activo al alumno.

Un objetivo pedagógico recomendado por psicólogos y pedagogos es el de crear situaciones de aprendizaje en las que el sujeto interaccione directamente con el objeto del conocimiento.

En los últimos años el profesorado ha realizado grandes esfuerzos para introducir la pedagogía activa en sus aulas a través, principalmente, de situaciones de aprendizaje fundadas en la investigación y en la experimentación. Una de las dificultades que surgen al planificar estos aprendizajes es que el material didáctico del que se dispone, no permite revelar más que una serie poco amplia de estrategias de aprendizaje.

Con un uso adecuado del ordenador, se pueden planificar aprendizajes en los que el sujeto decida qué proyectos quiere realizar y qué informaciones suplementarias necesita para realizarlos. Aquí, la tarea del educador consiste en responder a sus demandas, aportándole nuevos elementos si fuera preciso. Esto da lugar a que surjan distintas estrategias en la resolución de una situación y el ordenador facilita que ellas se respeten. Ello posibilita que el alumno sea autor y creador y ésto supone algo superior a la mera acción.

BIBLIOTECA - MINISTERIO DE EDUCACION Y CIENCIA

1.7. Fomentar el carácter lúdico del proceso de aprendizaje como uno de sus componentes esenciales.

Es muy importante recuperar el juego para la enseñanza, y no sólo en los niveles básicos, donde es elemento motivador irrenunciable. El ordenador permite el juego en una situación abstracta, sometido a reglas convencionales donde ganar y perder no significa victoria o derrota. Desaparecería así la ansiedad y frustración ante el posible fracaso (11).

La actuación lúdica produce en el alumno un sentimiento de competencia en un marco distanciado de los adultos; y un desarrollo de la capacidad de concentración, al tener que manejar bases de datos mentales altamente elaborados.

Como instrumento didáctico, los juegos contienen materiales de aprendizaje implícito de interés para el educador, especialmente aquellos de lógica y estrategia y los que utilizan simulaciones del mundo real (12).

Lograr este objetivo sería alcanzar la meta superior intemporal del "enseñar deleitando" de nuestros clásicos, y el "aprender a gozar de la existencia junto a los demás", objetivo de la actual Reforma de las Enseñanzas Medias.

1.8. Mejorar la calidad de enseñanza y los resultados educativos mediante la aplicación de recursos diferentes a los tradicionales, que provoquen nuevas propuestas metodológicas.

El hecho de que el profesorado revise críticamente sus métodos y medios didácticos, y reflexione sobre nuevas oportunidades y posibilidades ante el impacto de las nuevas tecnologías es ya un síntoma positivo y esperanzador.

Se plantean necesidades nuevas, tales como investigar las formas más apropiadas de uso del ordenador como elemento de apoyo a los procesos de enseñanza-aprendizaje. Esto obliga al profesorado a elaborar nuevas metodologías profundizando en el estudio de los mecanismos cognitivos del alumno y en la adecuación de los nuevos medios para conducir el aprendizaje (13).

1.9. Capacitar al alumno para resolver situaciones densas, que le ayuden a superar las dificultades objetivas de su actividad.

Estas dificultades van a presentársele al alumno, en principio, como situaciones complejas en su mundo real.

Una de las principales finalidades de la Informática, consiste en el desarrollo de algunos modelos de pensamiento, de cierto tipo de estrategias que el alumno puede emplear al enfrentarse con situaciones nuevas. Con la ayuda de ésto aprende a interiorizar el problema y posteriormente incidir en el proceso que lo lleva a diseñar el algoritmo que resuelve dicho problema. Por otra parte, la posibilidad del tratamiento de grandes cantidades de datos favorece la aproximación a este tipo de problemas.

1.10. Proporcionar un conocimiento general de la naturaleza y utilización del ordenador como instrumento principal de comunicación en la sociedad, valorándolo objetivamente en sus funciones social y educativa con relación a otros medios culturales afines, para contribuir tanto a su desmitificación como a su justo aprecio.

Este objetivo hace alusión directa a lo que se ha dado en llamar la "alfabetización informática". Debemos valorarlo como coyuntural, de primera fase y difícil alcance, en la medida que los ordenadores de la cuarta generación no atienden todavía las necesidades humanas de los usuarios, sino que requieren de unos conocimientos y habilidades previas. En el momento que el desarrollo de las computadoras llegue a obtener capacidades hasta ahora asociadas a la de los seres humanos (Inteligencia Artificial) como la comprensión del lenguaje natural, no se necesitará conocer y aprender su funcionamiento interno. Es más, al estar el hombre liberado del problema artificial de cómo funcionan los ordenadores, podremos volver al objetivo básico de la educación, es decir, a "la alfabetización humana para seres humanos". (14)

En esta etapa el ordenador estará ya incorporado de forma natural en el sistema social y en el subsistema educativo.

2) OBJETIVOS INTERMEDIOS

En un segundo nivel de especificación vamos a definir objetivos intermedios como resultado de la consideración de dos elementos:

- La persona en su dimensión individual y social (capacidades, relación interactiva, integración social, etc.)

- El currículo.

2.1. Favorecer la adquisición de las capacidades básicas de lectura, escritura, expresión y cálculo.

Es un objetivo para alcanzar en ciclos de Preescolar y E.G.B. mediante programas didácticos, juegos informáticos, etc. Una Enseñanza Asistida por Ordenador adecuada, puede surtir efectos valiosos. Es un medio útil para realizar aprendizajes repetitivos para conocer nociones y reglas simples bien estructuradas; para ampliar un vocabulario básico; para aprender unas reglas de gramática y cálculo aumentando su efectividad con el empleo de medios gráficos (15).

2.2. Orientar al alumno para que, a la salida del sistema educativo, pueda encontrar un sentido gratificador al puesto de trabajo que ocupe en una sociedad tecnológicamente avanzada.

La Informática está generando formas de vida distintas y determinando destrezas y técnicas nuevas. La Educación debe responder a estas nuevas necesidades con el desarrollo y fomento de actitudes y habilidades pertinentes a dichas necesidades, para contribuir a una próxima inserción social del alumno con un mínimo de coherencia. Esto no es posible lograrlo eficazmente si no es con el conocimiento y manejo del medio que produjo la necesidad.

2.3. Orientar el ocio resultante del reparto equitativo del trabajo.

- *Viviéndolo y asumiéndolo dentro del período de educación formal (procurando que lo disfrute).*

- *Disfrutándolo como un servicio de educación permanente.*

- *Debiendo enfocarlo como contribución a que la persona mantenga el debido respeto a sí misma y la suficiente autoestima en los períodos obligados de paro laboral, en tanto no se reestructure el trabajo y los patrones sociales. Se hará necesario, en este caso, un reciclaje y perfeccionamiento, sobre bases muy flexibles, para que sea duradero y polivalente.*

El ordenador puede contribuir a este objetivo desde su característica de máquina virtualmente inteligente, que brinde oportunidades y recursos a muy distintos niveles, para organizarse el ocio de una forma creativa.

2.4. Incrementar la capacidad de comunicación y, por tanto, de aprendizaje de los alumnos con deficiencias de lenguaje y expresión.

Dos capacidades hay que desarrollar en el alumno discapacitado: primero la de comunicar, es decir, leer, escribir e interaccionar con los "media" de la clase, con el profesor y sus compañeros. En segundo lugar, la de controlar su entorno educativo.

Ambas capacidades de comunicación y de control están muy relacionadas. La introducción del ordenador puede ser determinante en el proceso de integración, debido a su "paciencia", posibilidad de controlar otros dispositivos y capacidad mediante los programas adecuados, de adaptarse a cada individuo.

2.5. Contribuir a la adquisición de las habilidades de autoaprendizaje mediante la acción y el descubrimiento, estimulando la creatividad y respetando la particularidad de las estrategias de cada alumno.

El uso del ordenador puede favorecer estrategias heurísticas, especialmente si se utilizan metodologías como las basadas en LOGO.

2.6. Desarrollar capacidades lógicas.

A la consecución de este objetivo se puede llegar con las aportaciones de la PROGRAMACION LOGICA que realizadas en lenguaje de INTELIGENCIA ARTIFICIAL permiten el aprendizaje de la lógica como un lenguaje de programación (16).

En casi todas las actividades informáticas existe un aprendizaje lógico implícito, si bien es cierto que al pensar en capacidades lógicas a veces se piensa más en la "lógica de las máquinas" que en la propia lógica formal.

2.7. Incrementar la capacidad de organizar el saber por medio del acceso y tratamiento de la información.

Porque considerando que es esta una función educativa que ha venido impuesta desde siempre, en la situación actual se incrementa esta necesidad proporcionalmente a la cantidad y complejidad de la información.

Los Bancos de Datos contribuyen a facilitar las operaciones

memorísticas del hombre más que los libros y las bibliotecas.

La simple memorización de los conocimientos es insuficiente. Es necesario estructurarlos y organizarlos. Pero para poder asimilarlos, controlarlos y continuar adquiriéndolos se requiere la presencia de ciertas facultades tales como la observación, comparación, clasificación, simplificación, abstracción y razonamiento lógico inductivo y deductivo, que se desarrollan en el análisis y tratamiento informático de datos.

2.8. Desarrollar la capacidad de resolución de problemas.

Este objetivo puede conseguirse: desarrollando una metodología algorítmica y estrategias heurísticas de tratamiento de problemas. La capacidad de resolución de los mismos, especialmente en los aspectos de análisis y modularización, puede ser desarrollada de forma creativa utilizando lenguajes de tipo procedimental que por su propia naturaleza estimulan la modularización de los problemas.

2.9. Desarrollar la capacidad de aprendizaje permanente y autónomo.

Dada la aceleración del cambio que se está produciendo en casi todas las áreas del conocimiento, es hoy más necesario que nunca el logro de este objetivo, que la propia Informática puede proporcionar por sus posibilidades de adaptación al individuo y de acceso a datos, programas, etc. El ordenador es el instrumento adecuado para un aprendizaje autónomo.

2.10. Favorecer la interdisciplinariedad entre las diferentes materias del currículum".

En un sentido general, la Informática, como ya se comentaba anteriormente, es un elemento importante para el desarrollo de capacidades que, como la de resolución de problemas, se consideran hoy básicas en el currículum de las distintas materias. Igualmente favorece la interdisciplinariedad entendida en un sentido metodológico, ya que la informática va a dar una posibilidad de confluencia a profesores de distintas áreas en una metodología común. Desde el aspecto de los contenidos, el ordenador puede hacer converger a profesores de distintas áreas realizando proyectos comunes, simulaciones, etc. (17)

2.11. Introducir elementos nuevos en los currícula, desde la perspectiva de nuevas materias de enseñanza de nuevos contenidos y tratamiento metodológico y didáctico de las disciplinas ya existentes.

Aquí se centra la necesidad de afrontar la reforma curricular desde la perspectiva de unas metodologías y didáctica nuevas que trae consigo la introducción de la Informática y el uso del ordenador en los niveles de Enseñanza Básica y Bachillerato General.

La nueva metodología vendrá exigida tanto por los nuevos medios que el ordenador pone a disposición del profesor (tratamiento de textos, hojas electrónicas, representaciones gráficas) como por las posibilidades en las distintas actividades que el ordenador puede realizar asociadas a los diferentes niveles cognitivos: los ejercicios y prácticas (conocimiento y comprensión); actividad tutorial (aplicación); juegos educativos (análisis), simulaciones (síntesis): análisis y resolución de proble-

mas (evaluación). (18)

Una nueva metodología que puede desarrollarse con el uso del ordenador es la que proviene de la concepción metafórica del mismo como educando (como alumno del propio alumno).

Pensando en el Bachillerato Superior y en la Formación Profesional, el objetivo se dirige hacia la introducción de nuevos contenidos tales como control con microprocesadores, electrónica y sus aplicaciones, computadoras, programación, proceso de datos, proceso de textos y otras técnicas informático-administrativas, almacenamiento y recuperación de datos, etc. (19).

Con esto se pretende que el ordenador mejore los enfoques educativos actuales, introduzca nuevos enfoques deseables y elimine los enfoques obsoletos existentes.

2. 12. Atender a las necesidades individuales en la enseñanza.

La individualización de la enseñanza puede ser propiciada de manera óptima con el ordenador, tendiendo a restablecer el equilibrio de aquellos alumnos que proceden de medios con dificultades (bilingüismo/diglosia) o desfavorecidos socio-culturalmente. Asimismo, es un medio apropiado para la recuperación de alumnos con dificultades de aprendizaje.

Otro aspecto que puede cubrir el ordenador, es la atención a los alumnos superdotados.

10. CONCLUSION: HAGAMOS YA EL CAMINO EDUCATIVO CON LA INFORMATICA DENTRO DEL PROCESO ESCOLAR.

Dentro de los límites de la obligada brevedad de este acto hemos intentado resumir una dilatada reflexión de la que, por fuerza, hemos tenido que dejar fuera temores muy importantes que seguramente serán tratados desde otras perspectivas durante los días de este encuentro.

Concluímos, pues, en que las necesidades de la Sociedad demandan la incorporación de la Informática a la Educación. De la consideración de estas necesidades, en relación con las propias del individuo y las del currículum, se establecen una serie de objetivos que justifican dicha incorporación al sistema educativo, y proponen líneas de trabajo para que tal inserción obtenga logros valiosos para la persona y para la sociedad.

Queremos resaltar que esta definición de objetivos no pretende ser más que el principio general para dar paso a aplicaciones concretas que, junto con la investigación, nos van a aportar datos para una formulación más ajustada en el próximo futuro.

El mundo de la Educación se halla ante el reto inevitable de las nuevas tecnologías y estimo que debemos afrontarlo con serenidad, sano optimismo, ilusión razonable y mucha imaginación. No podemos dejarnos arrastrar ni por una utopía informaticista, que nos desvíe de problemas humanos y sociales profundos, ni tampoco por el acusado pesimismo de un mal entendido intelectualismo purista conservador en la práctica, que entona lamentaciones frente al progreso técnico.

La Educación tiene que optar por una apuesta progresista de futuro en sereno equilibrio crítico porque si no sale preparada al cami-

NUCLEO: ININ
UTI: 0010
PAG.: 24

no de la revolución tecnológica acabará imponiéndosele irremediablemente sin el adecuado equipaje pedagógico y didáctico prevenido.

Alcaláde Henares, 29 de Octubre de 1984

Fdo.: Esteban Orive Castro

REFERENCIAS BIBLIOGRAFICAS

1) - INFORME AL CLUB DE ROMA 1979
Botkin, J.W., El mandjra y M. MALITZA: Aprender, horizonte sin límites.

2) - CONFERENCIA INTERNACIONAL SOBRE "LA EDUCACION Y LAS NUEVAS TECNOLOGIAS DE LA INFORMACION." C.E.R.I. - O.C.D.E., 1984.

- I JORNADAS NACIONALES SOBRE "INFORMATICA EN LA ENSEÑANZA". Barbastro, 1984.

- PROYECTO ATENEA. Documento base para la discusión y redacción del Proyecto. Ministerio de Educación y Ciencia, Madrid, 1984.

- SEMINARIO Y JORNADAS SOBRE "INFORMATICA Y DERECHOS HUMANOS", Madrid, 1984

- SEMINARIO LOGO, Valladolid, 1984.

- SEMINARIO SOBRE "OBJETIVOS, METODOLOGIA Y PEDAGOGIA DE LA ENSEÑANZA DE LA INFORMATICA", Buitrago, 1983.

- SEMINARIO SOBRE "REFORMA EDUCATIVA Y FORMACION DEL PROFESORADO: un reto de la España de los ochenta. Universidad Internacional Menéndez y Pelayo, Santander, 1984.

- LUSSATO, B.: El desafío informático, 1981.

- SCHILLER, H.: El poder informático, 1981.

- SERVAN SCHREIBER, J.J.: El Desafío Mundial, 1982

- PAPERT, S.: Desafío a la mente, 1981

- TOFFLER, A.: La tercera ola, 1980.

- Apuntes de Educación, nº 12, 1984.

- Cuadernos de Pedagogía, nº 107, 1983

3) - SCHILLER, H.: o.c.

4) - O.C.D.E.: Conferencia Internacional sobre las Nuevas Tecnologías y la Educación.

5) - Ibidem.

6) - Simposio sobre "Teoría y Práctica de la innovación en la Formación y Perfeccionamiento del profesorado", Subdirección General del Perfeccionamiento del Profesorado, Madrid, 1984.

- GIMENO SACRISTAN, J.: La Pedagogía por objetivos: una obsesión por la eficiencia, 1982.

- GIMENO SACRISTAN, J. y A. PÉREZ GOMEZ: La enseñanza: su teoría y su práctica, 1983.

7) - ALFIERI, F.: "El municipio y la renovación pedagógica: análisis de la experiencia del Ayuntamiento de Turín"

- Simposio "Teoría y Práctica de la Innovación en la Formación y Perfeccionamiento del Profesorado", 1984

8) - ILLICH, J.D.: Une Société sans Ecole, 1971

9) - LANDSHERE: Objetivos de la Educación

10) - Ver programas especiales del Sistema Educativo Británico (Mycro Electronic Program: "Microelectronics and children With Special Education Meeds": 1982

NUCLEO ININ
UTI: 0010
PAG.: 27

11) - G. ATKINSON, J: Motivation and Achievement

- G. ATKINSON, J: Leistungsmessun und Verhalten

- HECKHAUSEN, H.: Haudlungstheorien in interdisziplinärer Perspektive

12) - ZAMORA, Ricardo M.: "The Pedagogy of Game", en The Intelligent Schoo House

13) - GIMENO SACRISTAN, J.: "El profesor como investigador en el aula" Educa ción y Sociedad, nº 2, 1983

14) - MASOUD YAZDANI: New Horizons in Educational Computing. 1984

15) - SIMON, J.C.: La Educación y la Informatización de la Sociedad, 1984.

16) - MASOUD YAZDANI: o.c.

17) - TITONE, R.: "Por una nueva didáctica: Hacia la integración inter trans-disciplinar"

18) - POIROT, J.L.: "What a Teacher of Computing Should Know" en Intelligent Shool House, 1984

19) - M.E.P.: o.c.

EXTRACTO DE LA CONFERENCIA "OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN L EDUCACION" PARA LAS JORNADAS SOBRE "INFORMATICA Y EDUCACION EN LA ENSEÑAN BASICA Y MEDIA".

Noviembre 1984

1. INTRODUCCION. OBJETIVOS: PEDAGOGIA Y EDUCACION

Los objetivos son concebidos en este contexto como fines o metas de Edu cación, que deben lograrse con una adecuada integración de la Informáti ca en el Sistema Educativo. Se ha obtenido al cabo de un análisis d la estructura social y sus demandas educativas, dentro de una concep ción pedagógica que denota una educación crítica y emancipadora de persona en libertad, pensando en el hoy y en el futuro próximo.

2. PREGUNTAS PARA UN ANALISIS SOCIAL

¿Qué está aconteciendo en nuestra era actual? ¿Qué transformaciones tie nen lugar en nuestro mundo y cómo influyen en el concepto de educació ¿Cómo lograr que la educación llegue a ser instrumento de ayuda al alu no para entender, adaptarse y mantenerse activo, dominándola, en la e tructura social?

3. ANALISIS DE LOS CAMBIOS: TRANSFORMACIONES SOCIALES, ECONOMICAS Y EDU CION.

Los cambios producidos en las dos últimas décadas, debidos a la irru ción de las nuevas tecnologías de la Información, son de tal índole q nos abocan a una nueva era. Los modelos y paradigmas establecidos todos los órdenes están siendo desbordados. El nuevo desarrollo de sociedad postindustrial ha forzado una sobredimensión cultural toda no suficientemente asimilada. Urge reflexionar sobre ella desde una c ciencia internacional para hallar soluciones al futuro.

4. CAMBIA EL CONCEPTO DEL SABER.

El desarrollo de la telecomunicación y la informática fuerzan a una reconsideración del concepto de "saber". Las facilidades para la memoria, reestructuración y reinterpretación de datos se han multiplicado de tal modo que son aminoradas las capacidades memorísticas humanas para realizarlo, y surge la necesidad de otras habilidades distintas para disponer de la información. Este saber, más utilitario, es explotado por los nuevos mercados de medios, a través de redes y sistemas que nos enfrentan a una economía ligada a la información; de donde, cambia también el concepto de "poder", consistente en la toma de decisiones a partir de la recogida, procesamiento, coordinación y expansión de la información.

Se hace imprescindible para la educación y formación del hombre moderno un saber fundado no en relatos o metarrelatos científicos sino en la capacidad para procesar información a partir de lenguajes formalizados.

5. LA ESCUELA SE ADAPTA MUY LENTAMENTE A LOS CAMBIOS SOCIALES.

Mientras que la casi totalidad de las técnicas de producción de bienes se han modificado profundamente bajo los efectos del progreso tecnológico, los servicios educativos permanecen en una evolución anacrónica en cuanto a espacios, contenidos, didácticas y metodologías. La eficacia de la escuela tradicional desciende a causa de diversos factores:

a) Expansión de los sistemas educativos, a causa de la democratización de la enseñanza y la prolongación de ofertas educativas.

b) Desnivel cualitativo entre países desarrollados y resto.

c) Demanda de mejora de calidad educativa en relación con otras técnicas de producción de bienes.

d) Inadaptación entre los niveles culturales y tecnológicos de la educación formal y las necesidades sociales.

e) Competencia con otras fuentes de información y con otros sistemas de formación.

No hemos llegado todavía al establecimiento de un modelo educativo adecuado a la sociedad postindustrial. Estamos en una fase preparadigmática, claramente crítica con respecto a los modelos inmediatamente pasados.

6. UNA ESCUELA ACTUALIZADA, CRITICA Y EMANCIPADORA.

La Escuela actual se halla ante el reto de cambiar radicalmente las ecologías educativas, adaptando las tecnologías del entorno al quehacer escolar y transformando, en consecuencia, los procesos educativos y los roles del profesor. De no afrontarlo, quedaría marginada del resto de los sectores sociales y culpabilizada por anacrónica e inadecuada.

La reconceptualización de la enseñanza requiere un debate previo entre los responsables; debe desarrollarse en fases sucesivas y coherentes, la primera de las cuales parte de una inmediata sensibilización profesional y política del problema, que se materialice en tres realizaciones:

1ª) Convertir la Escuela en espacio de síntesis donde los saberes populares del entorno se integren en cultura.

2ª) Plan de formación y perfeccionamiento del profesorado adecuado a las nuevas realidades sociales y educativas.

3ª) Cambio curricular inicial que introduzca la informática en los niveles educativos no universitarios, adecuando a ellos contenidos y metodologías de una forma escrupulosamente discriminada. Para ello precisa de establecer unos objetivos pedagógicos a los que se ajuste dicha adaptación.

7. EL ORDENADOR EN LA ESCUELA

Las demandas sociales actuales para su conocimiento y uso son patentes.

El Sistema Educativo debe analizar su conveniencia desde dos presupuestos:

a) Su incorporación a partir de principios educativos acordes con la formación integral del alumno como persona y como ciudadano libre.

b) Mediante la adecuación de la Escuela en sus métodos a los principios anteriores: Seleccionar los recursos más idóneos en relación con el perfeccionamiento de los aprendizajes educativos.

8. OBJETIVOS PEDAGOGICOS DE LA INFORMATICA EN LA EDUCACION

Desde este doble mandato, nuestra reflexión concluye en la propuesta de los siguientes objetivos, que requieren y justifican la inclusión de la Informática en la enseñanza no universitaria. Se han seguido criterios de organización relativos a valores socioculturales, individuales y curriculares.

9. FORMULACION DE OBJETIVOS

1) OBJETIVOS GENERALES

1.1. Contribuir a una educación emancipadora que permita a la persona, mediante una comprensión racionalmente crítica de los códigos y lenguajes formalizados de las Nuevas Tecnologías de la Información, entender el mundo en que vive, adaptarse activamente a la sociedad de cuya comunidad forma parte y defenderse de un consumo indiscriminado de mensajes que pueda recibir de su entorno inmediato.

Colaborar al desarrollo de un sistema educativo moderno, que garantice en la práctica el derecho de todos a la educación en pie de igualdad, en el proceso de educación formal y en los servicios de educación permanente y compensatoria.

NUCLEO: ININ
UTI: 0010
PAG.: 32

1.3. Favorecer la integración del alumno con dificultades para el aprendizaje (discapacitado), mediante el apoyo individual extraordinario que el ordenador le puede brindar.

1.4. Capacitar psicológicamente al individuo para que pueda autorregular y reequilibrar sus "umbrales perceptivos" con respecto a la ansiedad e inseguridad en el cambio.

1.5. Proporcionar al alumno nuevos elementos de motivación para el aprendizaje.

1.6. Conferir un rol activo al alumno.

1.7. Fomentar el carácter lúdico del proceso de aprendizaje como uno de sus componentes esenciales.

1.8. Mejorar la calidad de enseñanza y los resultados educativos mediante la aplicación de recursos diferentes a los tradicionales, que provoquen nuevas propuestas metodológicas.

1.9. Capacitar al alumno para resolver situaciones densas, que le ayuden a superar las dificultades objetivas de su actividad.

1.10. Proporcionar un conocimiento general de la naturaleza y utilización del ordenador como instrumento principal de comunicación en la sociedad, valorándolo objetivamente en sus funciones social y educativa con relación a otros medios culturales afines, para contribuir tanto a su desmitificación como a su justo aprecio.

2) OBJETIVOS INTERMEDIOS

2.1. Favorecer la adquisición de las capacidades básicas de lectura, escritura, expresión y cálculo.

2.2. Orientar al alumno para que, a la salida del sistema educativo, pueda encontrar un sentido gratificador al puesto de trabajo que ocupe en una sociedad tecnológicamente avanzada.

2.3. Orientar el ocio resultante del reparto equitativo del trabajo.

2.4. Incrementar la capacidad de comunicación y, por tanto, de aprendizaje de los alumnos con deficiencias de lenguaje y expresión.

2.5. Contribuir a la adquisición de las habilidades de autoaprendizaje mediante la acción y el descubrimiento, estimulando la creatividad y respetendo la particularidad de las estrategias de cada alumno.

2.6. Desarrollar capacidades lógicas.

2.7. Incrementar la capacidad de organizar el saber por medio del acceso y tratamiento de la información.

2.8. Desarrollar la capacidad de resolución de problemas.

NUCLEO: ININ
UTI: 0010
PAG.. 34

2.9. Desarrollar la capacidad de aprendizaje permanente y autónomo.

2.10. Favorecer la interdisciplinariedad entre las diferentes materias del currículum".

2.11. Introducir elementos nuevos en los currícula, desde la perspectiva de nuevas materias de enseñanza de nuevos contenidos y tratamiento metodológico y didáctico de las disciplinas ya existentes.

2.12. Atender a las necesidades individuales en la enseñanza.

10. CONCLUSION: Hagamos ya el camino educativo con la Informática dentro del proceso escolar.

Esta definición de objetivos no pretende ser más que el principio general para dar paso a aplicaciones concretas que, junto con la investigación, nos aportarán datos para una formulación más ajustada en el futuro.

Conviene no dejarse llevar ni por la moda de una utopía informaticista ni por el pesimismo de grupos intelectuales puristas que se pronuncian en tonos jeremíacos de cara el progreso. Precisamos decisión, sano optimismo, ilusión racional y mucha imaginación para poder llevarlo a cabo con provecho integral para nuestros alumnos.

INTRODUCCION A LOS ORDENADORES

UN POCO DE HISTORIA

Desde la más remota antigüedad, el hombre ha tratado de aligerar una de las tareas más pesadas, desde el punto de vista intelectual, el cálculo. Para conseguirlo ha ido inventando una serie de artificios que van desde lo puramente teórico: sistema de numeración, algoritmos de cálculo, etc. hasta los computadores, dispositivos capaces de realizar las operaciones automáticamente.

Los computadores digitales, contrariamente a los analógicos, trabajan directamente con los números (dígitos). Su historia se puede considerar que comienza en los dedos de las manos y sigue con el ábaco, que permanece como único dispositivo de cálculo hasta el siglo XVII en el que Pascal construye una máquina sobre la idea de la rueda dentada, que era capaz de realizar operaciones de suma y resta. Unos años más tarde Leibniz le añadió la posibilidad de multiplicar y dividir utilizando ahora como elemento básico unas ruedas dentadas escalonadas.

En el siglo XIX Babbage concibió una máquina que podría enlazar una operación tras otra, sin necesidad de, a cada paso, requerir la intervención del operador. Las operaciones a realizar estaban marcadas en unas tarjetas perforadas, idea que tomó de los telares construídos por Jacquard.

Sin embargo, este proyecto no se convirtió en realidad hasta el año 1937 en que Howard Aiken pensó llevarlo a la práctica. En el 1944 se terminó de construir el Automatic Sequence Controlled Calculator (calculador controlado con secuencia automática), conocido como ASCC o MARC I.

En 1945 el Dr. John Von Neumann añadió un elemento nuevo a los calculadores de la época: la memoria. La memoria es la parte del ordenador donde están guardados, no sólo los datos, sino también el programa con la secuencia de operaciones a realizar. En cuanto a la programación, creó las instrucciones de salto tanto condicional como incondicional.

Estas ideas expuestas por Von Neumann permanecen vigentes hoy en día y prácticamente, todos los ordenadores actuales están realizados sobre ellas.

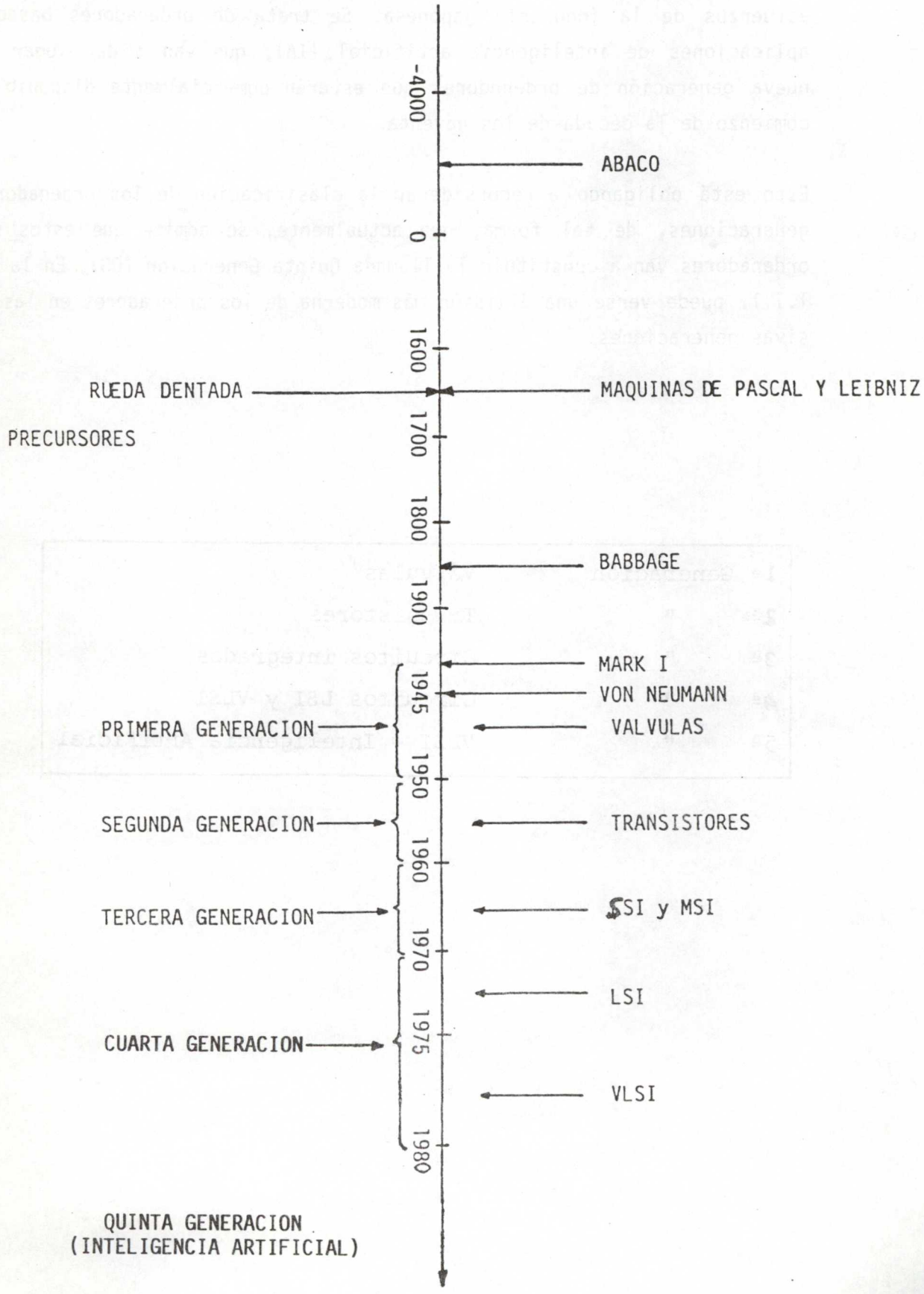
El diseño de los principales elementos de un ordenador está basado en el Algebra de Boole. El Algebra de Boole está formado por elementos que pueden tomar únicamente dos valores distintos: 0 y 1. Estos son los dígitos distintos existentes en el sistema de numeración binario que usan todos los computadores. Un dígito binario (binary digit) se denomina bit. Todos los computadores funcionan internamente en binario.

Los primeros computadores estaban realizados a base de válvulas electrónicas. Eran lentos, de dimensiones enormes y poco fiables. Constituyen la llamada primera generación. A continuación la válvula es sustituida por el transistor, mucho más pequeño y seguro en su funcioamiento. Es la segunda generación. Luego aparecen los circuitos integrados pequeños (SSI) dando lugar a la tercera generación. Más tarde estos circuitos integrados se van haciendo cada vez mayores en complejidad pero no en tamaño (MSI y LSI) y forman los ordenadores de la cuarta generación. En la actualidad se ha logrado introducir los principales elementos de un computador en un solo circuito integrado llamado microprocesador y estamos inmersos dentro de la quinta generación.

De una generación a otra se ha ido avanzando en miniaturización, fiabilidad, complejidad, velocidad y menor coste.

En la figura 1.1.1. se puede ver una cronología con los principales hitos de la historia de los computadores.

CRONOLOGIA DE LOS COMPUTADORES DIGITALES



Hay sin embargo un tema, que después de haber sido abandonado en 1968 por la empresa IBM, ha vuelto nuevamente a estar de actualidad gracias a los esfuerzos de la industria japonesa. Se trata de ordenadores basados en aplicaciones de inteligencia artificial (IA), que van a dar lugar a una nueva generación de ordenadores que estarán comercialmente disponibles al comienzo de la década de los noventa.

Esto está obligando a reconsiderar la clasificación de los ordenadores en generaciones, de tal forma, que actualmente, se admite que estos nuevos ordenadores van a constituir la llamada Quinta Generación (QG). En la tabla 1.1.1. puede verse una división más moderna de los ordenadores en las sucesivas generaciones.

1ª	Generación	Válvulas
2ª	"	Transistores
3ª	"	Circuitos integrados
4ª	"	Circuitos LSI y VLSI
5ª	"	VLSI - Inteligencia Artificial

NUCLEO: ININ
UTI: 0036
PAG.: 5

INTRODUCCION AL ESTUDIO DEL MICROORDENADOR

1.-Generalidades.

Los ordenadores actuales conservan básicamente la misma estructura que la máquina de Von Neumann.Se comenzará viendo la estructura interna de un ordenador y posteriormente se verán con más detalle cada una de sus partes.

Son tres los componentes básicos de cualquier ordenador:

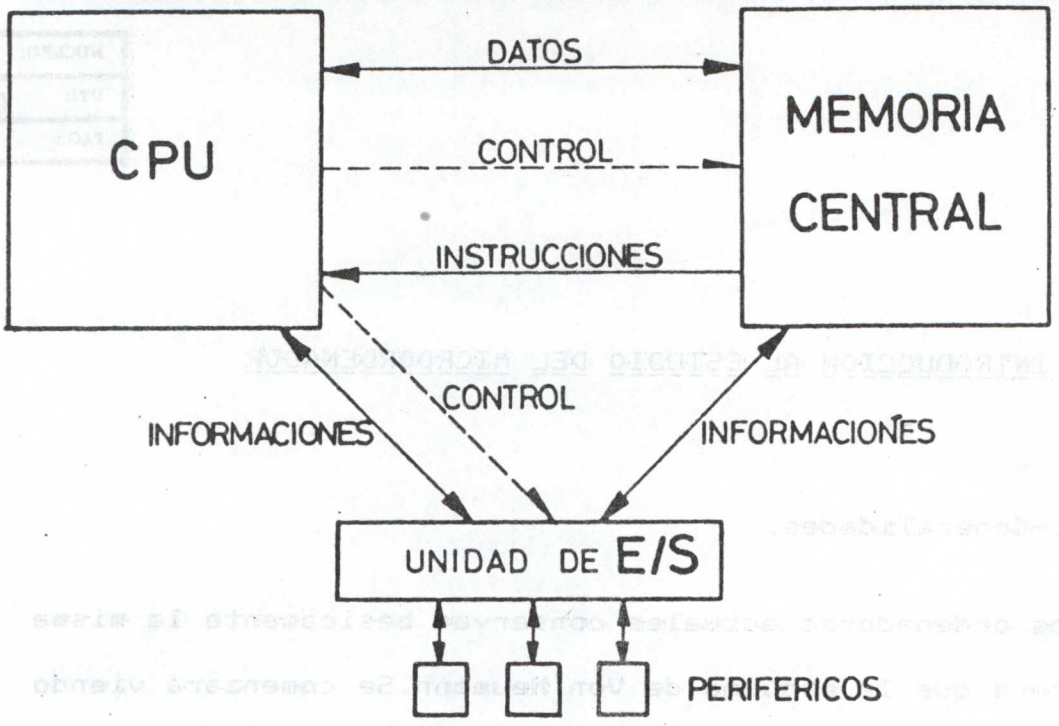
- unidad central de proceso (CPU).
- memoria principal.
- unidades de entrada/salida (periféricos).

La CPU es el componente encargado del funcionamiento de la máquina,lee las instrucciones,efectúa los cálculos y envía los resultados a la memoria principal o a los periféricos.

La memoria central contiene las instrucciones del programa y los datos necesarios para realizar dicho programa.

Las unidades de entrada/salida son el medio por el que se comunica el ordenador con el exterior.

En la siguiente figura se representa esquemáticamente la estructura funcional del ordenador:



2.-Memoria central.

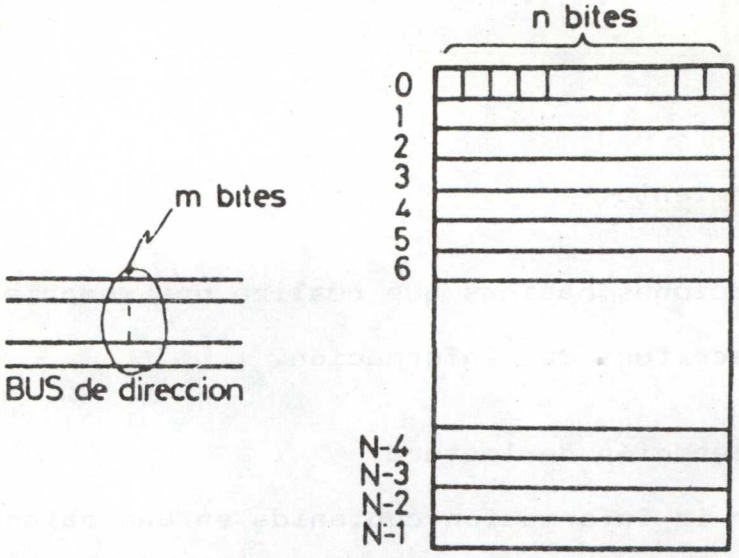
Como ya se ha dicho,la memoria contiene el programa constituido por las instrucciones y los datos.Se va a ver a continuación como está estructurada la memoria.

2.1.-Estructura de la memoria

Está formada por una serie de registros,cada uno de los cuales almacena una información digital.Cada registro forma lo que se llama,en el argot informático,una "palabra".Cada palabra,se identifica por un número al que llamaremos dirección de palabra(si existen N palabras,existirán N-1 direcciones de palabra o dirección de memoria).

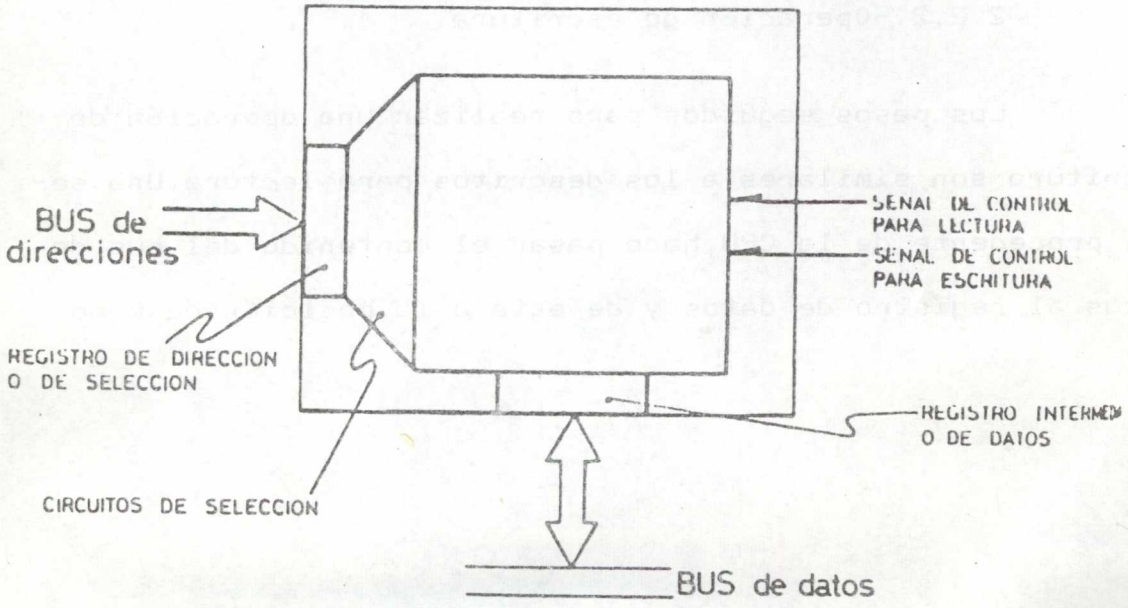
Según el tipo de ordenador,esta longitud de palabra es variable.Es frecuente que la palabra sea un octeto o byte,es decir,8 bits, aunque existen ordenadores cuya longitud de palabra es de 4, 6 y 16 bits.

En la figura siguiente,se representa la organización de una memoria de n bites:



Es importante señalar que el número de bits que componen la longitud de palabra,no tiene porqué coincidir con el número de bits que componen la selección de palabra.

Para leer o escribir la información en cada palabra de la memoria,es necesario disponer de un registro al que llamaremos registro de dirección o selección de memoria.Las informaciones que se deseen almacenar o leer de la memoria,pasan obligatoriamente por un registro intermedio llamado registro de datos.La información que llega al registro de selección lo hace a través del bus de direcciones(bus: conjunto de lineas eléctricas a través del cual fluye la información entre diferentes unidades del ordenador).Este bus de direcciones tendrá que tener m bits.La información que entra o sale del registro intermedio lo hará tambien a través de un bus,llamado bus de datos que tendrá que tener n bits.



2.2.-Funcionamiento.

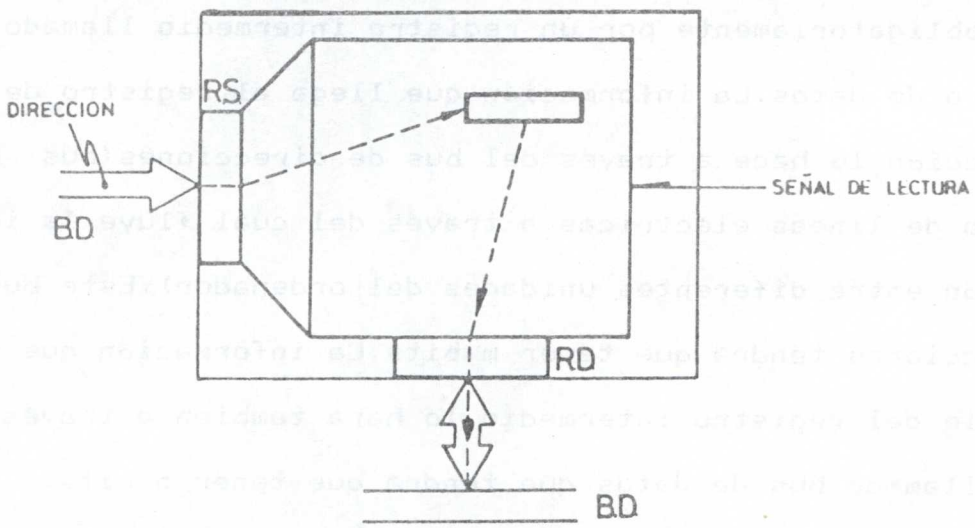
Las dos operaciones básicas que realiza una memoria son la lectura y la escritura de información.

2.2.1.-Operación de lectura.

Para leer la información contenida en una palabra de la memoria,se siguen los pasos siguientes:

a)desde el bus de direcciones se pasa al registro de selección de memoria la posición de la palabra cuyo contenido se desea leer.

b)el dispositivo de selección de memoria busca la palabra y debido a una señal de lectura procedente de la CPU la pasa al registro de datos y de este al bus de datos.Ver figura:



2.2.2.-Operación de escritura.

Los pasos seguidos para realizar una operación de escritura son similares a los descritos para lectura.Una señal procedente de la CPU,hace pasar el contenido del bus de datos al registro de datos y de este a la posición de memo-

NUCLEO: ININ
UTI: 0036
PAG.: 9

ria previamente seleccionada por el registro de selección.

3.-Caracteristicas funcionales.

Atendiendo al modo de operar de las memorias,se presentan a continuación algunas de sus caracteristicas.

3.1.-Modo de acceso.

Por el modo de obtener la información de una memoria se pueden distinguir:

a)Memoria de acceso directo o aleatorio.

Se asigna una dirección a cada palabra de la memoria y se accede a ella directamente.

b)Memoria de acceso secuerncial.

Para acceder a una determinada información,hay que pasar primero por las que hay delante de ella.

3.2.-Inscribilidad.

Se refiere a la posibilidad de modificar o no su contenido.Está directamente relacionada con la tecnologia de fabricación.

a)RAM (Random Access Memory).

Son memorias de lectura y escritura de acceso directo.

b)ROM (Read Only Memory).

Son memorias de acceso directo,pero de sòlo lectura.La

información que contienen se introdujo al fabricarlas y no se pueden borrar.

c)PROM (Programmable Read Only Memory).

Son memorias de sólo lectura programables una sóla vez.

d)EPROM (Erasable Read Only Memory).

Son tambien de sólo lectura,pero permiten ser borradas mediante rayos ultravioletas para grabarlas de nuevo.

3.3.-Capacidad.

Representa el número de palabras que puede contener.Su unidad de medida es el K que es igual a 1024.Una memoria de 32 Kbytes tendrá por tanto una capacidad de 32 x 1024 palabras de un byte.

4.-Unidad central de proceso.

Es la unidad que controla todo el sistema,es decir,tanto la memoria como los periféricos están gobernados por ella.

Desde un punto de vista funcional,la podemos dividir en dos módulos diferentes:

- unidad de proceso
- unidad de control

4.1.-Unidad de proceso.

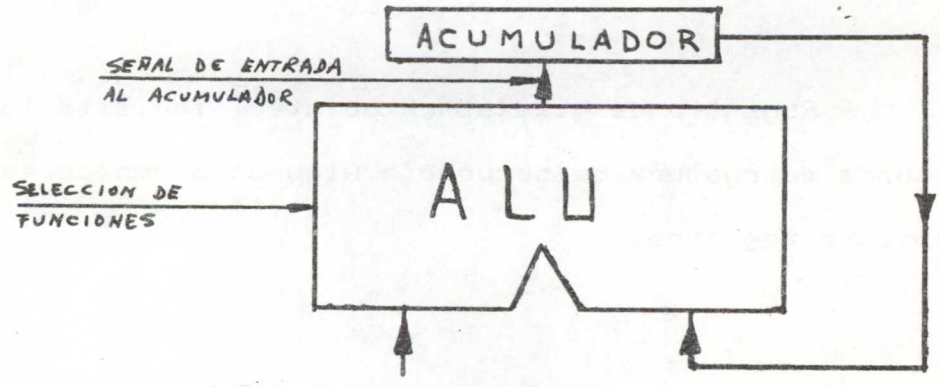
Realiza las operaciones aritméticas y lógicas que pida el programa y memoriza datos que aparecen entre las distintas instrucciones.Está constituido fundamentalmente por una memoria de acceso rápido(Scratch pad memory) y por la unidad aritmética-lógica(ALU).

4.1.1.-Unidad de acceso rápido.

Constituida por un grupo de registros que almacenan temporalmente informaciones de las operaciones que se están realizando.De estos registros destacaremos dos tipos de ellos:registros internos y registros de uso general.Los registros internos ayudan a la máquina en sus funciones y se ocultan al programador.Los registros de uso general pueden ser referidos por el programador,siendo el más típico de ellos el acumulador.

4.1.2.-Unidad aritmética-lógica.

Está compuesta por un conjunto de circuitos lógicos que bajo determinadas órdenes permiten realizar determinadas operaciones lógicas o aritméticas.En la actualidad la mayoria de los microordenadores adoptan,para la ALU,la siguiente configuración:



Esta configuración recibe el nombre de una dirección y su principal característica es que el acumulador es siempre uno de los operandos y recibe el resultado de la operación.

4.2.-Unidad de control.

Realiza las siguientes funciones:

- Recoger del programa las instrucciones que hay que ejecutar.

- Decodificar e interpretar información binaria contenida en la instrucción.

- Generar las microordenes precisas que correspondan a la decodificación contenida en la instrucción.

Para realizar estas funciones se ayuda de los siguientes elementos:

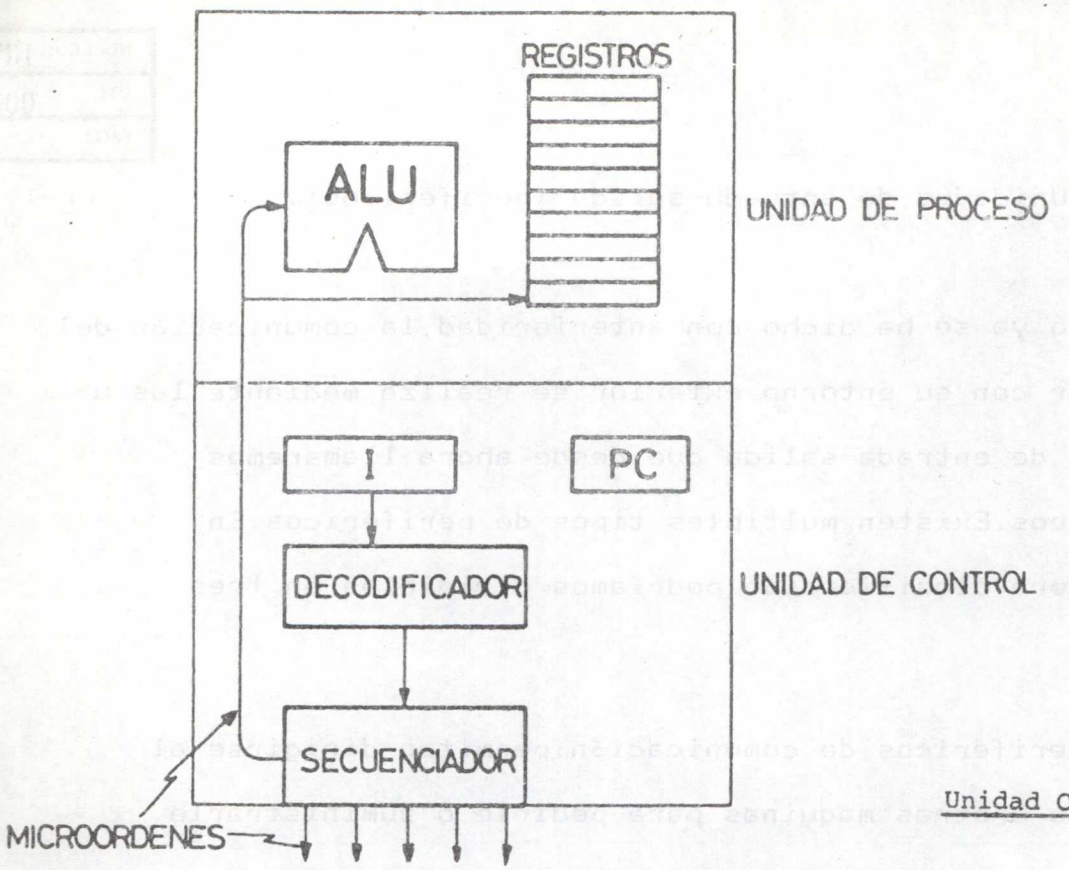
- Contador de programa(program counter):registro que contiene la dirección de la palabra de memoria donde se almacena la próxima instrucción a ejecutar.

- Registro de instrucción:recibe,desde la memoria central la instrucción a ejecutar.

- Decodificador:es un circuito conectado al registro de instrucción que generará las microordenes necesarias para ejecutar la instrucción.

- Secuenciador:genera la secuencia de esas microordenes.

- Registro de pila(stack pointer):facilita las instrucciones de ruptura de secuencia.Algunos ordenadores no poseen este registro.



Unidad Central de Proceso

4.2.1.- Busqueda y codificación.

Como ejemplo del funcionamiento de la unidad de control se va a describir el ciclo de busqueda y codificación de una instrucción de un programa.Simbólicamente y teniendo en cuenta que el símbolo () representa "contenido de",este ciclo se puede describir como:

(Contador de programa) ----------> Bus de direcciones

(Bus de direcciones) ---> Registro de selección de memoria

(Memoria) ----------> Registro de instrucción a través del bus de datos

(Contador de programa) + 1 ------> Contador de programa

A continuación,el registro de instrucciones se comunica con el decodificador para que averigue de que instrucción se trata.

5.-Unidades de entrada-salida (periféricos).

Como ya se ha dicho con anterioridad,la comunicación del ordenador con su entorno exterior se realiza mediante las u-nidaddes de entrada-salida que desde ahora llamaremos periféricos.Existen multiples tipos de periféricos.En una primera clasificación podríamos dividirlos en tres tipos:

- Periféricos de comunicación:permiten dirigirse al usuario o a otras máquinas para pedirle o suministrarle datos.
- Memorias auxiliares:son dispositivos de gran capa-cidad de almacenamiento que aumentan considerablemente la potencia del sistema.
- Periféricos especiales en control de procesos:se utilizan para leer señales presentes en un determinado proceso(temperaturas,presiones,tensiones,etc.).A través de ellos el ordenador puede enviar señales de control a dichos procesos.

5.1.-Descripción de periféricos.

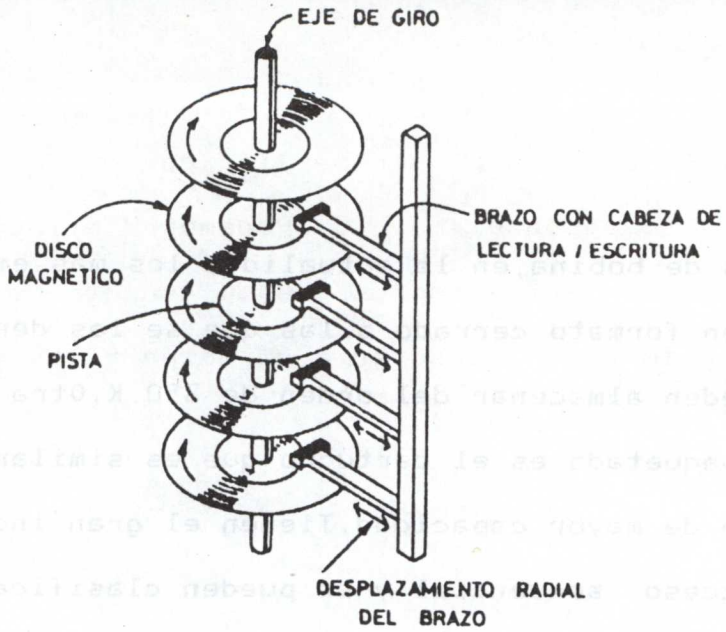
A continuación se van a describir los periféricos mas usuales.

5.1.1.-Unidad de cinta magnética: Aunque existen uni-dades de cinta de carrete abierto como las utilizadas en los

magnetófonos de bobina,en la actualidad los más empleados son las cintas en formato cerrado a las que se las denomina "cassettes".Pueden almacenar del orden de 350 K.Otra modalidad de cinta empaquetada es el cartucho que es similar a la "cassette" pero de mayor capacidad.Tienen el gran inconveniente de ser de acceso secuencial y se pueden clasificar como periféricos de memoria auxiliar.

5.1.2.-Unidad de disco magnético:como en el caso anterior se trata de unidades de entrada y salida de datos.El disco está constituido por un soporte plástico sobre cuyas caras se deposita material magnetizable.El disco está estructurado en coronas circulares concentricas denominadas pistas.La lectura o escritura se realiza mediante una cabeza que se desplaza a la pista donde esté situada la información que se desea leer o escribir.Son por tanto de acceso directo.

Existen dos tipos de discos.El disco normal(tambien llamado disco duro) y el flexible(floppy-disk o diskette). Se pueden agrupar sobre un soporte varios discos duros formamdo lo que se llama un "disk-pack"(ver figura).La capacidad de información de un disco duro normal es del orden de 5 a 20 Mb.



El disco flexible tiene unas dimensiones entre 3 y 8 pulgadas,siendo el más generalizado el de 5-1/4 de pulgada. La capacidad de almacenamiento está comprendida entre 160 y 600 Kb,segun el formato y la densidad.

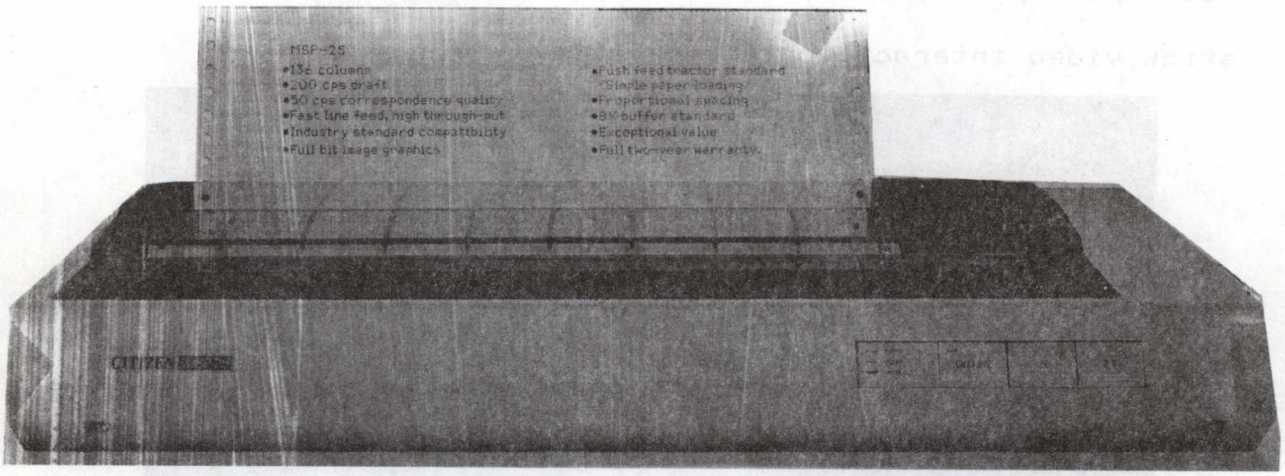
Las unidades de disco magnetico controlan el movimiento de las cabezas sobre los discos y envian o reciben la información del ordenador.

5.1.3.-Memorias auxiliares estáticas:son memorias basadas en dispositivos semiconductores que no necesitan mecanismos móviles para su funcionamiento.

Como ejemplo de estas memorias se presenta la memoria de carga CCD.Está formada por condensadores tipo MOS sobre un substrato que retiene carga.La información binaria se almacena por medio de dichas cargas.

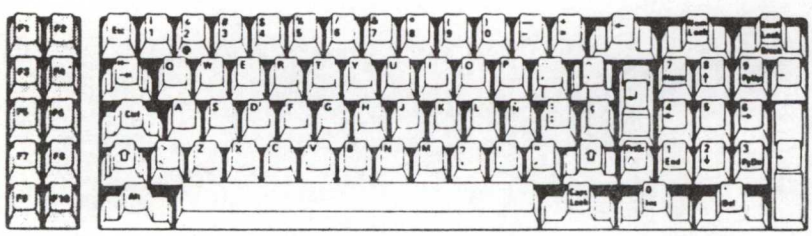
Existen memorias de este tipo de 64 Kbits,organizadas en sectores de 4 Kbits cada uno.

5.1.4.-Impresoras:son periféricos de salida.Existen multiples tipos en el mercado.Podemos distinguir por la forma de impresión entre impresoras de caracteres y de lineas ,las primeras imprimen caracter a caracter y las segundas líneas de una sóla vez.Por el mecanismo de impresión pueden ser térmicas,de matriz de puntos,de bola,de margarita,de chorro de tinta,laser,etc.

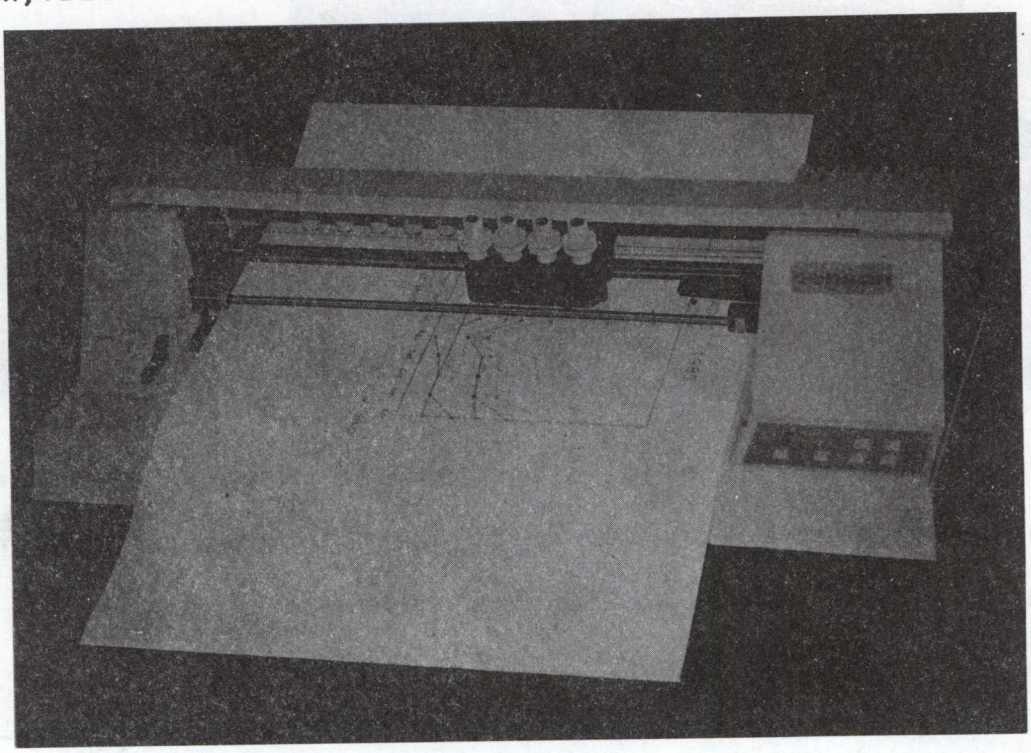


5.1.5.-Pantalla de rayos catódicos:también llamados monitores.Son periféricos de salida que no pueden guardar información.Sirven para visualizar la información.

5.1.6.-Teclado:dispositivo de entrada.Es el periférico que utiliza normalmente el usuario para comunicarse con el ordenador.Las teclas suelen estar dispuestas como en una máquina de escribir(teclado QWERTY).



5.1.7.-Otros periféricos.Existen otros muchos periféricos destinados a fines muy especificos,algunos de los cuales se citan a continuación:plotters,digitalizadores,lapiz óptico,ratón,lectoras ópticas y magneticas,joystick,video interactivo,terminales,etc.



Plotter-Printer.

FUNCIONAMIENTO DE UN ORDENADOR

Para ver el funcionamiento de un ordenador, supongamos este sencillo programa que suma dos números:

1) LEER A, B
2) S = A + B
3) ESCRIBIR S
4) PARAR
5) DATOS 3, 8

Las instrucciones llevan unas "direcciones" que, en primera aproximación, serán las direcciones de las palabras de una memoria donde se guardarán.

La primera operación a realizar es la "carga" del programa en memoria. Esto se realiza mediante las unidades de entrada (teletipo, por ejemplo).

Una vez en memoria, se da la orden de ejecución con lo que la unidad de control extrae de la memoria central la primera instrucción a ejecutar. A continuación interpreta su significado.

LEER A, B

que quiere decir que busque una instrucción de datos. Una vez buscada, asigna a la variable A el número 3 y a la variable B el número 8. Esto lo indicaremos mediante:

A ⟵ 3
B ⟵ 8

La flecha indica "asignación" y NO igualdad.

A continuación busca la siguiente instrucción:

S = A + B

y la U. de C. hace lo siguiente: lleva a la Unidad Aritmética y Lógica los valores de las variables A y B y ordena a la UAL que los sume, asignando a la nueva variable S el valor del resultado, o sea, 11.

Después busca ESCRIBIR S con lo que da órdenes a la unidad de salida de que escriba el valor que en ese momento tiene S que es 11. Por último se para.

NUCLEO: ININ
UTI: 0036
PAG.: 20

SOFTWARE

En el estudio de los ordenadores hay dos partes claramente diferenciadas: hardware y software.

El hardware, palabra inglesa sin traducción al castellano, es todo lo material, son los diversos componentes que forman físicamente el ordenador, es decir, la circuitería.

Por el contrario, software, que es otra palabra inglesa construída sustituyendo hard (duro) por soft (blando), constituye todo aquellos que no se puede tocar, lo que se le ha enseñado al computador, su comportamiento. Es, dicho brevemente, la programación. Este libro está dedicado al software.

El software es tanto o más complejo que el hardware. Para empezar hay que darse cuenta de que todo lo que hace el ordenador es producto de la ejecución de un programa. Algo aparentemente tan sencillo como cargar un programa con sus datos en memoria a través de una unidad de entrada, necesita para poder realizarse, que se ejecute un programa especial llamado precisamente cargador (boot strap) que va haciendo posible la introducción del programa y los datos. Es definitiva, un ordenador sin una serie de programas básicos que lo hagan funcionar, no vale absolutamente para nada. Esos programas básicos constituyen lo que se llama el sistema operativo.

Como ya hemos indicado el computador trabaja con toda la información en binario. Por tanto, las instrucciones y los datos, deben venir codificados por medio de bits. Estas instrucciones deben estar escritas en lo que se llama lenguaje máquina, o sea, lenguaje que puede ser comprendido directamente por el computador, compuesto por tanto por 0 y 1.

Ya se puede imaginar lo complicado y, sobretodo, lo pesado que puede resultar programar en este tipo de lenguaje.

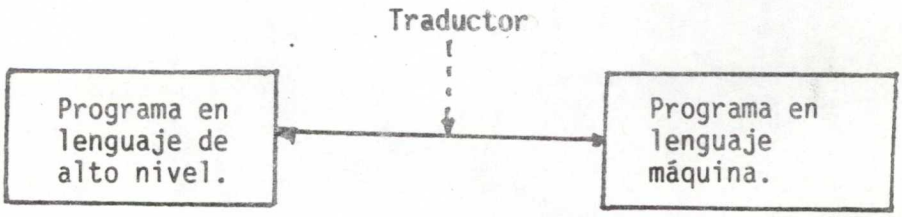
Baste para ello pensar, que un número que escrito en decimal tiene cuatro dígitos, en binario podrá tener hasta catorce.

Lo que se hace para simplificar la tarea al programador es, a un primer nivel, usar lo que se llama lenguaje ensamblador. En este lenguaje, las instrucciones tienen una estructura análoga a las instrucciones máquina, pero se permite usar códigos mnemotécnicos. Ya no hay que usar solamente bits, sino cualquier carácter alfabético numérico o especial, para codificar las instrucciones. Para ejecutar un programa, escrito en ensamblador, es preciso primero traducirlo al lenguaje máquina para que así pueda ser comprendido por el ordenador. Esta tarea la realiza un programa especial llamado también ensamblador. El ensamblador forma parte del sistema operativo y, todo ello, constituye lo que se llama programación de sistemas.

Todos estos lenguajes comentados hasta ahora se denominan lenguajes de bajo nivel. Para resolución de problemas complejos, tanto de tipo científico como de gestión, el uso del lenguaje ensamblador también resulta incómodo. Existen, para ellos, lo que se conoce como lenguajes de alto nivel, que están dirigidos a la resolución de los problemas y no a dar facilidades de comprensión al computador. Necesitan como es lógico de un traductor a lenguaje máquina, que se llama en este caso compilador. Ejemplos de este tipo de lenguajes con FORTRAN, APL, C, PASCAL, FORTH, ADA.

Una ventaja adicional en el uso de este tipo de lenguajes es que son idénticos para todos los ordenadores. Es practicamente igual un programa en el BASIC de un ordenador IBM que en el de un UNIVAC. Esto no ocurre con los de bajo nivel, en los que cada máquina, tiene su propio y particular repertorio de instrucciones.

Esta parte de la programación en lenguaje de alto nivel se emplea preferentemente en lo que se conoce como programación de aplicaciones.



POSIBILIDADES DE LA INFORMATICA EN LA ENSEÑANZA

El uso de los ordenadores en la enseñanza ha experimentado grandes transformaciones desde sus orígenes hasta hoy, dando lugar a un gran número de usos posibles diferentes.

Programas Lineales

Los primeros pasos consistieron en la realización de Programas Lineales, que de acuerdo con las teorías conductistas utilizaban el ordenador para resentar información escrita al alumno de modo secuencial. A éste se le realizaban preguntas para comprobar su rendimiento y si las respuestas eran correctas, se reforzaba con algún mensaje y caso de respuestas incorrectas se le suministraba algún tipo de apoyo. Nos encontrábamos en los comienzos del C.A.I. (Computer Aided Instruction) o E.A.O. (Enseñanza Asistida por Ordenador).

Programas Ramificados

Los programas lineales descritos fueron evolucionando dando lugar a programas en los que se permitían distintas alternativas en función de las respuestas de los alumnos o de sus intereses. Estos sistemas, conforme se fueron flexibilizando y se aumentaron sus posibilidades, permitieron registrar las respuestas de los alumnos, su historia de ejecución del programa, vueltas hacia atrás, generación de preguntas aleatorias, de forma que llegaron a hacer del ordenador un auténtico tutor.

El desarrollo de lenguajes específicos para realizar este tipo de programas, como el TUTOR o el COURSEWRITER, permitían que los profesores no tuvieran que ser programadores expertos.

Esta modalidad de uso de los ordenadores dió lugar a numerosos desarrollos, de los que destacaremos, por su mayor extensión, los proyectos PLATO y TICCIT a principios de los setenta. Ambos proyectos se basaban en grandes

ordenadores, y la llegada de los microordenadores hizo que la empresa creadora del proyecto PLATO, Control Data, creara versiones del sistema para uso en microordenadores. Estos sistemas trataron de aprovechar al máximo las capacidades de interacción de los ordenadores, PLATO utiliza pantallas táctiles de plasma, y daban grandes posibilidades al alumno de controlar su propia secuencia de aprendizaje.

La aparición de los microordenadores dió lugar a cambios en estos programas. La generalización de su uso en las escuelas e, incluso, privadamente en los hogares, trajo consigo un gran aumento y, a la vez, dispersión de la oferta de programas educativos para ordenadores, pero por sus limitaciones, en general podemos decir que con una gran disminución de la calidad educativa respecto de los desarrollados para ordenadores de mayor tamaño como los de los proyectos citados.

Por otra parte, se desarrollan lenguajes de autor más simples, como PILOT y sistemas de autor, que permiten a los profesores desarrollar autónomamente sus propios programas de enseñanza asistida. Se va poniendo más énfasis en gráficos, interacción del alumno, introducción de juegos, simulaciones que cada vez son más frecuentes en los programas de E.A.O. A la vez se van desarrollando sistemas que permiten controlar videocintas y videodiscos, multiplicando y diversificando las posibilidades de uso de los ordenadores en las escuelas.

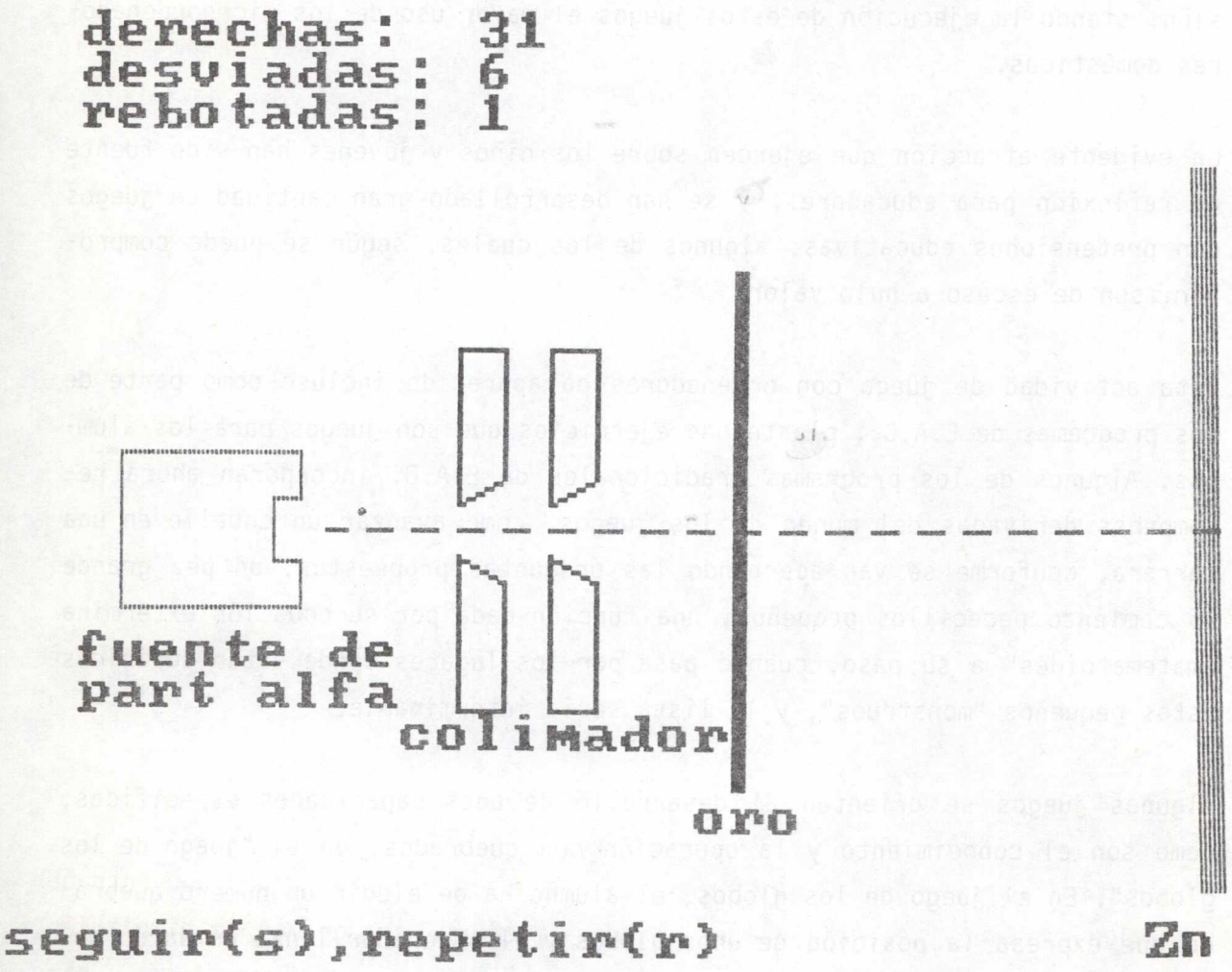
Simulaciones

La posibilidad de realizar modelos de situaciones reales, pero que entrañen riesgo o dificultad, ha llevado a la realización de programas de simulación, bien de experiencias no realizables fácilmente en el laboratorio, bien aprendizaje de técnicas antes de su uso real como son los simuladores de vuelo.

La realización de modelos en el ordenador por el propio alumno, es una actividad educativa de gran interés, y permite la comprensión del proceso a modelar de forma profunda, además de ser una actividad interdisciplinar y de alto valor intelectual.

La ejecución de un programa ya desarrollado de simulación también puede ser de gran interés si se permite la interacción variando distintos parámetros del modelo o representación gráfica de los procesos.

Ya se indicó que bastantes programas de E.A.O. llevan incluido algún tipo de simulación para que el alumno, al interaccionar con la misma, compruebe las teorías que se le han expuesto a lo largo del programa.



Simulación del experimento de Rutherford

Juegos

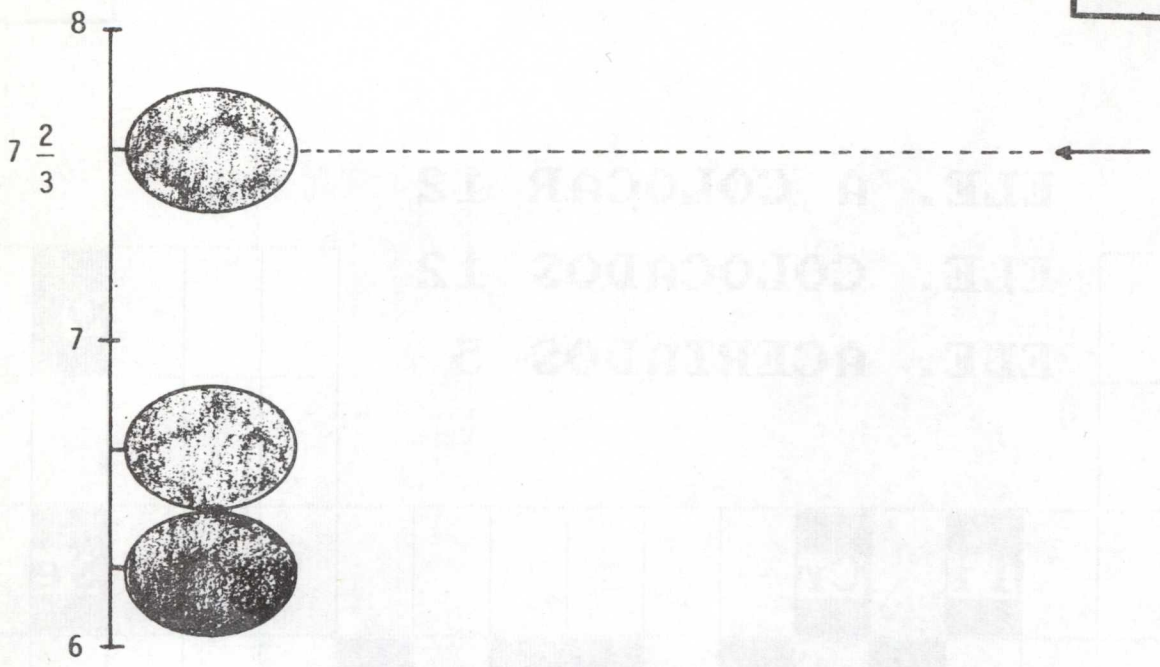
El uso de juegos de ordenador como instrumentos educativos se ha ido extendiendo, en algunos casos como actividades complementarias de programas de E.A.O., bien en las clases como ayudas al alumno, bien en las casas.

El desarrollo de programas de juegos de ordenador ha tenido una gran extensión, siendo la ejecución de estos juegos el mayor uso de los microordenadores domésticos.

La evidente atracción que ejercen sobre los niños y jóvenes han sido fuente de reflexión para educadores, y se han desarrollado gran cantidad de juegos con pretensiones educativas, algunos de los cuales, según se puede comprobar, son de escaso o nulo valor.

Esta actividad de juego con ordenadores ha aparecido incluso como parte de los programas de E.A.O., planteando ejercicios que son juegos para los alumnos. Algunos de los programas tradicionales de E.A.O. incorporan ahora recompenss derivadas del mundo de los juegos, como avanzar un caballo en una carrera, conforme se van acertando las preguntas propuestas, un pez grande va comienzo pececillos pequeños, una función dada por su ecuación extermina "matematoides" a su paso, cuando pasa por los lugares donde están definidos estos pequeños "monstruos", y la lista sería interminable.

Algunos juegos se orientan al desarrollo de unas capacidades específicas, como son el conocimiento y la operación con quebrados, en el "juego de los globos". En el juego de los globos, el alumno ha de elegir un número quebrado que expresa la posición de unos globos y el ordenador lanza un dardo sobre la posición indicada, pinchando el globo si acierta. El juego tiene varios niveles de dificultad, según el tamaño del globo y permite un gran ejercicio con los números quebrados y sus operaciones.

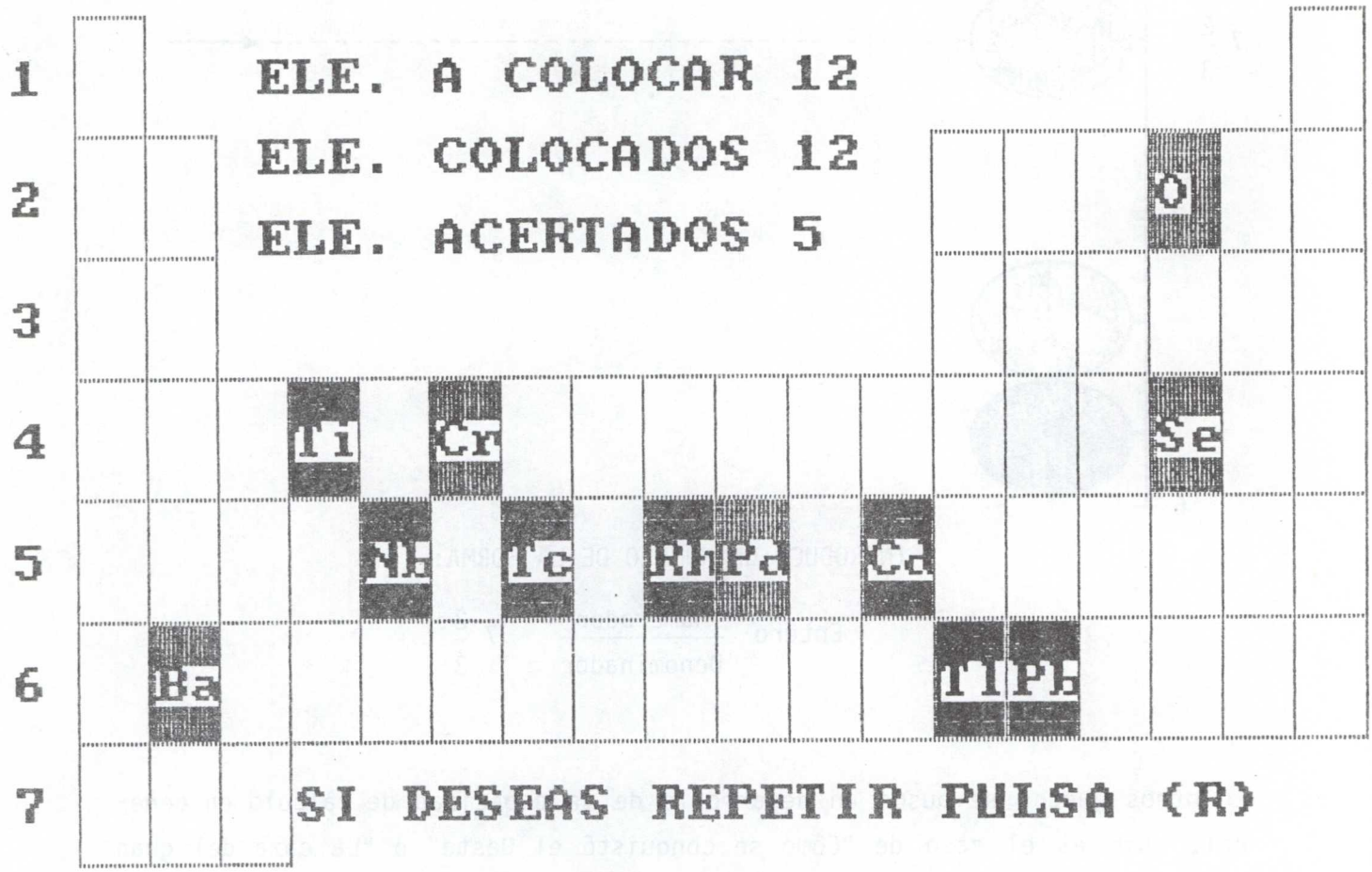


INTRODUCE UN NUMERO DE LA FORMA:

Entero $\frac{\text{Numerador}}{\text{Denominador}}$ $7\frac{2}{3}$

En otros juegos se busca un desarrollo de la capacidad de cálculo en general, como es el caso de "Cómo se conquistó el Oeste" o "La caza del gran número". En estos juegos, el alumno debe combinar con operaciones elegidas libremente un conjunto de números que aparecen en la pantalla y el número resultado le permite dirigirse a otro punto, como si fuese un dado, para cazar algo, llegar a la salida de un laberinto, con lo cual tiene que desarrollar estrategias de cálculo muy elaboradas para aprovechar lo mejor posible cada tirada.

Otros juegos tratan de desarrollar la capacidad de resolución de problemas en general, planteando distintas alternativas de estrategia de resolución de un problema-juego.



Juego para ayuda a un aprendizaje memorístico de la tabla periódica de los elementos.

Resolución de Problemas

Una de las aplicaciones más difundidas del ordenador en la enseñanza es la de realizar programas. En efecto, la mayor facilidad de aprendizaje de los lenguajes de ordenador como BASIC o Logo, este último utilizable incluso por los más pequeños, ha convertido la programación de ordenadores en algo sencillo y, además, útil para la enseñanza desde varios puntos de vista. La realización de programas es un instrumento de desarrollo de capacidades lógicas de resolución de problemas. Esta última especialmente en lenguajes que, como Logo, favorecen una estrategia de modularización de problemas. Por otra parte, permite utilizar el ordenador como instrumento de cálculo y para resolver problemas matemáticos o de otro tipo.

Además, permite un uso del ordenador como "educando", es decir, como alumno del propio alumno, ya que en el proceso de "enseñar" al ordenador a realizar dibujos o resolver problemas, el alumno aprende sobre el problema para el que está realizando el programa y sobre el proceso de aprendizaje en sí mismo, es decir, para "aprender a aprender", definiendo procedimientos en Logo o declarando hechos y reglas de inferencia de PROLOG, o realizando organigramas gráficos que le ayudarán a realizar programas en cualquier lenguaje. La descripción de un problema mediante un lenguaje simbólico altamente formalizado como son los lenguajes de programación es una actividad de alto valor intelectual.

Programas de Usuario

Los programas de uso más extendido en el mundo de la empresa son también de aplicación a la enseñanza.

Los alumnos pueden realizar algunos escritos manejando un procesador de textos, lo que cambia en cierto sentido la tradicional redacción en una composición en contínua transformación, pudiendo modificar partes de lo ya escrito, introducir módulos completos ya almacenados electrónicamente, borrar partes. La facilidad de modificación de la información electrónica puede introducir cambios en el proceso de redacción de los alumnos que utilizan procesadores de textos.

El uso de una base de datos puede servir de instrumento de aprendizaje tanto sobre los contenidos almacenados como sobre el propio proceso de selección y clasificación de la información relevante, así como los métodos de almacenamiento y recuperación de la misma, lo que sin duda son capacidades valiosas en la denominada sociedad de la "información", en la que se establece como objetivo de la enseñanza el desarrollo de un nuevo concepto del saber, en el sentido de conocimiento de los métodos de acceso y selección de la información frente al más tradicional de acumulación memorística de conocimientos.

El uso de hojas electrónicas de cálculo permite realizar modelos, previsiones, interrelacionar conceptos y ser, por tanto, un instrumento también de resolución de problemas y de simulación interactiva.

ENSEÑANZA ASISTIDA POR ORDENADOR

INTRODUCCION

Algunos de los usos más extendidos de los ordenadores en la enseñanza son:

- Programas lineales (Enseñanza Programada)
- Programas ramificados (tutoriales)
- Ejercicios y Prácticas
- Simulaciones
- Juegos
- Resolución de problemas
- Evaluaciones

Todos ellos formarían la que se podría llamar Enseñanza o Aprendizaje Asistido por Ordenador, ya que en todos los casos el ordenados "ayuda" al profesor y al alumno en su proceso de enseñanza y aprendizaje respectivamente, si bien se ha venido denominando Enseñanza Asistida por Ordenador (E.A.O.) a los usos tutoriales de los ordenadores. Es la traducción del inglés C.A.I. (Computer Aided Instruction)

En general podemos entender por E.A.O. el uso de ordenador para el aprendizaje/enseñanza de cualquier materia, aunque existen numerosas variantes de la misma e incluso aunque incluyan juegos educativos, simulaciones, ejercicios y prácticas como apoyo a alguna parte de la misma, la seguiremos denominando E.A.O.

Programas tutoriales

Los programas denominados tutoriales, han de tener las partes siguientes:

- Presentación del programa y de sus objetivos, lo que se pretende con el uso del programa.
- Menú de selección.

NUCLEO: ININ
UTI: 0027
PAG.: 2

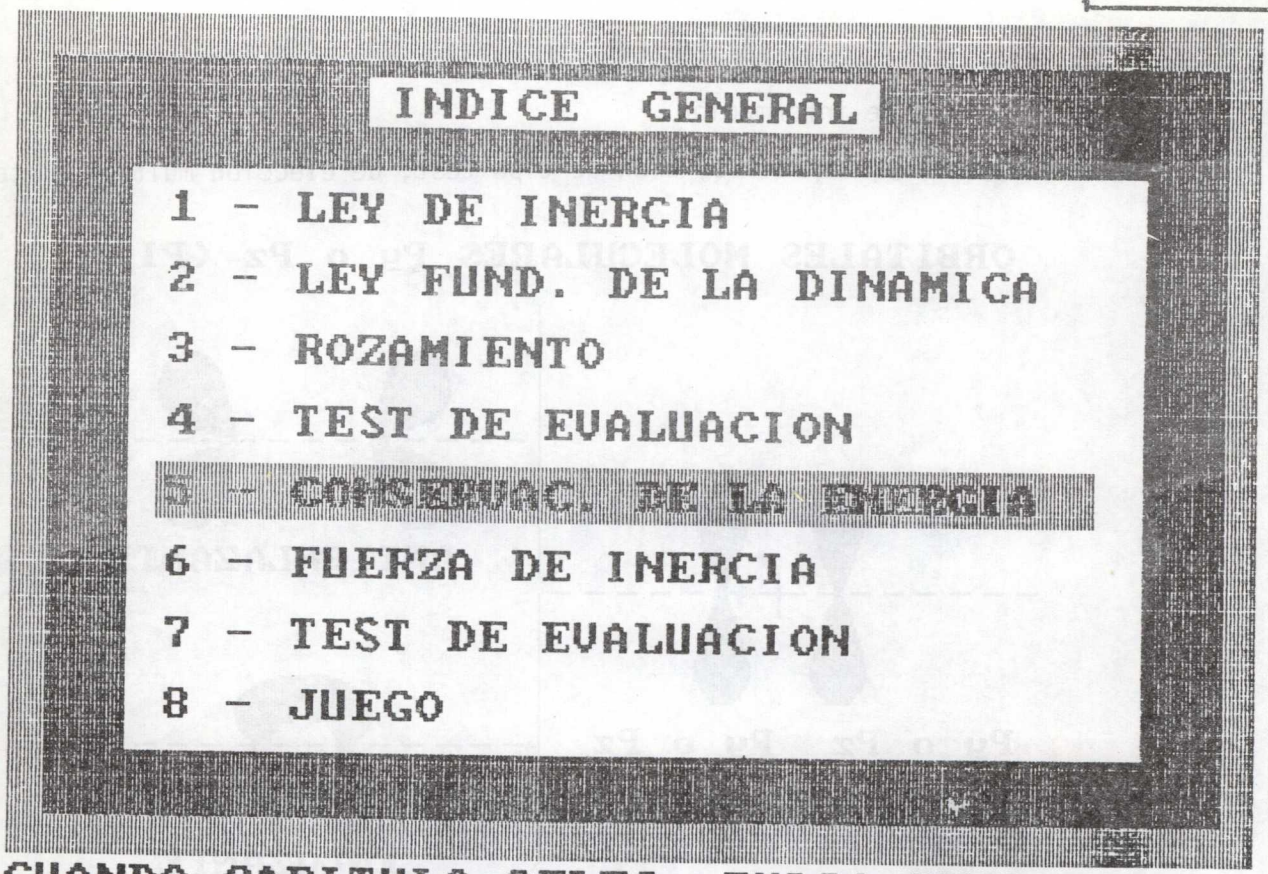


FIGURA 1

- Instrucciones de uso o para continuar

ROZAMIENTO, ENERGIA, FUERZA DE INERCIA

NIVEL 3º B.U.P. AUTOR Luis Alberto Calvo

Ten en cuenta las siguientes instruciones para usar el programa

1.- Sigue las instruciones que aparezcan en la ultima linea de la pantalla

2.- Contesta a las preguntas tecleando el dato y pulsando INNTRO

3.- Cumple las recomendaciones que aparezcan entre parentesis
Ej. si aparece (s/n) debes pulsar s ó n
Ej. si aparece (30-150) debes pulsar un nº comprendido entre 30 y 150

4.- Contesta con datos numericos o con letras, segun sea la pregunta

5.- Si tienes alguna duda consulta a tu profesor

P U L S A U N A T E C L A P A R A C O N T I N U A R

FIGURA 2

- Pantallas de información
- Preguntas y respuestas, bien de tipo test, de elección múltiple, etc.

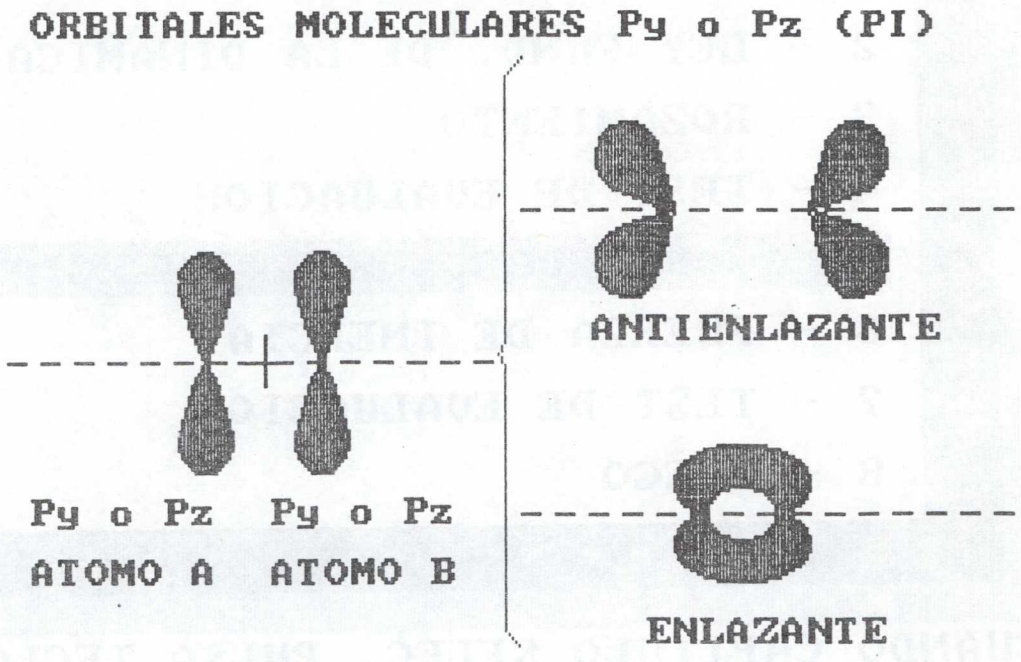


FIGURA 3

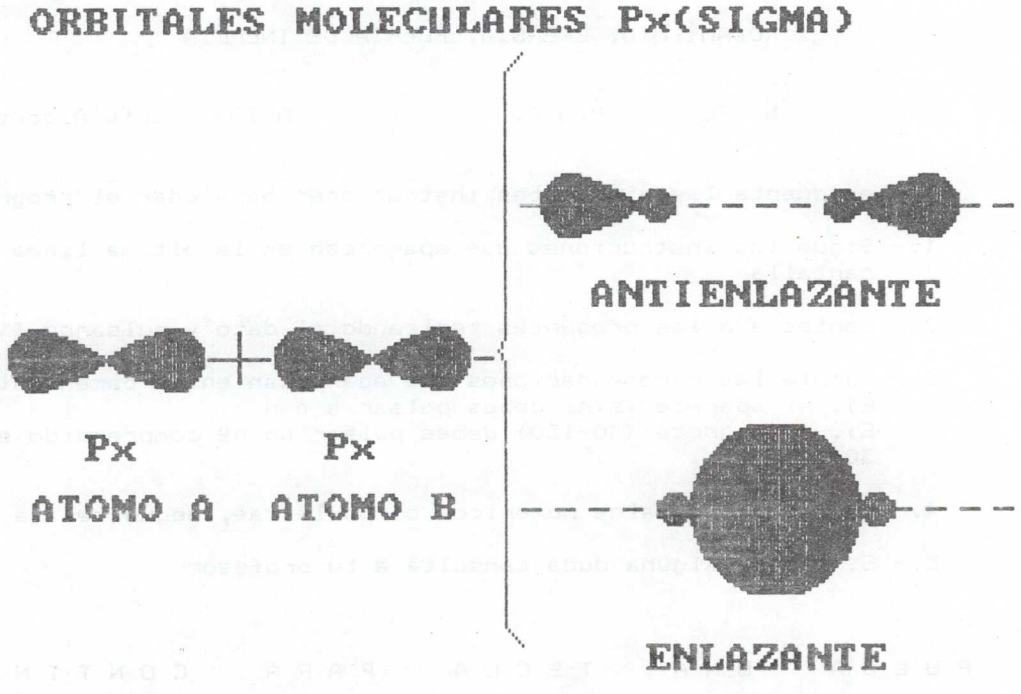


FIGURA 4

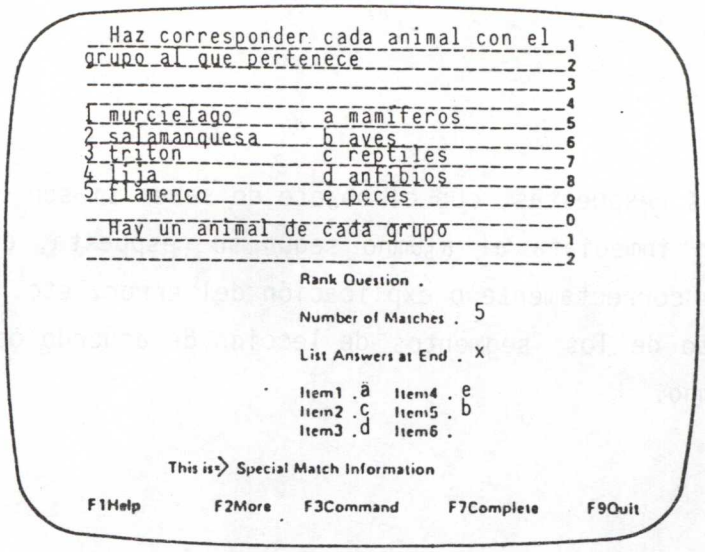


FIGURA 5

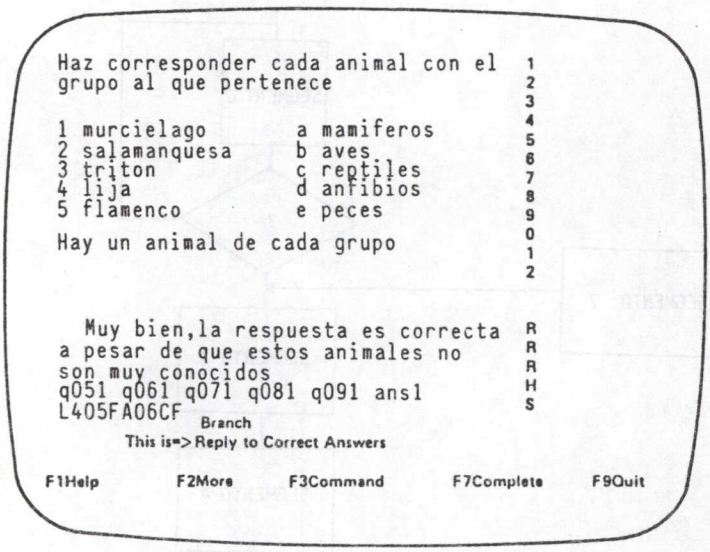


FIGURA 6

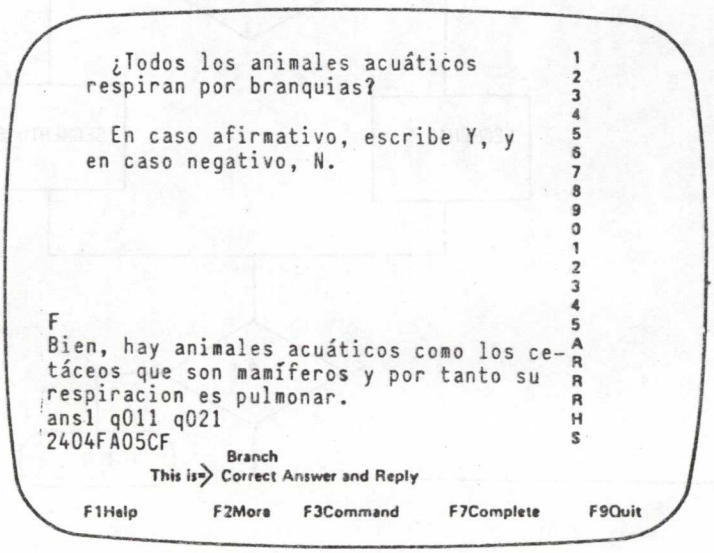
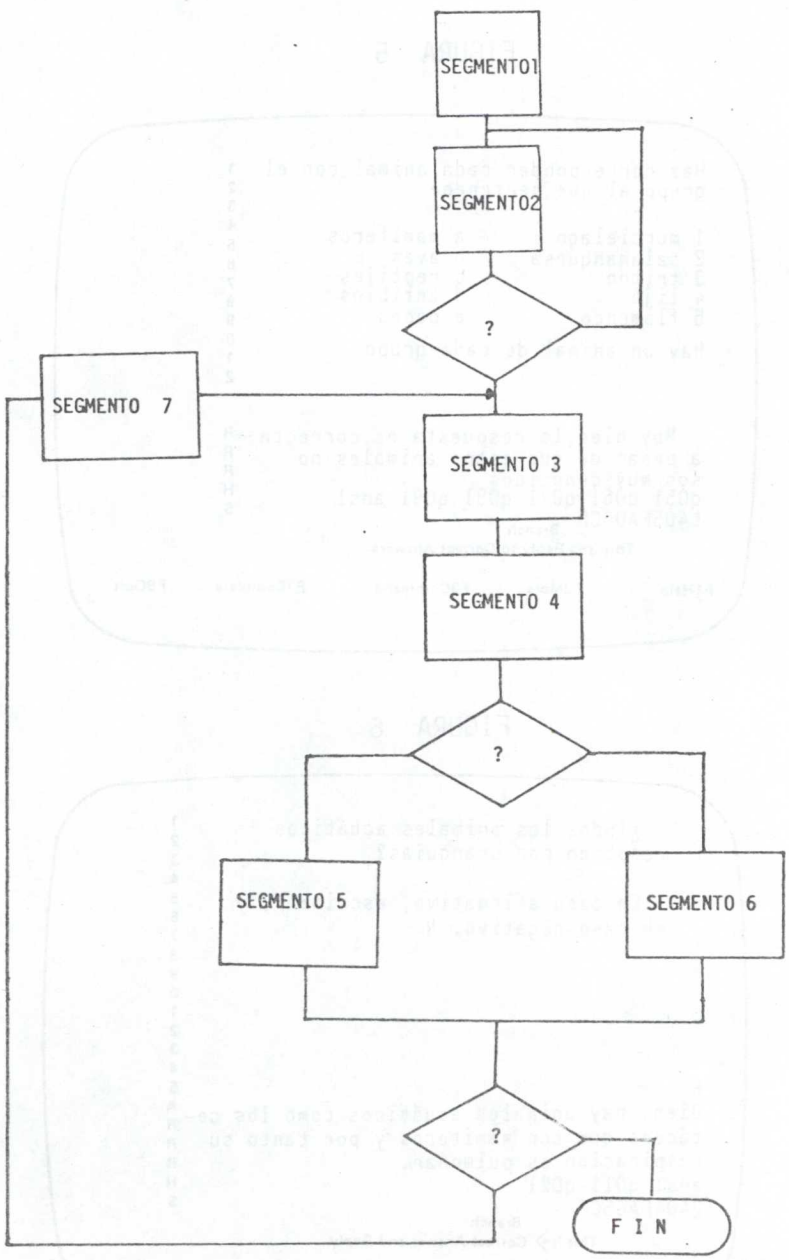


FIGURA 7

- Análisis de las respuestas, con el objeto de saber si son o no correctas
- Realimentación inmediata al alumno según su respuesta, es decir ayuda para responder correctamente o explicación del error, etc.
- Secuenciamiento de los segmentos de lección de acuerdo con las necesidades del alumno.



Diseño de lecciones

El diseño ha se realizarse en las siguientes fases:

1.- Definición de objetivos
2.- Selección de materiales adecuados
3.- Generación y aprendizaje de ideas.
4.- Elección de la o las metodologías a seguir
5.- Diseño global de la lección (con ayuda de un diagrama de flujo)
6.- Diseño de cada módulo hasta descomponerlo en pantallas.
7.- Programación de cada módulo de la lección.
8.- Unión de los módulos para formar la lección general.
9.- Evaluación de la calidad y efectividad de la lección.

El paso 1 es el más importante, ya que si los objetivos no se especifican con claridad no se puede llegar a un resultado aceptable.
Una vez elegidos los materiales y organizadas las ideas básicas, se pensará en la metodología a seguir: si se van a incluir juegos, si va a haber ejercitaría, si se van a producir problemas; test de entrada y salida, va a incluirse algún tipo de simulación, cómo va a interaccionar el estudiante con la lección.

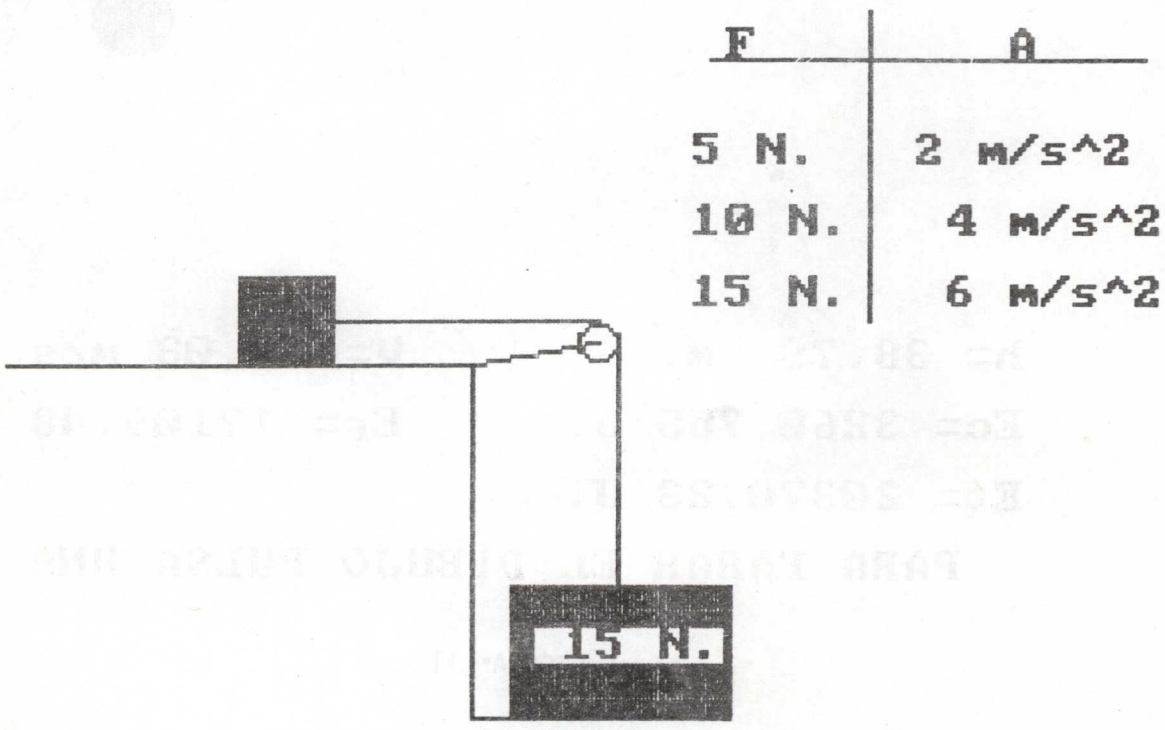
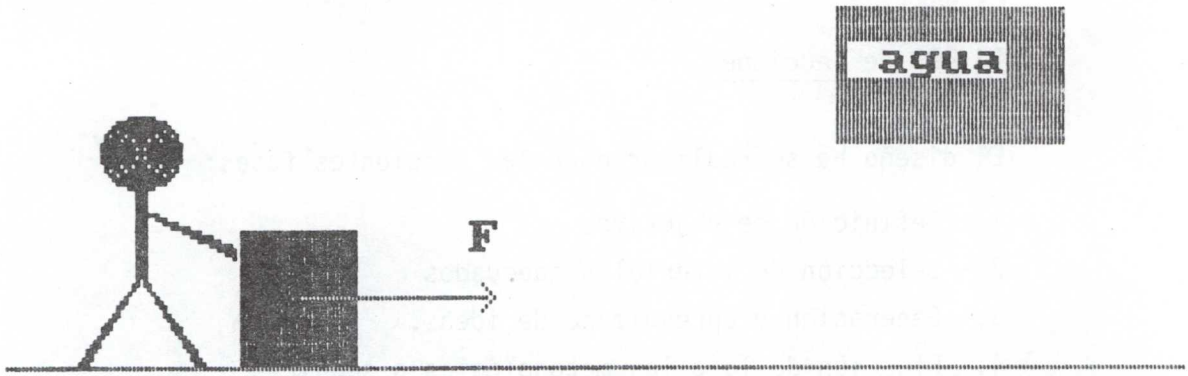
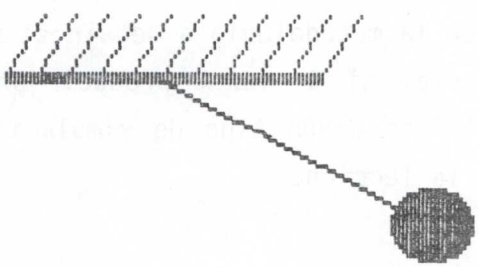


FIGURA 9



PARA CONTINUAR PULSA UNA TECLA

FIGURA 19



h= 38.79 m. V= 12.03 m/s

Ec= 3260.755 J. Ep= 17109.48 J.

Et= 20370.23 J.

PARA PARAR EL DIBUJO PULSA UNA TECLA

FIGURA 11

La realización del diseño de la lección (pasos 4 5) ha de hacerse con ayuda de diagramas de flujo generales y de detalle

Programación de la lección

La programación de la lección en un lenguaje entendible por la máquina ha de hacerse de acuerdo con las necesidades en un lenguaje adecuado. Por ejemplo si van a aparecer juegos o simulaciones que requieren gráficas, se realizará, al menos las partes correspondientes, en lenguajes con grandes posibilidades gráficas, puede incluso ser necesario recurrir a un programador profesional que trabaje con lenguaje ensamblador. Puede llevarse a cabo en lenguajes convencionales como BASIC, Logo u otros, o bien puede utilizarse un lenguaje de autor o sistema de autor diseñado específicamente para realizar enseñanza asistida por ordenador.

Lenguaje de Autor

Los lenguajes de autor son lenguajes de programación diseñados específicamente para la construcción de lecciones de E.A.O., de modo que permitan que un profesor sin conocimientos previos de Informática pueda desarrollar lecciones. Para ello suelen poseer instrucciones específicas para facilitar estas tareas que los lenguajes convencionales no poseen.
Veremos como ejemplo de lenguaje de autor uno de los más utilizados en el momento, el PILOT.

El lenguaje PILOT

PILOT (Programming Inquerg Learning o Teaching), es un lenguaje de autor, que en su versión más sofisticada llamada Super PILOT, ya es más que un lenguaje Autor, es un sistema de Autor, de los que hablaremos más adelante.

Este lenguaje permite desarrollar secuencias de preguntas con instrucciones del tipo:

```
r: ** lección de muestra **      (comentario)
t: Hablemos de tí  (texto que aparecerá en la pantalla)
* sexo              (etiqueta de la sección identificada por *)
```

```
t: ¿Eres hombre o mujer?        (texto que aparecerá en la pantalla)
a:                              (acepta la respuesta del estudiante)
m: mujer                         (busca una correspondencia o mat-
                                ching" con mujer)
jy: mujer                        (salto si coincide a la sección cuya
                                etiqueta es mujer)
m: hombre                       (busca una correspondencia o "matching")
jy: hombre                      (salto si coincide a la sección cuya
                                etiqueta es hombre)
t: Fijate en la contestación, tienes que ser hombre o mujer
                                (visualiza este texto)
t: pru e b a otra vez           (    "      "     "  )
j: sexo                         (salto al lugar cuya etiqueta es sexo)
* hombre
t: bien, eres un hombre         (texto visualizado)
t: ¿Cómo te llamas?             (  "       "      )
.........
.....
e:                               (fin de la lección)

* mujer
t: bien, eres una mujer          (texto visualizado)
t: ¿Cómo te llamas?              (  "       "     )
.....
.....
e:                               (fin de la lección)
```

Naturalmente, con un programa como el anterior se puede hacer en BASIC por citar un lenguaje común, pero existen posibilidades de análisis de respuestas más elaboradas con modificadores del tipo que se muestra a continuación.

```
t: ¿Cuál es tu color favorito?
    rojo, verde o azúl.
a:
mj:  rojo .....(salto si no ha contestado rojo al
                siguiente "match")
```

```
t:  las rosas son rojas, a mi también
    me gusta el rojo
j: parte 2                          (salto a parte2)
mj: azul .....                      (salto si no ha contestadoal siguien-
                                    te "match")
t: el mar es azul, es un color bonito
j: parte 2                          (salto a parte 2)
m: verde
ty: la hierba es verde, y además es (escriba el texto si la respuesta ha
    el color de la esperanza         sido verde)
j: parte 2

* parte 2                           (etiqueta de la sección parte 2)
  .
  .
  .
  .
```

otros ejemplos serían

```
m: % 9% !%NUEVE%
```

comodín general

comodín general

esta instrucción realiza una correspondencia con NUEVE o 9 o con SON NUEVE etc.

```
m: V*N*S
```

comodín de 1 carácter

(realiza una correspondencia en VENUS o con cualquier palabra que tenga las letras 1,3 y 5 iguales a VN y S respectivamente)

Es un lenguaje que además de tener instrucciones específicas de las que son una muestra los ejemplos anteriores, posee las características más comunes de BASIC y como manejo de "strings" (cadena de caracteres), listas y tablas, ficheros, etc.

Sistemas de Autor

Los sistemas de autor son un conjunto de programas que permiten una programación sencilla de lecciones de E.A.O., de forma similar a la de autor, pero a diferencia de éstos, tienen estandarizadas las funciones que pueden realizar, que son de mayor nivel (macros) que en el caso de los lenguajes, pero sus posibilidades puede ser limitadas.

Las funciones estandarizadas que admiten los sistemas de Autor son normalmente:

- Definición de pantallas de texto, lo que permiten introducir información de textos con el formato desarrollado.
- Definición de pantallas de pregunta/respuesta, que permite presentar una o más preguntas y permite programar ayudas, contestación a respuestas incorrectas previstas por el profesor, proveer elección entre múltiples respuestas.
- Definición de pantallas de tipo "test", en las que se debe hacer corresponder unas preguntas enunciadas con unas respuestas.
- Registro de los progresos del alumno, para facilitar la gestión del programa en las aulas.
- Encadenamiento de pantallas, combinando la secuenciación de presentación según las respuestas del alumno.

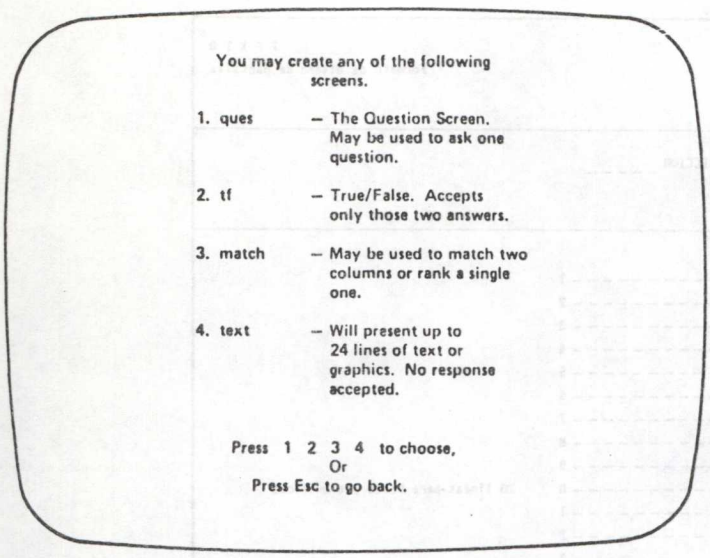


FIGURA 12

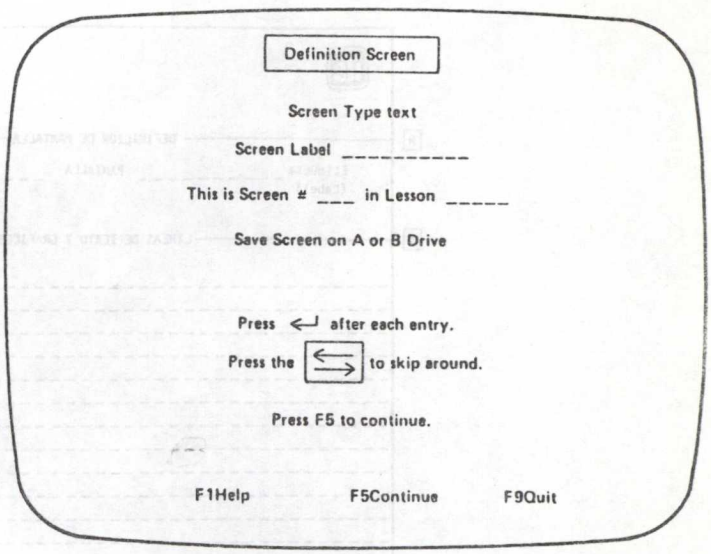


FIGURA 13

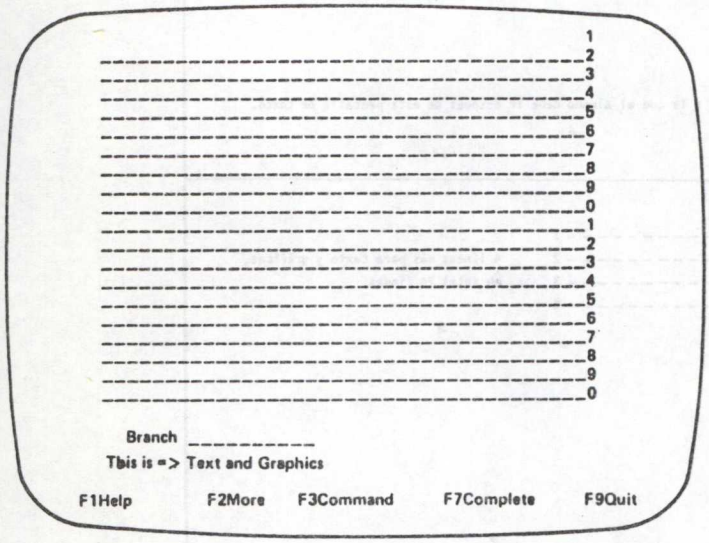


FIGURA 14

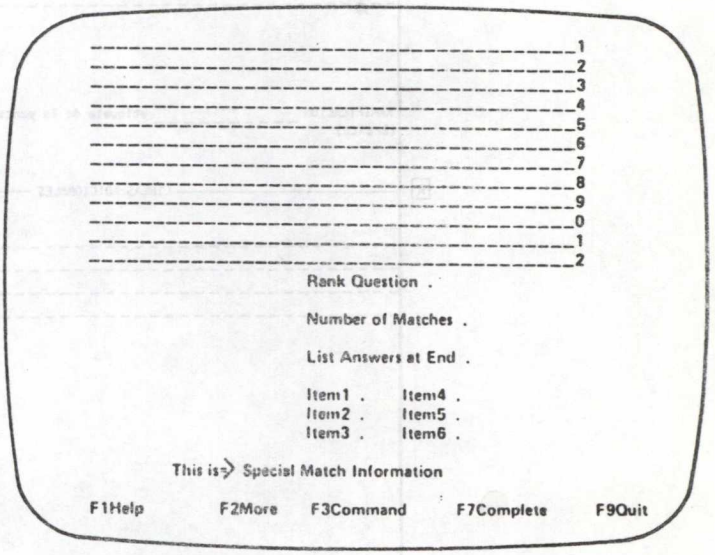


FIGURA 15

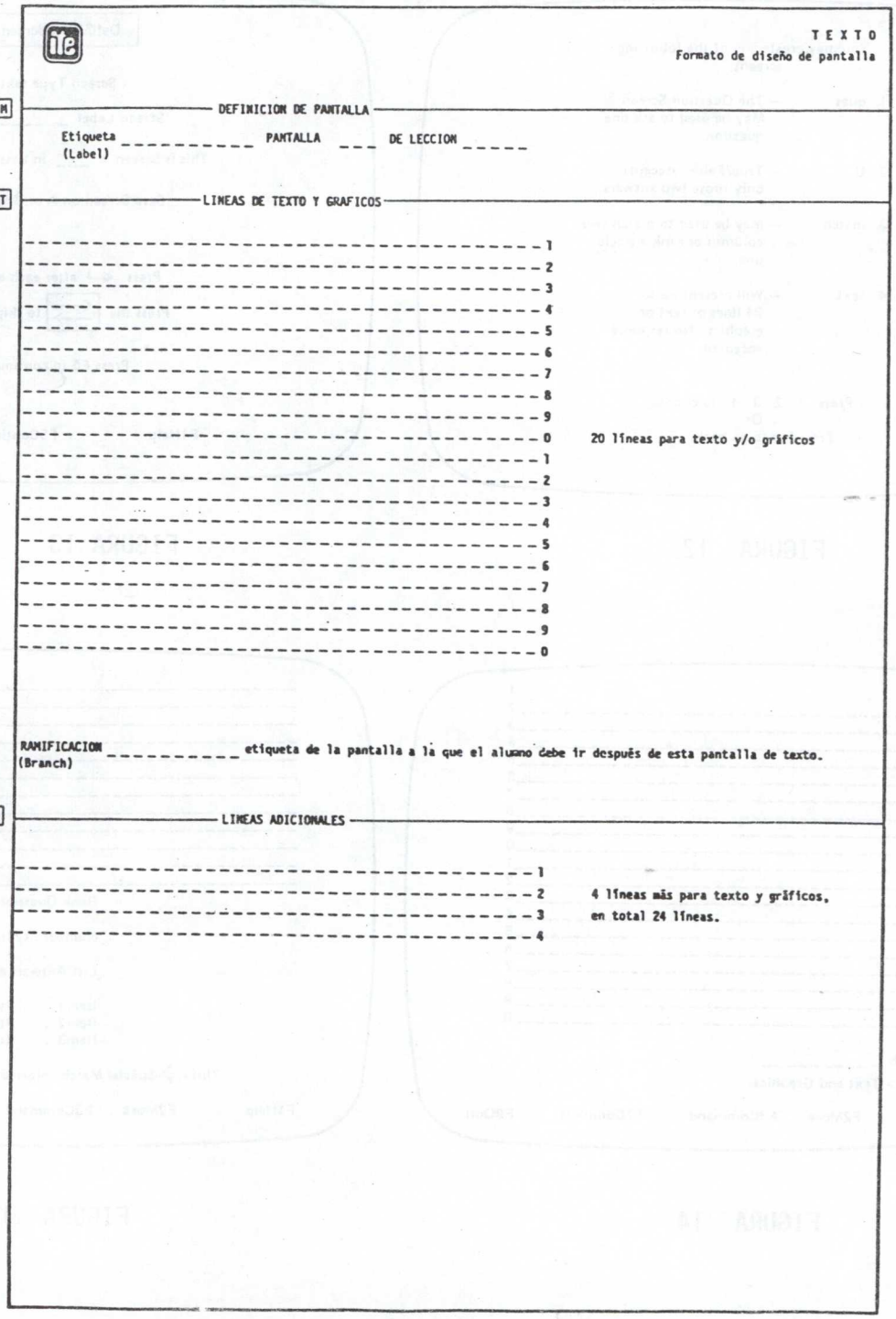

T E X T O
Formato de diseño de pantalla
M
DEFINICION DE PANTALLA
Etiqueta (Label) _____ PANTALLA ___ DE LECCION _____
T
LINEAS DE TEXTO Y GRAFICOS
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
20 líneas para texto y/o gráficos
RAMIFICACION (Branch) _____ etiqueta de la pantalla a la que el alumno debe ir después de esta pantalla de texto.
A
LINEAS ADICIONALES
1
2
3
4
4 líneas más para texto y gráficos, en total 24 líneas.

FIGURA 16

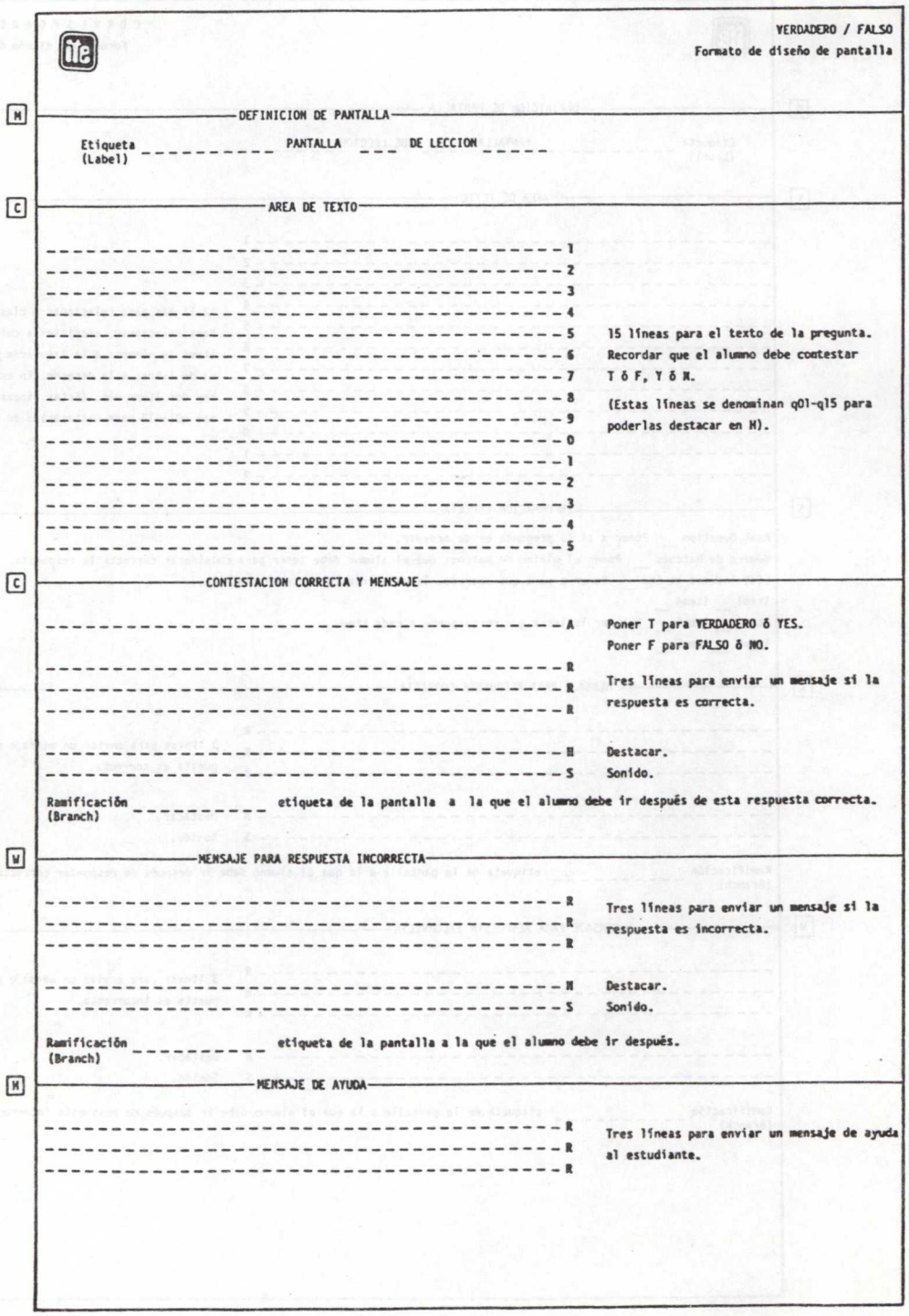

VERDADERO / FALSO
Formato de diseño de pantalla
M
DEFINICION DE PANTALLA
Etiqueta (Label) _____ PANTALLA ___ DE LECCION _____
C
AREA DE TEXTO
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
15 líneas para el texto de la pregunta.
Recordar que el alumno debe contestar T ó F, Y ó N.
(Estas líneas se denominan q01-q15 para poderlas destacar en M).
C
CONTESTACION CORRECTA Y MENSAJE
A
R
R
R
H
S
Poner T para VERDADERO ó YES.
Poner F para FALSO ó NO.
Tres líneas para enviar un mensaje si la respuesta es correcta.
Destacar.
Sonido.
Ramificación (Branch) _____ etiqueta de la pantalla a la que el alumno debe ir después de esta respuesta correcta.
W
MENSAJE PARA RESPUESTA INCORRECTA
R
R
R
H
S
Tres líneas para enviar un mensaje si la respuesta es incorrecta.
Destacar.
Sonido.
Ramificación (Branch) _____ etiqueta de la pantalla a la que el alumno debe ir después.
H
MENSAJE DE AYUDA
R
R
R
Tres líneas para enviar un mensaje de ayuda al estudiante.

FIGURA 17

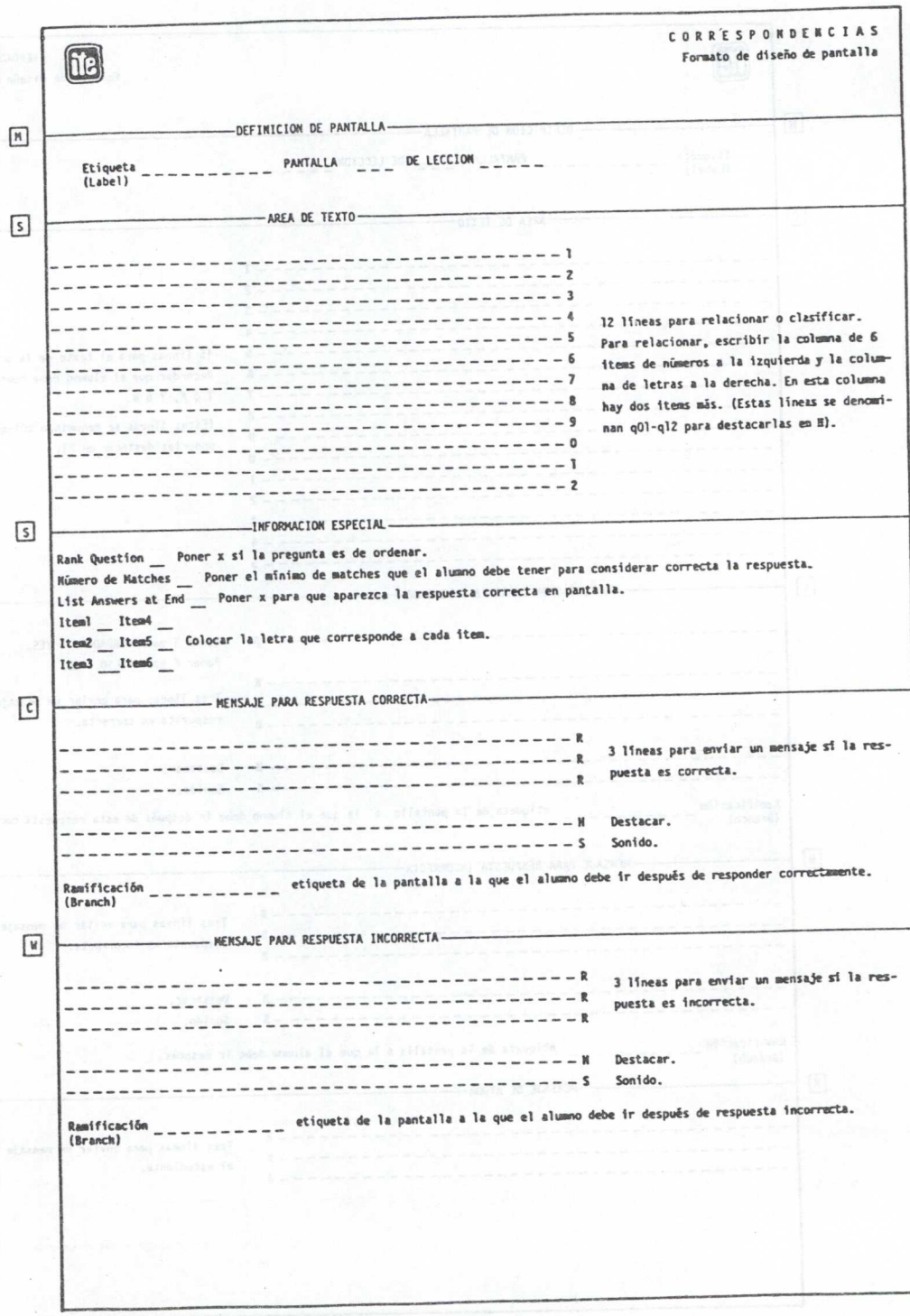

CORRESPONDENCIAS
Formato de diseño de pantalla
M
DEFINICION DE PANTALLA
Etiqueta (Label) _____ PANTALLA ___ DE LECCION _____
S
AREA DE TEXTO
1
2
3
4
5
6
7
8
9
0
1
2
12 líneas para relacionar o clasificar. Para relacionar, escribir la columna de 6 items de números a la izquierda y la columna de letras a la derecha. En esta columna hay dos items más. (Estas líneas se denominan q01-q12 para destacarlas en H).
S
INFORMACION ESPECIAL
Rank Question __ Poner x si la pregunta es de ordenar.
Número de Matches __ Poner el mínimo de matches que el alumno debe tener para considerar correcta la respuesta.
List Answers at End __ Poner x para que aparezca la respuesta correcta en pantalla.
Item1 __ Item4 __
Item2 __ Item5 __ Colocar la letra que corresponde a cada item.
Item3 __ Item6 __
C
MENSAJE PARA RESPUESTA CORRECTA
R
R
R
3 líneas para enviar un mensaje si la respuesta es correcta.
H Destacar.
S Sonido.
Ramificación (Branch) _____ etiqueta de la pantalla a la que el alumno debe ir después de responder correctamente.
W
MENSAJE PARA RESPUESTA INCORRECTA
R
R
R
3 líneas para enviar un mensaje si la respuesta es incorrecta.
H Destacar.
S Sonido.
Ramificación (Branch) _____ etiqueta de la pantalla a la que el alumno debe ir después de respuesta incorrecta.

FIGURA 18

PREGUNTA
Formato de diseño de pantalla (página 1)

DEFINICION DE PANTALLA

Etiqueta (Label) _____ PANTALLA # ___ DE LECCION _____

¿Se puntua la pregunta? (Y/N) Pulsar N si no quiere tener en cuenta la respuesta del alumno en la calificación de la lección.

[M] AREA DE TEXTO

_____ 1
_____ 2
_____ 3
_____ 4
_____ 5
_____ 6
_____ 7
_____ 8
_____ 9
_____ 0
_____ 1
_____ 2
_____ 3
_____ 4
_____ 5

15 líneas de texto para la pregunta.

(Estas líneas se denominan q01-q15 para destacarlas en H).

[C] RESPUESTA CORRECTA Y MENSAJE

_____ A Escribir aquí la lista de respuesta/s correcta/s con espacio intermedio.

_____ R
_____ R 3 líneas para el mensaje correspondiente.
_____ R

_____ H Destacar.
_____ S Sonido.

Ramificación (Branch) _____ la pantalla a la que el alumno debe ir después.

Para salir (To exit) __ Poner x para enviar al estudiante al mensaje de salida. Esto hace que se ignore la ramificación en caso de respuesta correcta.

[E] RESPUESTAS CORRECTAS ADICIONALES

_____ A
_____ A
_____ A
_____ A

4 líneas más para otras respuestas correctas.

[H1] MENSAJE DE AYUDA #1

_____ R
_____ R 3 líneas para ayuda al estudiante.
_____ R

[H2] MENSAJE DE AYUDA #2

_____ R
_____ R 3 líneas más de ayuda al estudiante.
_____ R

[U1] MENSAJE PARA LA PRIMERA RESPUESTA INCORRECTA

_____ R
_____ R 3 líneas para mensaje a una respuesta incorrecta.
_____ R

[U2] MENSAJE PARA LA SEGUNDA RESPUESTA INCORRECTA

_____ R
_____ R 3 líneas para mensaje a una respuesta incorrecta.
_____ R

FIGURA 19

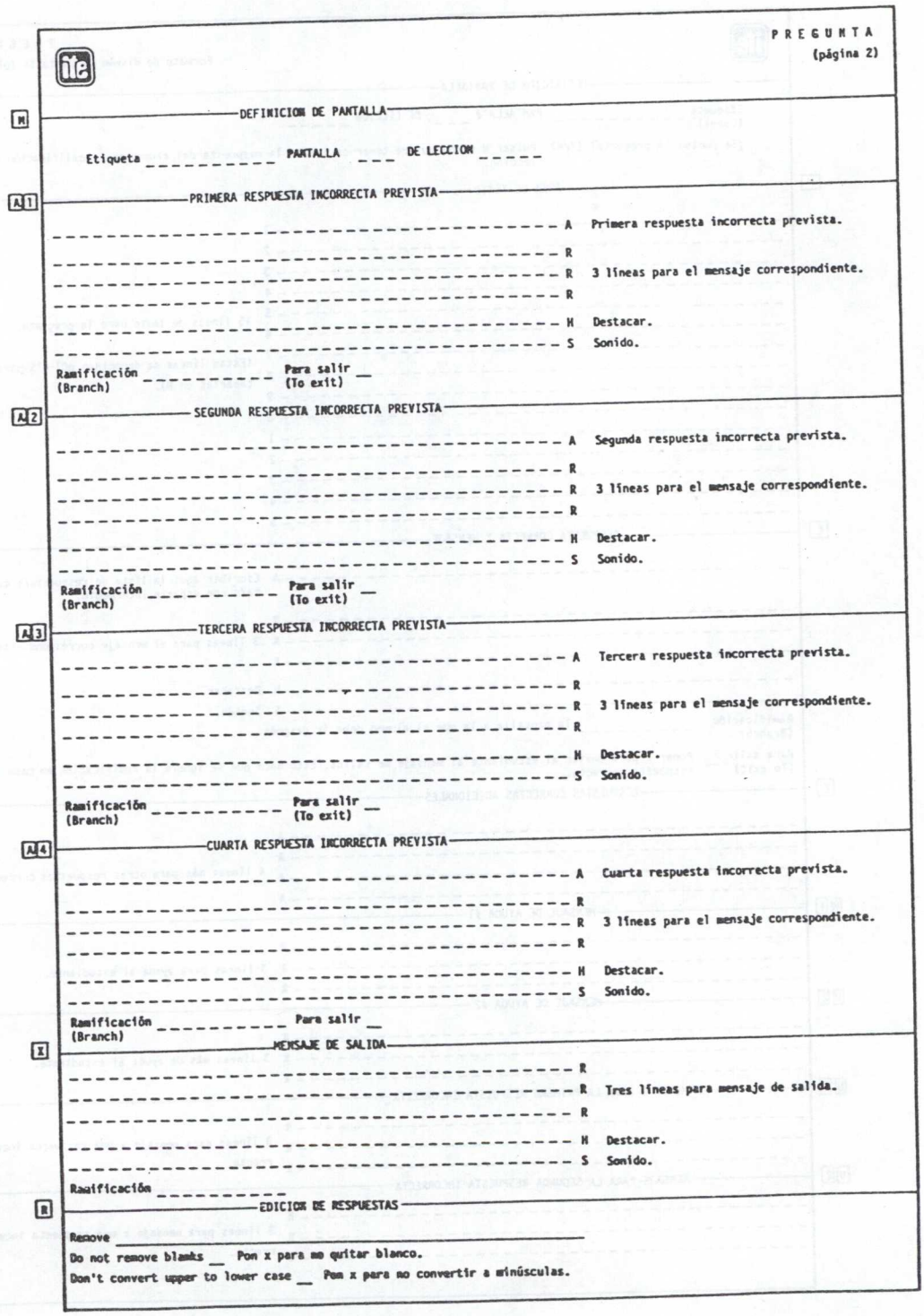

PREGUNTA
(página 2)
M
DEFINICION DE PANTALLA
Etiqueta _____ PANTALLA ___ DE LECCION _____
A1
PRIMERA RESPUESTA INCORRECTA PREVISTA
A Primera respuesta incorrecta prevista.
R
R 3 líneas para el mensaje correspondiente.
R
H Destacar.
S Sonido.
Ramificación (Branch) _____ Para salir (To exit) __
A2
SEGUNDA RESPUESTA INCORRECTA PREVISTA
A Segunda respuesta incorrecta prevista.
R
R 3 líneas para el mensaje correspondiente.
R
H Destacar.
S Sonido.
Ramificación (Branch) _____ Para salir (To exit) __
A3
TERCERA RESPUESTA INCORRECTA PREVISTA
A Tercera respuesta incorrecta prevista.
R
R 3 líneas para el mensaje correspondiente.
R
H Destacar.
S Sonido.
Ramificación (Branch) _____ Para salir (To exit) __
A4
CUARTA RESPUESTA INCORRECTA PREVISTA
A Cuarta respuesta incorrecta prevista.
R
R 3 líneas para el mensaje correspondiente.
R
H Destacar.
S Sonido.
Ramificación (Branch) _____ Para salir __
I
MENSAJE DE SALIDA
R
R Tres líneas para mensaje de salida.
R
H Destacar.
S Sonido.
Ramificación _____
R
EDICION DE RESPUESTAS
Remove _____
Do not remove blanks __ Pon x para no quitar blanco.
Don't convert upper to lower case __ Pon x para no convertir a minúsculas.

FIGURA 20

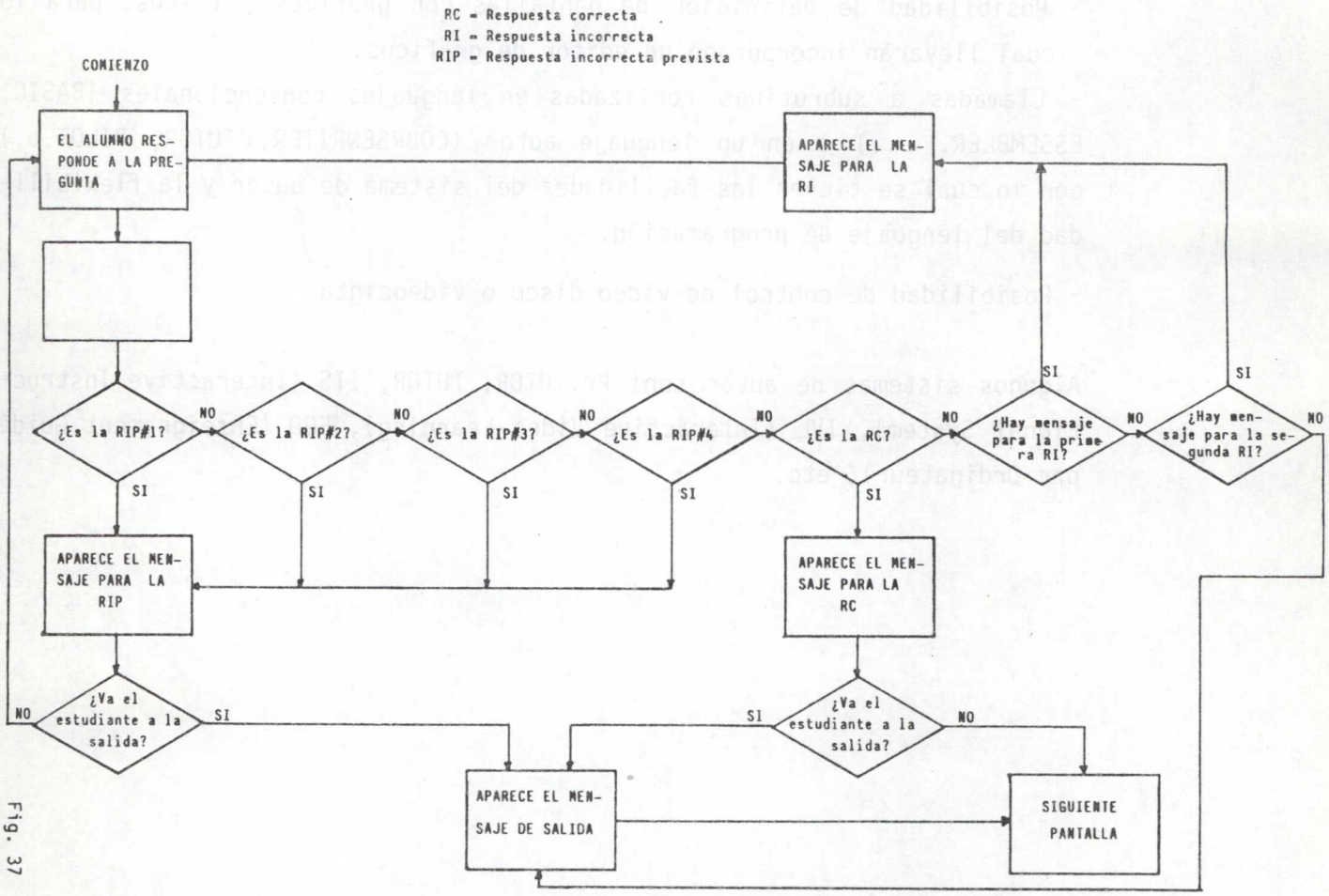


Fig. 37

FIGURA 21

Además de las características anteriores, los sistemas de autor suelen poseer otras facilidades como:

- Posibilidad de definición de pantallas con gráficos y textos, para lo cual llevarán incorporado un editor de gráficos.
- Llamadas a subrutinas realizadas en lenguajes convencionales (BASIC, ESSEMBLER, ...) o en un lenguaje autor (COURSEWRITER, TUTOR, PILOT...) con lo cual se tienen las facilidades del sistema de autor y la flexibilidad del lenguaje de programación.
- Posibilidad de control de vídeo disco o videocinta

Algunos sistemas de autor son: Pr.TUTOR, TUTOR, IIS (Interactive Instructional System), IVL (Interactive Video Learning), EGO (Enseignement Guidé par Ordinateur), etc.

REVISION Y EVALUACION DE PROGRAMAS DE ORDENADOR PARA LA ENSEÑANZA

CONSIDERACIONES GENERALES

En la actualidad se están produciendo constantemente, y a un ritmo cada vez mayor, gran cantidad de programas de ordenador para su uso en la enseñanza. Hay que constatar que, de la totalidad del material desarrollado, hay un tanto por ciento elevado de programas de escaso valor, lo que hace necesario disponer de instrumentos y criterios adecuados para revisar y evaluar estos programas.

La mejor forma de evaluar un programa de ordenador sería la de realizar una investigación empírica sobre su efectividad, que permitiera responder a preguntas como las siguientes: ¿Los estudiantes adquieren los conocimientos o capacidades deseados? ¿Las adquieren mejor que con otros medios?, etc.

Este tipo de evaluación sólo es posible hacerla en el desarrollo de unidades experimentales en las que es conveniente realizar un doble proceso de evaluación: en primer lugar, con una experimentación llevada a cabo por el propio equipo que ha desarrollado el programa, y más tarde con un proceso de evaluación realizado por especialistas ajenos al desarrollo del programa.

Para la selección de material ya desarrollado, y para tener en cuenta en la construcción del mismo unos criterios de valoración que influyan en el proceso de su desarrollo, es conveniente disponer de un formato de revisión y evaluación que recoja las líneas maestras y criterios que sirvan de guía.

Los puntos fundamentales a considerar se pueden agrupar en los siguientes bloques:

- Operación técnica del programa, en la que se ha de considerar todos aquellos aspectos funcionales como menús adecuados, si están previstas todas las posibilidades, si no hay errores de funcionamiento, etc.

- Contenidos, en los que se considerará si son adecuados y se adaptan al curriculum actual, si tienen calidad técnica, si están actualizados, etc.

- Presentación, de forma que se considere si es fácil de lectura para el nivel de los alumnos a los que va dirigido, las pantallas están diseñadas de forma atractiva y sin distractores, con gráficas y sonido de apoyo, si es autoexplicativo, etc.

- Documentación en la que se ha de valorar la calidad de la guía de operación del programa, en la que se estudiará si contiene ejemplos de muestra adecuados, aplicaciones, ideas para su uso en clase, etc.

- Aspectos didácticos en los que ha de tener en cuenta si el enfoque pedagógico que subyace es adecuado al seguido habitualmente por el profesor, si se consiguen los objetivos educativos perseguidos, si posibilita un control por parte del estudiante, permitiendo al mismo cambiar de niveles de dificultad, interaccionar con el programa, dirigirse al módulo de lección que desee, si permite modificar al profesor alguna parte del programa, si la realimentación es buena, si es motivadora, si la enseñanza está individualizada, si se aprovechan adecuadamente las posibilidades del ordenador, si hay una evaluación final, si hay pruebas de entrada, si admite respuestas libres, si tutoriza adecuadamente el proceso de aprendizaje, si favorece un estilo de trabajo de cooperación o por el contrario es competitivo, si estimula la creatividad, etc.

A la hora de diseñar y de evaluar el Logical educativo debemos tener en cuenta los distintos fallos más comunes que suelen detectarse en los programas educativos que recoge Alfred Bork y que enumeramos a continuación:

- No se hace un buen uso de la capacidad de interacción del ordenador.
- El proceso de enseñanza no está individualizado.
- El texto es excesivo.
- Contenidos extracurriculares.
- Excesivas preguntas de elección múltiple.
- Se utilizan gráficos innecesarios.
- Se utilizan las pantallas del ordenador como si fueran páginas de libros.
- El material, aún siendo atractivo, es muchas veces escasamente educativo.

- Los juegos, aún siendo divertidos, no son, en absoluto, educativos.
- Las instrucciones para el manejo del programa no son adecuadas, son difíciles de llamar durante la ejecución del programa y no se recuerdan fácilmente.
- El material auxiliar necesario (diapositivas, video, etc.) es excesivo.
- La presentación, aunque sea buena, está fuera de contexto.
- Se utilizan materiales que sólo interesan al profesor y no a los alumnos.

Para realizar una visión crítica del Logical educativo que incluya todos los aspectos anteriormente citados, es preciso que el profesor lleve a cabo un proceso de ejecución del programa, valorándolo en el doble aspecto de usuario-alumno, juzgando el papel que puedan jugar los diferentes alumnos en el programa, respondiendo incorrectamente para estudiar el comportamiento del programa en todas las ramificaciones posibles del mismo, y de usuario-profesor para poder valorar la calidad técnica, el enfoque educativo y demás aspectos enumerados anteriormente.

BIBLIOGRAFIA

BITTER, Gary G. and Ruth A. CAMUSE: **Using a microcomputer in the classroom.** Reston (Virginia), Reston Publishing Co., A Prentice Hall Co., 1984.

LATHORP, Ann and Bobby GOODSON: **Courseware in the Classroom.** Addison Wesley, 1983.

WALLACE, Joy and Rose RAYMOND M.: "A hard look and software: What to examine and evaluate (With an evaluation form). **Educational Tecnology**, october 1984.

CRISWELL, Eleanor L. and Robert W. SWEZEY: "Behavioral learning theory-based computer courseware evaluation". **Educational Technology**, November 1984

Microcomputer software evaluation instrument. National Science Teachers Association, 1984.

BORK, Alfred: **Compendium of bad but common practices in computed based learning.** En "Computers and Learning: A compendium of papers", 1983.

BORK, Alfred: **Personal Computers for education.** Harper & Row, 1985.

ALESSI, Stephen M. and Stanley R. TROLLIP: **Computer-Based Instruction: Methods and Development.** Prentice-Hall Inc., 1985.

FORMATO DE REVISION Y EVALUACION DE PROGRAMAS
DE ORDENADOR PARA LA ENSEÑANZA

DESCRIPCION: RESUMEN GENERAL

Nombre del programa:

Autor/es _____

Editor _____

Fecha _____ Precio _____
(Incluyendo disco seguridad)

Ordenadores para los que está disponible _____

Memoria necesaria _____ K

Soporte físico: ◯ Cassette ◯ Diskette (___") ◯ Cartucho

Periféricos y Hardware necesarios:

- Unidades de disco: ◯ una ◯ dos
- Impresora ◯
- Monitor color ◯
- Interface cassette ◯
- Otros periféricos: _____
  (Sintetizador de voz, mandos, lápiz óptico, _____
  plotter, teclados especiales,...) _____
- Tarjetas extras _____
  (Adaptador de color/gráficos, alta resolución,...) _____

Logical necesario:

- Sistema Operativo _____
- Otros programas _____

Material adicional requerido _____
_____

Lenguaje en el que se ha desarrollado _____

Documentación:

- Guía del profesor ◯
- Guía del alumno ◯
- Instrucciones ◯
- Aplicaciones ◯
- Programa muestra ◯

Tiempo medio de ejecución _____

Perteneciente a la serie _____

Programas anteriores _____

Programas siguientes _____

Nivel Educativo _____

Area temática _____

Modalidad:

- Ejercicios y prácticas ○
- Simulación ○
- Juego educativo ○
- Tutorial ○
- Bases de datos ○
- Resolución de problemas ○
- Otros:

_____

_____

Conocimientos previos necesarios (Prerrequisitos) _____

_____

_____

Uso:

- Individual ○
- Pequeño Grupo ○
- Clase ○
- Recuperación ○
- Enseñanza Reglada ○
- Perfeccionamiento ○

Objetivos principales del programa: _____

_____

_____

_____

_____

Indice extractado de contenidos

Descripción del programa

CARACTERISTICAS TECNICAS: OPERACION Y PRESENTACION

	V	F
El programa no tiene errores de ejecución.	○	○
El programa indica con claridad si está esperando una respuesta y de qué tipo.	○	○
Está previsto que el programa no se pueda abortar por uso indebido de las teclas.	○	○
Es correcta la temporalización del programa.	○	○
Se puede salir del programa sin completarlo.	○	○
Se pueden saltar las instrucciones de funcionamiento.	○	○
La presentación de las pantallas son claras y con buen formato.	○	○
Es de lectura fácil para el nivel de los alumnos a los que va dirigido.	○	○
Es autoexplicativo.	○	○
Permite la elección fácil de opciones, por medio de "menús" claros y bien presentados.	○	○
Utiliza gráficos, sonido, animación de forma adecuada.	○	○

CONTENIDOS, OBJETIVOS Y DOCUMENTACION

	V	F
Los contenidos del programa se adaptan al curriculum actual.	○	○
El programa no posee errores de contenido.	○	○
El contenido tiene calidad técnica y está actualizado.	○	○
Los objetivos del programa están formulados con claridad.	○	○
Los objetivos perseguidos son de interés.	○	○
Se alcanzan los objetivos educativos previstos.	○	○
El manual del profesor es útil: contiene sugerencias de uso, aplicaciones, ejemplos de muestra, etc.	○	○
El manual del alumno es claro y de uso sencillo.	○	○

ASPECTOS DIDACTICOS

	V	F
El enfoque pedagógico es conforme a las técnicas actuales.	○	○
El profesor puede modificar alguna parte del programa.	○	○
El profesor puede determinar los niveles de cada estudiante dentro del programa.	○	○
El programa permite una enseñanza individualizada.	○	○
Incluye pruebas de entrada.	○	○
Incluye evaluación final.	○	○
Admite respuestas libres.	○	○
Tutoriza adecuadamente el aprendizaje.	○	○
Registra las puntuaciones de los alumnos.	○	○
Si incluye ejercicio, realiza una generación aleatoria de problemas.	○	○
Si es una simulación, el program proporciona una imagen adecuada de la realidad.	○	○
Se puede utilizar el programa sin conocimientos previos sobre informática.	○	○
El alumno puede acceder libremente a las distintas partes del programa.	○	○
El alumno puede cambiar los niveles de dificultad del programa.	○	○
El programa está modularizado adecuadamente.	○	○
Genera mensajes motivadores y de refuerzo como contestación a las respuestas tanto correctas como incorrectas del alumno.	○	○
Estimula la creatividad del alumno.	○	○
Favorece un estilo de trabajo cooperativo y, por tanto, no es competitivo.	○	○
Se aprovechan adecuadamente las posibilidades del ordenador.	○	○
El uso del ordenador aporta elementos positivos y nuevos para los fines perseguidos respecto del uso de otros medios.	○	○

VALORACION RESUMIDA

	MUY BUENO	BUENO	ACEPTABLE	MALO
OPERACION	○	○	○	○
PRESENTACION	○	○	○	○
DOCUMENTACION	○	○	○	○
CONTENIDOS	○	○	○	○
ASPECTOS DIDACTICOS	○	○	○	○

Aspectos más positivos del programa

Aspectos más negativos del programa

Recomendaciones de uso

EJEMPLOS DE PROGRAMAS DE E.A.O.

(Documentación anexa al disco)

0 PRESENTACION

El disco "ININ" es autoarrancable en Computec.

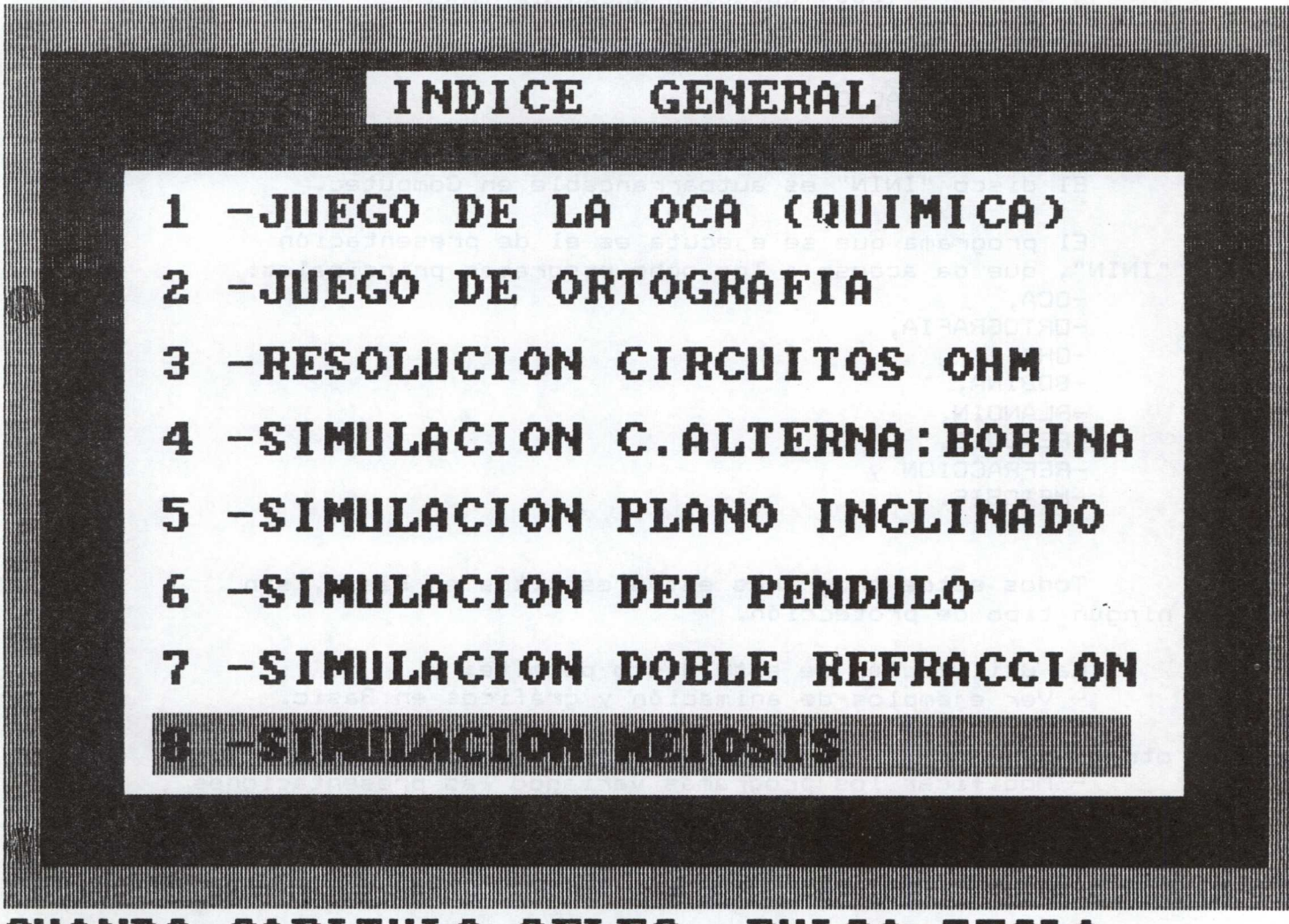
El programa que se ejecuta es el de presentación "ININ", que da acceso a los ocho programas principales:

-OCA,
-ORTOGRAFIA,
-OHM,
-BOBINA,
-PLANOIN,
-PENDULO,
-REFRACCION y
-MEIOSIS.

Todos estos programas están escritos en Basic, sin ningún tipo de protección.

La utilización de este disco permite:

- Ver ejemplos de animación y gráficos en Basic.
- Sacar los listados para utilizar como subrutinas de otros programas.
- Modificar los programas variando las presentaciones o mejorándolos.
- Presentar simulaciones y juegos como punto de partida para discutir sus posibilidades en la enseñanza.



CUANDO CAPITULO SELEC. PULSA TECLA

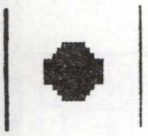
1 JUEGO DE LA OCA (QUIMICA)

Basandose en el conocido juego de la OCA se le formulan al alumno una serie de preguntas de Química.

El juego comienza con la presentación de un tablero de la OCA.

Se simula el lanzamiento de un dado y se avanza según sea su resultado.

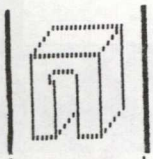
El significado de cada casilla es el siguiente:



De átomo a átomo y sumas 10 puntos.



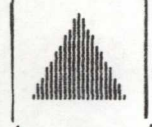
Pregunta. Sumas o restas tantos puntos como indica la ruleta , según si aciertas o no la pregunta.



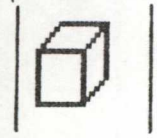
De puente a puente por la corriente.



Restas los puntos que obtienes con la ruleta.



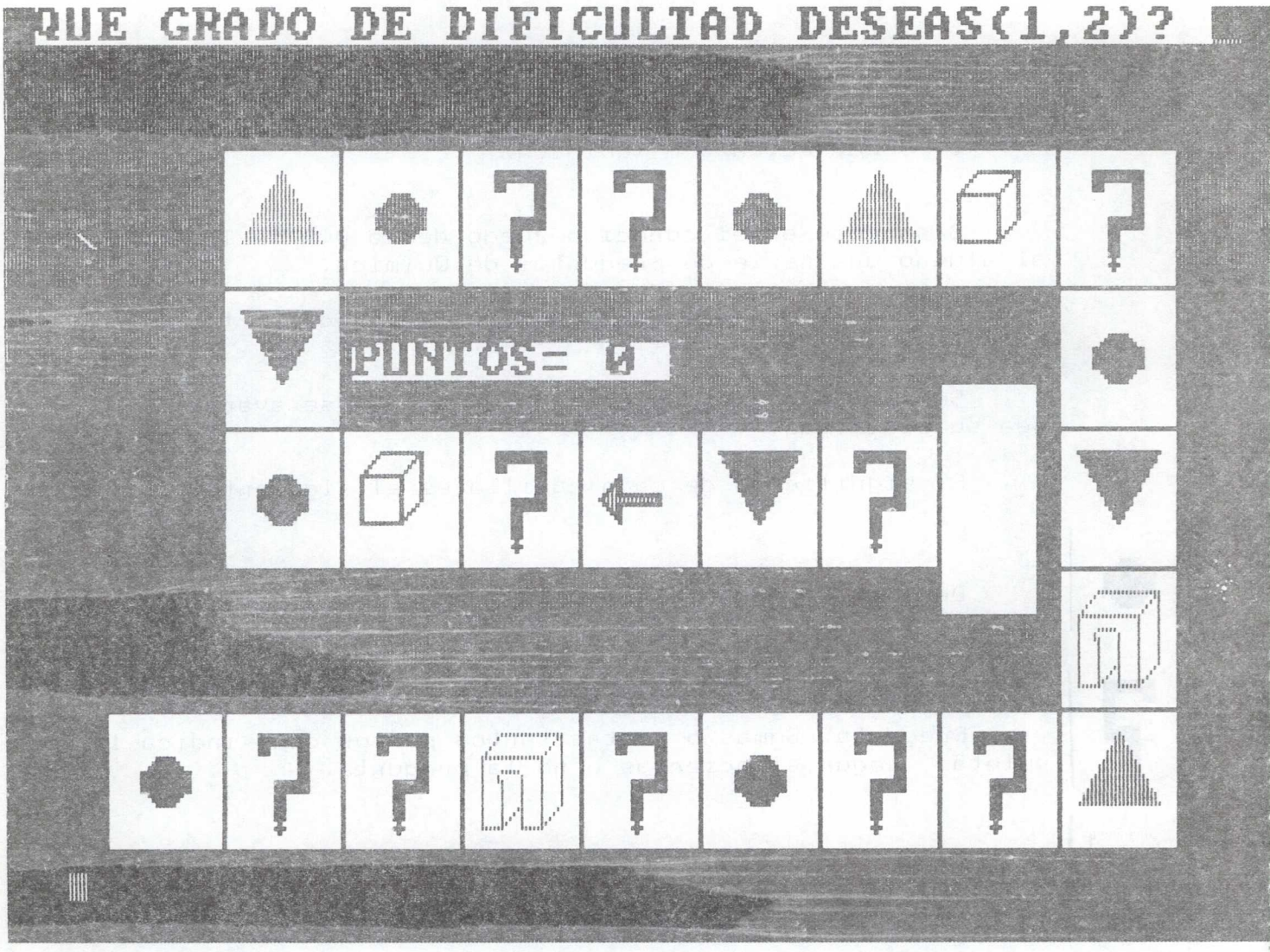
Sumas los puntos que obtienes con la ruleta.



De dado a dado.

Al finalizar el juego pasas a un laberinto.

¡Pulsar la barra espaciadora para avanzar en el juego!


QUE GRADO DE DIFICULTAD DESEAS(1,2)?
PUNTOS= 0

2 JUEGO DE ORTOGRAFIA

Es un programa para ayudar en el aprendizaje de las reglas de la "b" y de la "v".

Van apareciendo cuatro palabras por pantalla y el alumno debe colocar la correspondiente letra. A la vez aparecen mensajes para fijar las diferentes reglas.

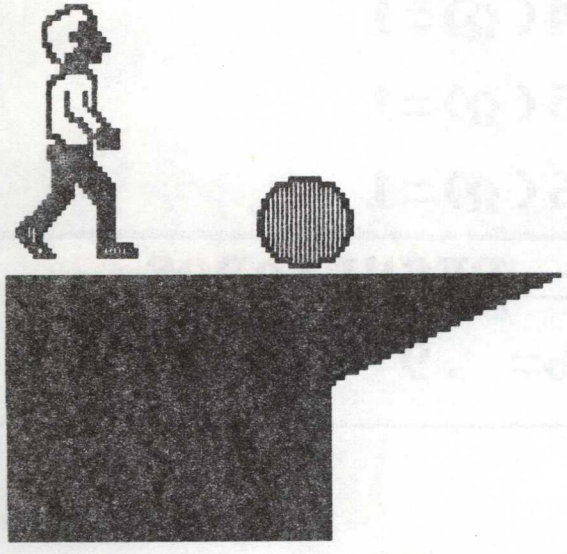
El juego consiste en que según sea el número de aciertos un muñeco alcanza una pelota antes de que se caiga.

mo ilidad

irgen

be er

ser ir

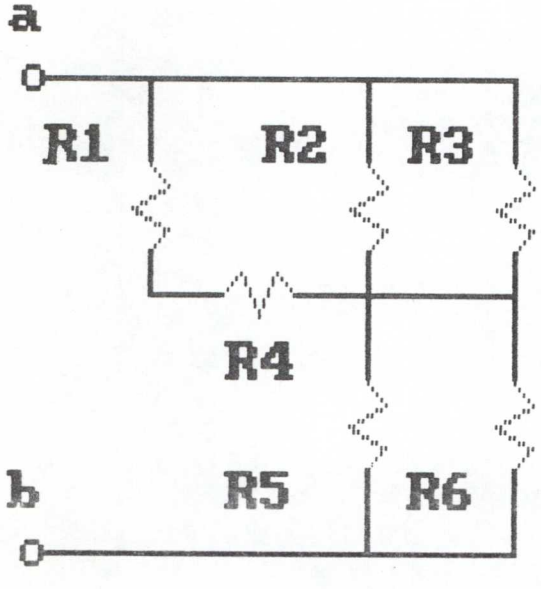


3 RESOLUCION CIRCUITOS OHM

Tras una breve explicación de la ley de Ohm el programa nos lleva a un apartado de asociación de resistencias.Así nos presenta distintos circuitos en los que tenemos que dar valores a las resistencias debiendo calcular la resistencia equivalente.

La segunda parte sigue los mismos pasos para la ley de Ohm generalizada simulando la corriente en el circuito.

- CIRCUITO -



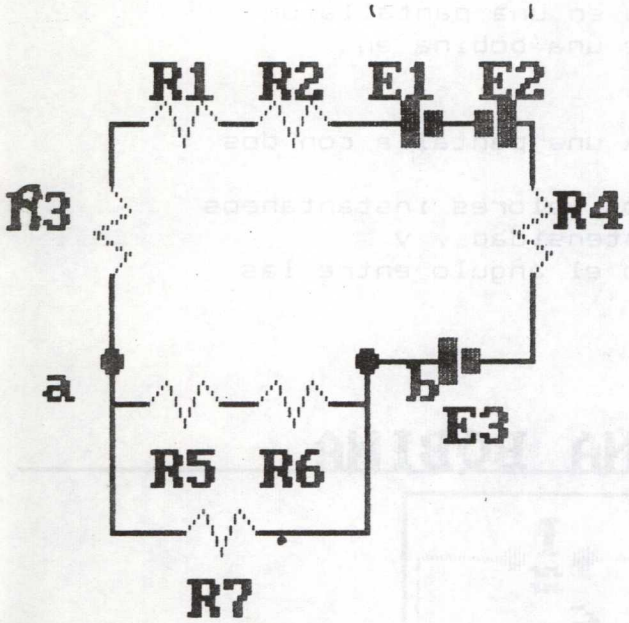
- DATOS -

R1(Ω)=1
R2(Ω)=1
R3(Ω)=1
R4(Ω)=1
R5(Ω)=1
R6(Ω)=1

- RESULTADOS -

Rab= .9 (Ω)

PARA REPETIR PULSA(R),PARA FINALIZAR(F)



- CIRCUITO -

- DATOS -

R1(Ω)=1
R2(Ω)=2
R3(Ω)=1
R4(Ω)=3
R5(Ω)=1
R6(Ω)=1
R7(Ω)=1

E1(V)=1
E2(V)=2
E2(V)=3

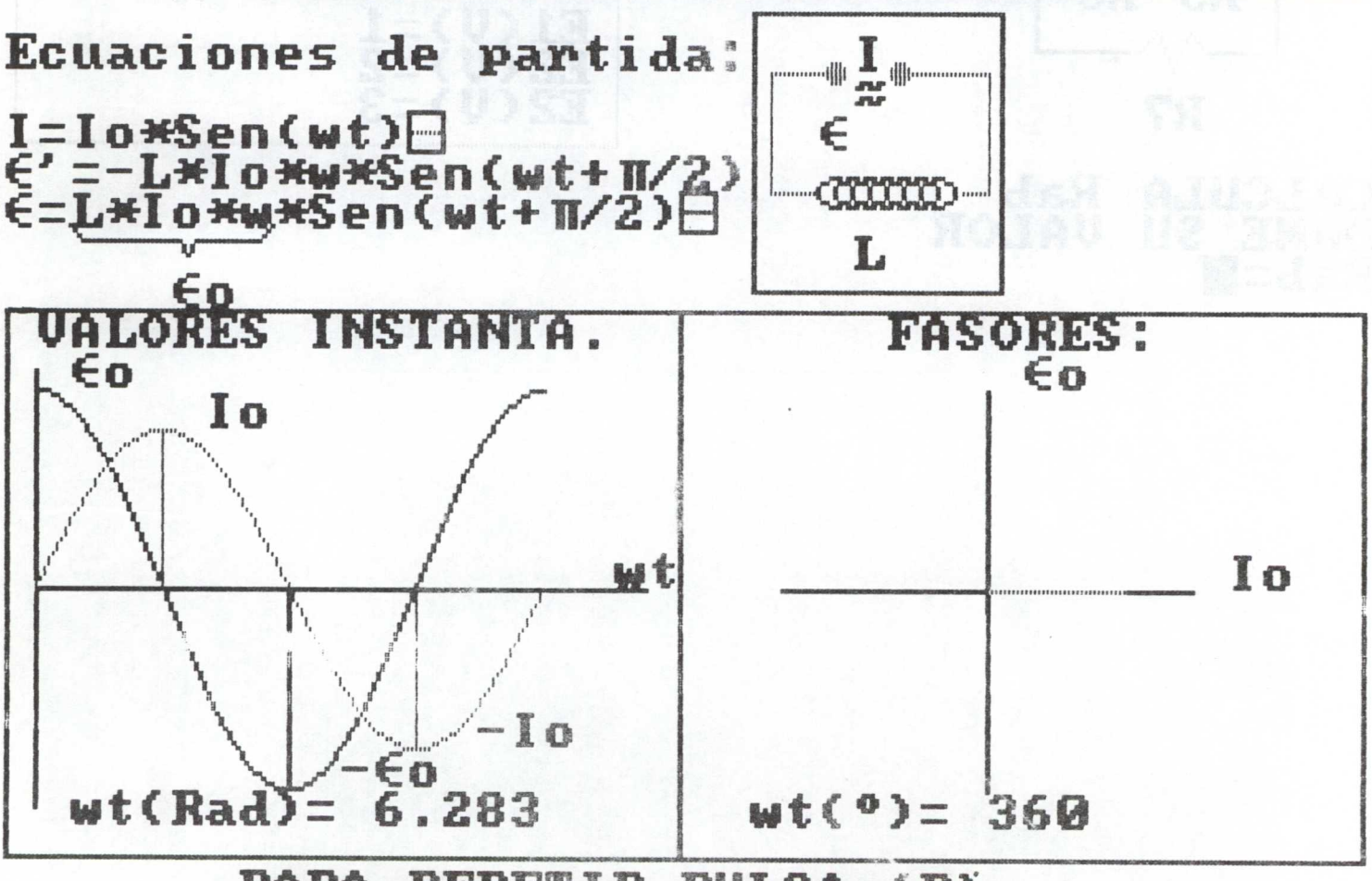
CALCULA Rab
DAME SU VALOR
Rab=

4 SIMULACION C. ALTERNA : BOBINA

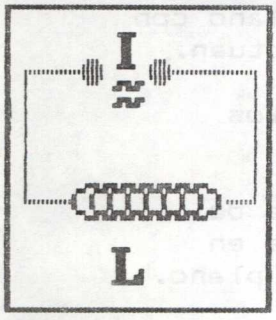
Comienza el programa presentando en una pantalla un resumen teórico del comportamiento de una bobina en corriente alterna.

En el siguiente paso se presenta una pantalla con dos gráficos :

- el primero va representando los valores instantaneos de la fuerza electromotriz y de la intensidad , y
- el segundo muestra paso a paso el ángulo entre las fases en el tiempo.



COMPORTAMIENTO DE UNA BOBINA



Suponemos un circuito por el que circula una corriente :

$$I = Io*Sen(wt)$$

Sabemos que la autoinducción L crea una f.e.m que se opondrá a la variación sinusoidal de I,siendo su valor :

$$\epsilon' = -L*(dI/dt) = -L*Io*w*Cos(wt)$$

Esta expresión puede escribirse como:

$$\epsilon' = -L*Io*w*Sen(wt+\Pi/2)$$

El valor de la f.e.m del alternador es:

$$\epsilon = -\epsilon' = L*Io*w*Sen(wt+\Pi/2)$$

PARA REPETIR PULSA (R)
PARA CONTINUAR PULSA (C)

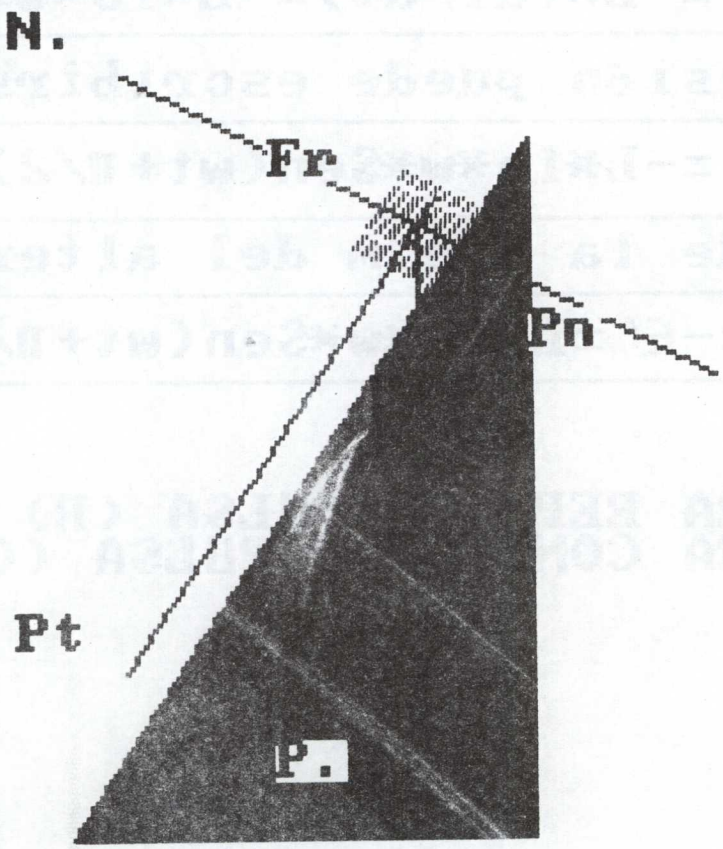
5 SIMULACION PLANO INCLINADO

Empieza el programa pidiendo datos (ángulo , masa y coficiente de rozamiento) para el clásico problema del plano inclinado.

En la siguiente pantalla aparece dibujado el plano con el bloque encima y el diagrama de las fuerzas que actuan.

Tras pular una tecla aparece una pantalla con los valores calculados de las distintas fuerzas.

Al pulsar de nuevo una tecla se simula la caída del cuerpo por el plano , calculando el tiempo que tarda en bajar y la velocidad con la que llega a la base del plano.



CUANDO DESEES CONTINUAR PULSA UNA TECLA

FUERZAS PERPENDICULARES AL PLANO

Pn=P*sen(A)= 220.3677 N.

N=-Pn= 220.3677 N.

FUERZAS PARALELAS AL PLANO

Fr=-Kct*N= 22.03677 N.

Pt=P*cos(A)= 381.9501 N.

Resultante fuerzas normales=0 N.

El movil descendera con a fuerza de arrastre de 359.9133 N.

CUANDO DESEES CONTINUAR PULSA UNA TECLA

6 SIMULACION DEL PENDULO

Este programa simula la ley de conservación de la energía,con la ayuda de un péndulo.

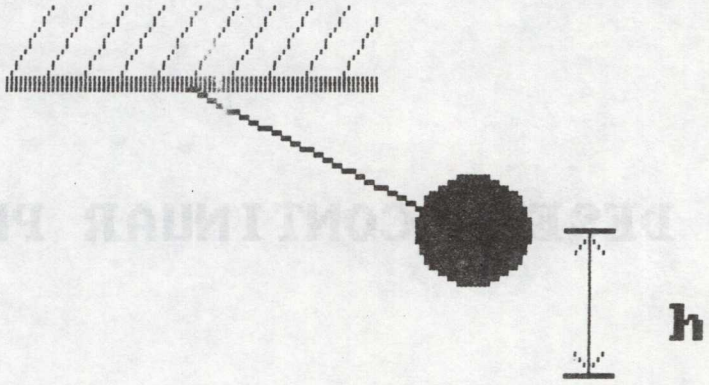
Al iniciar el programa se piden los siguientes datos: ángulo inicial, longitud del hilo, radio de la bola y masa de la bola.

Cuando sa pone en movimiento el péndulo van apareciendo los datos de altura, velocidad, energía cinética, energía potencial y energía total del péndulo en cada momento.

Podemos parar el péndulo (pulsando una tecla) apareciendo su representación en cuatro posiciones diferentes con sus valores correspondientes.

Finaliza la simulación con una pantalla en la que están todos los valores tomados por las variables, comprobando el valor constante de la energía total.

El programa concluye con dos pantallas de texto resumiendo la ley inducida mediante la simulación.



Altura de la bola= 35.001 m.
Velocidad= 0 m/s
E.Cinetica= 0 J.
E.Potencial= 27440.78 J.
E.total= 27440.78 J.

PARA CONTINUAR PULSA UNA TECLA

Masa de la bola= 80 kg. Longuitud del hilo= 70 m.

Angulo	Altura (h)	Velocidad (V)
60	35.001 m.	0 m/s
40	16.377 m.	19.10577 m/s
20	4.222 m.	24.56152 m/s
	0 m.	26.19198 m/s

E.Cinetica (Ec)	E.Potencial (Ep)	E.Total (Et)
0 J.	27440.78 J.	27440.78 J.
14601.22 J.	12839.57 J.	27440.79 J.
24130.73 J.	3310.048 J.	27440.78 J.
27440.79 J.	0 J.	27440.79 J.

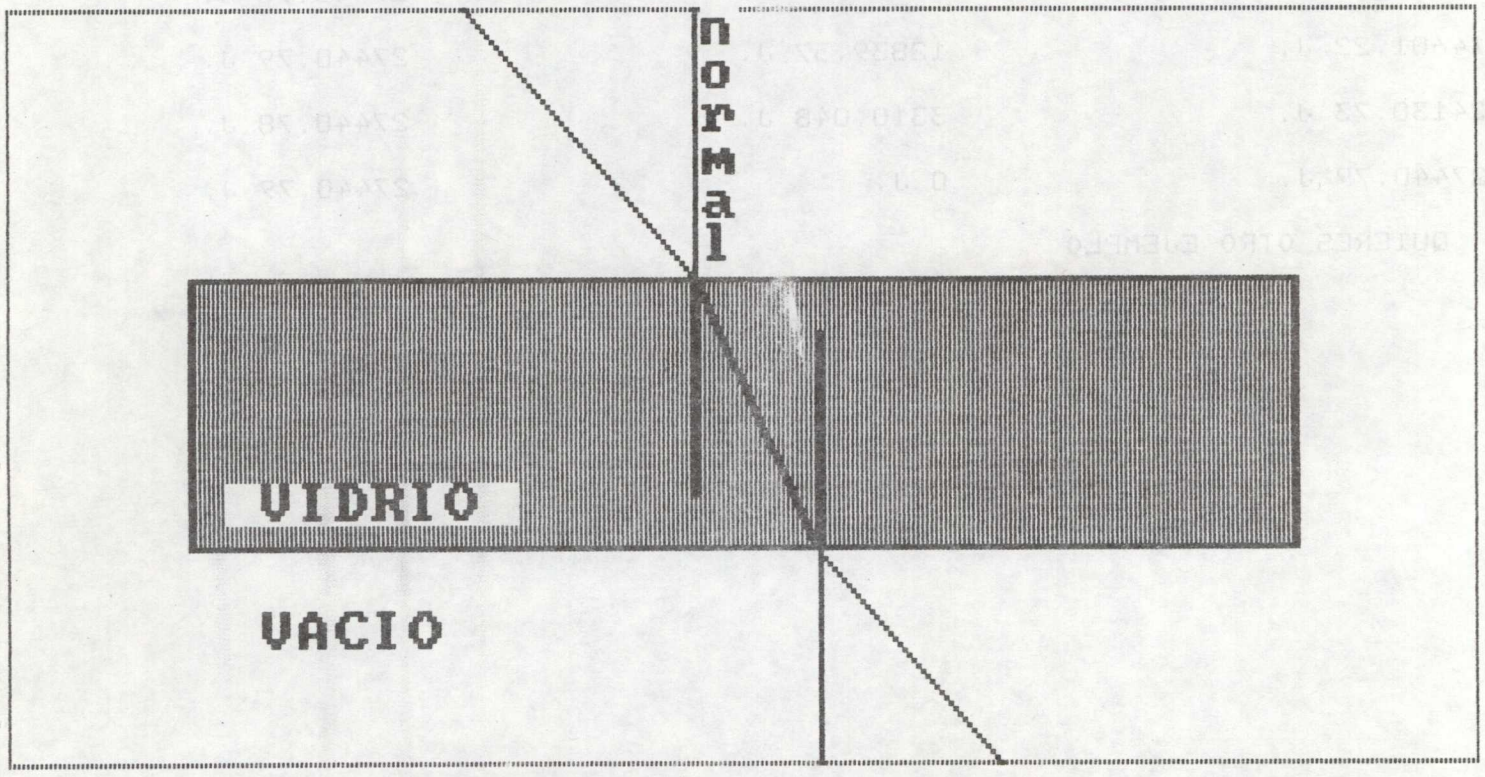
QUIERES OTRO EJEMPLO

7 SIMULACION DOBLE REFRACCION

Consiste en la simulación del paso de un rayo de luz a través de una lámina de caras plano paralelas y de un prisma

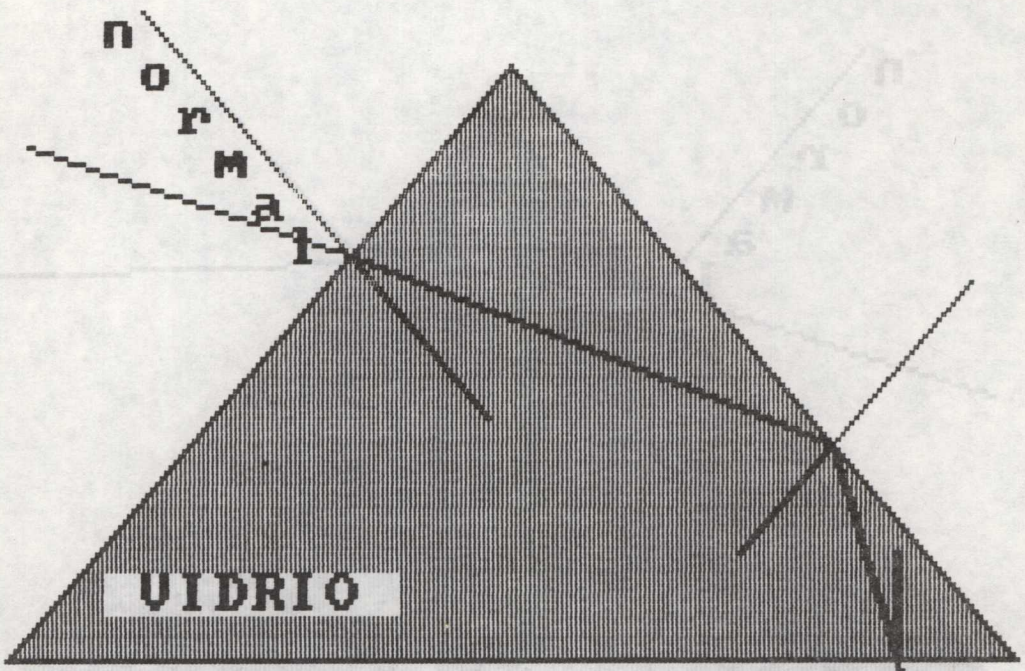
En la lámina de caras plano-paralelas se elige el ángulo de incidencia (entre 0° y 90°) obteniendose el ángulo de salida . Este proceso se puede repetir ciantas veces se desee.

En el prisma óptico se simula la inmersión de un prisma de vidrio en agua . De forma automática se prueban ángulos de incidencia de 30°, 35°, 45°, 60° y 75°



Ang i1	Ang r1	Ang i2	Ang r2
45	27.97325	27.97325	45

PULSA TECLA PARA CONTINUAR

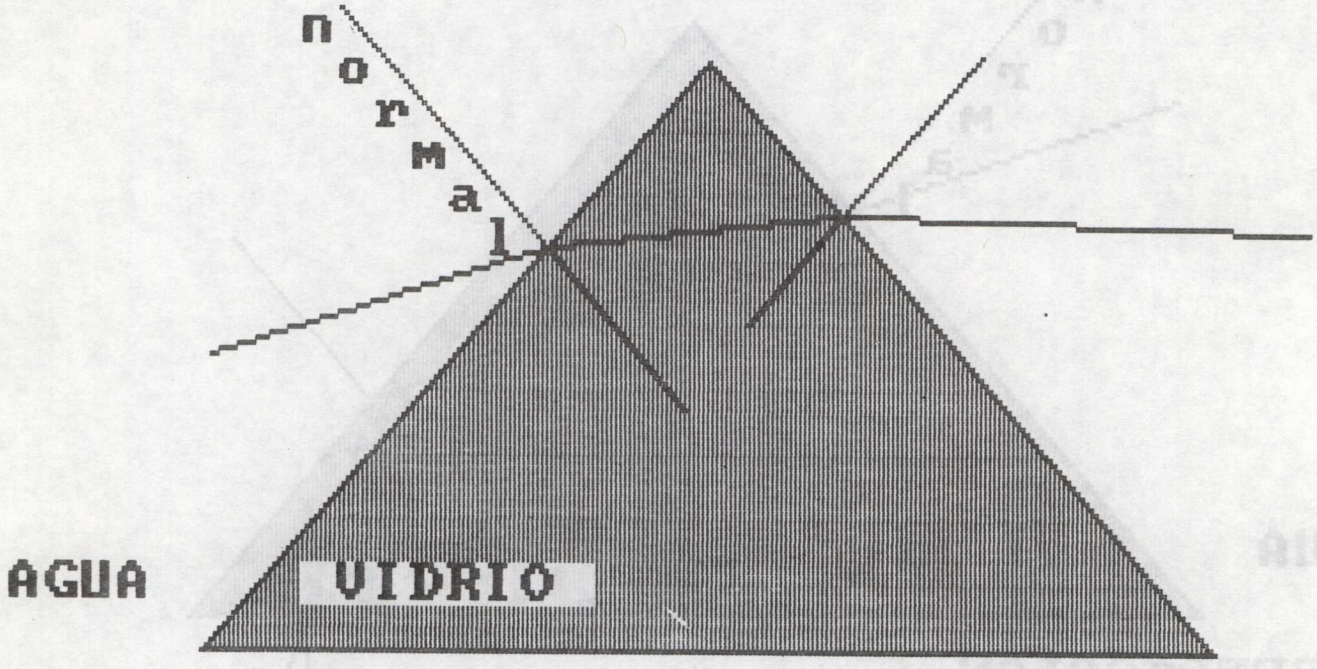


REFRACCION 1

Ang i	Ang r
30	26.24

HAY REFLEXION
Ang i2=63.75
Ang. limite=62.18
ang i2 > ang lim

PARA CONTINUAR PULSA TECLA



REFRACCION 1		REFRACCION 2	
Ang i	Ang r	Ang i	Ang r
60	50	40	46.61

PARA CONTINUAR PULSA UNA TECLA

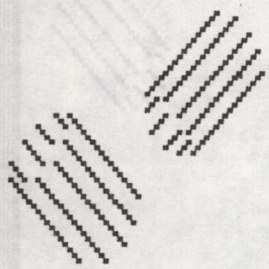
8 SIMULACION MEIOSIS

En la primera pantalla se presenta un menú con cuatro opciones.Al final de cada parte se puede volver al menú.

La primera parte es una introducción que consta de cuatro pantallas de texto explicativo sobre la meiosis.

En la segunda comienza la simulación con la presentación de los elementos que intervienen en la meiosis.

Y a continuación se simula el comportamiento de los cromosomas sexuales durante las dos divisiones meióticas en todas sus fases.



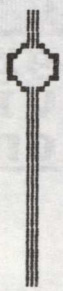
---Simbolizan los CENTRIOLOS



---Simboliza el NUCLEOLO



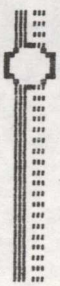
---Cromosoma X



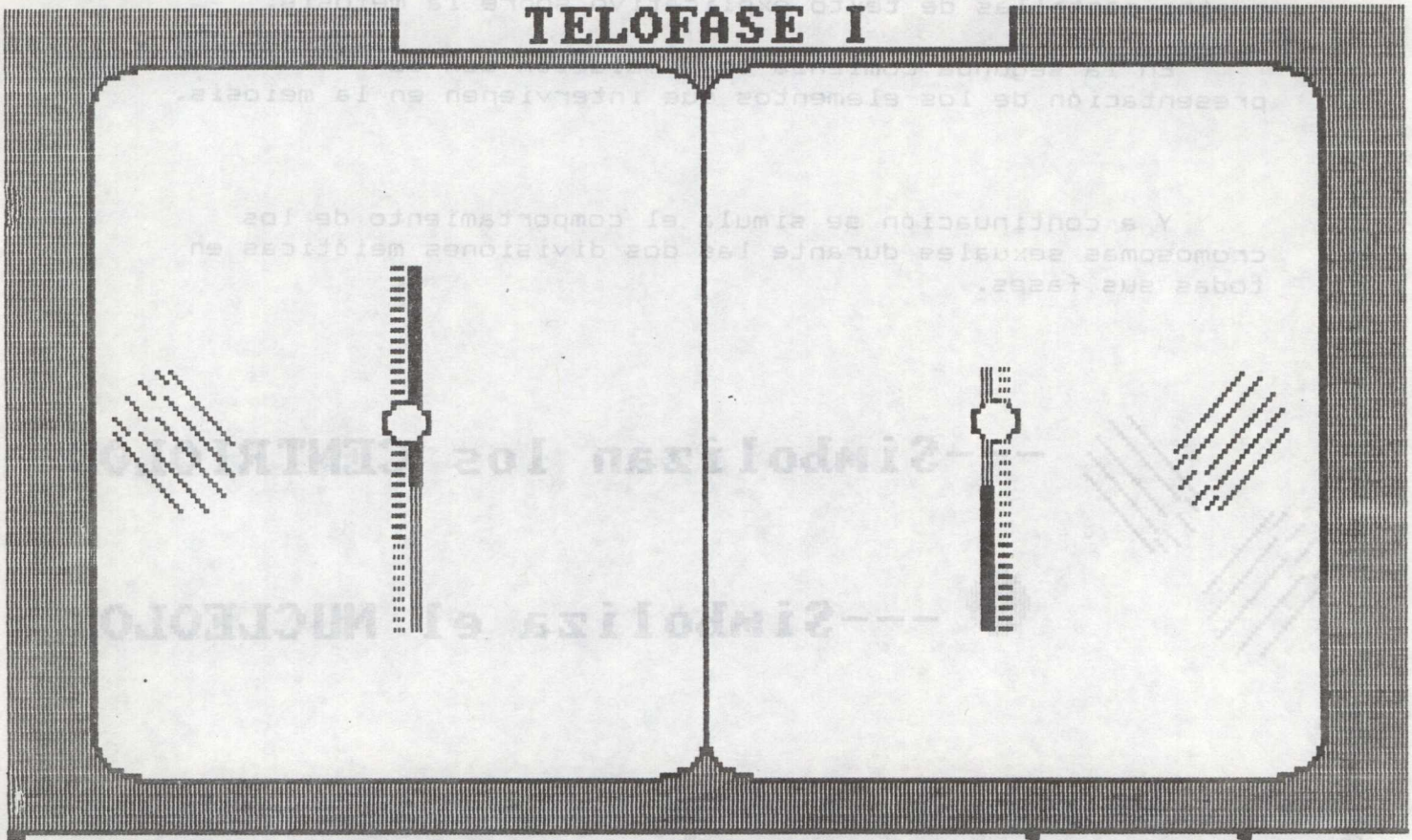
---Cromosoma Y



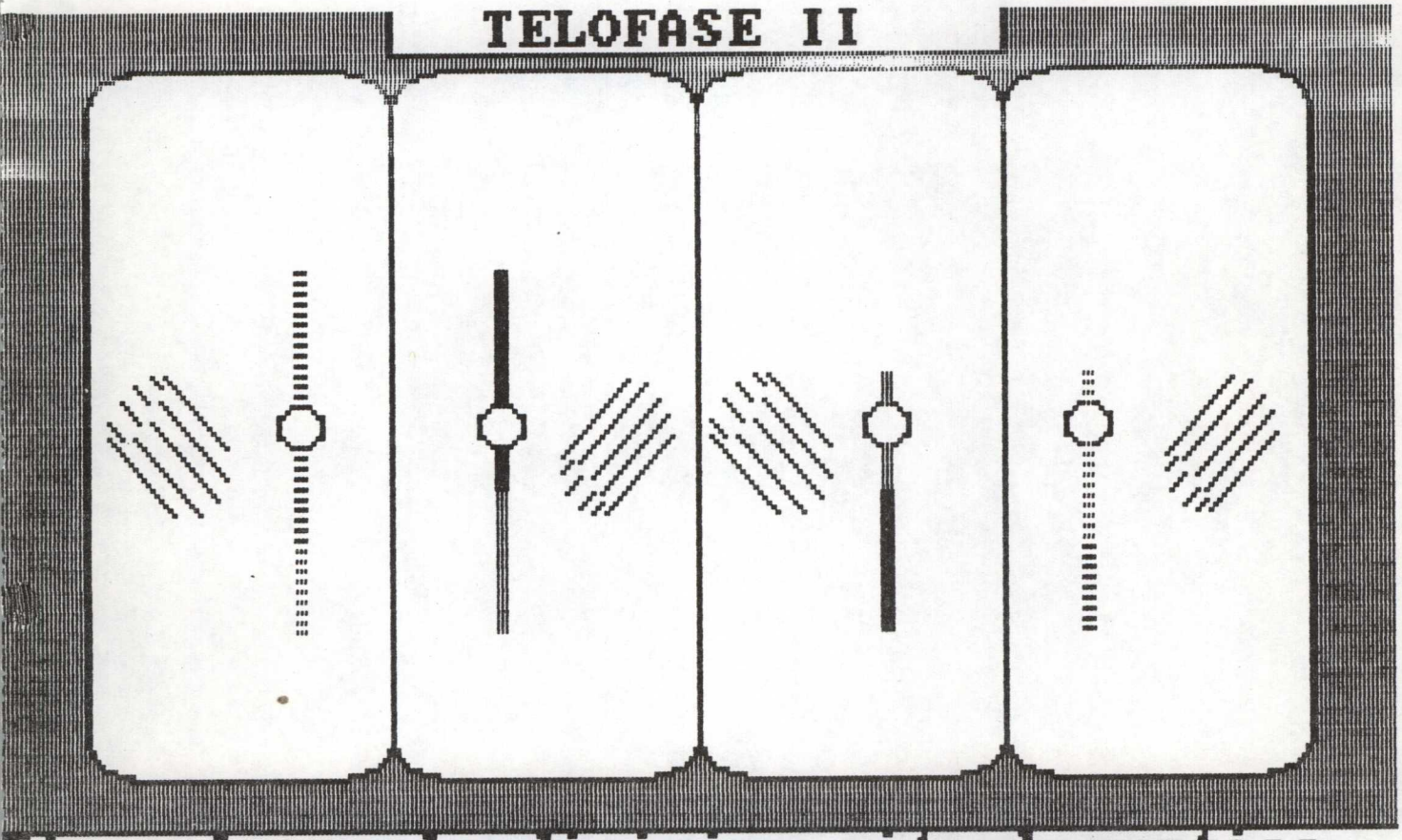
---Cromosoma X con el ADN duplicado (2 cromatidas)



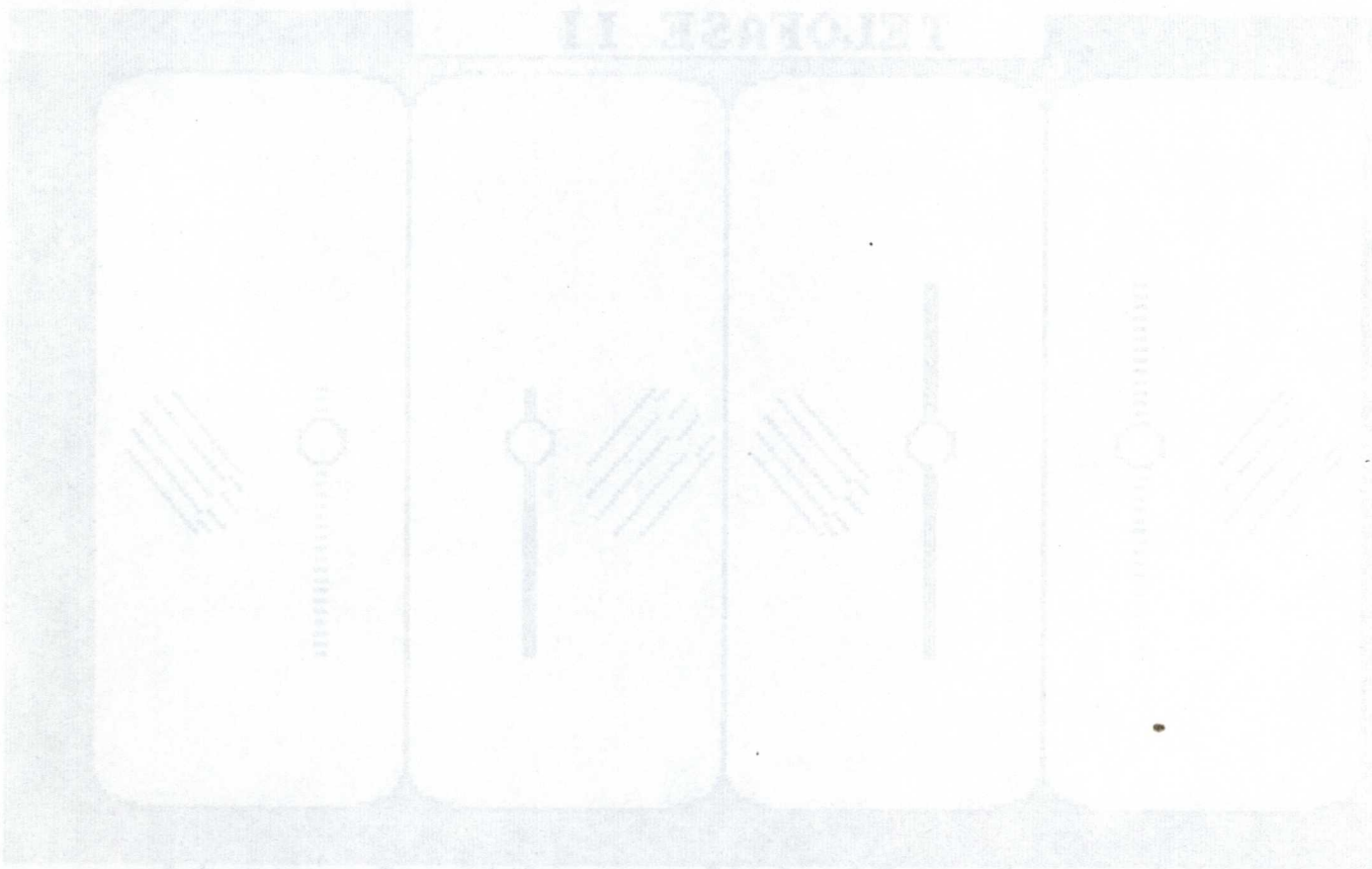
---Cromosoma Y con el ADN duplicado (2 cromatidas)



Los cromosomas se agrupan en los polos
y tiene lugar la division citoplasmatica
Las celulas hijas son haploides



Tiene lugar la division citoplasmatica, quedando constituidas cuatro celulas con dotacion haploide de cromosomas



INTELIGENCIA ARTIFICIAL Y EDUCACION

La Inteligencia Artificial es un tema de moda, representa una esperanza de un nuevo estilo de comunicación del hombre con las máquinas que permita un uso inteligente de las mismas, que posibilite la realización de tareas verdaderamente inteligentes que hasta ahora eran reservadas a los humanos. Si lo anterior es un nuevo espejismo o una utopía de imposible alcance es algo que el desarrollo futuro de los ordenadores de la denominada Quinta Generación aclarará. Por el momento, la realización de sistemas expertos en distintas facetas del conocimiento representa el mayor logro que puede exhibir.

Desde un punto de vista educativo su interés actual puede circunscribirse en dos aspectos: la realización de programas "inteligentes" de apoyo a la enseñanza y la organización de conocimientos que un alumno realiza al efectuar un programa él mismo en un lenguaje de I.A.

El primero de ellos pretende desarrollar sistemas expertos, o aplicar los ya existentes, para situaciones de aprendizaje. Se trataría de aprender de los "conocimientos" de un experto y de sus modos de razonar. El desarrollo de estas aplicaciones está hoy día restringido a las investigaciones de muy pocos departamentos universitarios y sus escasas realizaciones están en su mayoría en tésis doctorales de difícil localización. La mayor dificultad se encuentra en el hecho costatado de que, si bien las conclusiones del sistema experto pueden ser correctas, sus procesos de razonamiento difieren grandemente de los que siguen los expertos humanos.

El segundo enfoque se sustenta en la creencia de que realizar programas en un lenguaje de I.A ayuda a desarrollar procesos de pensamiento inteligente, y de que al explicitar la multiplicidad de factores que inciden en un razonamiento definiendo los conocimientos en forma de reglas y de hechos, produce una positiva organización del saber en quien desarrolla el programa. Hay además dos beneficios culturales añadidos en este segundo enfoque: Se produce un enriquecimiento de la materia en la que trabajan, al verse precisados a explicitar las reglas y el cuerpo de sus conocimientos relevantes en la definición del problema. Se adquiere, además, una aproximación a la dimensión cultural de la I.A, se comprende el proceso de razonamiento de la máquina.

Los lenguajes de I.A se caracterizan por poseer una gran capacidad de manipulación de símbolos, frente a los que se utilizan para realizar cálculos numéricos. Se basan normalmente en estructuras abstractas, denominadas listas, cuyos elementos pueden ser, a su vez, otras listas, posibilitando el tratamiento de estructuras de datos generales. Entre estos lenguajes cabe destacar el LISP, acrónimo de LISt Processing; su descendiente para uso educativo: el Logo; el Smalltalk, que realiza una metodología de programación orientada al objeto; y el PROLOG, acrónimo de PROgramación LOGica.

Más adelante daremos una visión más concreta de algunas aplicaciones educativas de los lenguajes de la I.A, por el momento estudiaremos uno de estos lenguajes, el PROLOG, tal vez el más desconocido de los mismos. La aparición de versiones de este lenguaje para microordenadores (MICRO-PROLOG) facilitará su difusión.

Introducción al lenguaje PROLOG

PROLOG, derivado de PROgramming in LOGic, es un lenguaje de ordenador desarrollado hacia 1970 por un equipo dirigido por Alain Colmerauer en la Universidad de Marsella. A partir de su creación se han ido desarrollando traductores cada vez más eficaces, y la aparición de versiones para microordenadores ha hecho aumentar su interes. Su ámbito de aplicación se encuentra en aquellos tipos de problemas que involucran representaciones simbólicas de objetos y de relaciones entre los mismos. La base fundamental de PROLOG es la noción de programación lógica desarrollada por Kowalski, en la que la computación puede verse como una secuencia de inferencias lógicas. En PROLOG no existe diferencia entre base de datos y programa, ni entre recuperación de datos y computación. Algunas de las aplicaciones actuales del lenguaje son:

-Bases de datos relacionales
-Sistemas expertos
-Lógica matemática
-Demostración de teoremas
-Semántica
-Comprensión del lenguaje natural
-Resolución de problemas

PROLOG es un lenguaje de estructura muy simple. La concepción del mismo es muy diferente de la del resto de los lenguajes, de forma que el estilo del programador de PROLOG es muy distinto del de cualquier lenguaje convencional, ya que, en lugar de tratar de obtener un algoritmo para resolver un problema (estilo imperativo), declara las relaciones que existen en el problema entre los distintos elementos que intervienen (estilo declarativo) e inferir, en un sentido lógico, la respuesta.
Así, un programa en PROLOG consiste en una serie de cláusulas que, o bien reflejan hechos y relaciones que son conocidas por haberse declarado, o bien representan reglas que al aplicarse a los hechos anteriores permiten deducir nuevos hechos o conocimientos.
La resolución de un problema escrito en PROLOG tiene relación directa con el análisis del problema y no con el funcionamiento de la máquina, como ocurre con los lenguajes imperativos. En este sentido se puede considerar como un paso más en la evolución de los lenguajes desde una orientación hacia la máquina a una orientación hacia el problema.

Iniciación a la programación en PROLOG

Un programa en PROLOG consiste básicamente en:

-Declaración de hechos acerca de objetos y sus relaciones
-Declaración de reglas
-Realización de consultas

Por tanto, consiste en una base de datos que contiene un conjunto de conocimientos relevantes para la resolución de un problema. Para proceder a la resolución del mismo, basta con preguntar acerca de la verificación o no de determinadas condiciones.

-Hechos

La forma en que el programador debe especificar los hechos consiste en introducir listas que expresan relaciones entre objetos, siendo, en sintaxis estándar, el primer elemento de la lista el nombre por el que se ha de reconocer la relación. Estas relaciones pueden referirse a uno o mas elementos. Algunos ejemplos son:

```
(( varón Jose ))
(( hembra Maria ))
(( gusta Maria dulces )) .... A Maria le gustan los dulces
(( padre-de Juan Luis ))
(( padre-de Juan José )) .... Juan es el padre de José
(( padre-de Jose David ))
(( padre-de Luis Antonio ))
```

Como puede observarse la notación estándar es poco natural, ya que exige poner la relación antes que los objetos relacionados. Por eso se han desarrollado modulos, como el denominado SIMPLE, que permiten una notación más acorde con la forma habitual de expresar relaciones.

Una vez introducidos los hechos, ya se puede efectuar consultas sobre los mismos, bien para averiguar si algunos hechos son o no ciertos, bien utilizando incógnitas para averiguar qué objetos verifican una relación determinada. A continuación se muestran algunos ejemplos:

```
es (( padre-de Juan Luis ))
SI

es (( hembra Maria ))
SI

cual ( _persona ( padre-de Juan _persona ) )
Luis
José
No hay (más) respuestas

cual ( (_pers1 _pers2) ( padre-de _pers1 _pers2 ) )
(Juan Luis)
(Juan José)
(José David)
(Luis Antonio)
No hay (más) respuestas
```

Obsérvese cómo la interrogación puede utilizarse para preguntar acerca de cualquiera de los elementos que falten por conocer en la relación, siempre que no se origine un número infinito de respuestas. Esto permite poner de manifiesto el carácter reversible de las operaciones. Por ejemplo, la suma puede utilizarse para sumar o para restar:

```
cual ( _número ( suma 3 5 _número ) )
8
No hay (más) respuestas
```

```
cual ( _número ( suma 3 _número 8 ) )
5
No hay (más) respuestas
```

-Reglas

La definición de reglas se efectúa traduciendo a la sintaxis del lenguaje sentencias lógicas (cláusulas de Horn) introduciendo una lista de hechos, el primero de ellos es el consecuente de la regla y los siguientes los antecedentes de la misma. Seguidamente se muestran algunos ejemplos.

Para expresar que toda mujer es una hembra, es decir:

```
x hembra <-- x mujer

cosecuente   antecedente
```

se escribiría de la forma :

```
( ( hembra _persona ) ( mujer _persona ) )
     consecuente          antecedente
```

Para definir la relación abuelo-de expresada de forma natural sería: si la persona 1 es padre de la persona 2 y la persona 2 es padre de la persona 3, entonces la persona 1 es abuelo de la persona 3.
En forma de proposición lógica sería:

```
x abuelo de z <-- x padre de y & y padre de z
 consecuente              antecedentes
```

Su equivalente en sintaxis estándar de PROLOG sería:

```
( ( abuelo-de _p1 _p3 ) ( padre-de _p1 _p2 )( padre-de _p2 _p3 ) )
       consecuente                  antecedentes
```

Si ahora preguntamos

```
es (( abuelo-de Juan David ))
```

se realizará una búsqueda para comprobar si existe algún elemento _p2 para el cual se verifiquen (padre-de Juan _p2) y (padre-de _p2 David). En este caso existe, y , por tanto, contestará afirmativamente.

Las siguientes cláusulas expresan en sintaxis estándar la definición de la nueva relación "hija-de":

```
(( hija-de _pers1 _pers2 ) ( padre-de _pers2 _pers1 )( hembra _pers1 ))
(( hija-de _pers1 _pers2 ) ( madre-de _pers2 _pers1 )( hembra _pers1 ))
```

En este caso se ha establecido una disyunción, basta que se verifiquen las dos condiciones de la primera cláusula **o** de la segunda para que sea cierta la relación consecuente.

NUCLEO: IA
UTI: 0001
PAG.: - 5

Aplicación: Bases de datos relacionales

Las Bases de Datos relacionales se han considerado hasta ahora como un concepto de difícil adquisición y manejo para alumnos de enseñanza no universitaria. Veremos como el uso de un estilo de programación declarativa permite un acercamiento simple a las mismas. Un alumno puede crear sus propios conocimientos y relaciones, enseñándoselos al ordenador, definiendolos a su antojo para consultarlas después y realizar proyecciones de las mismas, adquiriendo asi conocimientos básicos sobre la organizacion de la informacion y su lógica asociada. Ilustraremos con un ejemplo el proceso de creación de una base de datos relacional. Crearemos una base de hechos que estara formada por los siguientes asertos:

```
((identidad Luis 35 profesor Madrid))
((identidad Pedro 26 estudiante Barcelona))
((identidad Juan 20 estudiante Madrid))
((identidad Maria 25 estudiante Madrid))
((identidad Jose 32 profesor Alcalá))
((identidad Cristina 26 secretaria Torrejón))
((identidad Andrés 25 estudiante Alcalá))

((comunidad Madrid Madrid))
((comunidad Barcelona Cataluña))
((comunidad Torrejon Madrid))
((comunidad Alcala Madrid))
```

Efectuaremos una búsqueda de todos los individuos que cumplan unas caracteristicas determinadas. Por ejemplo:
¿Qué personas viven en Madrid?
Para interrogar a la base de hechos se escribiria:

```
cual ( _persona ( identidad _persona _edad _profesión Madrid ) )
Luis
Juan
Maria
No hay (más) respuestas
```

Se puede, igualmente, establecer una relacion entre dos campos correspondientes a tipos de ficha diferente buscando un enlace entre ambos. Por ejemplo: ¿Qué personas son estudiantes y viven en la comunidad de Madrid? Interrogamos a la base de la forma siguiente:

```
cual ( _pers ( identidad _pers _edad estudiante _ciudad)(comunidad
     _ciudad Madrid))
Juan
Maria
Andrés
No hay (más) respuestas
```

Posibilidades educativas de PROLOG

Las escasas experiencias realizadas hasta hoy no permiten establecer de forma suficientemente concluyente las ventajas posibles del uso de este lenguaje en la educación. Tan sólo podemos, por el momento, realizar conjeturas, si bien el desarrollo de investigaciones como la de integración curricular que está llevando a cabo en Inglaterra el Imperial College, y el previsible incremento de su uso en las escuelas, permitirá una mayor clarificación.

La programación lógica es interesante por derecho propio, al tratarse de una combinación de lógica y computación, y puede servir, además, como un instrumento para una enseñanza más efectiva de otras áreas del curriculum. De acuerdo con Piaget, la lógica es el espejo del pensamiento, y no a la inversa. Es decir, la función de la lógica es hacer explícitos aquellos procesos mentales que se producen de modo natural en el nivel más elevado del desarrollo humano. La enseñanza del modo de pensamiento lógico se encuentra en todas las áreas del curriculum.

Existen dos tendencias actualmente en el uso educativo de este lenguaje: la de los que utilizan programas desarrollados en el mismo dando al alumno el papel de usuario y la de aquellos que confieren al alumno el papel de creador de programas utilizando los principios de la programación lógica. El uso de programas suele orientarse a la linguística y a las ciencias sociales principalmente, en este último caso es predominante el trabajo con bases de datos relacionales. La realización de programas por los propios alumnos parte de la base de un conocimiento de la sintaxis básica del lenguaje - en ocasiones se incorporan al mismo gráficos que ayudan a visualizar los procesos -para, a partir de éste, desarrollar programas de creación de bases de datos relacionales o aprender la lógica a través de un lenguaje de ordenador. No obstante, la complejidad del lenguaje obliga a trabajar tan sólo con subconjuntos del mismo y realizar pequeños programas de creación y consulta de bases de conocimientos. En este caso podemos establecer una similitud con Logo, del que habitualmente se utiliza solamente la parte de gráficos de tortuga, pero, a diferencia de éste, es más difícil progresar en su dominio. Sería, por tanto, valioso como una especie de micromundo para el aprendizaje de la lógica o de conceptos básicos de Inteligencia Artificial.

La reciente aparición de instrumentos para el desarrollo de sistemas expertos en microordenadores, basados en PROLOG, aumentan las posibilidades en este último campo y posibilitarán el desarrollo de sistemas expertos para facilitar otros aprendizajes e, incluso, que el propio alumno los cree. El estilo de la programación declarativa representa una curiosa innovación y, además, un reto a la inteligencia, y, ésto, unido al carácter mítico que posee la Inteligencia Artificial hoy, hará crecer el interés de profesores y alumnos hacia este lenguaje. Su desarrollo futuro en las escuelas dependerá tanto de los resultados educativos como de factores tecnológicos e industriales, vinculados a los grandes proyectos de investigación y desarrollo de máquinas PROLOG, en el contexto de los planes que se están llevando a cabo en distintas naciones para la creación de máquinas "inteligentes". La propia aparición de éstas, se basen en PROLOG, LISP o cualquier otro lenguaje,

puede hacer desaparecer, paradójicamente, el interés actual por todos los lenguajes, incluso el de los propios especialistas en ordenadores. En cualquier caso cada vez se considera de menor interés la programación de ordenadores en la enseñanza general, pero paralelamente se están desarrollando nuevos ambientes activos de aprendizaje por ordenador en los que se pone el énfasis en la resolución de problemas y el desarrollo de capacidades más que en la programación en sí. Algunos de estos desarrollos profundizan la línea abierta por S. Papert con el lenguaje Logo, al que añaden la posibilidad de uso de la programación declarativa (PROLOGO) y la programación orientada al objeto propia de Smalltalk, entre otros instrumentos educativos. Este es el caso del entorno AIBLE (Artificial Intelligence Based Learning Environment) que se realiza en el marco del proyecto europeo ESPRIT.

BIBLIOGRAFIA

MICRO-PROLOG: Programming in logic, K.L. Clark and F.G.Mc Cabe. C.A.R. Hoare Ed., Prentice Hall Int. 1984.

Logic for Problem Solving, R. Kowalski. North Holland. 1979

Logic as a Computer Language for Children, R. Kowalski

Teaching Logic as a Computer Language in Schools, R. Ennals. Proceedings of the First International Logic Programming Conference, Marseille. Sept 1982.

An introduction to PROLOG, W.F. Clocksin

Enseñanza y Aprendizaje con Ordenadores: Inteligencia Artificial en Educacion. Tim O'Shea y John Self. Ed. Anaya. 1985.

Quelques Réflexions sur Intelligence Artificielle et Education, Jacques Perriault. Education et Informatique 23. Nov/Dec 1984.

Une utilisation de MICRO-PROLOG en classe d'Histoire, R. Ennals. Education et Informatique 23. Nov/Dec 1984.

Pupils, Computers and History Teaching, J. Nichol and J. Dean.

A Prototype Grammar kit in PROLOG, K. M. Kahn. Instruccional Science 13. 1984.

PROLOG, Lenguaje de la Inteligencia Artificial, Alain Colmerauer. Mundo Cientifico 41. 1984.

A Query the User Facility for Logic Programming, Marek Sergot, North Holland. 1983.

Smalltalk-80, A.Goldberg. Reading Mass.Addison Wesley. 1984.

Beginning Micro PROLOG, Richard Ennals.

NUCLEO: IA
UTIL: 0002
PAG.: 1

INSTITUTO DE TECNICAS EDUCATIVAS
INTELIGENCIA ARTIFICIAL

BIBLIOGRAFIA

SISTEMA OPERATIVO MS-DOS

INTRODUCCION

Un sistema de microordenador, en unas de las configuraciones más típicas que se encuentran en la práctica, está formado por el microordenador, la consola (combinación del teclado y la pantalla), una o dos unidades para discos y una impresora. Se pueden introducir numerosos cambios en este sistema.

Estas componentes forman lo que se llama el hardware del sistema. Para controlar las operaciones entre estos dispositivos físicos los ordenadores ejecutan unos programas que, en conjunto, se denomina Sistema Operativo. Por ejemplo, permiten enviar información a la pantalla, a una impresora o a una puerta de comunicaciones para que se envíe a otro sistema.

MS-DOS (Sistema Operativo de Disco Microsoft) es el nombre genérico del sistema operativo patentado por Microsoft Corporation para su uso sobre distintas marcas de ordenadores. Algunos fabricantes de ordenadores han modificado el MS-DOS para sus máquinas y le han dado un nuevo nombre, PC-DOS o Z-DOS.

A la denominación de un sistema operativo le acompañan unos números que indican la versión (MS-DOS 1.2). Todo sistema operativo experimenta modificaciones en cada lanzamiento al mercado (corrección de algunos programas, incorporación de otros nuevos, etc.). Cada lanzamiento supone una versión. Si la diferencia entre dos versiones es la última cifra representa pequeñas variaciones respecto a la anterior. Sin embargo, si es en la primera cifra representa cambios fundamentales.

NIVELES EN QUE TRABAJA EL MS-DOS

El MS-DOS trabaja a dos niveles :

Primer nivel.- Es un sistema que gestiona el hardware. Coordina la unidad central de proceso (C.P.U.: chip microprocesador que actúa como cerebro) del ordenador con el resto del hardware. Así, el MS-DOS toma el carácter que se pulsa en el teclado, lo codifica en forma comprensible para la CPU y,a continuación, lo visualiza en el monitor en forma comprensible para nosotros. Si se está usando un programa de hoja de cálculo o un procesador de texto, el MS-DOS actúa como intermediario que convierte las señales electrónicas, que genera el teclado, en códigos de control utilizables por el programa de aplicación.

Segundo nivel.- El MS-DOS ejecuta "órdenes" que permiten interactuar directamente con el ordenador. Estas órdenes realizan

funciones de utilidad, como nombrar los ficheros almacenados en disco o copiar ficheros de un disco a otro. Nosotros nos vamos a referir principalmente a este nivel.

El MS-DOS, como conjunto de programas, al ejecutar un programa se convierte en una orden a la cual trata como un programa de aplicación, aunque con más limitaciones.

Cada orden tiene un nombre de acuerdo con su función y fácil de recordar. Por ejemplo, para hacer un "chequeo" del disco se utiliza la orden CHKDSK.

DISCOS Y FICHEROS

Cuando se guardan datos en un disco, la información se almacena en un fichero.

Un fichero es una colección de información identificada con un nombre único que nosotros le asignamos.

La información puede consistir en un texto (e.,un informe), datos (e.,una lista de direcciones) o un programa (e.,un procesador de texto).

El fichero puede tener cualquier longitud, limitada solamente por el espacio disponible en el disco. Este tamaño se mide en bytes.

Un byte u octeto es una unidad de memoria capaz de guardar un carácter. Así, la letra "A" o el dígito "1" pueden almacenarse en un byte de memoria. Como múltiplos del byte hay el Kilobyte(K) y el Megabyte(M); 1K = 1024 bytes que aproximadamente son 1000 bytes; 1M = 1024 x 1024 bytes, aproximadamente 10^6 bytes.

Los ficheros se pueden almacenar en dos tipos de disco: discos flexibles (diskettes) y discos duros o rígidos. Casi todas las órdenes del MS-DOS operan de la misma manera sobre discos duros que sobre diskettes.

Un disco flexible puede tener entre 150 y 1.200 K de capacidad, mientras que un disco duro de 5 a 20 M.

DISCOS Y FICHEROS

En cuanto a su tamaño y cantidad de información que se pueden almacenar, hay las siguientes clases de diskettes:

Diskettes de 8, $5^{1/4}$, $3^{1/2}$ y 3 pulgadas.

Diskettes de doble cara o simple cara.

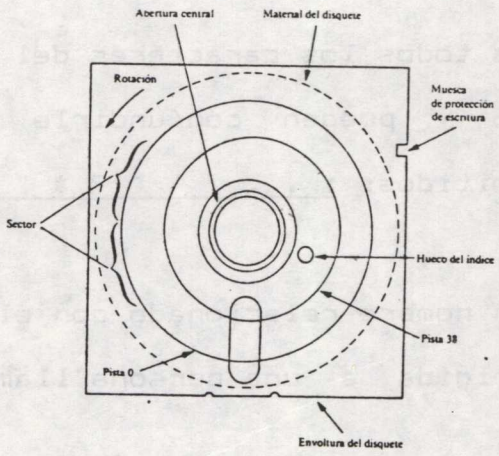
Diskettes de doble densidad o simple densidad.

Diskettes de alta capacidad. (1.2 MB)

Cuando la información se almacena sobre un diskette, se considera que la superficie del mismo está constituída por una serie de circunferencias concéntricas, llamadas pistas. Pista es un número que indica la distancia del fichero al eje del disco. La circunferencia mayor se enumera como 0 y la menor como un número entre 39 y 80, dependiendo del tamaño del disco.

Una pista se divide en sectores. Dependiendo del disco, una pista puede tener entre 5 y 27 sectores y un sector puede almacenar 128 bytes (caracteres).

Pistas y sectores en un disquete



Para almacenar un fichero en disco, hay que darle un nombre que se grabará, junto con otros parámetros, en una zona especial del disco llamada directorio. Cada fichero se referencia con un único nombre llamado especificador de fichero. Este especificador está dividido en dos partes: el nombre del fichero y la extensión o tipo, separados por un punto (ej. Informe.TXT). La extensión es opcional pero un fichero ha de tener un nombre.

Ambas se usan para describir el contenido del fichero. La combinación del nombre del fichero y la extensión tiene que ser única en el disco; es decir, un disco no puede tener dos especificadores idénticos.

Existen limitaciones respecto a la longitud del nombre del fichero y la extensión. Un nombre puede tener hasta 8 caracteres como máximo y una extensión hasta 3 caracteres. También hay que tener en cuenta las siguientes recomendaciones:

- No están permitidos todos los caracteres del teclado. Hay muchos que los utiliza el MS-DOS y pueden confundirle si apareciesen en el nombre. No están permitidos: : , . / \ ' " ? * " .

- Conviene elegir un nombre relacionado con el contenido del fichero. Por ej. una carta dirigida a una persona llamada Ana podría llamarse CARTANA.TXT.

- El MS-DOS no admite nombres de ficheros iguales a los nombres de dispositivos que utiliza:

CON: Consola
LPT1:
y
PRN: Ambos indican la puerta de impresora en paralelo. Las puertas paralelo segunda y tercera se llaman LPT2: y LPT3:
NUL: Dispositivo inexistente que se usa exclusivamente para probar programas de aplicación.

- Es conveniente utilizar las extensiones estándar convencionales:

ASM: Programa fuente en Assembler

BAK: Copia de seguridad de otro fichero

BAS: Programa en Basic

BAT: Fichero batch

BIN: Fichero binario usado por un programa

C : Programa fuente en lenguaje C

COB: Programa fuente en Cobol

COM: Programa comando

DAT: Fichero de datos

EXE: Programa ejecutable

OBJ: Fichero objeto, resultado de la compilación

OVR: Programa suplementario para programa de una aplicación

PAS: Programa fuente en Pascal

SYS: Programa especial del Sistema Operativo

TEX: Fichero de texto

TXT: Fichero de texto en formato ASCII

- Hay que evitar el uso de algunas extensiones que tiene significados especiales para el MS-DOS (COM, EXE, BAT) pues pueden confundir al sistema operativo y asumir estos ficheros como órdenes.

UTILIZACION DE CARACTERES COMODIN O AMBIGUOS

Para referenciar o llamar a un grupo de ficheros, en lugar de especificar los nombres completos y la extensión, se pueden utilizar los caracteres ? y *.

El ? se utiliza para emparejar con cualquier carácter en la posición en que se encuentra, dentro del nombre del fichero o extensión.

FEL?.DAT representa un fichero cuyo nombre tiene 4 caracteres de los cuales desconocemos el último.

El carácter * sustituye a cualquier nombre o extensión de un fichero.

FELT.* sustituye a todos los ficheros de nombre FELT, sea cual sea su extensión.

LIMITACIONES DE MS-DOS

A pesar de las mejoras que se realizan en cada versión el MS-DOS no es capaz de realizar algunas tareas. Nos referimos a dos situaciones que ya resuelven algunos sistemas operativos, como el UNIX y otros.

Una de ellas es la que se denomina multitarea o concurrencia. El MS-DOS no permite ejecutar varios programas simultáneamente.

La otra situación que no permite el MS-DOS es la denominada multiusuario que consiste en ejecutar y dar órdenes a un mismo programa, desde terminales diferentes.

ARRANCANDO EL MS-DOS

Si se pudiera mirar dentro del MS-DOS veríamos una masa complicada de instrucciones escritas en lenguaje máquina, lenguaje que la CPU sabe cómo leer.

Afortunadamente, no se necesita conocer dicho lenguaje, ni siquiera cómo hace el MS-DOS su trabajo.

No hay un método estándar para cargar el S.O. en el ordenador; cada marca tiene su método.

Normalmente hay dos formas de cargar el S.O. en la memoria del ordenador:

-Insertando el disco del S.O. en una de las unidades de disco (unidad A:), antes de encender el ordenador.

-Si el ordenador ya está en funcionamiento se pulsan simultáneamente las teclas CTRL+ALT+DEL.

Después de cargar el MS-DOS, es común que se vea en el margen superior izquierdo de la pantalla la petición de nueva fecha:

```
Current date is Tue  1-01-1980
Enter new date (mm-dd-yy):
```

Se escribe la fecha separada por guiones (-) o barras (/). Así, son formatos válidos para el 10 de Abril de 1986:

4-10-86

4/10/86

04/10/86

Observar que se pone primero el mes y después el dia.

A continuación el MS-DOS nos pide la hora.

```
Current time is  0:08:12.95
Enter new time:
```

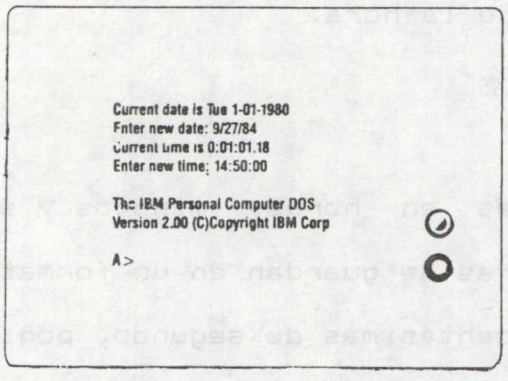
El único formato aceptable es en horas, minutos y segundos separados por dos puntos (:). Las horas se guardan en un formato de 24 horas. Incluso se pueden introducir centésimas de segundo, poniendo un punto a continuación de los segundos y después las centésimas:

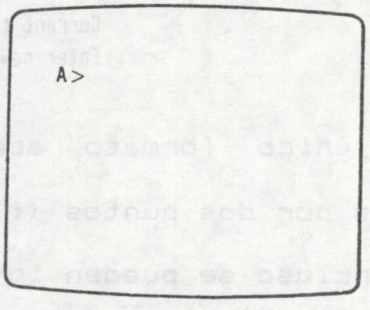
10:11:12 15:45 0:0:0.5

¿Por qué motivo el DOS quiere conocer la fecha? Cuando el S.O. graba ficheros en el disco, lo hace también con la fecha y la hora de forma que podamos saber cuándo hemos creado el fichero. No obstante, no es necesario grabar la fecha y la hora, pues basta pulsar la tecla INTRO. El MS-DOS asumirá la fecha que haya sido visualizada en la pantalla.

Puede ocurrir que al escribir, nos hayamos equivocado y deseemos borrar un carácter; lo hacemos pulsando BACKSPACE. Si es una línea la que queremos eliminar, con ESCAPE. Cualquier modificación posterior en la fecha y la hora, se pueden realizar con las órdenes DATE y TIME (en algunas versiones del DOS el mandato es WTDATIM).

Después de dar la fecha y la hora aparecerá un mensaje parecido al siguiente:





A> es la petición de orden y nos indica que el control de la máquina ha pasado al sistema operativo, esta petición es originada por el procesador de comandos COMMAND.COM, que mencionaremos posteriormente. A partir de este momento, está esperando a que le demos una orden.

ORDENES DEL MS-DOS

¿Qué es lo que ha ocurrido hasta la aparición de A>? El S.O., como hemos dicho anteriormente, es un conjunto de programas, parte de los cuáles han sido cargados en la memoria del ordenador desde el disco, como por ejemplo los ficheros ocultos IBMBIO.COM, encargado de la transferencia de datos de memoria a las unidades e IBMDOS.COM, que gestiona los archivos. Así pues, se distinguen dos tipos de ordenes:

-Permanentes o internas: están siempre presentes en la memoria y por tanto, ejecutadas inmediatamente sin consultar el disco.

-Transitorias o externas: no están almacenadas en la memoria del ordenador, cuando es arrancado el MS-DOS.

Quizás resulte extraño que el S.O. no sea cargado íntegramente en la memoria. Cuando se invoca una orden, el proceso expuesto en una forma sencilla sería el siguiente: el MS-DOS busca en la memoria un fichero con ese nombre; si no lo encuentra, acude al disco, lo carga, ejecuta la orden y libera la memoria, con lo cual quedará más capacidad en la RAM para el ususario.

Es importante saber cuáles son órdenes internas y cuáles son externas; es decir, cuáles están permanentemente en memoria y cuáles hay que recurrir al disco del sistema para encontrarlas.

INTERNAS

ERASE (DEL)	RENAME (REN)	COPY
TYPE	DIR	DATE
TIME	VERIFY	VER
VOL	PROMPT	BREAK
CTTY	MKDIR (MD)	CHDIR (CD)
RMDIR(RD)	PATH	ECHO
PAUSE	IF	FOR
GOTO	REM	SHIFT
CLS	SET	

EXTERNAS

COMP	EDLIN	PRINT
MORE	FORMAT	SYS
CHKDSK	RECOVER	DISKCOPY
FDISK	BACKUP	RESTORE
MODE	ASSIGN	TREE
COMMAND	FIND	SORT
GRAPHICS	LINK	DEBUG
EXE2BIN		

Estas son algunas de las órdenes externas aunque puede haber variaciones según la versión del S.O.

Las órdenes internas y externas se pueden clasificar en grupos funcionales. Grupo de órdenes para el mantenimiento de ficheros, salida de herramientas de programación...

ORDENES MAS USUALES PARA EL MANTENIMIENTO DE FICHEROS

Estas son órdenes que tienen que ver con discos y directorios.

ORDEN DIR

La orden DIR visualiza en la pantalla una lista de los ficheros de un disco, excepto los ficheros ocultos del Sistema Operativo (normalmente IBMBIO.COM e IBMDOS.COM), y da información de sus tamaños y de cuando han sido actualizados por última vez.

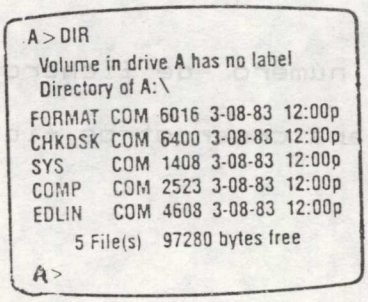
Se permiten varias sintaxis de la orden. Los ficheros pueden ser explorados usando caracteres ambiguos.

Primera expresión:

A>DIR

Nos mostrará la lista de todos los ficheros que están en la unidad A:, por defecto.

Ejemplo:

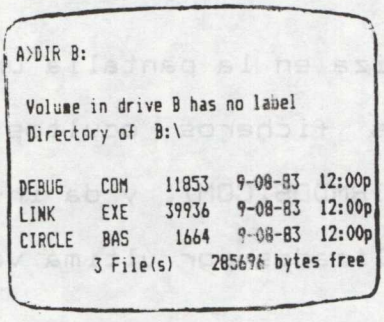


Segunda expresión:

A>DIR B:

Nos mostrará la lista de todos los ficheros que están en la unidad B:.

Ejemplo:



La primera línea es para especificar el rótulo del disco (Volume in drive A is) y si no está rotulado aparece " Volume in drive A has no label ".

En las dos primeras columnas está el nombre el nombre del fichero y la extensión. La tercera columna contiene el tamaño del fichero en bytes.

Las dos últimas columnas dan la fecha y la hora de la actualización más reciente del fichero.

Finalmente, aparece el número de ficheros y que capacidad del disco, en bytes, queda libre para crear otros ficheros.

Tercera expresión:

-/P

Este argumento hace que el listado del directorio haga una pausa al final de cada página.

```
A>DIR/P
 Volume in drive A has no label
 Directory of A:\
ANSI    SYS  1766 3-08-83 12:00p
FORMAT  COM  6016 3-08-83 12:00p
CHKDSK  COM  6400 3-08-83 12:00p
SYS     COM  1408 3-08-83 12:00p
COMP    COM  2523 3-08-83 12:00p
EDLIN   COM  4608 3-08-83 12:00p
MODE    COM  3139 3-08-83 12:00p
FDISK   COM  6177 3-08-83 12:00p
BACKUP  COM  3687 3-08-83 12:00p
RESTORE COM  4003 3-08-83 12:00p
PRINT   COM  4608 3-08-83 12:00p
Strike a key when ready ...
```

ORDEN FORMAT

La mayoría de los discos que se compran no están preparados para su uso. Necesitan ser "formateados". El formateo consiste en poner marcas electrónicas en el disco, de forma que el MS-DOS pueda encontrar los lugares correctos donde colocar los datos. Normalmente cada Sistema Operativo tiene su propio formato de discos, y este es un factor importante en cuanto a compatibilidad de periféricos

Los discos duros usualmente vienen ya formateados.

Primera expresión de FORMAT:

A>format B:

Formatea el disco que está en la unidad B:

El MS-DOS responderá a la orden FORMAT con la frase

A>DIR nombrefichero.extensión

Ej.:

```
A>DIR COMMAND.COM

 Volume in drive A has no label
 Directory of  A:\

COMMAND  COM    22042   6-14-84   8:00a
       1 File(s)     101376 bytes free

A>
```

Si el fichero se halla en el disco B:

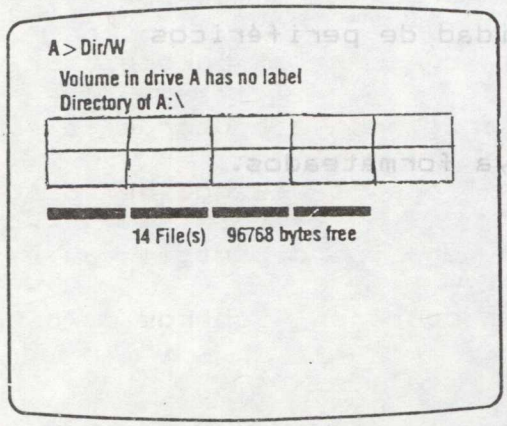
A>DIR B:nombrefichero.extensión

Uso de DIR con argumento:

-/W

Cuando el directorio tiene gran número de ficheros y no tenemos gran interés en el tamaño y fecha de actualización.

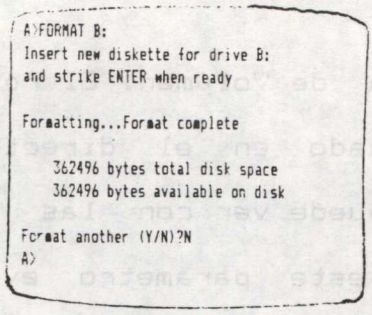
Ej.:




```
A>DIR/W

 Volume in drive A has no label
 Directory of  A:\

COMMAND  COM    ANSI     SYS    SORT     EXE    SHARE    EXE    FIND     EXE
ATTRIB   EXE    MORE     COM    ASSIGN   COM    PRINT    COM    SYS      COM
CHKDSK   COM    FORMAT   COM    VDISK    SYS    BASIC    COM    BASICA   COM
FDISK    COM    COMP     COM    TREE     COM    BACKUP   COM    RESTORE  COM
LABEL    COM    DISKCOPY COM    DISKCOMP COM    KEYBSP   COM    KEYBIT   COM
KEYBGR   COM    KEYBUK   COM    KEYBFR   COM    MODE     COM    SELECT   COM
GRAPHICS COM    RECOVER  COM    EDLIN    COM    GRAFTABL COM    CIRCULO  BAS
      35 File(s)     101376 bytes free
```



En algunas versiones aprecerá este y otros mensajes en castellano, por ejemplo: Otro formato (S/N)?

Si queremos formatear otro disco, lo colocamos en dicha unidad y pulsamos Y (yes), o en su caso S.

Segunda expresión, con parámetros:

-/s (A>FORMAT B:/S

A veces es interesante que el disco que nosotros formateamos contenga los ficheros del sistema, llamados ocultos porque no se listan cuando se da la orden DIR; es decir, con este disco se puede hacer la carga inicial del sistema conteniendo además de los ficheros ocultos un fichero externo, COMMAND.COM, que es el procesador de comandos que analiza los mismos, necesario para ejecutar el MS-DOS. Esta opción de cargar en memoria los tres ficheros basicos del Sistema Operativo, es normalmente más conveniente en los diskettes que van a contener programas y no en los que vayan a contener ficheros de datos.

-/V

Este parámetro permite poner un rótulo de volumen al disco,

denominado a veces Etiqueta de Volúmen. El rótulo es un dato de 11 caracteres como máximo grabado en el directorio del disco para identificar su contenido. Se puede ver con las órdenes DIR, VOL y CHKDSK. La utilización de este parámetro evita el tener que usar posteriormente el mandato LABEL, si se desea incluir en el diskette una etiqueta de Volúmen. (El mandato LABEL sirve para crear,cambiar o suprimir la etiqueta).

-/B

Este parámetro hace que FORMAT cree un diskette de 8 sectores por pista, con espacio asignado para los módulos ocultos del sistema (denominados en IBM como IBMBIO.COM e IBMDOS.COM). Con esta opción se crea un diskette sin sistema, pero en el que posteriormente se podrá crear cualquier versión del DOS a traves del mandato SYS (en este caso >SYS B:). De no utilizarse esta opción posteriormente no tendrá utilidad el mandato SYS, para instalar el DOS en un disco ya formateado, pues los ficheros ocultos necesitan estar en unos sectores de unas pistas determinadas que habría que reservar con esta opción al formatear el disco.

Observaciones:

- Si formateamos un disco usado, borra toda su información previa.
- Si queremos formatear un diskette de una sola cara y nuestra unidad de disco es de doble cara, hay que usar el argumento /1 (A>FORMAT B:/1). Los diskettes formateados por ambas caras no son compatibles

con unidades físicas de una sola cara, pero sí a la inversa.

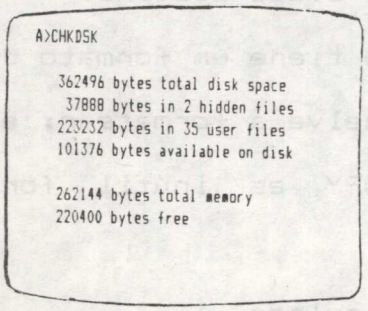
- Los diskettes formateados con antiguas versiones del DOS, son compatibles con la última, excepto los recientes de alta capacidad que son incompatibles por Hardware con unidades que no sean de alta capacidad, los diskettes formateados con recientes versiones del DOS no de pueden utilizar con la versión DOS.1 (El formato es de 9 sectores por pista mientras que en la antigua versión era de 8).

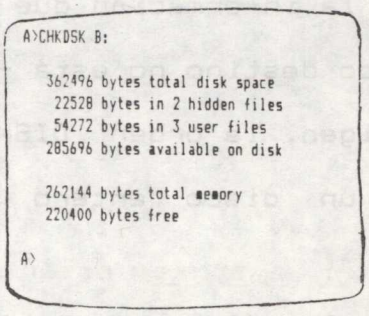
ORDEN CHKDSK

La orden CHKDSK informa de lo siguiente:

- Del tamaño del disco.
- Del espacio ocupado en el disco por ficheros ocultos y por ficheros visibles en el directorio.
- Del espacio libre del disco.
- De la RAM disponible.
- Avisa y, a veces, repara muchos errores internos del disco.

Sintaxis, para el chequeo del disco en la unidad A, y posteriormente en la unidad B:





La orden CHKDSK se puede encontrar con errores internos del disco. Si queremos que nos vaya informando de los nombres de los ficheros que está examinando en el directorio, mientras avanza, usamos junto a ella el argumento /V. Se usaría tal argumento después de examinar el disco y encontrar errores con el CHKDSK. Es decir, primeramente

A>CHKDSK

y si aparecen errores nuevamente

A>CHKDSK/V.

Se puede utilizar dicha orden con el nombre y extensión de un fichero.

ORDEN DISKCOPY

La orden DISKCOPY copia un disco entero en otro.

Observaciones:

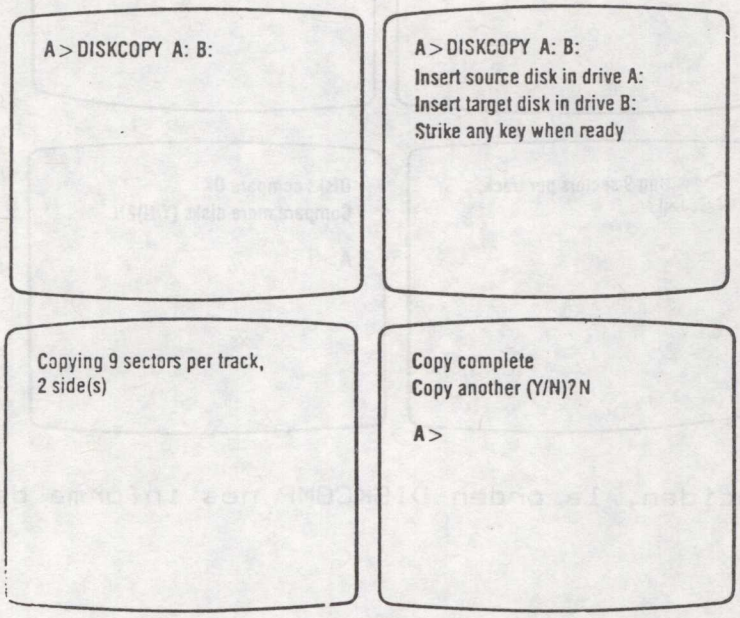
- Borra toda la información que había en el disco destino.
- Si el disco destino no está formateado o tiene un formato diferente del disco origen, la orden DISKCOPY lo vuelve a formatear; es decir, para copiar un disco entero con DISKCOPY es inútil formatearlo previamente.
- Copia todos los ficheros, incluidos los ocultos.

- Verifica los datos después de escribirlos.
- No funciona con disquetes protegidos contra copia.

Sintaxis:

A>DISKCOPY A: B:

Copia el disco origen, que está en la unidad A:, en el disco destino que está en B:



ORDEN DISKCOMP

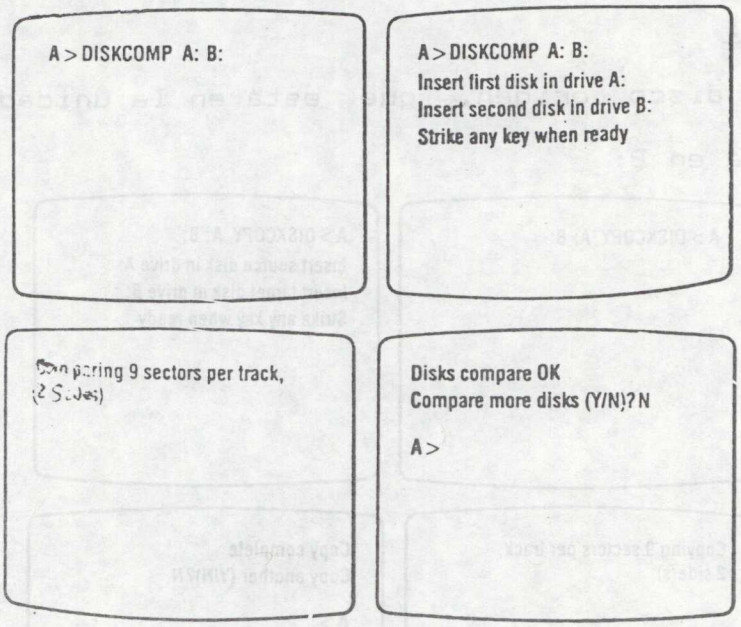
Para asegurarnos de que una copia está bien hecha, aunque DISKCOPY verifica los datos después de escribirlos, podemos usar DISKCOMP para comparar los discos sector a sector.

Sintaxis:

A>DISKCOMP A: B:

Compara sector a sector el disco que se halla en la unidad A: con el que se halla en la unidad B:.

Si los discos coinciden exactamente aparecerá en pantalla Diskettes compare OK



Si no coinciden, la orden DISKCOMP nos informa de la pista y cara donde hay error

ORDENES DE MANTENIMIENTO DE FICHEROS

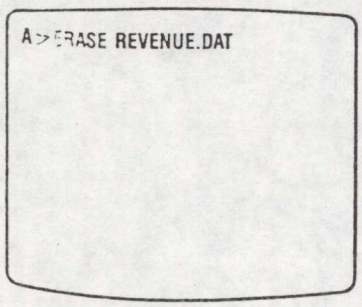
ORDEN ERASE O DEL

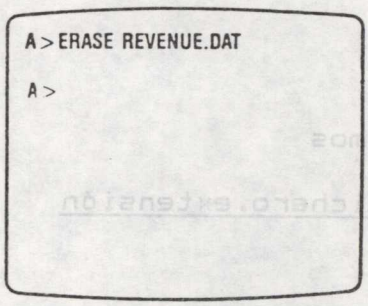
La orden ERASE o DEL elimina un fichero o un grupo de ficheros del directorio de un disco.

Primera expresión:

A>ERASE nombrefichero.extensión

El MS-DOS entiende, por defecto, que deseamos borrar el fichero con especificador "nombrefichero.extensión" del disco que se halla en la unidad A:.

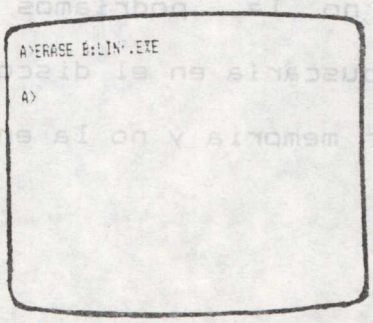




Si queremos eliminar un fichero del disco que se halla en la unidad B:, el formato sería

A>ERASE B:nombrefichero.extensión







O bien, como se trata de una orden permanente, podemos cambiar de unidad de disco de la forma:

A>B:

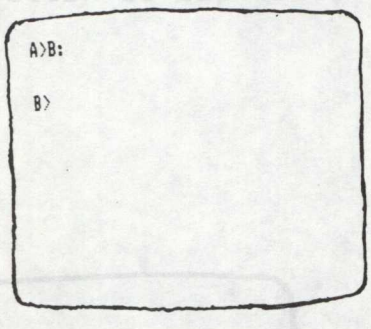
aparecerá

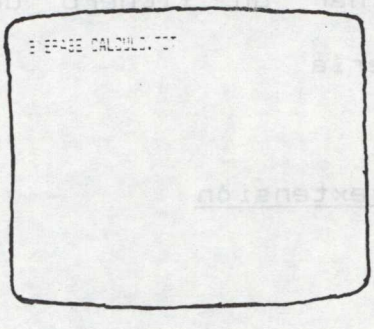
B>

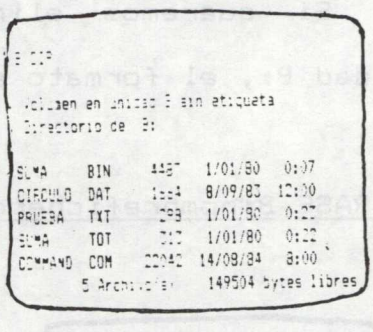
y ahora escribimos

B>ERASE nombrefichero.extensión

Ej.:

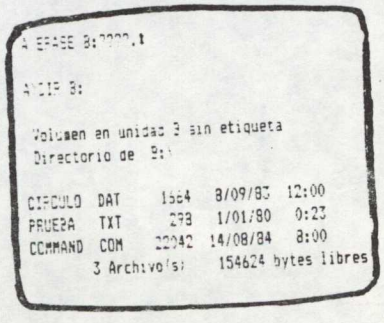






Esta operación no la podríamos hacer si la orden fuese transitoria. El S.O. buscaría en el disco que está en la unidad B: tal orden para cargarla en memoria y no la encontaría puesto que está en A: con el MS-DOS.

Segunda expresión, con caracteres ambiguos:



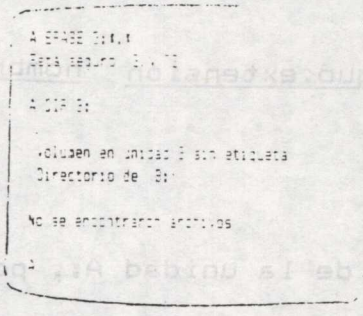
Como es extraño borrar todos los ficheros de un disco, al dar la orden :

A>ERASE *.*

nos preguntará

Are you sure (Y/N)? Y

En algunas versiones aparecerá en castellano. ¿Está seguro? ¿(S/N)?.



Mensajes de error al utilizar la orden ERASE

- Invalid number of parameters (número de parámetros no válido).
- File not found (fichero no encontrado).



ORDEN RENAME O REN

Al utilizar la orden RENAME o REN cambia el nombre de un fichero sin modificar el contenido.

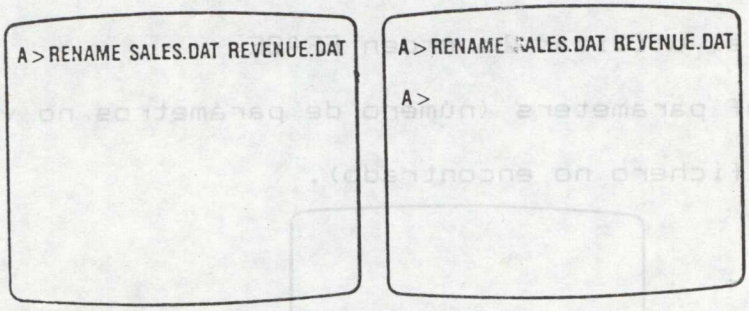
Hay muchas razones por las que queremos cambiar el nombre de un fichero. El contenido de un fichero puede haber cambiado tanto que el nombre antiguo ya no es adecuado. O bien, nos podemos dar cuenta que tenemos dos ficheros con nombres similares; cambiamos el nombre a uno de ellos para evitar confusión.

Sintaxis:

Primera expresión:

A>RENAME nombreficheroantiguo.extensión nombreficheronuevo.extensión

Se refiere a ficheros de la unidad A:, por defecto.

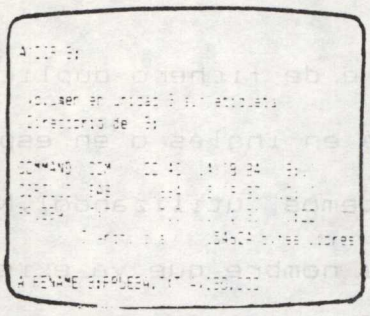


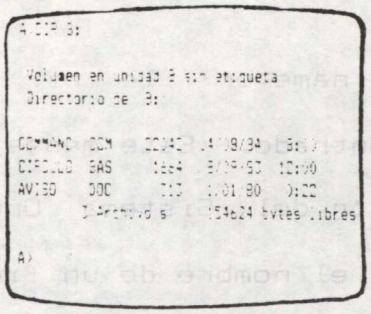
Segunda expresión:

A>RENAME B:nombreantiguo.extensión nombrenuevo.extensión

Observemos que no es necesario la unidad en la que va a estar el fichero con nombre "nuevonombre". Entiende, por defecto, que se trata de la unidad B:, puesto que no se puede cambiar los nombres de ficheros que están en discos diferentes.

Ej.:





Tercera expresión:

A veces necesitamos intercambiar los nombres de dos ficheros. En este caso, hay que utilizar un fichero temporal o intermediario.

Por ejemplo, supongamos que queremos intercambiar los nombres de los ficheros CARTANA.TXT y CARTANA.BAK (copia de seguridad). Se escribirá:

A>RENAME CARTANA.TXT TEMPORAL.TXT

A>RENAME CARTANA.BAK CARTANA.TXT

A>RENAME TEMPORAL.TXT CARTANA.BAK

Cuarta expresión:

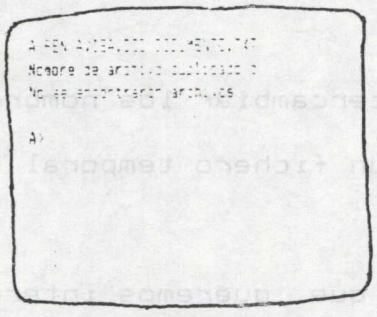
Se pueden utilizar caracteres ambigüos si dos o más ficheros comparten o bien el nombre o bien la extensión.

Ej.:

A>RENAME CARTANA.* CARTA.*

Mensaje de error:

-Duplicate file name or file not found. Nombre de fichero duplicado o fichero no encontrado. (Este mensaje aparecerá en inglés o en español, según que versión del Sistema Operativo estemos utilizando).No se puede cambiar el nombre de un fichero a otro nombre que ya exista en el disco.



ORDEN COPY

La frecuencia con que se utiliza la orden COPY viene explicada por las siguientes operaciones que realiza:

- Copia ficheros de un disco a otro .
- Copia un fichero en el mismo disco pero con nombre diferente.
- Copia informción en los dispositivos físicos (dispositivos hardware) y de los dispositivos.
- Combina muchos ficheros en uno solo.

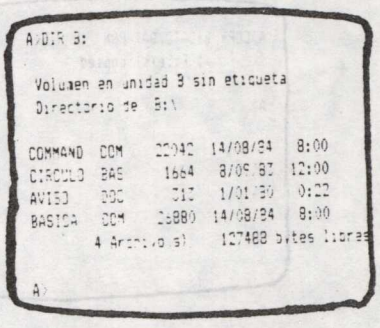
Primera expresión.

A>COPY nombrefichero.extensión B:

Copia el fichero de especificador "nombrefichero.extensión" en el disco de la unidad B:

Ej.





La orden COPY siempre nos dice cuántos ficheros ha copiado. Se observa que no es necesario dar el nombre del fichero en el disco de destino; es suficiente con nombrar la unidad. La orden COPY asume que el fichero copiado tiene el mismo nombre.

Segunda expresión

Si el fichero se halla en el disco de la unidad B: y queremos copiarlo en la unidad A:, con el mismo nombre, escribiríamos :

A>COPY B:nombrefichero.extensión

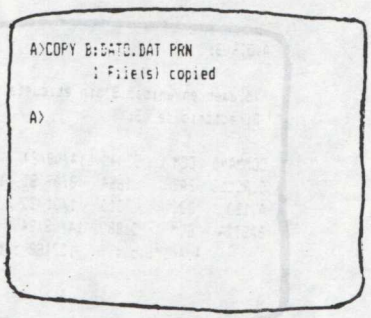
Tercera expresión

Cuando queremos que el MS-DOS lea o escriba en los dispositivos físicos:

A>COPY nombrefichero.extensión dispositivofísico:

Copia el fichero con especificador del nombre indicado en el dispositivo de nombre "dispositivofísico".

La orden COPY tratará a los dispositivos físicos como si fueran ficheros.



Caso recíproco:

A>COPY dispositivofísico: nombrefichero.extensión

Lee del dispositivo y lo graba en el fichero con especificador "nombrefichero.extensión". Por ejemplo si introducimos >COPY CON: nombrefichero.TXT, estaremos creando un fichero a través del teclado.

¡OBSERVACION! La orden COPY continuará leyendo del dispositivo hasta que seintroduzca una señal que indique fin de fichero. Esta es "Control+Z" y a continuación "return".



Cuarta expresión:

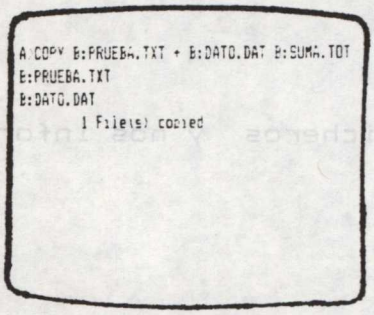
Sintaxis de la orden COPY cuando queremos concatenar ficheros; es

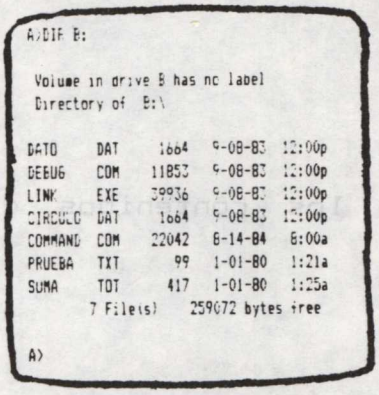
decir, conectar un fichero al final de otro:

A>COPY nombrefichero1.extensión +
nombrefichero2.extensión nuevo nombre.extensión

Copia el fichero de especificador "nombrefichero1.extensión" y a continuación el fichero "nombrefichero2.extensión" y le da el nombre "nuevonombre.extensión".

Ej.:





Quinta expresión:

La orden COPY se puede utilizar con el argumento /V para asegurarnos de que la copia ha sido correcta.

Ej.:

A>COPY cartana.txt B: /V

1 file(s) copied (1 fichero copiado, en versiones españolas).

El MS-DOS vuelve a leer lo que ha escrito y nos asegura que, efectivamente, ha copiado ese fichero.

Nos dará un mensaje de error cuando intentamos copiar un fichero que no existe:

File not found (Fichero no encontrado).

Es necesario prestar atención para no confundir el disco destino con el origen.

ORDEN COMP

Compara los contenidos de dos ficheros y nos informa de las diferencias.

Primera expresión:

A>COMP nombrefichero1.extensión1 nombrefichero2.extensión2

Compara ambos ficheros de la unidad A:, por defecto.

Segunda expresión:

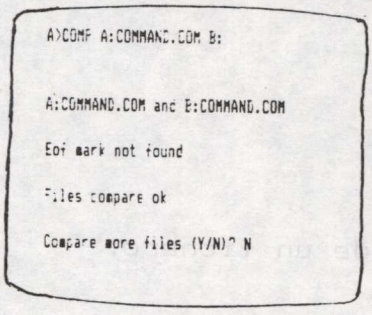
A>COMP A:nombrefichero.extensión B:

Compara el fichero del disco de la unidad A:, con el nombre indicado con otro fichero del mismo nombre en la unidad B:.

Generalmente esta expresión se utiliza después de hacer una

copia.

Ej.:



Tercera expresión:

A>**COMP** B:nombrefichero1.extensión1 B:nombrefichero2.extensión2

Compara ambos ficheros de la unidad B:.

Se pueden utilizar caracteres ambiguos.

Ej.:

A>COMP *.* B:

Compara todos los ficheros del disco A: con los de B:.

Entre cada conjunto de ficheros comparado nos preguntará si deseamos seguir con la comparación.

ORDEN EDLIN

La orden EDLIN es un editor de líneas que se puede usar para:

-Crear ficheros.

-Modificar el contenido de un fichero.

Es una orden transitoria, por tanto, si quiero crear un fichero en la unidad B: , que no tiene incorporada la orden EDLIN, necesitamos el disco con el S.O. en la unidad A:.

La sintaxis de la orden sería la siguiente:

A>EDLIN B:nombrefichero.extensión

Nos abre en el disco de la unidad B: un fichero con el especificador "nombrefichero.extensión". A continuación, le damos la orden "I" (insertar) seguido de "return", quedando ya dispuesto para escribir el fichero o programa.

La grabación hay que efectuarla con la orden "E", después de haber indicado el fin del programa pulsando simultáneamente ,CONTROL + C, o bien la tecla funcional F6. Al salir con la orden "E" del Edlin se graba el fichero en disco, así como una copia del anterior, en el caso de que previamente ya existiera el fichero con la extensión .BAK

Si queremos listar por pantalla el contenido del fichero a partir

de una determinada linea se teclea el número de línea y "L" (listar), si se desea borrar una determinada línea se teclea el número de línea y "D" (delete).

Si durante la escritura del fichero nos arrepentimos y queremos destruirle, pulsamos CONTROL + C y a continuación le damos la orden "Q", que nos pedirá confirmación del deseo de destruir el fichero.
Si disponemos de un procesador de Textos o un Editor de Pantalla tendremos mayores facilidades que con el Editor de Líneas, si bien son programas Software que no incorpora el Sistema Operativo.

Ejemplo de Edlin:

```
A>EDLIN B:CALCULO.BAS
New file
*I
       1:*LET S = 0
       2:*LET S = S + 1
       3:*PRINT "LA SUMA ES..."; S
       4:*^C

*E

A>
```

```
A>EDLIN B:CALCULO.BAS
End of input file
*2L
       2: LET S = S + 1
       3: PRINT "LA SUMA ES..."; S
*2D
*1L
       1: LET S = 0
       2:*PRINT "LA SUMA ES..."; S
*E

A>
A>
```

```
A>EDLIN B:CALCULO.BAS
New file
*I
       1:*LET S = 0
       2:*LET S = S + 2
       3:*^C

*Q
Abort edit (Y/N)? Y
A>
```

ORDENES DE SALIDA DE FICHEROS

Son las órdenes TYPE, PRINT y MORE. Se usan para visualizar ficheros en la pantalla o imprimirlos en el papel.

El inconveniente que tienen estas órdenes es que son para visualizar un fichero de principio a fin, no permitiéndonos volver atrás y releer un trozo de un fichero. Por ello, para hacer una modificación, usaremos el editor EDLIN.

ORDEN TYPE

La orden TYPE visualiza el contenido de un fichero en la pantalla.

No todos los ficheros pueden ser visualizados en la pantalla; han de estar escritos en el conjunto de caracteres ASCII, sino visualizará caracteres ilegibles en la pantalla.

Primera expresión:

A>TYPE nombrefichero.extensión

Visualiza el fichero de especificador "nombrefichero.extensión" que está en la unidad A:, por defecto.

Segunda expresión:

A>TYPE B:nombrefichero.extensión

Visualiza el fichero con el especificador indicado, que está en la unidad B:.

Ejemplo:

```
A>TYPE B:CALCULO.BAS
LET S = 0
PRINT "LA SUMA ES..."; S

A>
```

Si el fichero tiene más lineas de las que caben en la pantalla, la orden TYPE hace un desplazamiento tan rápido que es imposible leerlo. Para fijar la pantalla se pulsa simultáneamente las teclas CTRL + NUMLOCK.

ORDEN PRINT

La orden PRINT imprime ficheros en la impresora y permite continuar trabajando con el MS-DOS. Amite también, una cola de ficheros que después se irán imprimiendo uno detrás de otro.

Primera expresión:

A>PRINT nombrefichero.extensión

Nos preguntará el nombre del dispositivo al cual le enviaremos los ficheros. Para la puerta de la impresora en paralelo el nombre es PRN.

Segunda expresión

A>PRINT B:nombrefichero.extensión

```
A>PRINT B:CALCULO.BAS
Name of list device [PRN]:
Resident part of PRINT installed

    B:\CALCULO.BAS is currently being printed

A>
```

```
LET S = 0
PRINT "LA SUMA ES..."; S
```

- En castellano, estos mensajes serían:

Nombre del dispositivo de impresión [PRN]:

Parte residente del PRINT instalada

B:\CALCULO.BAS está siendo impreso

Tercera expresión:

A>PRINT B:nombrefichero1.extensión1 B:nombrefichero2.extensión2

Se pueden introducir hasta diez ficheros e incluso, usar caracteres ambiguos. Es muy util para poder utilizar la consola, al tiempo que se efectúan impresiones de ficheros.

Ejemplo:

```
A>PRINT B:SUMA.TOT B:CALCULO.BAS

   B:\SUMA.TOT is currently being printed
   B:\CALCULO.BAS is in queue

A>
```

```
Este texto es un fichero de texto en ASCII que se grabará con el nombre "prueba" y la extensión txt
LET S = 0
PRINT "LA SUMA ES..."; S
```

- En castellano, estos mensajes serían:

B:\SUMA.TOT está siendo impreso

B:\CALCULO.BAS está en cola

Para volcar la pantalla en la impresora no es necesario utilizar esta orden. Se pulsa simultáneamente las teclas SHIFT + PRTSC.

Se puede activar la impresora pulsando simultáneamente CONTROL+P

ORDEN MORE

La orden MORE escribe un fichero en el monitor, ocupando una pantalla cada vez y espera a que pulsemos una tecla para continuar.

Sintaxis:

A>MORE <B:nombrefichero.extensión

```
A>MORE <CUENTA.BAS

FOR N=1 TO 10
PRINT N
NEXT N

A>
```

No debemos olvidarnos del símbolo <

FICHEROS BATCH PARA TRATAMIENTO POR LOTES

Si queremos ejecutar varias órdenes, una tras otra, se puede ahorrar tiempo creando un fichero de extensión BAT, denominado fichero batch o fichero para el tratamiento por lotes, en el cuál están contenidas dichas órdenes.

Se crea el fichero utilizando el editor EDLIN, dándole un nombre y la extensión BAT.

¿Qué programas se pueden incluir en un fichero batch? Cualquier orden o programa del MS-DOS de los que se ejecutan a partir de la petición de orden A>. cada orden tiene que estar en una línea separada.

Ejemplo:

```
A>EDLIN B:START.BAT
New file
*I
       1:*CLS
       2:*REM "BIENVENIDO AL MS-DOS"
       3:*KEYBSP
       4:*^C

*E

A>
```

Además de las ordenes vistas, se pueden introducir en el fichero batch, las siguientes órdenes específicas de ficheros batch:

- ECHO: Escribe un mensaje en la pantalla. Con la opción ECHO OFF no se ven los mensajes del fichero batch cuando este se ejecute.
- PAUSE: Visualiza el mensaje "Strike a key when ready" (Pulse cualquier tecla para continuar). Se usa para dar tiempo a cambiar tipo de papel, diskette, etc...
- REM: Añade comentarios a un fichero batch.
- IF
- FOR
- GOTO
- SHIFT
- COMMAND

FICHERO AUTOEXEC.BAT

Es un fichero batch que se ejecuta automáticamente cuando se hace la carga inicial del sistema desde el disco. Este fichero puede contener todas las órdenes que uno quiera. A continuación, cada vez que se haga la carga inicial del sistema el MS-DOS buscará el fichero AUTOEXEC.BAT y si lo encuentra ejecutará inmediatamente las órdenes, de no existir este fichero se realizará la carga por defecto del S.O.

Vamos a crear un fichero AUTOEXEC.BAT en un disco nuevo:

- Formateamos el disco con el parámetro /S, con lo cual copiamos los programas ocultos del sistema operativo y el programa COMMAND.COM.

```
A>FORMAT B:/S
Insert new diskette for drive B:
and strike ENTER when ready
```

```
Formatting...Format complete
System transferred

    362496 bytes total disk space
     60416 bytes used by system
    302080 bytes available on disk

Format another (Y/N)?N
A>
```

- Abrimos un fichero ,en el disco usuario, AUTOEXEC.BAT con el editor EDLIN, introducciendo las órdenes que queramos.(conviene introducir al principio el comando KEYBSP, para inicializar el teclado español, al objeto de que funcione correctamente el teclado de los caracteres especiales).

```
A>EDLIN B:AUTOEXEC.BAT
New file
*I
       1:*KEYBSP
       2:*DATE
       3:*TIME
       4:*CLS
       5:*VER
       6:*^C

*E

A>
```

```
A>COPY KEYBSP.COM B:
        1 File(s) copied

A>
```

Se puede comprobar la autoejecución apagando la máquina y poniéndola en funcionamiento con nuestro disco.

Ejemplo, si pulsamos CTRL+ALT+DEL:

```
A>KEYBSP

A>DATE
Current date is Tue  1-01-1980
Enter new date (mm-dd-yy):

A>TIME
Current time is  0:00:17.02
Enter new time:
```

```
A>VER

IBM Personal Computer DOS Version  3.00

A>
A>
```

PRACTICAS DE SISTEMA OPERATIVO MS-DOS

Se trabajará con un diskette que contenga información, con varios ficheros y cuyos datos no se consideren valiosos (Diskette de Prueba),realizando las siguientes prácticas.

1.- Manejo del **DIR** con y sin parámetros (fijarse en la información de cada columna).

2.- ERASE

a) Ver el directorio del diskette.

b) Borrar un fichero.

c) Ver de nuevo el directorio y comprobar que se ha borrado el fichero.

d) Repetir estos pasos utilizando caracteres ambigüos

3.- RENAME

I. a) Visualizar el directorio.

b) Cambiar el nombre a un fichero.

c) Visualizar de nuevo el directorio comprobando que la operación ha sido bien realizada.

II. Utilizar caracteres ambigüos para cambiar genéricamente el nombre de la extensión de un grupo de ficheros a otra. Comprobación

de la bondad de la operación.

III. Intercambiar los nombres de dos ficheros en la misma unidad de diskette, creando para ello un fichero intermedio.

4.- COPY

a) Copiar un fichero de un diskette en la unidad A a un diskette en la unidad B. Comprobar el resultado de la operación con el comando DIR.

b) Repetir la opción, pero cambiando el nombre al grabar el fichero en la unidad B, utilizando además el parámetro //V.

c) Copiar un fichero que esté en un diskette de la unidad B a la unidad A.

d) Realizar una copia utilizando caracteres ambigüos

e) Grabar directamente un texto creado con el dispositivo consola (a través del teclado) en fichero del diskette.

f) Imprimir directamente un texto creado con el dispositivo consola en la impresora (PRN:).

g) Concatenar dos ficheros en uno nuevo.

5.- TYPE

a) Visualizar el contenido de un fichero en ASCII

b) Utilizar el MORE en caso de visualizar un fichero de formato ASCII (texto), que no quepa en una pantalla.

6.- PRINT

- Imprimir dos ficheros, formando una cola de impresión.

7.- CHKDSK

a) Chequear un diskette.

b) Chequear un fichero del diskette, comprobando si son o no contigüos los sectores que ocupa en el diskette.

8.- DATE Y TIME

a) Visualizar y cambiar la fecha.

b) Visualizar y cambiar la hora.

9.- VOL

a) Introducir una nueva Etiqueta al Volúmen

10.- EDLIN

a) Crear un pequeño fichero con la extensión .BAS (es decir, crear un programa en BASIC con el Editor del Sistema Operativo).

b) Crear un fichero del tipo .TXT.

c) Abandonar la creación del fichero de txto

11.- FICHEROS BATCH

I. Crear un fichero Batch, que efectúe las siguientes operaciones:

a) DIR

b) PAUSE

c) TYPE, de alguno de los ficheros creados anteriormente

II. Hacer un fichero AUTOEXEC.BAT que se ejecute automáticamente cuando se hace la carga inicial del Sistema (No olvidar el COMMAND.COM, ni el WTDATIM)

III. Crear un fichero (AUTOEXEC.BAT) que lance directamente la ejecución del programa en Basic, creado con el Editor (No olvidar el comando BASICA).

ANEXO : SISTEMA OPERATIVO MS-DOS

A continuación se incluye un Resúmen de las funciones y teclas de control, de utilidad para el manejo del teclado del ordenador con el Sistema Operativo MS-DOS. Asimismo se muestran las teclas de Edición del DOS que se pueden utilizar a partir del símbolo >.

Asimismo se acompaña una guía de Referencia de los comandos del MS-DOS y escueta descripción de los mismos.

Finalmente una guía de los comandos específicos para tratamiento por lotes (ficheros .BAT), y del Editor de líneas Edlin.

FUNCIONES Y TECLAS DE CONTROL

Alt + Ctrl + Del	Reinicializa el sistema
Alt + Ctrl + Esc	Llama al monitor en ROM
Ctrl + Scroll Lock/Break	Interrumpe y termina la operación en curso
Ctrl + C	Interrumpe y termina la operacion en curso
↑ + PrtSc	Imprime el contenido de la pantalla en la impresora
Ctrl + P	Activa/desactiva eco sobre impresora
Ctrl + ↵	Continua el comando en la siguiente linea de pantalla
Ctrl + J	Continua el comando en la siguiente linea de pantalla
Esc	Cancela la linea escrita
Ctrl + S	Suspende la salida de datos en pantalla. Para continuar pulse cualquier tecla
Ctrl + Num Lock	Suspende la salida de datos en pantalla. Para continuar pulse cualquier tecla
↹	Mueve el cursor ocho espacios a la derecha
←	Borra carácter y retrocede un espacio
Ctrl + H	Borra caracter y retrocede un espacio
Alt + Ctrl + F1	Activa el controlador de teclado americano
Alt + Ctrl + F2	Activa el controlador de teclado espanõl

TECLAS DE EDICION DEL DOS

→ o F1	Visualiza un caracter de la plantilla
F2	Visualiza los caracteres de la plantilla hasta el carácter especificado
F3	Visualiza el resto de los caracteres de la plantilla
Del	Salta un carácter en la plantilla. (No lo copia)
F4	Salta (no copia) los caracteres de la plantilla hasta el carácter especificado (lo contrario de F2)
Esc	Cancela la linea escrita. No cambia el contenido de la plantilla
Ins	Activa y desactiva la introducción
F5	Pone la linea tecleada como nueva plantilla

NOTACIONES

Para llegar a una mejor comprension de como introducir los comandos del DOS, usaremos las siguientes notaciones.

1. Las palabras en MAYUSCULAS (palabras clave) son las que Vd. debe teclear tal y como se expresan aqui. Estas palabras clave pueden introducirse usando tanto las teclas mayusculas como las minusculas o bien una combinacion de ambas. DOS convierte todas las palabras clave a mayúsculas.

2. Los campos expresados en **letra minúscula**, indican el texto que debe Vd. suministrar. Por ejemplo Vd. debera teclear el nombre de un fichero cuando el campo "nombre del fichero" aparezca en el formato.

3. Los campos que van enmarcados por corchetes [ ] son opcionales. Si Vd. desea incluir alguna información opcional, no incluya los corchetes, sólo deberá incluir la información enmarcada por éstos.

4. Los puntos suspensivos ... indican que se puede repetir un campo tantas veces como se desee.

5. Incluya todos los signos de puntuación tales como comas, signos de igualdad, interrogaciones, dos puntos, barras inclinadas, etc. en el lugar en que se muestran.

6. Una barra vertical sirve para separar múltiples argumentos de los cuales Vd. debe elegir solamente uno por ejemplo ON OFF indica que se puede seleccionar uno de los dos parámetros como parte del comando.

COMANDOS DEL DOS

ASSIGN [d = d[...]]	*Redirecciona las peticiones de una unidad a otra*
ATTRIB [+/-R][d:] [itinerario] nombre del fichero [.ext]	*Cambia un fichero pasando a poseer proteccion contra escritura*
BACKUP [d:] [itinerario] [nombre del fichero] [.ext] d: [/S][/M][/A][/D:dd-mm-aa]	*Traspasa ficheros del disco duro a discos flexibles con formato DOS*
BREAK [ON|OFF]	*Activa o desactiva la comprobación de la función de Interrupción (o Ctrl-C)*
CHDIR [d:][itinerario]	*Cambia el directorio de la unidad actual o la designada al directorio indicado por "itinerario". Visualiza el directorio actual*
CHKDSK [d:][nombre del fichero][/F][/V]	*Explora el directorio de la unidad de disco especificada y comprueba su consistencia; proporciona un informe del espacio en memoria y disco*
CLS	*Borra la pantalla del monitor*
COPY [/A][/B][d:] [itinerario] nombre del fichero [.ext][/A][/B][d:] [itinerario][nombre del fichero][.ext][/A][/B][/V]	*Copia uno o más ficheros al mismo disco o a otro. Se pueden dar nombres diferentes a las copias*
CTTY dispositivo	*Cambia el dispositivo a través del cual se introducen los comandos (TTY representa el monitor y el teclado)*
DATE [dd|mm|aa]	*Permite visualizar o poner la fecha. Esta fecha quedará registrada en el directorio de cualquier fichero que se cree o se modifique*

DIR[c:][itinerario] [nombre del fichero] [/P][/W]	*Visualiza una lista o directorio de todos los ficheros que estan en el disco flexible*
DISKCOMP [d:][d:] [/1][/8]	*Compara el contenido de dos discos flexibles*
DISKCOPY [d:][d:][/1]	*Copia el contenido de la unidad de origen en la unidad de destino*
ERASE [d:][itinerario] [nombre del fichero]	*Borra todos los ficheros que tengan el nombre indicado*
EXE2BIN nombre del fichero [d:][nombre del fichero [.ext]]	*Convierte ficheros del tipo .EXE, ejecutables, a formato ordinario binario. Con esto se consigue ahorrar espacio en disco y una carga mas rapida del programa*
EXIT[c:][itinerario]	*Sale del programma COMMAND.COM, (el procesador de comandos) y vuelve al nivel anterior al de COMMAND.COM, si existe un procesador secundario*
FIND [/V][/C][/N] "cadena caracteres" [[d:] [itinerario]nombre del fichero]	*Busca la cadena de caracteres especificada, en uno o varios ficheros*
FORMAT [d:][/S][/8][/1] [/V][/I][/C]	*Inicializa y da formato al disco de la unidad especificada, de forma que pueda aceptar ficheros con formato DOS*
GRAPHICS	*Habilita la impresión del contenido de una pantalla de graficos en una impresora matricial (volcado de imagen de elementos)*
LABEL[d:][etiqueta de volumen]	*Crea, cambia o borra una etiqueta de volumen en un disco*
MKDIR [d:]itinerario	*Crea un nuevo directorio*
MORE	*Visualiza los datos de salida dirigidos al monitor, pantalla a pantalla*
PATH[[d:]itinerario[;[d] itinerario]...]	*Define un itinerario de busqueda de los comandos*
PRINT [[d:]nombre del fichero][/T][/C][/P]...]	*Imprime un fichero de texto mientras se procesan otros comandos del DOS. (Esta caracteristica se denomina "impresion simultanea")*
PROMPT [texto]	*Cambia el mensaje del DOS en pantalla*
RECOVER[d:][itinerario] nombre del fichero	*Recupera un fichero o un disco completo que contenia sectores en mal estado*
REN[**AME**] [d:][itinerario] nombre del fichero	*Cambia el primer nombre de fichero por el segundo nombre del fichero*
RESTORE d:d:[itinerario] [nombre del fichero] [.ext][/S][/P]	*Restaura al disco duro los ficheros de un disco flexible grabados previamente con el comando BACKUP*
RMDIR [d:]itinerario	*Elimina un directorio de una estructura jerarquica de directorios*
SET [cadena caracteres = cadena caracteres]	*Hace equivalentes dos cadenas de caracteres para su uso en programas posteriores*

COMANDOS PARA TRATAMIENTO POR LOTES

SETUP SETUP LPTn:[c],[l],[P] SETUP modalidad,[s],[T] SETUP COMn[:[baudios], [paridad],[bits de datos], [bits de parada],[P]] SETUP LPTn:=COMn	*Fija los parámetros de la impresora, monitor y comunicaciones asíncronas*
SORT [/R][/+n] [‹antiguo nombre fichero› nuevo nombre fichero]	*Clasifica datos en código ASCII, hacia delante o hacia atrás, escribiéndolos a continuación en la pantalla o en un fichero*
SPINDOWN	*Prepara el disco duro para su transporte*
SYS d:	*Transfiere los ficheros del sistema DOS del disco en la unidad por omisión, al disco en la unidad especificada por d:*
TIME [hh[:mm:ss:xx]]	*Visualiza o fija la hora*
TREE [d:][/F]	*Visualiza la etiqueta del volumen y cada uno de los subdirectorios que existan en la unidad designada*
TYPE [d:][itinerario] nombre del fichero	*Visualiza el contenido de un fichero*
VDISK	*Divide una parte de la memoria del sistema en unidades de disco simuladas*
VER	*Imprime el número de versión del DOS*
VERIFY [ON]IOFF]	*Activa o inhibe la verificacion de los datos que se graban en el disco*
VOL[d:]	*Visualiza el nombre o número de identificación del volumen, si existe*

ECHO [ONIOFF Imensaje]	*Activa y desactiva la caracteristica de eco en ficheros de tratamiento por lotes*
FOR%% cIN[conjunto] DO comando	*Iteraciones en proceso de tratamiento por lotes*
FOR% c IN [conjunto] DO comando	*Iteraciones en proceso interactivo*
GOTO etiqueta	*Transferencia de control en proceso de tratamiento por lotes*
IF comando condicion	*Ejecucion condicional en proceso de tratamiento por lotes*
PAUSE [comentario]	*Detiene el proceso de tratamiento por lotes hasta que se pulse una tecla*
REM[comentario]	*Visualiza el contenido durante la ejecucion de un fichero de tratamiento por lotes*
SHIFT	*Permite acceder a mas de diez parametros sustituibles en el proceso de ficheros de tratamiento por lotes*

COMANDOS DEL EDLIN

[n] A	*Añade líneas*
[línea],[línea],línea, [reces] C	*Copia línea*
[línea][,línea] D	*Borra línea*
[línea]	*Edita línea*
E	*Finaliza la edición*
[línea] I	*Introduce línea*
[línea][,línea] L	*Lista línea*
[línea],[línea],línea M	*Mueve líneas*
[línea][,línea] P	*Página hasta 23 línea*
Q	*Abandona la edición*
[línea][.línea][?] R [cadena caracteres 1] [F6 cadena caracteres 2]	*Sustituye texto*
[línea][,línea][?] S cadena caracteres ↵	*Busca texto*
[línea] T nombre del fichero	*Transfiere línea*
[n] W	*Escribe línea*

EL PROCESADOR DE TEXTOS

INTRODUCCION

Los procesadores de texto son programas de ordenador que permiten realizar todas las operaciones necesarias desde que se crea un texto hasta su impresión o archivo en soporte magnético.

El usuario por medio del teclado del ordenador introduce el texto que aparece en pantalla, allí puede corregirse, cambiarse o darle el formato deseado a través de comandos específicos que difieren de un Ptx a otros. Cada vez que se hace una modificación el programa la incorpora y la pantalla muestra la versión corregida.

Las órdenes que se dan en el teclado mueven el cursor en la pantalla permitiendo señalar las palabras que desean cambiarse o corregir. También a través del teclado se organiza el texto, márgenes, sangrado, número de líneas por página, tipo de letra, etc., para su posterior impresión en papel.

Las funciones que se realizan con el PTX son: edición, impresión, almacenamiento y comunicación con otros programas si el PTX es parte de un paquete integrado.

EDICION DE UN TEXTO

El programa permite borrar, cambiar o suprimir caracteres, palabras o frases, eliminando los huecos que ocasionen estas correcciones. También permite añadir palabras, frases o párrafos dando formato al texto automáticamente. Muchos procesadores permiten también recuperar partes de texto eliminadas previamente.

Mediante procedimientos sencillos pueden marcarse frases o párrafos y moverlos a otro sitio del texto, copiarlos en otro sitio o eliminarlos.

Al escribir con un procesador no hay que preocuparse del final de línea ya que el programa pasa automáticamente a la línea siguiente (plegado de línea), ni del final de página. El ordenador escribe todo el texto y será en el momento de la imrpesión cuando se delimitará el número de líneas por página. En cualquier momento el usuario puede ver en pantalla la parte del texto que desee desplazando verticalmente el texto hacia arriba o hacia abajo con la tecla específica para ello.

IMPRESION DE UN TEXTO

Antes de sacar el texto por impresora hay que indicar las características con las que queremos que se imprima, es decir: márgenes, izquierdo y derecho, espacios entre líneas, número de líneas por página, cabeceras y pie de páginas, tipo de letra, subrayado, etc...

ALMACENAMIENTO

Una vez editado un documento si se desea guardar para su posterior utilización puede archivarse en un disco utilizando las instrucciones que para ello tenga establecido el Ptx.

COMUNICACION CON OTROS PROGRAMAS

Algunos procesadores tienen acceso a otros programas y pueden comunicarse con ficheros, bases de datos, diccionario o programas educativos interaccionando con ellos de modo que aumentan sus posibilidades de empleo en distintos campos.

CARACTERISTICAS GENERALES DE LOS PROCESADORES DE TEXTOS

La mayor parte de los procesadores de textos tienen una serie de características comunes que facilitan el trabajo de corregir y modificar un texto aparte de la posibilidad de guardarlo en un disco y editarlo o imprimirlo con posterioridad. Sin embargo, el modo en que estas funciones se realizan varía de unos procesadores a otros, y también, hay diferencias en el modo

de presentar el texto en el momento de imprimirlo. Unos tienen muchos tipos de letras, otros menos, permiten subrayados, permiten notaciones especiales, exponentes, signos científicos, etc.

Estas diferencias harán que unos procesadores de textos sean más apropiados para la gestión de una empresa, otros tengan más aceptación entre profesionales, periodistas, escritores, etc. y habrá que intentar encontrar los más idóneos para su empleo en los centros de enseñanza para uso del profesorado y de los alumnos.

Conviene conocer cuáles son las distintas operaciones que pueden realizarse con un PTX para poder valorar en un momento determinado la adecuación o no de uno de ellos a los fines que se pretenden conseguir y su facilidad de manejo.

Se establecen los criterios de evaluación en tres apartados: documentación, edición de textos y operación (archivo e impresión de los mismos).

DOCUMENTACION

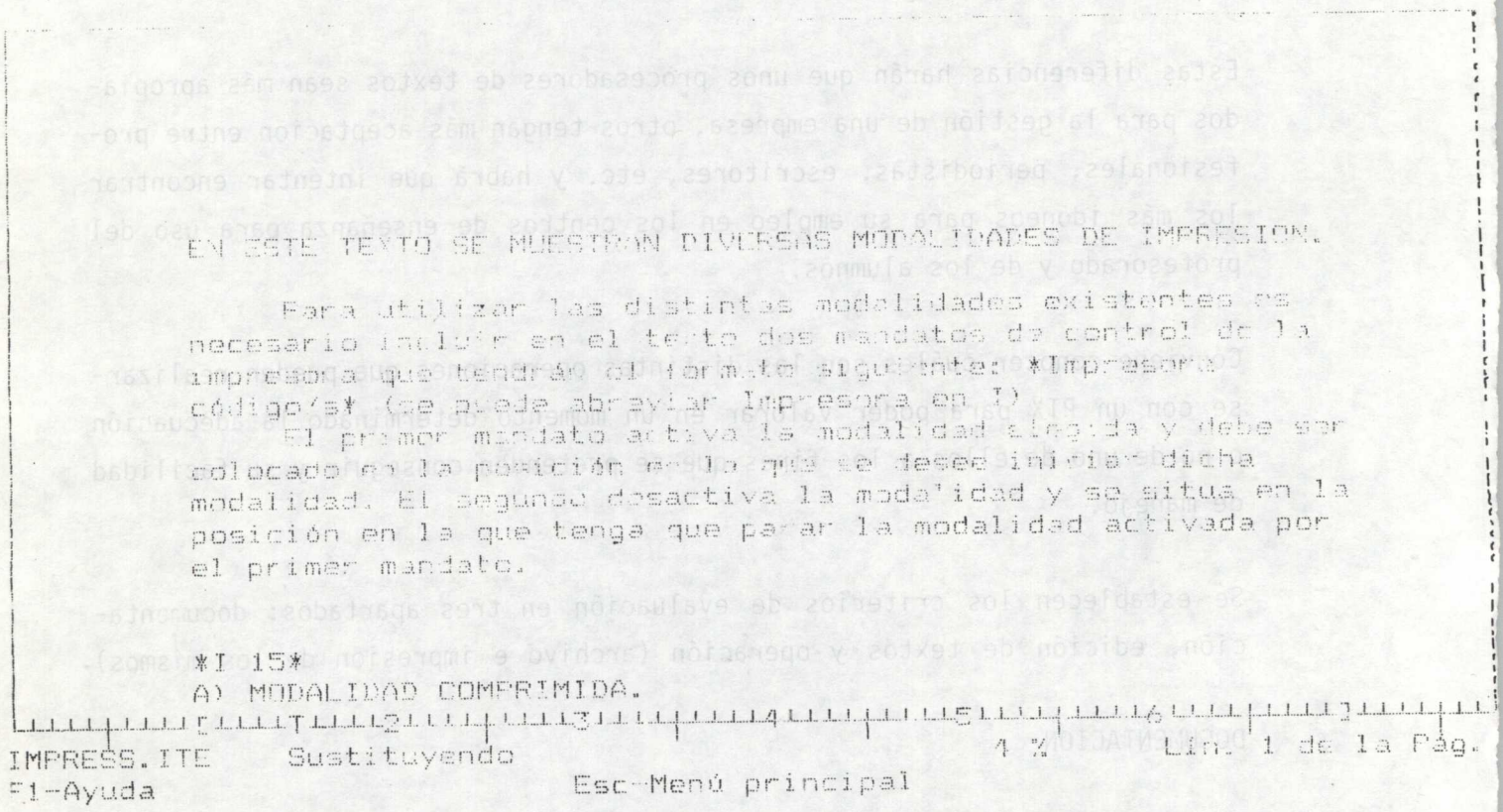
Lo primero que debemos valorar si existe en español y si está expresada de modo claro y correcto, bien organizado con un índice de contenidos que facilite la búsqueda. Algunos PTX ofrecen una plantilla resumen de los comandos que parece colocarse sobre el teclado cuando se está empleando y que facilita su aprendizaje. También suele ser muy útil que el manual tenga una parte tutorial que por medios de unos ejercicios de dificultad progresiva, enseñe al usuario todas las posibilidades que ofrece.

EDICION

El segundo bloque de características que se incluyen bajo la denominación de edición, hacen referencia al momento en que se redacta o se modifica un documento en pantalla.

La edición o creación de un documento, se hace en la pantalla de texto.

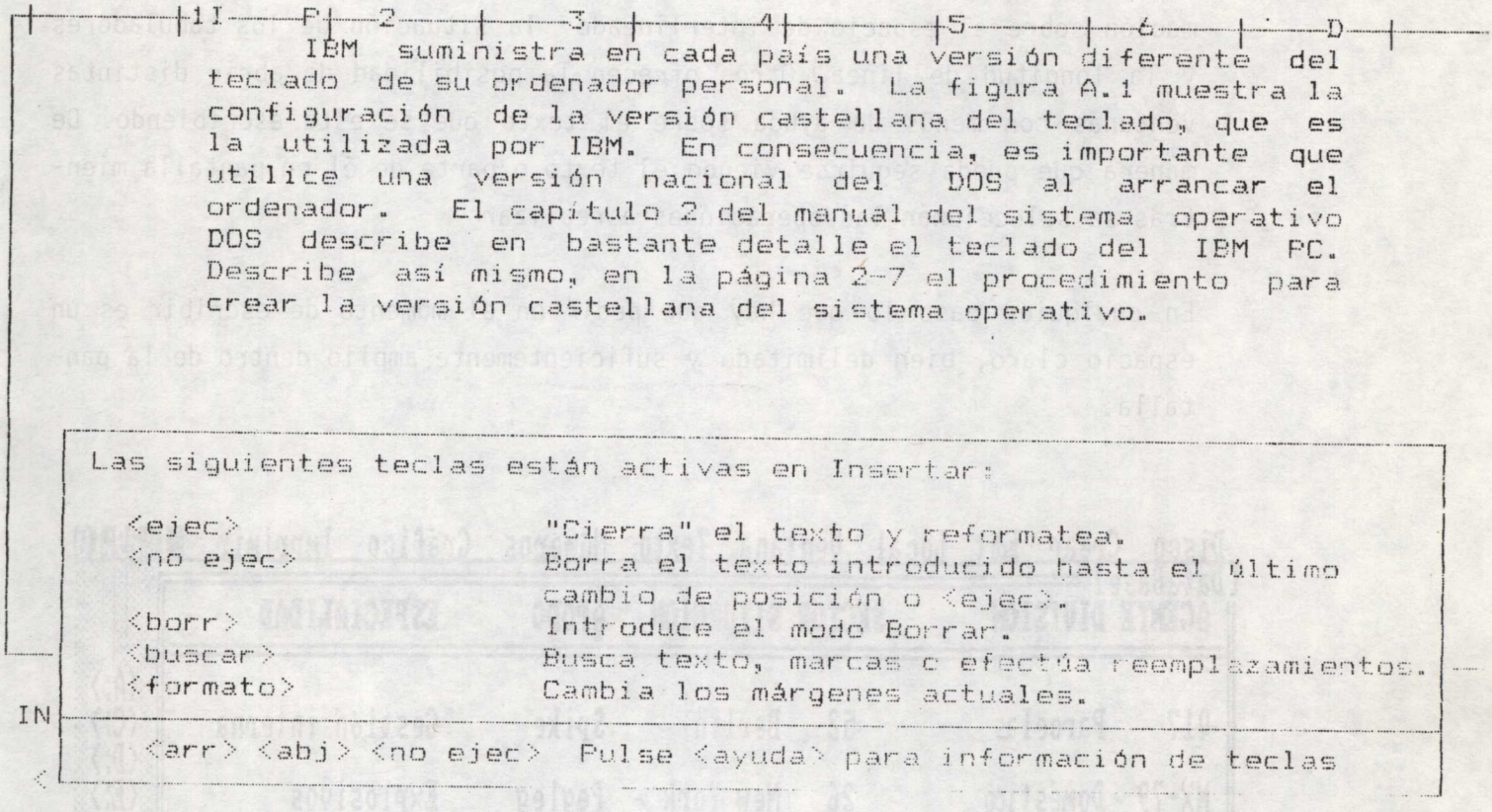
Algunos PTX dejan la pantalla entera para escribir,



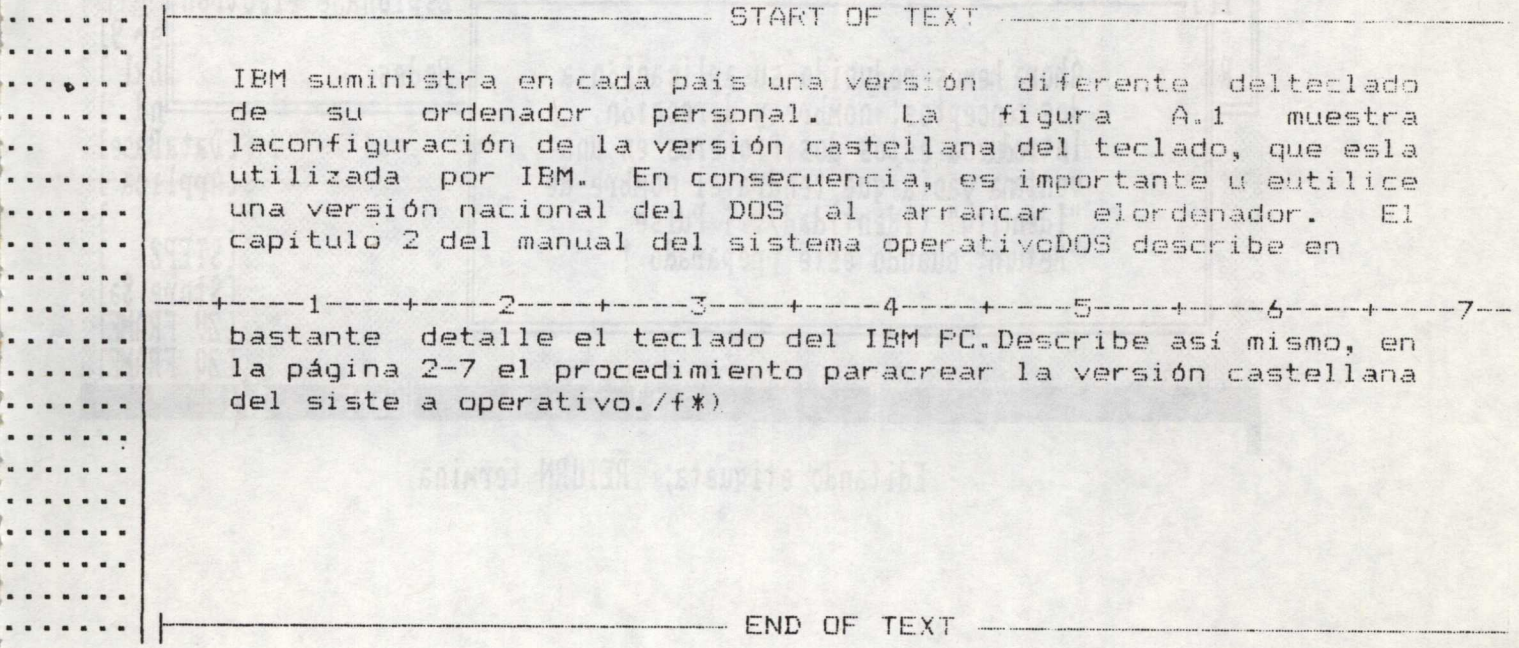
otros dejan solamente una parte reservando otra parte para que los menús de ayuda puedan visualizarse al mismo tiempo que se escribe.

Algunos contienen en la parte superior una línea de estado que informa sobre el documento con el que se está trabajando, y sobre el número de página, el número de línea y de columna en los que se encuentra el cursor.

Ejemplo de pantalla con menú de ayuda.



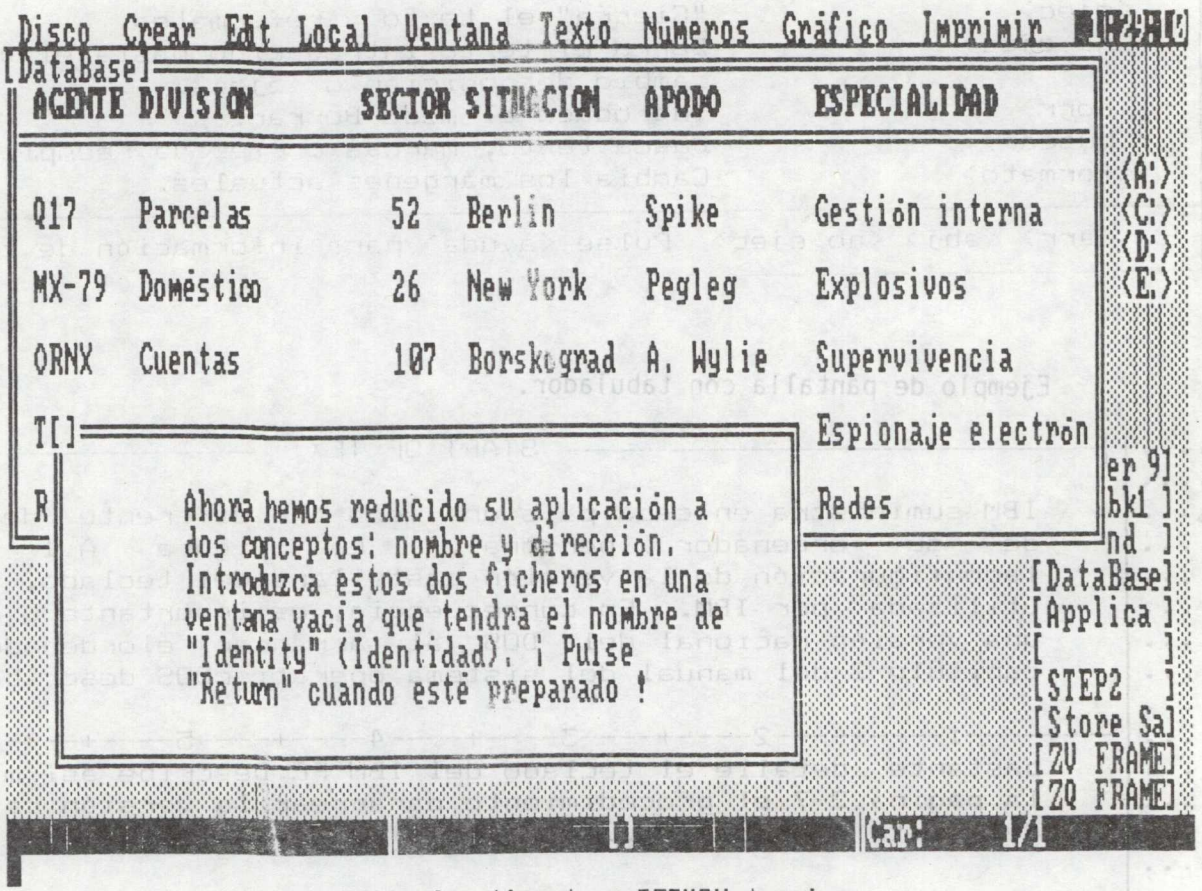
Ejemplo de pantalla con tabulador.



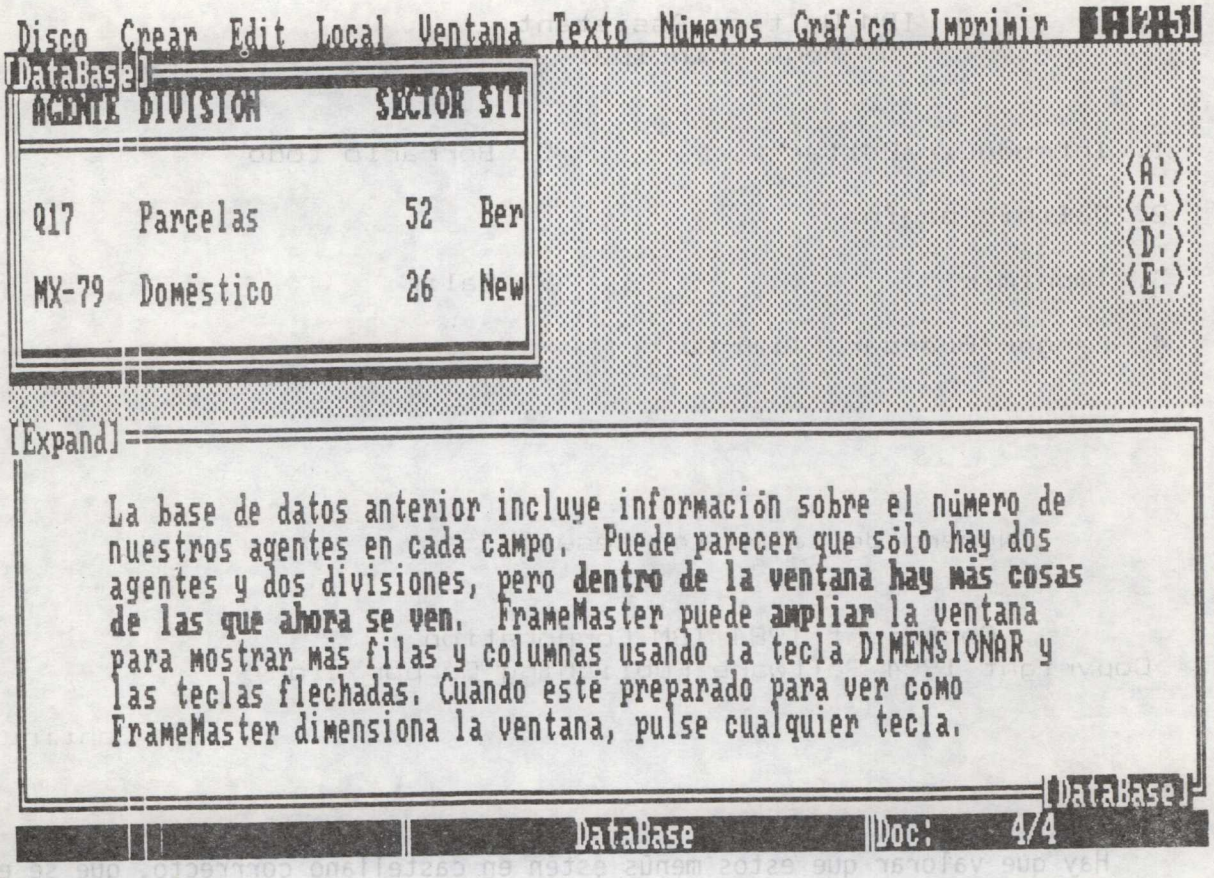
COMMAND MD I VS12 HS20 BN 1:140 WI 1: 72 SP64963
01 KEYS 2 SWAP 3 LINE 4RECALL5INSERT6DELETE7 8 9 0 KB

(En su parte inferior pueden tener una línea de formato que facilita información sobre el espacio de interlineado la situación de los tabuladores y la longitud de línea.) Otros ofrecen la posibilidad de abrir distintas ventanas con menús de ayuda sobre el texto que se está escribiendo. De manera que pueda seguirse viendo el texto o parte de él en pantalla mientras se seleccionan las operaciones a realizar.

En cualquier caso lo que hay que pedir en el momento de escribir es un espacio claro, bien delimitado y suficientemente amplio dentro de la pantalla.



Editando etiqueta; RETURN termina



Las distintas opciones que ofrece un PTX (editar, imprimir, copiar, borrar documentos, etc.) se presentan por medio de menús.

Menú de impresión

```
Desde la página: 1

Hasta la página: 1

Imprimir en: IMPRESORA

Pausa entre páginas (S/N): N

Número de copias: 1

Un espacio/dos espacios/sobre (U/D/S): U

Sangrar: 0

Nombre del archivo Filing Assistant:

Esc-Menú principal                    ↵ Continuar
```

IBM Writing Assistant
Menú Principal

1. Introducir/Editar
2. Definir página
3. Imprimir
4. Cargar/Guardar/Eliminar
5. Borrarlo todo
9. Salir

Número de la opción deseada:

Copyright 1984 IBM Corporation
Copyright 1984 Software Publishing Corporation

↵ Continuar

Hay que valorar que estos menús estén en castellano corrrecto, que se entiendan y que sean fáciles de ejecutar.

Del mismo modo ha de buscarse que las pantallas de ayuda, si no están visibles al mismo tiempo que el texto, sean de fácil acceso en cualquier momento y que realmente sean una ayuda, se entiendan y los comandos sean fáciles de recordar, es mejor si emplean la letra o letras iniciales de la acción que ejecutan, por ejemplo: BB (borrar bloque), IB (imprimir bloque), G (grabar en memoria de disco), etc.

```
+-------------------------------+-------------------------+-------------------------+
|                               | Impresora Preparada     | Impresora IBM           |
|  PEACHTEXT IMPRESION          |                         |                         |
|  Pantalla de Estado           |  Impres     Act         |  A:PRINT.DFT            |
|                               |  Disk       Des         |                         |
|                               |  Pantalla   Des         |                         |
+-------------------------------+-------------------------+-------------------------+
| Ajuste: Alineado izq.         | SU : Ignora Caract.     | Estado                  |
|                               | BF : Blanco Fijo        |                         |
| LP: Longitud de Pág 66        | NE : Negrita        1   | Copia       1           |
| MS: Margen Superior 6         | SR : Subrayado      C   | Página      1           |
| MI: Margen Inferior 6         | NPC: Nueva Pág Cond 0   | Renglón     1           |
| MQ: Margen Izq.     0         | RPP: Renglones/plg  6   | Posición    1           |
| AR: Anchura de Reng 60        | CPP: Caract/plg     10  | Registro    0           |
| SA: Sangrado        0         | COP: # Copias       1   | Fuente      Texto       |
| SP: Sangrar Párrafo 0         | FORM: Continuo          | Alipag      Act         |
| ES: Espaciado       1         | # Variables         0   | Pieap       Act         |
+-------------------------------+-------------------------+-------------------------+
```

```
Teclear 'FIN' para volver al menú principal o 'EDIT' para ir al Editor
Introducir mandato de impresión, pulsar tecla 'INTRO' para imprimir:
\just
```

También es deseable que no se necesite emplear muchas teclas a la vez para realizar una operación, sobre todo si es para niños el tener que pulsar CTRL-SHIFT-W por ejemplo les ocasiona muchos problemas. Es mejor si estas teclas están colocadas racionalmente, es decir, las teclas usadas para borrar no deben colocarse en un lugar donde puedan tocarse inadvertidamente. Algunos procesadores emplean las teclas de la parte superior para insertar y las de la parte inferior del teclado para borrar.

Como norma general habrá que valorar positivamente todo lo que el PTX ofrezca de facilidad para corregir errores ya que lo que se va a intentar es conseguir mejores textos escritos, sin faltas de ortografía, con buen estilo, etc... Para ello, habrá que analizar si el cursor se mueve con facilidad, en general la mayoría de los ordenadores lo hacen con las teclas de las flechas, en el caso de que no haya flechas habrá que considerar si están bien elegidas las teclas a las que se ha asignado el movimiento del cursor. Con el PTX se pueden buscar palabras, frases o párrafos que queremos mover, borrar o modificar la facilidad de ejecución de estas acciones son fundamentales y harán que el empleo del ordenador y el PTX sea preferible al uso de una máquina de escribir y aprovechen al

máximo la capacidad del ordenador.

Respecto al formato del texto hay posibilidad de definir muchas cosas:

Longitud de línea, en general la mayoría de los PTX permiten escribir sin preocuparse del salto de línea que es realizado automáticamente y cuando la última palabra no cabe se escribe en la línea siguiente de modo que no hacen falta guiones de división de palabras. Otros dividen palabras con guión. También puede pedirse en algunos de ellos que ajusten el margen derecho, con lo que automáticamente reparten el hueco que quede al final en los espacios de la línea.

El usuario emplea la tecla INTRO únicamente cuando quiere indicar el final de un párrafo o cuando está escribiendo listas.

Es habitual que permitan el sangrado de párrafo por lo que lo que habrá que considerar es la facilidad de ejecución.

La posibilidad de establecer cabeceras y piés de páginas y paginen automáticamente facilitan las tareas de impresión.

Algunos ordenadores escriben en pantalla en mayúsculas y señalan las mayúsculas en vídeo inverso, esto aunque en el momento de la impresión se subsana, ocasiona problemas si los usuarios son niños o no están acostumbrados a trabajar con este tipo de PTX es preferible que tengan ambos tipos de letra en pantalla. Así como la posibilidad de escribir simultáneamente a uno o más espacios y puede escribir notaciones especiales subíndices, exponentes o signos científicos que requieren alterar la línea normal de escritura.

A menudo es la impresora la que determina el que puedan emplearse este tipo de notaciones, así como la posibilidad de emplear subrayado o distintos tipos de letras (negrilla, itálica, etc...) Hay PTX que ofrecen hasta 15 tipos distintos de letra.

Algunos tienen posibilidad teórica pero en la práctica es tan complicado el mezclar tipos de letras en un documento que la hacen impracticable.

Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras
Estos son distintos tipos de letras

Aunque las pantallas tienen 80 caracteres por línea, según el tipo de letra con el que se está escribiendo al imprimir pueden salir líneas de más de 80 caracteres, con lo que no se podrá visualizar toda la línea en pantalla a menos que el PTX permita un desplazamiento horizontal del texto de un modo similar al que hace al desplazar el texto hacia arriba o hacia abajo (SCROLLING).

Una vez escrito un documento un buen procesador de textos permitirá cualquier tipo de modificación del formato (márgenes, sangrado, espacios, paginación, etc.) en el momento de la impresión mediante una serie de mandatos especiales recogidos en la pantalla de ayuda de impresión.

Una función interesante que tienen algunos PTX es la de fusión de documentos con la que es posible crear una serie de documentos similares pero no idénticos y originales. Algunas aplicaciones típicas para esta función son los envíos por correo, difusión de memorandums, etc. Los documentos de fusión se crean combinando un Documento Primario (Fijo) que permanece invariable durante las series de impresiones de fusión y un Documento Secundario (variable) que es insertado en los lugares prefijados del Documento.

Otra característica útil que pueden tener, es la de emplear Procedimientos Clave (Macros) que permite integrar en procedimientos que son archivados, las secuencias de función y órdenes o palabras que son utilizadas con mayor frecuencia, para que puedan ser ejecutadas en cualquier momento sin que sea necesario reconstruirlas cada vez.

La función de Biblioteca permite crear y almacenar grandes cantidades de textos que pueden insertarse automáticamente en cualquier documento.

Diccionario. Los PTX que tienen esta opción suelen ofrecerle en el menú principal. El programa revisa, en una primera fase, el documento completo o las páginas señaladas por el usuario. Después de que el programa ha calculado el número de palabras mál ortografiadas y el número total de palabras del documento, presenta en pantalla la primera página con el primer caracter de la palabra incorrecta parpadeante. El usuario puede corregir las palabras utilizando la sobreimpresión, o la función de corrección ortográfica si la hay. (Este suele ofrecer una lista de ortografías alternativas para cada palabra y la opción de seleccionar una de ellas para sustituir la actual.).

OPERACION

Incluímos dentro de este apartado todo lo que tiene que ver con el uso del PTX dentro del ordenador y la seguridad que debe ofrecer al usuario para no perder el documento en ningún momento o la posibilidad de tener la privacidad de un documento protegiéndolo con una contraseña.

En primer lugar debe de ser fácil de cargar en el ordenador, por supuesto es necesario que tenga una copia de seguridad o posibilidades de hacerla. Cuando el PTX vaya a emplearse con los alumnos de un aula hará falta que cada grupo de alumnos tenga un PTX o una copia, es valorable las facilidades que ofrezcan las casas comerciales para sacar las copias que hagan falta o la compensación económica en el caso de que se compren todas las necesarias.

También pueden ocurrir que una vez cargado el programa en el ordenador, sea residente en memoria y no haga falta tener el disco dentro de la uni-

dad porque no requiere acceso intermitente al mismo con lo que cada grupo podría cargar con el mismo disco el PTX en memoria. Pocos PTX son residentes en memoria.

Según el o los tipos de ordenadores que se posean habrá que valorar si el PTX vale para todos los modelos o sólo para uno en concreto.

Escribir y editar un texto es sólo una parte de los PTX. Los usuarios deben poder guardar los textos y recuperarlos para su posterior trabajo con ellos o para su impresión. Estas operaciones de archivo y recuperación de documentos tienen que ser fáciles y ofrecer alguna medida de protección que impida la pérdida involuntaria de un texto, para ello, cuando se selecciona del menú la opción **BORRAR FICHERO** el procesador requiere confirmación solicitando el nombre completo del fichero. A pesar de eso algunos PTX guardan una copia del fichero borrado.

También el usuario puede proteger su trabajo mediante una contraseña que permite el acceso a su texto únicamente a quien haya codificado esa contraseña.

El procesador debe tener una ejecución correcta y sin errores puesto que los errores en la lógica del programa pueden detener la ejecución del mismo causando la pérdida de los textos con los que se está trabajando.

También debe permitir realizar programas distintos o abandonar el PTX sin tener que desconectar el ordenador.

INTRODUCCION

Los procesadores de texto son programas construidos para facilitar la creación de ficheros de texto,permitiendo realizar con gran comodidad modificaciones futuras.

Entre las utilidades básicas de un procesador, se encuentran comunmente, las de insertar,borrar o cambiar caracteres, mover o copiar trozos de texto, buscar o reemplazar cadenas de caracteres, pudiendo en los más potentes establecer de forma automatica la estructura de los parrafos de acuerdo con los formatos que se definan, y especificar los atributos de impresión, tales como negrilla, subrayado, italica, cabeceras o pies de página, espaciado entre lineas, etc.

Los procesadores de texto , son por tanto la herramienta ideal para escribir cartas, informes, libros, contratos, etc.

El procesador de texto del Open Access, permite trabajar con dos tipos de ficheros, el denominado documento (extensión **.doc**) y el denominado texto (extensión **.txt**).

El primero dispone de todas las caracteristicas de los potentes procesadores, permitiendo realizar escritos con perfecta presentación de forma muy sencilla.

El segundo es compatible con los demas módulos del Open Access, para lo cual tiene una concepción mas sencilla, no pudiendo definir los márgenes que permitan dar formatos automaticos a los parrafos, ni diferentes tipos de letras (negrilla, subrayado e italica).Sin embargo la compatibilidad antes mencionada con los otros módulos le da una gran potencia, al poder incluir en un informe, resultados obtenidos en Hoja de

Calculo, gráficos previamente conseguidos en el módulo de Gráficos, o bien enviar este mismo informe o cualquier carta de forma individualizada a una serie de personas cuyos datos se encuentran almacenados en un fichero de la Base de Datos.

La documentación desarrollada, a continuación, sobre el Procesador de Textos del Open Access, esta agrupada en 3 grandes apartados:

-COMANDOS. Se incluyen todos los comandos, con una breve explicación de los mismos.Estan ordenados alfabeticamente, para una facil localización.

-GUIA DE OPERACION. Se explican los pasos a seguir para realizar las operaciones fundamentales.

-APENDICES. Se da un resumen sobre los aspectos basicos para el manejo del Procesador de Textos.

-COMPORTAMIENTO DEL TECLADO.

-FORMATO .

-CARACTERISTICAS DE IMPRESION.

-CARACTERES DE SUSTITUCION.

COMANDOS

BORRAR (BORR.). Permite eliminar caracteres.

CALCULADORA (=CALC). Accede a una calculadora.

CAMBIAR. Permite reemplazar caracteres.

COPIAR. Copia un trozo de texto en el lugar especificado del fichero de trabajo.La información copiada, puede proceder del propio fichero actual o de otro cualquiera almacenado en el disco.

FICHERO. Permite guardar en el disco el contenido del fichero de trabajo.Puede hacerlo con el mismo nombre o con otro nuevo.Pasa al Menú de Selección de Ficheros del Procesador de Textos.

FORMATO. Permite definir 8 tipos diferentes de estructura de parrafo, para la presentación de documentos.Tambien permite definir abreviaciones.

GUARDAR. Permite guardar en el disco el contenido del fichero de trabajo, manteniendolo activo.Puede hacerlo con el mismo nombre o con otro nuevo.Vuelve al Menú de Proceso de Texto.

INSERTAR (INS). Permite pasar al modo inserción.

MARCAR. Permite establecer hasta 8 marcas invisibles dentro del texto.Las marcas pueden tener hasta un máximo de 3 letras.

MOVER TEXTO (MOVTEXTO). Permite trasladar un trozo de texto de una parte a otra del fichero de trabajo.

OPCIONES. Permite guardar en el disco el contenido del fichero de trabajo.Puede hacerlo con el mismo nombre o con otro nuevo.Pasa al Menú Principal del Open Access (ventana de **Opciones**).

GUIA DE OPERACION

El objeto de esta guia es iniciar el uso del Procesador de Textos, definiendo los procesos a seguir para realizar las operaciones fundamentales que permitan familiarizarnos con el modo de proceder de este módulo del Open Access.

Vamos a desarrollar los siguientes apartados:

1. Acceso al Procesador de Textos.
2. Creación de un fichero.
3. Operaciones en un fichero.
4. Operaciones con ficheros.
5. Impresión de ficheros.
6. Integración con otros módulos. Cartas personalizadas.

1. ACCESO AL PROCESADOR DE TEXTOS

a) Con dos unidades de diskette.

Desde ambiente del sistema operativo (A>), introducir el disco 1 del Open Access en la unidad A y teclear **OA** y <RET>. Aparecerá una pantalla como la de la figura 1.

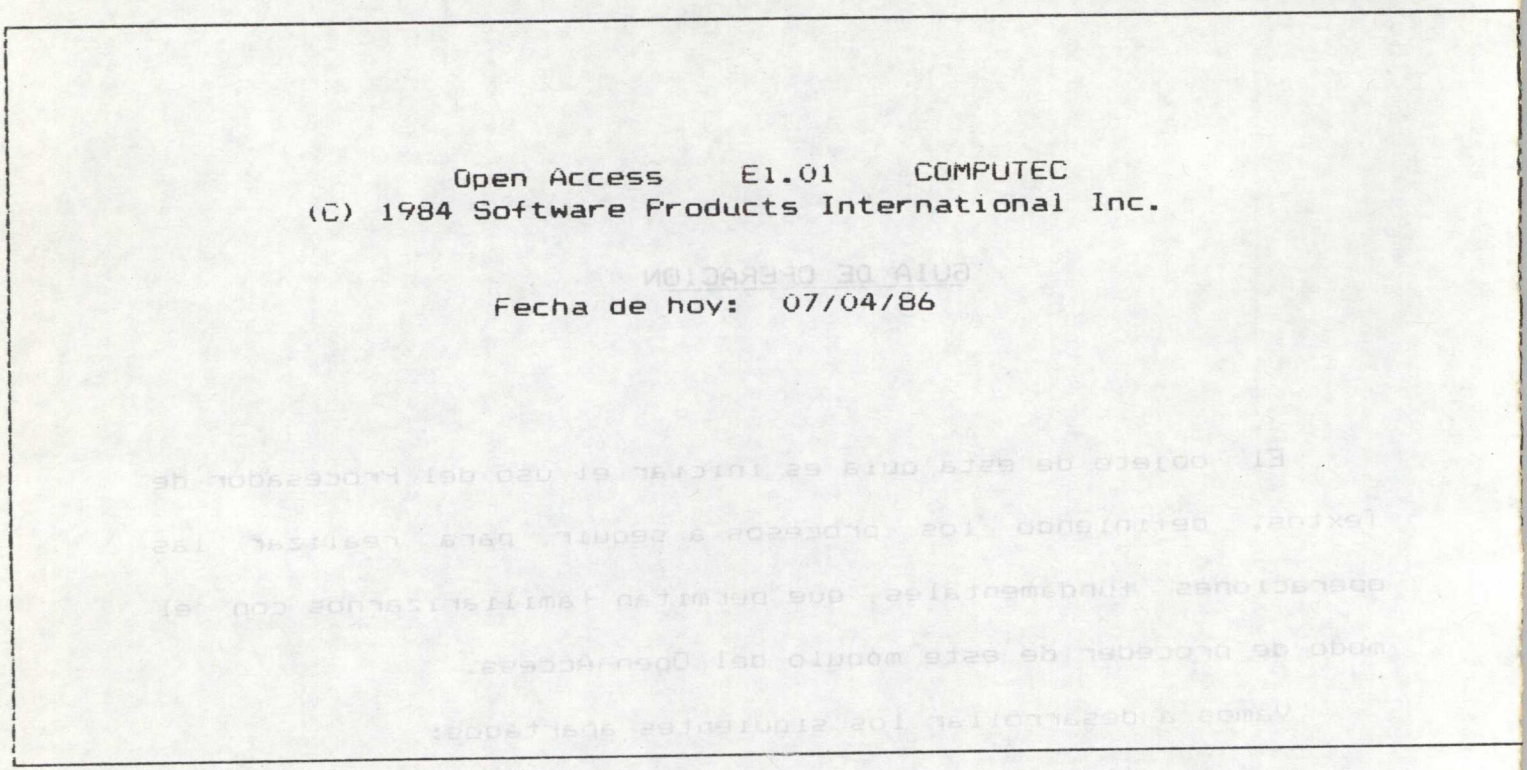


figura 1

Actualizar ó confirmar la fecha y pulsar **<RET>**, la pantalla presentará el Menú de **OPCIONES** del Open Access (figura 2).

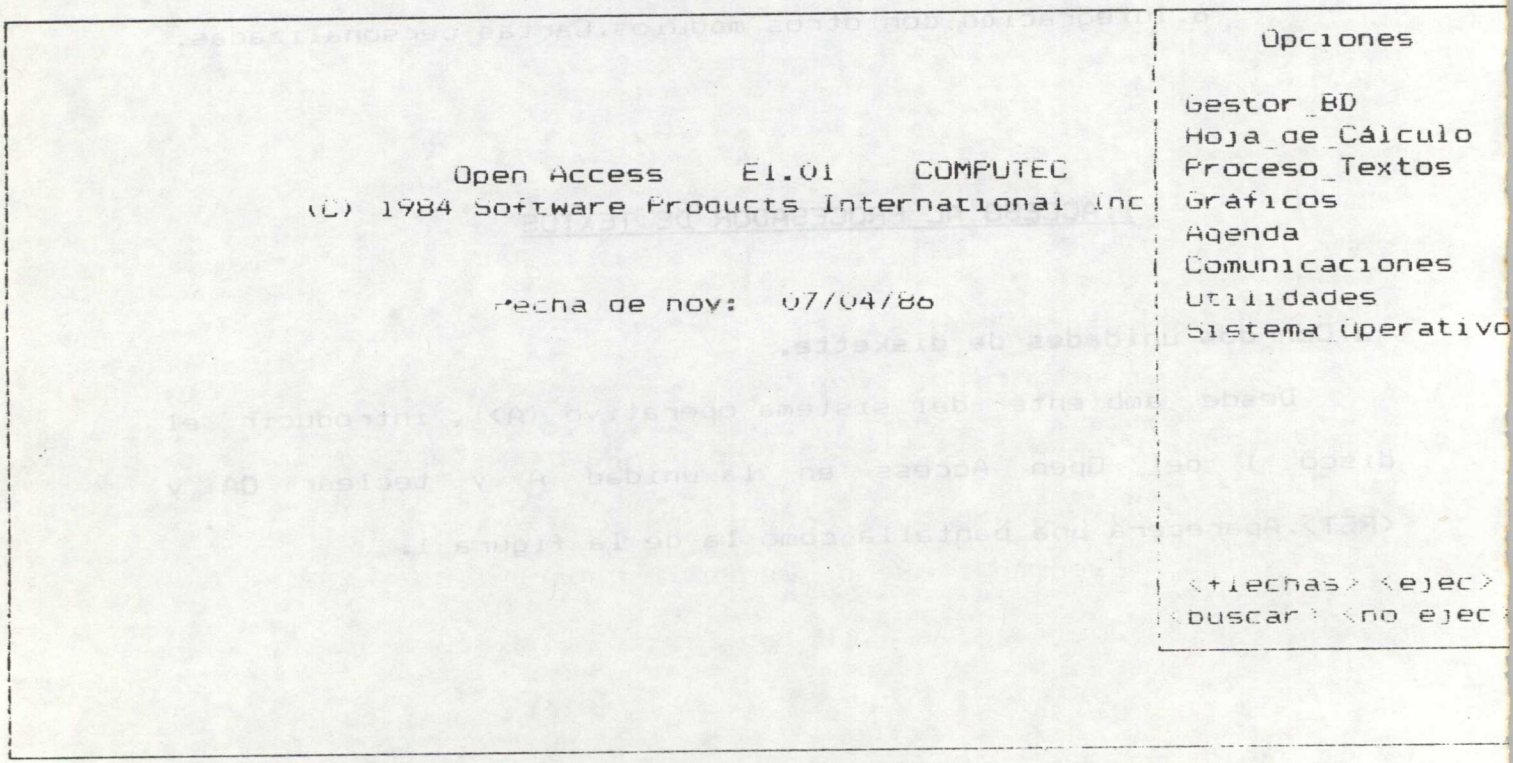


figura 2

Situar el cursor en Proceso Textos y pulsar **<RET>** (ó pulsar P).El Open Access pedirá el disco de Proceso de Textos (Codigo_2), que habrá que introducir en la unidad A, sustituyendo al de arranque, pulsando a continuación **<ejec>**=F10.La pantalla presentará el **Menú de Selección de Fichero** (figura 3).

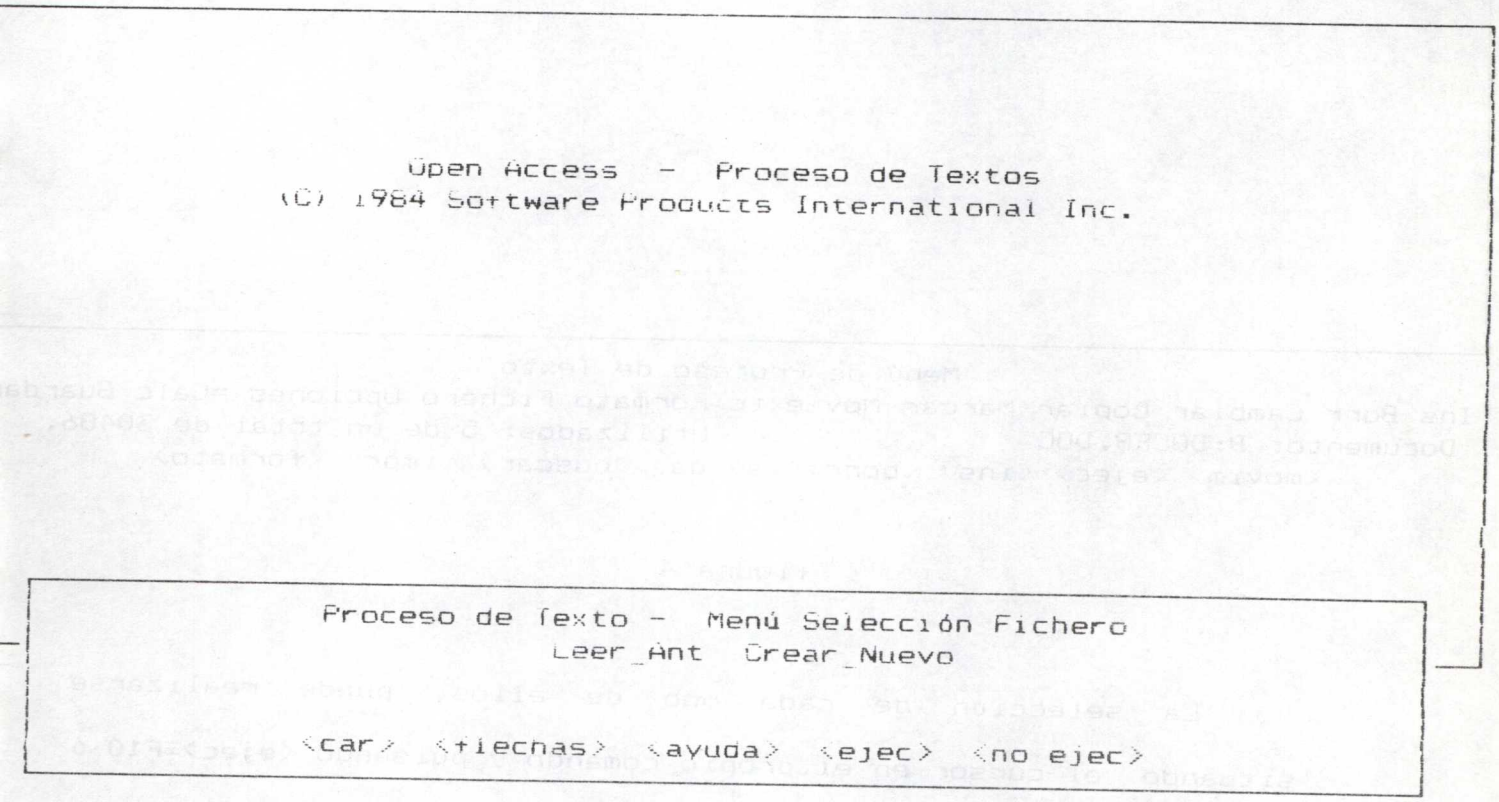
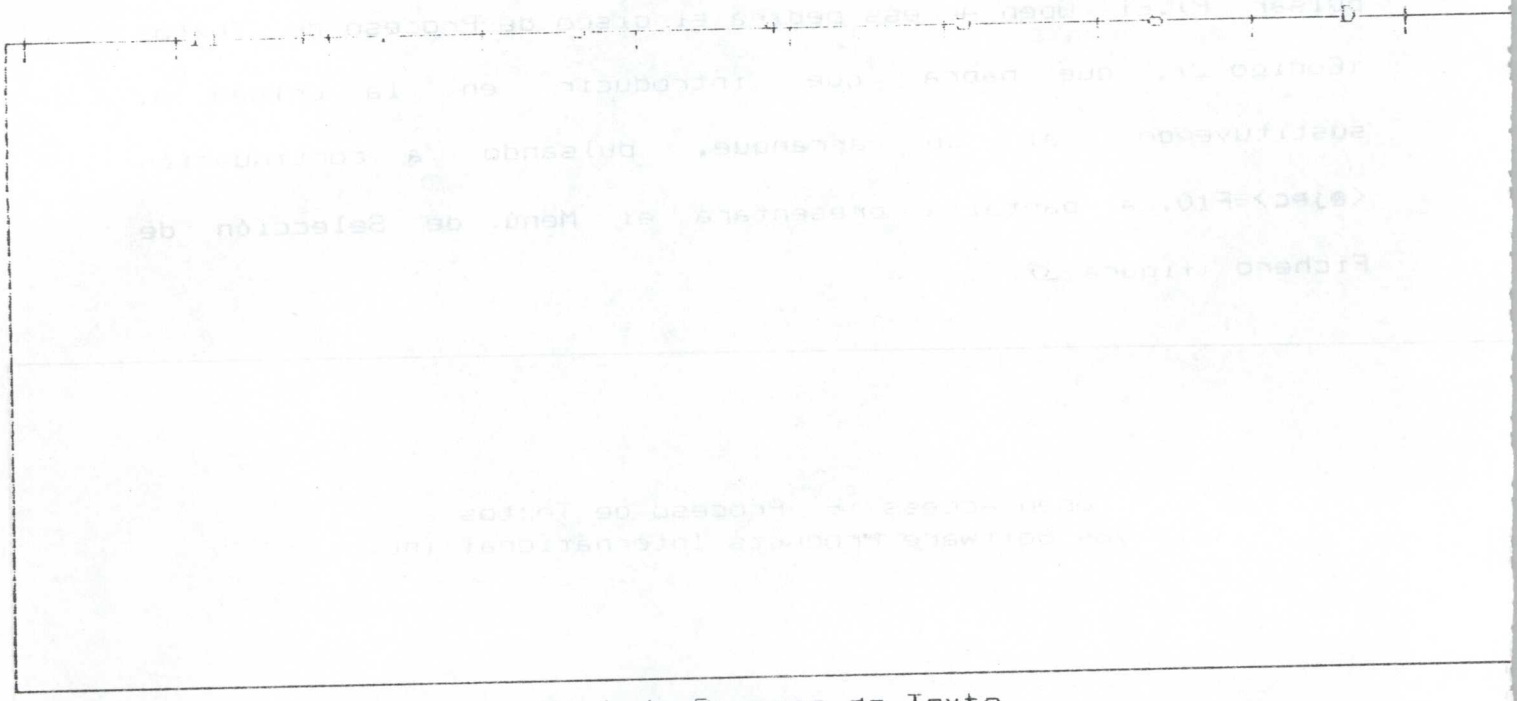


figura 3

Introducir el diskette de usuario en la unidad B, y ya nos encontramos en condiciones de trabajar con el Procesador de Textos.

Con cualquiera de las dos alternativas, tras indicar el nombre del fichero con el que se va a trabajar, se llega al **Menú de Proceso de Texto**, en donde, pulsando **F2** aparecen todos los comandos existentes, figura 4.



Menú de Proceso de Texto
Ins Borr Cambiar Copiar Marcar MovTexto Formato Fichero Opciones =Calc Guardar
Documento: B:DOCFP.DOC Utilizados: 5 de un total de 30486.
<movim> <ejec> <ins> <borr> <ayuda> <buscar> <impr> <formato>

figura 4

La selección de cada uno de ellos, puede realizarse situando el cursor en el propio comando y pulsando **<ejec>≡F10** ó bien pulsando la inicial del comando, si bien en este ultimo caso, será necesario añadir la siguiente letra cuando coincida la inicial en comandos distintos.

Tanto en esta posición, como durante la ejecución de cualquier comando,la tecla de **<ayuda>≡F1** nos informa de lo que podemos hacer ó contestar en ese punto.Una nueva pulsación de dicha tecla, nos presenta una ventana con el comportamiento del teclado (figura 5).En ambas ventanas de ayuda, la flecha de movimiento del cursor | sigue presentando la información, si no cabe.Para regresar al punto de trabajo, hay que pulsar la tecla Esc.

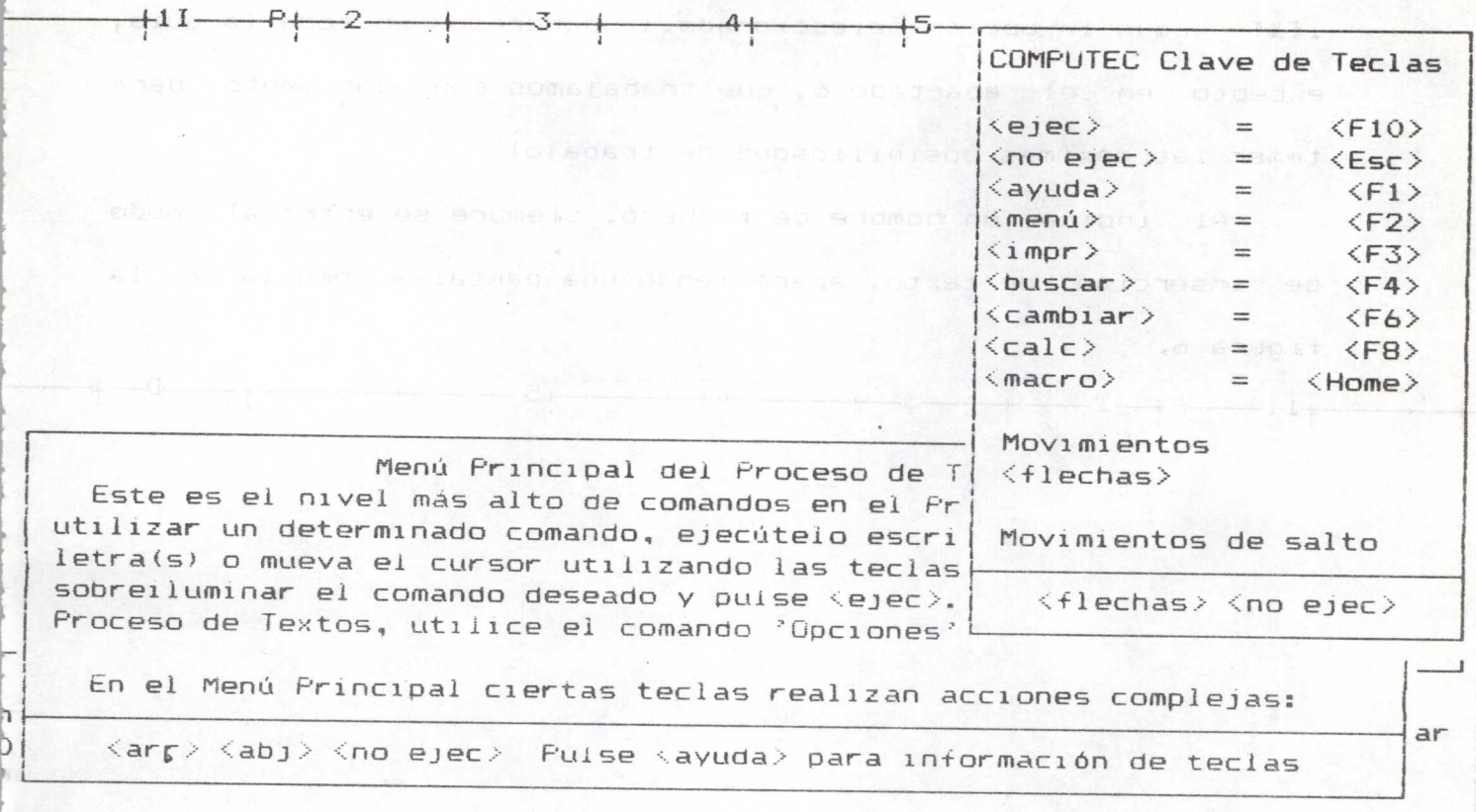


figura 5

b)Con unidad de disco duro.

El proceso es análogo, con las siguientes diferencias:

-El punto de partida es la situación en el directorio del Open Access.

-El unico diskette necesario es el de arranque, ya que los módulos del Open Access y el área de trabajo del usuario se encuentran en el disco duro.

2.CREACION DE UN FICHERO

Desde el **Menú de Selección de Fichero** seleccionar **Crear_Nuevo**, y según el fichero a crear **Documento** o **Texto**.En ambos casos solicitará el nombre del nuevo fichero, a lo que debemos responder con un nombre valido para fichero de DOS, sin extensión, ya que el Open Access añadirá automáticamente .DOC o

.TXT segun la opcion seleccionada.(Supondremos en toda la guia, excepto en el apartado 6, que trabajamos con documento para tener las máximas posibilidades de trabajo)

Al indicar un nombre de fichero, siempre se entra al modo de inserción de texto, apareciendo una pantalla como la de la figura 6.

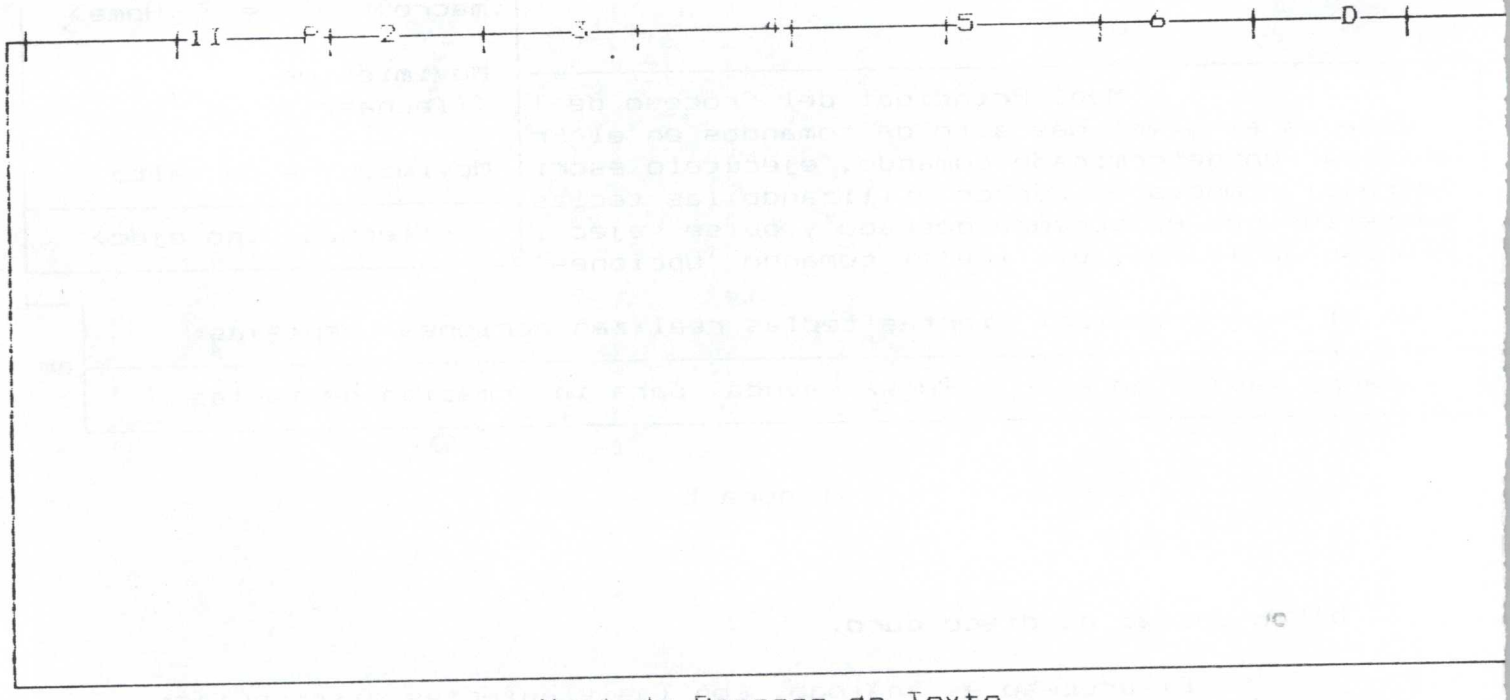


figura 6

En la linea superior, pueden observarse:

-Números del 1 al 7, que indican las decenas correspondientes a la posición de las columnas.

-Rayas verticales, que indican las posiciones de tabulación asumidas por defecto.

-La letra I, que indica el margen izquierdo.

-La letra P, que indica el sangrado de parrafo.

-La letra D, que indica el margen derecho.

Para modificar los margenes y las tabulaciones, se dispone del comando **Formato** y de tecla de función **F8** (ver el apendice B).

En este modo inserción, introducimos la información que constituya el fichero, pudiendo en cualquier momento cambiar el tipo de los caracteres que se vayan a escribir.Para ello basta pulsar la tecla **F6**, que nos presenta una pantalla como la de la figura 7.

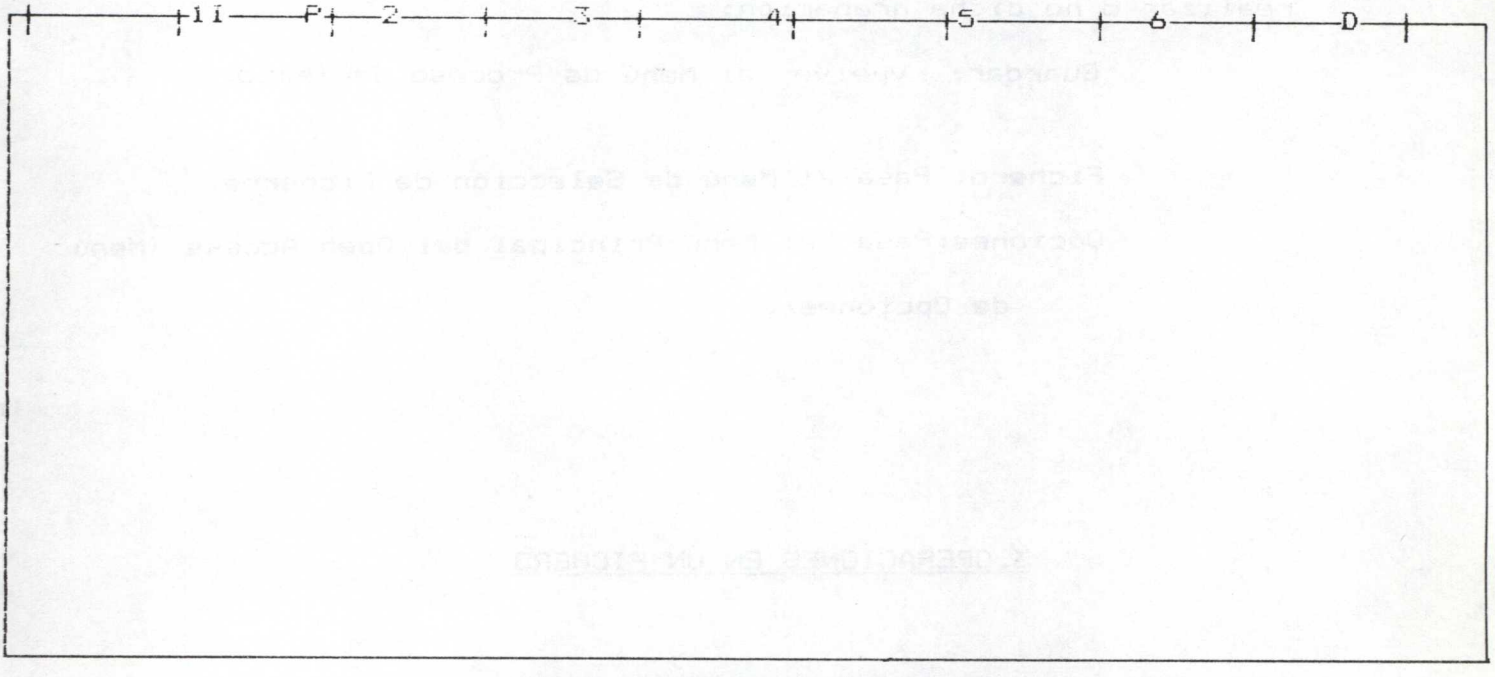


figura 7

Para modificar el tipo de escritura, hay que situar el cursor sobre **Negrilla, Subrayado** o **Italica** y pulsar **F10**, que, alternativamente hace cambiar el si por el no.Cuando se encuentre la combinación deseada, se vuelve al modo inserción pulsando de nuevo **F6.**

Para finalizar la sesión de edición, pulsamos **F2**, que nos lleva al **Menú de Proceso de Texto.**

Existen tres comandos **Guardar, Fichero** y **Opciones,** que permiten grabar el texto editado en el disco, la diferencia entre ellos corresponde a la situación a la que conducen tras realizar o no dicha grabación:

Guardar: Vuelve al **Menú de Proceso de Texto.**

Fichero: Pasa al **Menú de Selección de Ficheros.**

Opciones: Pasa al **Menú Principal del Open Access (Menú de Opciones).**

3.OPERACIONES EN UN FICHERO

3.1.Mover Texto.

-Ejecutar el comando **MovTexto.**

-Definir el bloque a mover.Para ello situar el cursor al inicio del bloque y pulsar **F10**, a continuación situar el cursor al final del bloque y pulsar **F10.**Observar como el bloque definido como origen queda sobreiluminado.

-Definir el lugar de destino y pulsar **F10.**El bloque queda trasladado.

3.2.Copiar.

-Ejecutar el comando **Copiar.**

-Definir el bloque a copiar:

-Interno:

-Cursor al inicio del bloque y pulsar **F10.**

-Cursor al final del bloque y pulsar **F10.**

-Externo:

-Pulsar la tecla **F6.**

-Dar el nombre del fichero origen y **F10.**

-Realizar los pasos como si fuera interno.

-Definir el lugar de destino y pulsar **F10.**El bloque queda copiado.

3.3.Buscar/Reemplazar.

-Situado el cursor en el origen de busqueda, pulsar **F4.**Aparece un sub-menú con **Caracteres Marca Reemplazar.**El comportamiento al seleccionar cada una de las opciones es el siguiente:

Caracteres. Permite localizar caracteres que formen parte del texto.Si se selecciona, lo primero que nos pide es a las cuantas veces da la localización.Teniendo a continuación que indicar si el patron es una muestra (palabra completa) o un literal.Se pasa de una a otra opción pulsando **F6.**

Marca. Permite localizar marcas invisibles que previamente se han introducido en el texto.El proceso de trabajo con marcas, tiene por tanto 2 fases:

Establecer marcas.

-Estando en modo inserción, posicionar el cursor en el lugar a marcar y pulsar **F2.**

-Ejecutar el comando **Marcar.**

-Introducir la marca que se desee.Presenta las que ya existen.Pueden haber hasta un máximo de 8, y han de tener hasta 3 caracteres.

Buscar marcas.

-Estando en modo inserción, posicionar el cursor en el origen de la busqueda y pulsar **F4.**

-Seleccionar **Marca.** Solicita la marca que deseamos buscar, presentando relación de las existentes.

Reemplazar.

-Solicita a las cuantas veces se realiza el reemplazamiento.

-Solicita el literal o muestra patron.

-Permite realizar el proceso con verificación (pide confirmación para realizar el cambio), o sin verificación.El paso de una a otra alternativa es con **F6.**

-Solicita el literal o muestra por el que se cambia.

4.OPERACIONES CON FICHEROS

4.1.Cambiar nombre.

-Desde el **Menú de Selección de Fichero** ejecutar **Leer_Ant**, utilizando **F4** para buscar el nombre del fichero al que queremos cambiarle su nombre.Cuando este localizado en la ventana que presenta los nombres de los ficheros, pulsar la tecla **F6**.Pedirá el nuevo nombre para el fichero.

4.2.Borrar.

-Desde el **Menú de Selección de Fichero** ejecutar **Leer_Ant**, utilizando **F4** para buscar el nombre del fichero a borrar.Cuando este localizado en la ventana que presenta los nombres de los ficheros,

pulsar las teclas **END-DEL**.

4.3.Copiar.

-Desde el **Menú de Selección de Fichero** ejecutar **Leer_Ant**, utilizando **F4** para buscar el nombre del fichero a copiar.Cuando este localizado en la ventana que presenta los nombres de los ficheros, pulsar las teclas **END-INS**.Pedirá el nombre para el nuevo fichero.

5. IMPRESION DE FICHEROS

Desde el modo insercion, se pulsa **F3**, con lo que aparece una pantalla como la de la figura 8.

```
+---------+1+---+2+---+3+---+4+---+5+---+6+---+D+---+
Cabecera :      Número linea:    1        Especial: Nada

Subcabecera:    Numero linea:    2        Especial: Nada

   Pie:         Número linea:    2        Especial: Nada
:-#-:

        Ancho papel:    80    Margen Superior:    3 Primera página :     1
     Longitud papel:    24    Margen Inferior:    4 Página comienzo:     1
Cabeceras alternas:    No.   Bloque  Párrafo:    4 Página de final: 32767
          Espaciado:    1     Desplazam. izda:    0

Dispositivo de impresión: CONSOLA
```

Menú de Proceso de Texto
IMPRIMIR: <flechas> moverse, <impr> imprimir, <menú> vuelve a editar
Escribir nuevos valores o <cambiar> para si/no
<flechas> <ret> <cambiar> <impr> <ejec> <no ejec> <menú> <ayuda> <buscar>

FIGURA 8

Se definen los parámetros de impresión (ver Apendice C) y se vuelve a pulsar la tecla **F3**.

6.INTEGRACION CON OTROS MODULOS.CARTAS PERSONALIZADAS

El Procesador de Textos tiene la posibilidad de utilizar e incluir en su texto, informaciones preparadas en los otros módulos del Open Access (Base de Datos, Hoja de Calculo y Graficos) gracias a la integración existente entre ellos.

Una de las aplicaciones más interesantes de esta integración, consiste en la elaboración de **Cartas Personalizadas.**Para ello se crea un modelo de carta, incluyendo caracteres de sustitución (ver apendice D), que se utiliza como plantilla y el contenido de un fichero de Base de Datos que suministra los datos que hay que particularizar.Se describe a continuación el Proceso basico de realización:

1.-Desde el Procesador de Textos, crear un **Fichero de Texto**, con caracteres de sustitución.Es imprescindible que sea tipo Texto, si bien puede crearse tipo Documento y posteriormente realizar una copia a tipo Texto.

2.-Desde Base de Datos, ejecutar el comando **Cartas.**Con las clausulas de interrogación se seleccionaran los afectados por el envio de la carta.

-A continuación nos pedirá la **Plantilla**, que es el nombre del fichero de texto.

-Por ultimo nos pide que definamos el dispositivo de salida, tras lo cual realiza, de forma continuada, la impresion de las cartas correspondientes.

APENDICE B

FORMATO

Hay 2 formas de definir un formato, con el comando **Formato** y con la tecla **F8**.

1.-Comando Formato. Desde el Menú de Proceso de Texto, seleccionar **Formato**. Aparecera una pantalla como la de la figura 1.

	1 Margen Izdo	2 Sang Parr	3 Margen Dcho	4 Salto Pal	5 AutoSangrado	6 Ajuste
A	11	16	70	Si	No	Izdo.
B	11	16	70	Si	No	Izdo.
C	11	16	70	Si	No	Izdo.
D	11	16	70	Si	No	Izdo.
E	11	16	70	Si	No	Izdo.
F	11	16	70	Si	No	Izdo.
G	11	16	70	Si	No	Izdo.
H	11	16	70	Si	No	Izdo.

Abreviaciones: Texto Corto Texto Completo

1

Menu de Proceso de Texto
escribir nuevos valores, o pulsar <cambiar> para si/no
Tipos de párrafo
<flechas> <car> <cambiar> <ayuda> <menú>

figura 1

En ella se pueden definir hasta 8 formatos diferentes, referenciados por las letras A a H.En cada uno de ellos existen los siguientes atributos:

Margen_Izdo. Define el margen izquierdo.Se visualiza en la linea de formato con la letra **I**.

Sang_Parr. Establece el margen izquierdo de la primera linea del parrafo.Se visualiza en la linea de formato con la letra **P**.

Margen_Dcho. Define el margen derecho.Se visualiza en la linea de formato con la letra **D**.

Salto_Pal.

SI Provoca el cambio de linea, cuando la escritura alcanza el margen derecho.

NO No tiene en cuenta el margen derecho.

Autosangrado

-Con Salto_Pal en **SI**.Al dar **<RET>**, el cursor se situa, como siempre, en la columna **P**, pudiendo con la barra espaciadora moverlo a la derecha "arrastrando la **P**".Es decir, se pueden adentrar más los parrafos, sin pasar por las definiciones de formatos.

-Con Salto_Pal en **NO**.Se puede mover el cursor, situado en la columna **P**, hacia la derecha, pero sin arrastrar la **P**.

Ajuste. Pulsando la tecla **F6**. se pueden obtener:

Izdo

Dcho

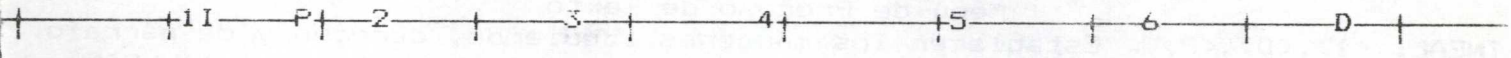
Ambos

Centro

Abreviaciones Es una facilidad que permite almacenar textos cortos (hasta **39** caracteres), a los que se asocia una abreviación de hasta 3 caracteres.De modo que al escribirla, entre blancos, en cualquier parte del texto, queda sustituida por el texto completo.

Hay que respetar la forma de escribir la abreviación, ya que distingue minusculas de mayusculas.

2.-Tecla F8. Desde el estado de inserción pulsar la tecla **F8**, aparecerá una pantalla análoga a la de la figura 2.



Menú de Proceso de Texto

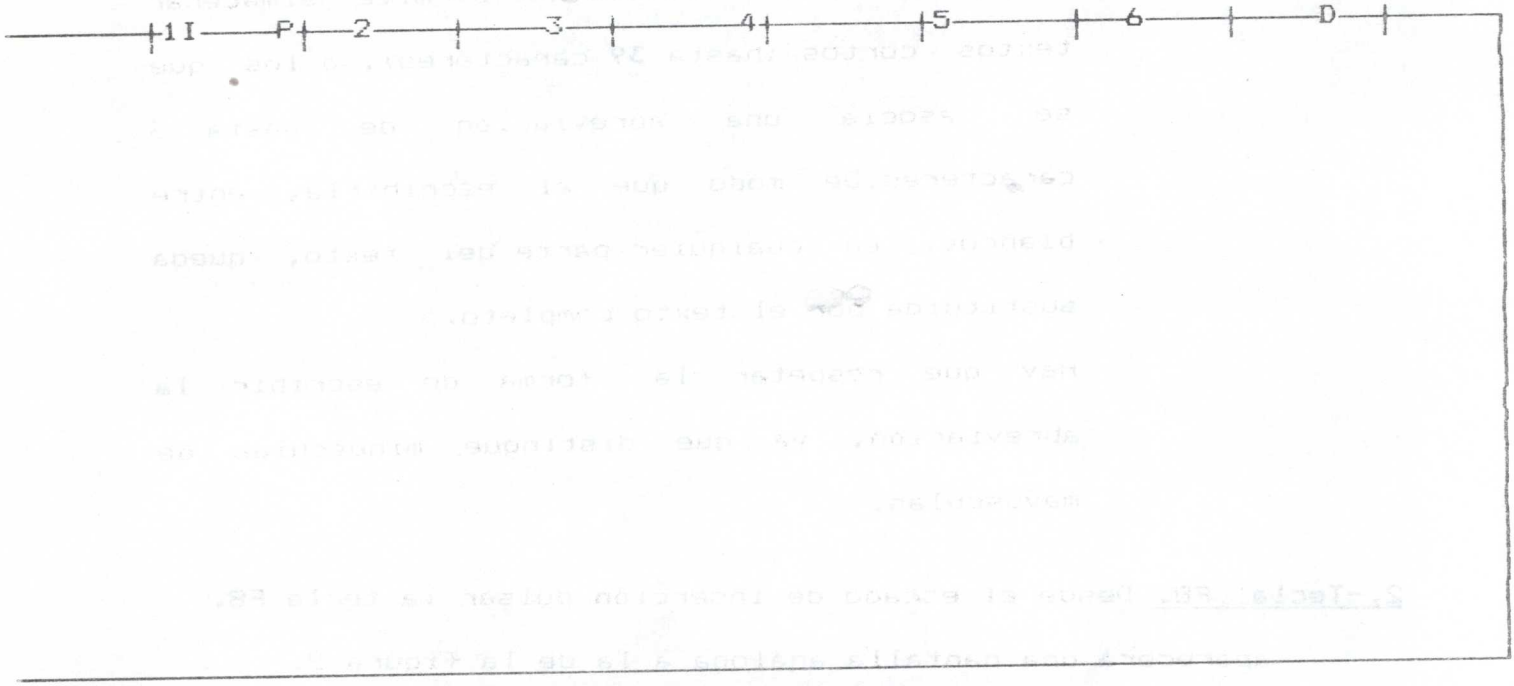
FORMATO: Formato Márgenes_y_Tabul Salto_Pal AutoSangrado Ajuste Guardar
Sí No Izdo.

<izda> <dcha> <ejec> <menú> <formato> <ayuda>

figura 2

Formato. Permite ejecutar un formato previamente definido.

Margenes_y_Tabul. Permite definir margenes y tabulaciones.Al ejecutarlo, presenta una pantalla como la de la figura 3.



Menú de Proceso de Texto
INEAL: <I>,<D>,<P> Establecer los márgenes izquierdo, derecho y de párrafo.
,1 1<->,<+>,<.>,<:> Borra Tab, Pone Tab, Pone Tab Decimal, Pone Tab SIF.
<I> <D> <P> <-> <+> <.> <:> <flechas><tab><tab atrás><ayuda><menú><borr lin>

figura 3

Se puede, moviendo el cursor:

-Determinar nuevos margenes.Pulsando I,P o D.

-Indicar un tope de tabulación.Pulsando +.

-Borrar un tope de tabulación.Pulsando -.

-Determinar una tabulación decimal.Pulsando ..

-Determinar una tabulación tipo SIF.Pulsando :.

Salto_Pal. Tiene el mismo significado que en el comando **Formato.** Se cambia la situación con **RET** o **F10.**

Autosangrado. Tiene el mismo significado que en el comando **Formato.** Se cambia la situación con **RET** o **F10.**

AjusteTiene el mismo significado que en el comando **Formato.** Se cambia la situación con **RET** o **F10..**

Guardar. Permite almacenar el formato así definido.Hay que indicar una letra de la A a la H.

Hay diferencias importantes entre el comando **Formato** y la tecla **F8** :

-El comando solamente define formatos, mientras que la tecla los ejecuta, permitiendo tambien definirlos.

-Las abreviaciones solo pueden definirse con el comando **Formato.**

-Las tabulaciones solo pueden modificarse con la tecla **F8.**

APENDICE C

CARACTERISTICAS DE IMPRESION

La impresión de un fichero dependerá del contenido basico que se haya introducido en el texto (margenes,tipos de letras, etc), pero tambien de los parámetros de impresión.Estos pueden definirse a traves de la **Ventana Imprimir** o mediante los **Comandos de Formato En_texto.**

1.-Ventana Imprimir. Desde el estado de inserción pulsamos la tecla **F3** y aparece una pantalla como la de la figura 1.

```
┌─────┬1I──P┬─2──┬──3─┬──4┬──5──┬──6──┬──D──┐
 Cabecera :      Número línea:      1          Especial: Nada

 Subcabecera:    Número línea:      2          Especial: Nada

    Pie:         Número línea:      2          Especial: Nada
¦-*-¦

        Ancho papel:     80    Margen Superior:      3 Primera página :      1
     Longitud papel:     24    Margen Inferior:      4 Página comienzo:      1
Cabeceras alternas: No         Bloque  Párrafo:      4 Página de final: 32767
          Espaciado:      1    Desplazam. izda:      0

 Dispositivo de impresión: CONSOLA

                        Menú de Proceso de Texto
IMPRIMIR: <flechas> moverse, <impr> imprimir, <menú> vuelve a editar
        Escribir nuevos valores o <cambiar> para sí/no
  <flechas> <ret> <cambiar> <impr> <ejec> <no ejec> <menú> <ayuda> <buscar>
```

figura 1

El significado de los diferentes parámetros es el siguiente:

Cabecera: Permite introducir un texto que aparecerá, en las páginas que especifique el atributo **Especial.**

Número de línea: Especifica la linea en que se debe colocar la cabecera.

Especial: Permite elegir una opción, de entre las siguientes, para la impresión de la cabecera.

Nada Aparece en todas las páginas.

Saltar primera

Saltar impares

Saltar pares

Subcabecera:

Número de línea:

Especial:

Pie:

Número de línea: Cuenta las lineas desde abajo.

Especial:

Ancho papel: Número de columnas que caben en el ancho de papel (normalmente 80 o 132).

Longitud papel: Número de lineas en una hoja de papel completa (normalmente 66).

Cabeceras alternas: Permite cambiar la posición de la cabecera según que la pagina sea impar o par.Se activa o desactiva con F6.

Espaciado: Establece el espaciado entre lineas.

Margen superior: Determina el número de lineas del margen superior.Afecta al texto, pero no a la cabecera y subcabecera.

Margen inferior: Determina el número de lineas del margen inferior.Afecta al texto, pero no al pie.

Bloque de párrafo: Es el tamaño, en número de lineas, de los párrafos que no se quieren cortar al cambio de página.

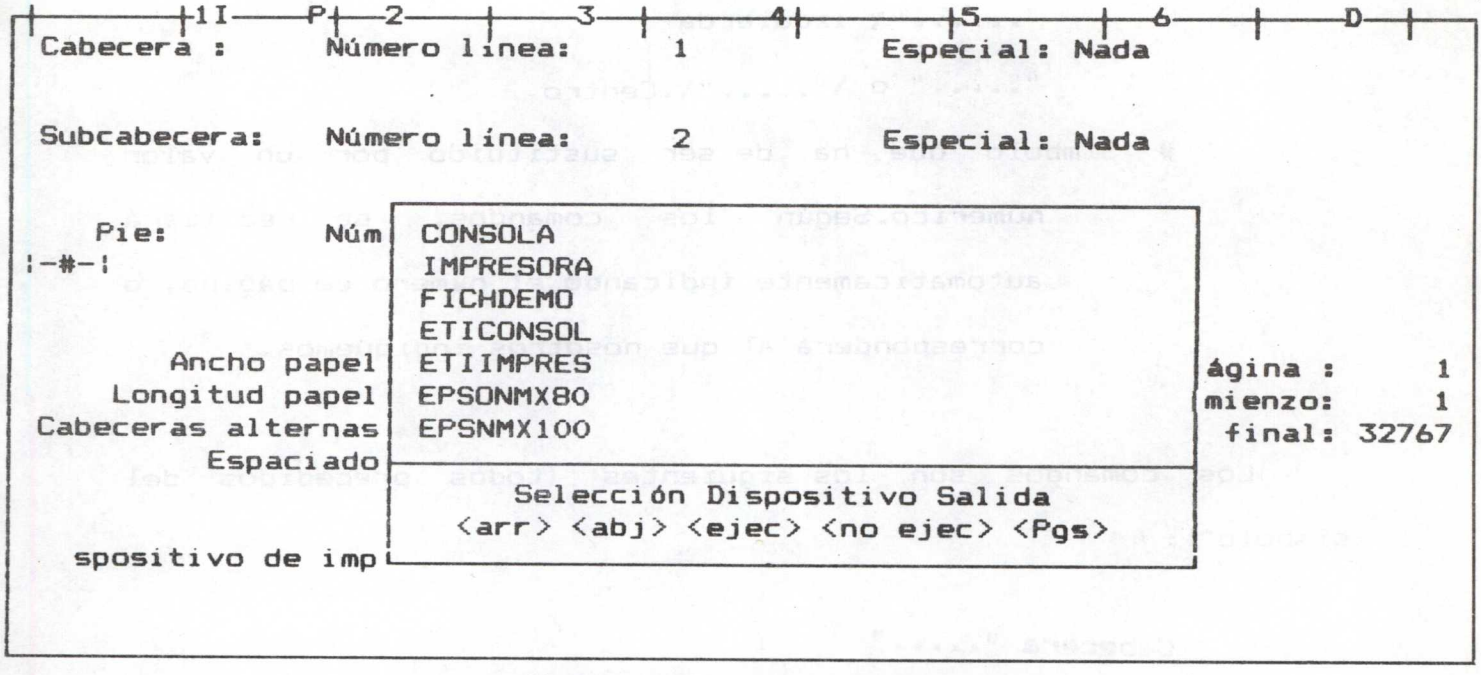
Desp. izdo: Desplaza el texto hacia la derecha, el número de columnas que se indica.

Primera página: Es el número con el que va a empezar la numeración de las páginas.

Página comienzo: Es la primera pagina que imprime.

Página de final: Es la ultima página que imprime.

Dispositivo de impresión: Permite selecionar el dispositivo por el que se va a realizar la impresión.Por defecto asume CONSOLA, pero pulsando la tecla **F4** aparece una ventana como la de la figura 2 que permite seleccionar cualquiera de los disponibles.



Menú de Proceso de Texto

IMPRIMIR: <flechas> moverse, <impr> imprimir, <menú> vuelve a editar
Escribir nuevos valores o <cambiar> para sí/no
<flechas> <ret> <cambiar> <impr> <ejec> <no ejec> <menú> <ayuda> <buscar>

figura 2

2.-Comandos de Formato En_texto. Son comandos que permiten definir los parámetros de impresión, y que como el nombre indica, pueden introducirse en el texto.

Permiten mayor flexibilidad que la **Ventana de Impresion**, pudiendo redefinir las opciones establecidas en ella, y cambiar **comandos En_texto** introducidos previamente.

Todos los comandos han de estar precedidos del simbolo (^).Según los comandos, pueden ser necesarios tambien los siguientes simbolos de control:

"....." Comillas para delimitar un texto usado como parámetro por el comando correspondiente.

\ Barra que permite indicar la posición de escritura en una linea.

\"....." Derecha

"....."\ Izquierda

"....." o \"....."\ Centro

Simbolo que ha de ser sustituido por un valor numerico.Según los comandos, se realizará automaticamente indicando el número de página, o corresponderá al que nosotros indiquemos.

Los comandos son los siguientes (todos precedidos del simbolo^):

Cabecera "....."

Subcabecera "....."

Pie "....."

Linea cabecera #

Linea subcabecera #

Linea pie #

Linea #. Indica la línea en que empieza a imprimirse el texto.(Debe tener acento sobre la í).

Margen sup #. Determina el margen superior.

Margen inf #. Determina el margen inferior.

Página #. Indica con que número comienza la numeración de las páginas.

Página. Provoca el salto de página.

Alterna sí.

Alterna no

Bloque #

Espaciado #

Incluir Fichero.doc. Permite continuar la impresión con el fichero aquí llamado.

APENDICE D

CARACTERES DE SUSTITUCION

Los caracteres de sustitución son los que en vez de imprimirse como parte del texto, actuan como variables que son sustituidas por el valor correspondiente en el momento de impresión.

En el Procesador de Textos del Open Access, para indicar que se trata de un carácter de sustitución se utiliza el simbolo @: unido a cualquiera de las siguientes variables de sustitución:

1.-Sustituciones de un CAMPO de Base de Datos.

Sustituye el nombre del campo por el contenido del campo del registro correspondiente.

@:nombre de campo.

@:nombre de campo +1. Sustituye por el valor del campo más uno.Es valido para campos que contienen valores numéricos.

@:nombre de campo@:"cadena de caracteres". Une al contenido del campo la cadena de caracteres especificada.

2.-Sustituciones de FUNCIONES Especiales.

@:FECHASIS. Queda sustituida por la fecha del sistema.

@:PAGINA. Queda sustituida por el número de pagina.

@:ORDEN. Queda sustituida por el número de orden, que se va incrementando de uno en uno.

3.-Sustituciones condicionales.

Permiten construir una estructura del tipo **IF-THEN-ELSE.** Es decir si la condición se cumple, escribir una cosa (@+=Verdad) y si no se cumple escribir otra (@-=Falso).

@?nombre de campo="expresión"@+"expreA"@-"expreB"

POSIBILIDADES EDUCATIVAS DE LOS PROCESADORES DE TEXTOS

1. INTRODUCCION

Los ordenadores se han introducido ya en muchos colegios y están empezando a tener un papel importante en la enseñanza, muchos profesores especialmente de Ciencias y Matemáticas están interesados en el tema y ven las posibilidades que pueden tener de aplicaciones directas en sus asignaturas.

Sin embargo los profesores de otras áreas están, en general, poco motivados para su uso , muchos de ellos piensan que el ordenador no tiene ninguna utilidad para las clases de humanidades, a excepción de los programas didácticos de Geografía o Historia que puedan encontrarse en el mercado y que quizás no encajan muy bien en el currículo o no se ve cómo emplearlos con clases numerosas y pocos ordenadores.

Es especialmente en el campo de las humanidades donde va a tener su campo de acción el procesador de textos. El ordenador puede servir como un ayudante del profesor, incansable y paciente para guiar a los alumnos y ayudarles a desarrollar técnicas de lectura y escritura de un modo personalizado y es con un procesador de textos donde incluso el profesor más inexperto en este campo informático puede desarrollar programas y actividades para sus alumnos con gran facilidad.

2. USO POR EL PROFESOR

Una primera aplicación que ofrece un procesador de textos en la escuela para su uso por el propio profesorado es la elaboración de las programaciones, temas y ejercicios en el procesador de textos. Todas las modificaciones que, sobre la marcha, o en un futuro quiera hacer en estos materiales serán mucho más sencillas de introducir que si tuviera que hacerlo de nuevo con la máquina de escribir.

El profesor puede crearse archivos con ejercicios, preguntas o actividades sobre los distintos temas de su asignatura que irá aumentando o corrigiendo a lo largo de cursos sucesivos. En cualquier momento podrá sacar de los distintos ficheros preguntas para un examen, textos para hacer comentarios, el plan y las instrucciones para un trabajo o proyecto con los alumnos, etc.

El procesador de textos puede acceder a una base de datos documental con lo que las posibilidades de seleccionar unos temas con otros se amplian considerablemente.

El profesor puede tener en un fichero aparte las respuestas a los ejercicios protegidos con una contraseña, de modo que en un momento determinado puede, si le interesa, entregarlas a los alumnos para que autocorrijan un ejercicio..

3. UTILIZACION DE LOS PROCESADORES DE TEXTOS CON ALUMNOS

En las manos de un profesor creativo y con imaginación, el procesador de textos puede ser una herramienta poderosa, motivadora, sugestiva, abierta y autoevaluadora para los alumnos.

Las dos destrezas básicas que condicionan cualquier tipo de aprendizaje, lectura y escritura, pueden ser practicados a todos los niveles, desde preescolar hasta adultos con programas apropiados hechos en el procesador de textos.

Hay dos tipos de aplicaciones que van a poder hacerse; unas que requieren únicamente el uso de un procesador de textos y otras que estarán hechas en otros lenguajes (Basic o algún lenguaje de autor) bien por el profesor o por casas comerciales y que implican el procesador de textos para el trabajo del alumno. Estas últimas aunque son más costosas de preparar ofrecen una mayor variedad y riqueza de actividades que las primeras.

3.1. Composición de textos, escritura, lectura.

Muchos de los objetivos que tienen los programas de Lengua Española y de Lengua Extranjera, podrían conseguirse más fácilmente si los

alumnos tuvieran acceso a un procesador de textos. Veamos como ejemplo algunos de ellos del programa de Reforma de las Enseñanzas Medias:

- Hacer resúmenes y esquemas de textos escritos y orales de naturaleza diversa.

- Saber redactar cartas, instancias y otros documentos usuales así como monografías, informes y recensiones.

- Saber realizar fichas de lectura, citar adecuadamente la bibliografía y manejar ficheros.

- Crear textos de carácter literario.

Indudablemente todos estos objetivos pueden conseguirse de la manera tradicional, con lápiz y papel. Sin embargo, los alumnos (los adultos también) son reacios a escribir, les molesta tener que organizar un trabajo, escribirlo, corregirlo, pasarlo a limpio y en general, tienden a escribir poco y a corregir lo menos posible.

El texto en papel es estático, es difícil cambiarlo a menos que se reescriba de nuevo por lo que la tarea es penosa y no resulta gratificante.

El texto en un ordenador parece vivo, las palabras son luces en la pantalla y pueden cambiarse inmediatamente, el equivocarse no tiene demasiada importancia.

El alumno se siente más libre cuando habla que cuando escribe. Al hablar expresa muchas más ideas porque no se preocupa demasiado del modo en que las dice y puede ir matizándolas o modificándolas a lo largo del discurso. Al escribir se encuentra atado buscando la manera y el estilo correcto para explicarlas. Una solución es practicar la escritura tan libremente como el hablar, sin preocuparse de fal-

tas, estructuras o corrección gramatical. Después de este primer intento y una vez que tiene todas las ideas podrá organizarlas, cambiar el estilo, corregir la ortografía tantas veces como quiera y hasta el último momento. Cuando entregue el trabajo, después de imprimirlo, la copia no tendrá tachaduras o enmiendas; cualquier modificación que sugiera el profesor al trabajo ya terminado, no le implicará volver a escribirlo sino editarlo de nuevo y modificarlo en la pantalla.

3.2. Situaciones de uso del procesador de textos en el aula.

Hasta aquí las ventajas que podrían derivarse del uso de los procesadores de textos serían meramente mecánicas, un mejor manejo de los textos y una facilidad para corrección de errores.

Veremos algunos aspectos de los procesadores de textos que permitirían su empleo en distintas situaciones en el aula. La posibilidad de modificar un texto en pantalla sin necesidad de reescribirlo da lugar a nuevas situaciones de aprendizaje que representan una aportación con respecto a los medios tradicionales. Algunos ejemplos serían:

- Dado un documento base, una instancia, una carta comercial, etc., el alumno lo modificará para solicitar algo distinto de lo planteado en el original. Al no tener que reescribirlo completo sería posible que el alumno hiciera varios ejercicios del mismo tipo o de otro similar concentrando todas sus energías en el contenido del ejercicio y no en la actividad manual de escribirlo.

- Estas mismas aplicaciones podrían utilizarse en la clase de idiomas modernos. El alumno tendría que sustituir una serie de datos en un texto de modo que el nuevo texto ofreciera una versión personalizada del documento o cubriera algún punto específico del programa del curso modificar los verbos de un párrafo escrito en presente que la narración estuviera en un tiempo verbal diferen-

te, pasado o futuro por ejemplo.

La posibilidad de mover bloques dentro de un texto puede permitirnos realizar con los alumnos experiencias en el campo del análisis y organización de textos. Algunos ejemplos de utilización que hacen uso de esta característica son:

- Se ofrece al alumno un texto en el que los párrafos no estén ordenados. El alumno lee los distintos párrafos y decide cual es el orden lógico de esa narración. Esto puede hacerse a distintos niveles: para niños de E.G.B. una narración tipo cuento o fábula, un texto literario más complejo en BUP o incluso un razonamiento o ensayo de tipo filosófico.

  El alumno empleará las teclas de función para marcar los párrafos y los moverá a los lugares que él crea convenientes. Después puede sacar su versión por la impresora y comentarla con la de otros compañeros, finalmente pueden comprobar si corresponde con la versión original del texto cargando en el ordenador el texto original que le proporcionará el profesor en un disco de respuestas una vez que haya terminado su ejercicio.

  El mismo ejercicio puede aplicarse a lengua extranjera siempre que el texto sea más sencillo el alumno tendrá que identificar palabras como: "primero", "luego", "más tarde", "por último", etc.

- La misma aplicación realizada con un poema al que se le han alterado el orden de las estrofas, o incluso, el orden de los versos, si no es demasiado largo, puede dar lugar a que los alumnos aprendan o practiquen la versificación de un modo creativo. Según el tipo de poema que elijamos los objetivos de la actividad serán distintos. Si les damos los versos que componen un soneto desordenado y les pedimos que lo reconstruyan en forma de soneto, probablemente se acercarán mucho al poema original. Si les damos los versos de un poema más libre o poesía moderna posiblemente obtengamos una variedad de composiciones que una vez impresas pueden ser de gran valor en clases sucesivas para su comentario en pequeños grupos o en gran grupo o la elaboración de un manual.

- Si se quiere sacar más partido educativo de este tipo de actividades, y una vez que el alumno está más familiarizado con el manejo del procesador de textos, el profesor puede preparar un ejercicio haciendo de dos textos diferentes uno único mezclando párrafos de uno y otro o estrofas si son poemas.

  Los alumnos tendrán que decidir qué párrafos juntos tienen una unidad y reparar los dos textos. Esto implica una lectura comprensiva, análisis de estilo si son textos de épocas diferentes, organización de un texto etc.

  El típico ejercicio de las líneas cruzadas en una conversación de teléfono en la que el alumno tiene que escribir las dos conversaciones que se cruzan es mucho más fácil de realizar si se va visualizando los cambios en pantalla que haciéndolo por escrito.

  La facilidad que ofrece un procesador de textos para este tipo de actividades es innegable. Algunas de ellas es muy difícil realizarlas sin él.

  Se describe a continuación un ejemplo de situación de uso en el aula:

  El profesor prepara varios textos cortos de 8 a 10 líneas, textos literarios o periodísticos que no sean conocidos por los alumnos y carga en cada ordenador un texto diferente. El alumno o alumnos que trabajan en cada ordenador leen su texto atentamente y tratando de mantener el estilo y la coherencia del párrafo, insertan en él una o varias frases elaboradas por ellos mismos, y sacan una copia impresa del texto con la frase que ellos han redactado incorporada en algún lugar del mismo. Después pasan este trabajo a otro grupo de alumnos que no conoce el texto original porque ha estado trabajando con otro diferente, éstos intentarán adivinar cual es la frase "intrusa" en el texto. La validez del ejercicio residirá en el interés que pongan los alumnos en escribir sus frases del modo más semejante y coherente al del autor del texto.

  Todos estas ideas pretenden tan sólo introducir algunos ejemplos del uso más sencillo del Procesador de textos. Sin duda cuando este instrumento sea conocido por los profesores de las distintas áreas, podrán desarrollar diferentes actividades acordes con sus necesidades.

4. PROGRAMAS CON PROCESADORES DE TEXTOS

Aunque son de difícil localización en España, existen ya programas que ayudan en la redacción de un texto, estimulando la marginación, la asociación de ideas, la organización de un tema en concreto, mediante un esquema, etc.

Veamos algunos ejemplos descritos en artículos de revistas:

Burus en su artículo "Simulating Invention in English Composition throug Computer Assisted Instruction" (English Technology Agosto 1980) describe cóm, a través de mensajes en pantalla los programas pueden ayudar al usuario sugiriendo ideas y ayudando a organizarlas una vez escritas.

Por ejemplo: si un escritor se para más de diez minutos, el crdenador puede animarle a que escriba más, identificando la palabra más frecuente del texto y enviando el mensaje: ¿PUEDES DECIR ALGO MAS SOBRE X?" asumiendo que si X es una palabra repetida, posiblemente sea interesante como tópico. En un primer paso el programa por medio de preguntas puede ayudar al usuario a hacer un primer borrador con todas las ideas sobre el tema. El programa hará las preguntas y recogerá las respuestas del que escribe ayudándole a organizar el guión del tema con indicaciones del tipo:

"EN ESTA PRIMERA FASE DE EXPLORACION SE PIDE QUE CLARIFIQUES EL OBJETIVO DE TU ENSAYO SOBRE 'X' ¿PODRIAS DESCRIBIRLO BREVEMENTE COMPLETANDO LA SIGUIENTE FRASE: EL PROPOSITO DE ESTE ENSAYO ES .....

Una vez que el usuario ha contestado, el ordenador responde de nuevo sugiriendo alguna otra manera de continuar:

"BIEN, VAMOS A HABLAR DE ESTE OBJETIVO. VAMOS A COMENZAR CON UNA SESION DE BRAINTORMING. ¿QUE OBJETOS O IDEAS ASOCIAS CON 'X'? ¿COMO PODRIAN INCLUIRSE EN ESTE TEMA.

De este modo interaccionando con el ordenador puede ir aprendiendo a esquematizar las ideas sobre cualquier tópico, para posteriormente desarrollarlas en un ensayo. Indudablemente estos programas no pueden responder sustancialmente a los aspectos lógicos del tema pero hacen sentir al escribir que hay alguien leyendo.

Hay programas que pueden identificar errores superficiales de ortografía y estilo.

Las correcciones ortográficas las realiza comparando todas las palabras del texto con un diccionario que contiene el mismo programa. No identificará errores de palabras que están mal escritas en un contexto pero serían válidas en otro, por jemplo: vaca y baca, a, ha, ¡ha!, etc....

Las correcciones de estilo se reducen a las más elementales, tales como repetición de palabras, palabras innecesarias, vulgarismos. El programa puede sugerir suprimir o cambiar algunas de las palabras con mensajes del tipo:

"LAS PALABRAS PARPADEANTES PUEDEN NO SER NECESARIAS"

"LAS PALABRAS PARPADEANTES SE REPITEN VARIAS VECES ¿PUEDES SUSTITUIR ALGUNAS POR SINONIMOS?

En esta línea están:
WRITER'S WORKBENCH diseñado para gestión y para escribir informes científicos aunque también se estaba probando en Colorado para clases de composición y estilo en la Universidad.

Otros programas THE EPISTLE de IBM y CATCH pueden identificar partes de frases y hacer un análisis gramatical elemental, pudiendo detectar errores sencillos de construcción. Por ejemplo: si falta la segunda parte de una condicional, un relativo sin su antecedente o consecuente etc., CATCH tiene mensajes que alertan al escritor sobre posibles problemas en el texto. Después de escribir un documento, el usuario de

catch seleccione una lista de sugerencias para revisión referentes a claridad, organización, estructura de frases, puntuación, etc. CATCH envía mensajes del tipo: "¿HAY UNA FRASE QUE INDIQUE EL OBJETIVO DEL TEXTO?"

"¿HAS ESCRITO UNA CONCLUSION?"

Al ir revisando los distintos apartados el usuario va automatizando el proceso de revisión de un documento literario o científico, analizando cada problema por separado y puede mejorar sus posibles deficiencias personales.

En los últimos meses han proliferado programas de este tipo para escolares "THE WRITER'S WORKSHOP" de la casa Milliken aparte del procesador de textos tiene discos adicionales que incluyen ejercicios previos a la escritura, para general voleas, organizar las líneas generales del tema, objetivos a quien va dirigido el tipo de lenguaje que hay que emplear, etc.

QUILL también es un programa que facilita la redacción ayudando a los usuarios a desarrollar ideas para escribir con la parte que se llama "Planner", tiene también otra opción la de "library" que le permite almacenar información de distintos tipos a la que puede acceder en un momento determinado.

Desde luego, el profesor puede crear sus propios programas para ayudar a sus alumnos a escribir con ordenador.

Pueden diseñar sus modelos de composición. Escribirlos en su procesador de textos y grabarlos para que sus alumnos las empleer en clase. Ejercicios previos de generar ideas y organizarlas similares a las de los programas QUILL y WRITER'S WORKSHOP
pueden escribirse con el procesador de textos y guardarse en archivos aunque estos programas no serán tan interactivos como el QUILL o el WRITER'S WORKSHOP tiener la ventaja de que los profesores pueden crearlos de acuerdo con las necesidades espcíficas de sus alumnos.

Otros programas favorecen la lectura y la escritura al mismo tiempo por ejemplo los que ofrecen textos dinámicos del tipo de "Elige tu

aventura" Estos programas ofrecen una historia de estructura flexible en la que se les dé un párrafo inicial tras el cual el lector ha de optar por una de entre varias posibilidades para continuar leyendo o escribir la continuación que le parezca más oportuna y enlazarla con otras páginas de la historia o continuar escribiendo hasta el final.

Otros permiten personalizar los textos de modo que antes de comenzar a leer un cuento o historia el programa pide al lector algunos datos (nombre, nombre de sus padres, amigos, colegio, etc.) Después al presentar la narración el niño que está leyendo y sus amigos serán los protagonistas.

También hay los que permiten que el niño haga dibujos de tipo comic y pongan texto a las ilustraciones (kidwriter, story maker, bank street story book)

A nivel de Enseñanzas Medias, hay programas que pueden ayudar al aprendizaje de lectura rápida o lectura comprensiva o incluso crítica literaria, pero éstos ya son específicos de Enseñanza Asistida por Ordenador y no propiamente de procesadores de textos.

MODELO DE APLICACION: REDACCION

Corrección de una redacción en pequeño grupo

Objetivos:

- Mejorar la expresión escrita, estilo y ortografía.
- Comparar dos versiones de un mismo trabajo

Este tipo de actividades requiere que los alumnos trabajen en grupos de dos a cuatro alumnos. Hay un trabajo previo en el que los alumnos escriben individualmente redacciones sobre un tema propuesto por el profesor. Después cada pequeño grupo lee las composiciones de cada alumno y elige la más adecuada para meterla en el ordenador, mejorarla y hacer de ella un trabajo de equipo.

Hay que estimular a los alumnos para que experimenten con la composición y prueben todas las posibilidades que el Procesador de Testos les ofrece para organizar y expresar sus ideas. Previamente habrán hecho una copia de seguridad para evitar destruir la composición si cometen algún error.

La práctica de la aplicación con los alumnos sería la siguiente:

1. Leer las composiciones preparadas por los compañeros de grupo y elige una para meterla en el ordenador.

2. Escribe la redacción en el ordenador usando el editor del procesador de textos. No importa si hay faltas en esta primera versión.

3. Guarda en un archivo la redacción, asignándole un nombre.

4. Intenta mejorar la redacción cambiando lo que haga falta, usando el procesador de textos para corregir la ortografía, borrar palabras o

frases, trasladar palabras, frases o párrafos. Añade nuevas ideas. Incluir, si es posible, alguna idea sobre el tema que hubiera en las otras composiciones que no se han metido en el ordenador.

5. Dar un nuevo nombre a esta composición final, guardarla en un archivo y sacar una copia por la impresora.

6. Comparar esta versión con la composición original.

Organización y elaboración de una redacción con el Procesador de Textos

Objetivos:

- Generar ideas en torno a un tema
- Organizar las ideas en un esquema
- Agrupar las ideas bajo epígrafes
- Dar forma al bosquejo creado para iniciar una redacción
- Modificar la composición añadiendo, anulando frases y párrafos.
- Guardar en archivos las distintas fases de la composición y la versión final de la misma.

Los alumnos trabajarán en grupos de dos a cuatro y se pretende con esta actividad, iniciarles en las técnicas de generar ideas (brainstorming) en torno a un tema, para después organizarlas y agruparlas hasta darles forma en una composición.

Muchos alumnos se quedan parados ante el papel porque no saben qué decir o cómo empezar.
La actividad les irá guiando desde una sesión inicial corta, en la que escriben en el ordenador todo lo que se les ocurre sobre el tema propuesto sin pensar mucho, ni preocuparse del modo de expresarse, hasta una sesión final para conseguir una versión lo más perfecta posible.

Hay que inculcar en los alumnos la teoría de que hay que empezar a escribir para que surjan ideas y que buenas ideas generan mejores

ideas. El hecho de poder mover y corregir las frases sin tenerlas que reescribir será un aliciente para ellos. El hacerlo con otro o más compañeros les estimula a pensar.

Es aconsejable dejar una línea en blanco entre cada idea para facilitar el movimiento de bloques y la inserción de frases en la fase de organización

Después de la sesión inicial se sugiere a los alumnos que agruper las ideas en apartados bajos distintos epígrafes funcionando las ideas que sean similares en ura sola y guardando en un apartado distinto aquellas que, de momento, no encajen bajo ninguno de los epígrafes creados por si, se considerase oportuno emplearlas en fases posteriores de la composición.

Los procesadores de textos que tienen la posibilidad de guardar textos en distintas ventanas, favorecen este tipo de actividad ya que se puede visualizar un apartado distinto en una ventana a la vez que se escribe en otra parte de la pantalla.

Los alumnos pasarán a ordenar lógicamente los distintos apartados y las ideas dentro de cada apartado estudiando la posibilidad de añadir o suprimir algunas si el esquema general no queda bien equilibrado.

Después de esta fase de preescritura guardan lo que han hecho en un archivo y comienzan a dar forma literaria a las ideas generales, modificándolas con los procedimientos habituales de los procesadores de textos.

Cuando tienen la versión final sacan una copia en la impresora. El profesor podrá evaluar las distintas fases del trabajo si pide una copia de los archivos guardados en los distintos momentos.

La práctica con los alumnos sería la siguiente:

1. Reúnete con tus compañeros de grupo e ir escribiendo en el editor del Procesador de Textos todas las ideas que se os ocurran sobre el tema.

No os preocupeis del orden en que se os ocurren ni el modo en que las expresais. Dejar una linea en blanco entre cada idea. Trabaja generando ideas durante 10 minutos. Asigna un nombre al documento y guardalo en un archivo.

2. Agrupar las ideas en distintos bloques. Pensar para ello cuáles tienen una relación entre sí. Si hay varias parecidas, redactar una que las resuma. Para hacerlo, emplear todas las posibilidades de corrección y movimiento de palabras que os proporciona el Procesador de Textos. Guarda las ideas que os queden sueltas en un archivo aparte.

3. Organizar los apartados que os han salido de un modo lógico. Piensa que la composición debe tener una introducción, un desarrollo y la parte final o conclusión.

4. Organiza las ideas dentro de cada apartado dándoles forma para elaborar un primer borrador. Comprueba que el tema tiene un equilibrio y no hay una parte que quede demasiado escueta o demasiado larga. Añadir o suprimir frases según os convenga.

5. Guardar una copia de seguridad, dándole un nombre distinto del anterior en un archivo.

6. Modifica esta primera versión, revisando la ortografía, reformando o desplazando las frases, si es necesario.

7. Poner nuevo nombre a la versión final y guárdala en el disco usando el nuevo nombre.

8. Saca una copia por la impresora. Considerar el aspecto del texto sobre el papel. El menú de impresión contiene algunas ideas sobre formatos, practicar distintas modalidades de márgenes espaciados, encabezamientos, etc. Ensayar diferentes salidas hasta conseguir una que os guste.

APLICACION DIDACTICA

LENGUA ESPAÑOLA: TEXTOS EN PROSA

Objetivos:

- Lectura comprensiva de un texto.
- Organización de textos en prosa.
- Resumen de un texto.
- Debate sobre un texto leído.

Aplicación 1ª: Ordenación de textos

Se presenta a los alumnos en el ordenador un texto corto que tenga una unidad en sí mismo, en el que se han desordenado los párrafos.
El alumno deberá leer con atención el texto y sirviéndose de la facilidad que proporciona el movimiento de bloques con el tratamiento de textos lo ordena del modo que él crea más oportuno. El profesor puede seleccionar los textos de acuerdo con el nivel de sus alumnos, siendo posible esta actividad desde E.G.B. hasta B.U.P. dependiendo de la dificultad del texto elegido.

Los alumnos después de realizar el ejercicio comentan y razonan su versión antes de conocer la versión original y finalmente comparan su trabajo con el texto original.

Los textos que se presentan como modelo de esta actividad están pensados para alumnos de 7º y 8º de E.G.B. y son los siguientes:

Londres de Pio Baroja que se presenta alterado del siguiente modo:

LONDRES1

Al acercarnos a la ciudad las casas eran ya más altas, la niebla se hacía más densa y más turbia.

LLoviznaba.Las grandes chimeneas de las fábricas vomitaban humo denso y negro;el rio amarillo manchado de vetas oscuras arrastraba al impulso de la marea tablas,corchos,papeles y haces de paja.

La animación y el movimiento en el Támesis comenzaban a ser extraordinarios.La niebla y el humo iban espesándose a medida que nos acercabamos a Londres, y en la atmòsfera opaca y turbia apenas si se distinguian los edificios de las dos orillas.

A uno y otro lado se veian grandes almacenes simétricos,montones de carbón piedra,pilas de barricas de distintos colores.Parecía que se iba pasando por delante de varios pueblos levantados en las orillas...

Los vapores entraban y salían de los Docks, el horizonte se veía surcado por palos de barco, en el río se mezclaban gabarras y botes;cruzaban el aire chorros de vapor, silbaban las calderas de las máquinas y en medio de la niebla y del humo subían suavemente,izados por las grúas que giraban con la caseta del maquinista, barricas de colores diversos, sacos y fardos.

Los alumnos realizan las siguientes actividades:

- Carga en tu ordenador uno de los siguientes documentos: LONDRES1 o LIBER1

  En estos textos los párrafos no están ordenados del mismo modo que los escribieron sus autores.

- Lee el texto: ¿Cuál crees tú que debería ser el primer párrafo? Emplea el procedimiento de marcar y mover bloques y colócalo en primer lugar.
- Haz lo mismo con los otros párrafos.
- Saca una copia de tu versión con la impresora.
- Compara y comenta con tus compañeros tu versión y la de ellos. ¿En qué se ha basado cada uno para reconstruir el texto?
- Carga en tu ordenador la versión original del texto. Están archivadas como LONDRES2 y LIBER2.
- Saca una copia por la impresora y compara la versión del autor con la tuya y la de tus compañeros.
- Las versiones originales de los dos documentos son las siguientes:

LONDRES2 ("La Ciudad de la Niebla" Pio Baroja)

La animación y el movimiento en el Támesis comenzaban a ser extraordinarios.La niebla y el humo iban espesándose a medida que nos acercabamos a Londres, y en la atmòsfera opaca y turbia apenas si se distinguian los edificios de las dos orillas.

LLoviznaba.Las grandes chimeneas de las fábricas vomitaban humo denso y negro;el rio amarillo manchado de vetas oscuras arrastraba al impulso de la marea tablas,corchos,papeles y haces de paja

A uno y otro lado se veian grandes almacenes simétricos,montones de carbón piedra,pilas de barricas de distintos colores.Parecía que se iba pasando por delante de varios pueblos levantados en las orillas...

Parece conveniente que los párrafos que se desordenen sean largos y que no haya más de 5 ó 6 párrafos en cada texto, de modo que sean fáciles de manejar en una primera fase.

El texto libertad de Platero y yo, se presentó en cuatro bloques desordenados.

LIBER1

Monté en Platero, y, obligándolo con las piernas, subimos, en un agudo trote, al pinar. En llegando bajo la sombra cúpula frondosa, batí palmas, canté, grité. Platero , contagiado, rebuznaba una vez y otra, rudamente. Y los ecos respondían, hondos y sonoros, como en el fondo de un gran pozo. Los pájaros se fueron a otro pinar, cantando.

Platero, entre las lejanas maldiciones de los chicos violentos, rozaba su cabezota peluda contra mi corazón,dándome las gracias, hasta lastimarme el pecho.

Llamó mi atención,perdida por las flores de la vereda, un pajarillo lleno de luz que sobre el húmedo prado verde, abria sin cesar su preso vuelo policromo.Nos acercamos despacio, yo delante, Platero detrás.Había por allí un bebedero umbrío, y unos muchachos traidores le tenían puesta una red a los pájaros.El triste reclamillo se levantaba hasta su pena,llamando,sin querer ,a su hermano del cielo.

La mañana era clara, pura, traspasada de azul.Caía del pinar vecino un leve concierto de trinos exaltados,que venía y se alejaba,sin irse, en el manso y aúreo viento marero que ondulaba las copas.¡Pobre concierto inocente, tan cerca del mal corazón!

Aplicación 2ª: Inserción de textos

Otra aplicación con textos en prosa que aparte de los objetivos señalados incluiría un análisis más detallado del texto, sería la siguiente:

Se ofrece al alumno un texto en el que se han suprimido algunas frases de la versión original que se le ofrecen en una lista aparte.

El alumno tiene que ver en qué parte del texto pueden tener sentido esas frases y colocarlas donde cree que deben ir, insertándolas con el procesador de textos.

La posibilidad de insertar con facilidad en cualquier lugar del texto y que se formatee automáticamente el párrafo hace que sea un ejercicio adecuado para realizarlo en un procesador de textos.

El texto que se presenta para la actividad es el siguiente:

IDILIO1

Los niños han ido con Platero al arroyo de los chopos,y ahora lo traen trotando,todo cargado de flores amarillas.Allá abajo les ha llovido -aquella nube fugaz con sus hilos de oro y plata, en los que tembló el arco iris-.Y sobre la empapada lana del asnucho, las campanillas mojadas gotean todavía.

¡Idilio fresco, alegre, sentimental! ¡Hasta el rebuzno de Platero se hace tierno bajo la dulce carga llovida ! vuelve la cabeza y arranca las flores que su bocota alcanza.Las campanillas, le cuelgan,un momento, entre el blanco babear verdoso ¡Quién, como tú,Platero, pudiera comer flores ... y que no le hicieran daño!

¡Tarde equívoca de Abril!... Los ojos brillantes y vivos de Platero copian toda la hora de sol y lluvia, en cuyo ocaso,se ve llover,deshilachada, otra nube rosa.

Al acercarnos a la ciudad las casas eran ya más altas, la niebla se hacía más densa y más turbia.

Los vapores entraban y salían de los Docks, el horizonte se veía surcado por palos de barco, en el río se mezclaban gabarras y botes;cruzaban el aire chorros de vapor, sibaban las calderas de las maquinas y en medio de la niebla y del humo subían suavemente,izados por las grúas que giraban con la caseta del maquinista, barricas de colores diversos, sacos y fardos.

LIBER2

Llamó mi atención ,perdida por las flores de la vereda, un pajarillo lleno de luz que sobre el húmedo prado verde, abria sin cesar su preso vuelo policromo.Nos acercamos despacio, yo delante, Platero detrás.Había por allí un bebedero umbrío, y unos muchachos traidores le tenían puesta una red a los pájaros.El triste reclamillo se levantaba hasta su pena,llamando,sin querer, a su hermano del cielo.

La mañana era clara, pura, traspasada de azul.Caía del pinar vecino un leve concierto de trinos exaltados,que venía y se alejaba, sin irse, en el manso y aúreo viento marero que ondulaba las copas.¡Pobre concierto inocente, tan cerca del mal corazón!

Monté en Platero, y, obligándolo con las piernas, subimos, en un agudo trote, al pinar. En llegando bajo'la sombra cúpula frondosa, batí palmas, canté, grité. Platero , contagiado, rebuznaba una vez y otra, rudamente. Y los ecos respondían, hondos y sonoros, como en el fondo de un gran pozo. Los pájaros se fueron a otro pinar, cantando.

Platero, entre las lejanas maldiciones de los chicos violentos, rozaba su cabezota peluda contra mi corazón, dándome las gracias, hasta lastimarme el pecho.

Inserta las siguientes frases en el lugar que te parezca más oportuno del texto anterior:

-sobre el campo de San Juan,
-entre juegos sin razón y risas desproporcionadas,
-que veló el prado verde
-como en una lira de llanto,
-De cuando en cuando,
-níveas y gualdas,
-y luego se van a la barrigota cinchada.

Los alumnos realizan las actividades que se relatan a continuación:

Carga en tu ordenador el documento Idiliol

- Las frases que hay al comienzo del documento se han entresacado del texto original.
- Lee atentamente el párrafo y mediante el procedimiento de marcar y mover bloques, coloca cada uno en el lugar del texto que te parezca más apropiado.
- Saca por la impresora una copia de tu versión y coméntala con tus compañeros razonando tus decisiones.
- Carga en tu ordenador la versión original que está archivada con el nombre Idilio2
- Saca una copia por impresora y compárala con tu ejercicio y el de tus compañeros.

La versión original del texto es la siguiente:

IDILIO2

Los niños han ido con Platero al arroyo de los chopos,y ahora lo traen trotando, entre juegos sin razón y risas desproporcionadas,todo cargado de flores amarillas.Allá abajo

les ha llovido -aquella nube fugaz que veló el prado verde con sus hilos de oro y plata, en los que tembló, como en una lira de llanto, el arco iris-.Y sobre la empapada lana del asnucho, las campanillas mojadas gotean todavía.

¡Idilio fresco, alegre, sentimental! ¡Hasta el rebuzno de Platero se hace tierno bajo la dulce carga llovida ! De cuando en cuando, vuelve la cabeza y arranca las flores que su bocota alcanza.Las campanillas, níveas y gualdas, le cuelgan,un momento, entre el blanco babear verdoso y luego se van a la barrigota cinchada.¡Quién, como tú,Platero, pudiera comer flores ... y que no le hicieran daño!

¡Tarde equívoca de Abril!... Los ojos brillantes y vivos de Platero copian tyoda la hora de sol y lluvia, en cuyo ocaso, sobre el campo de San Juan, se ve llover, deshilachada, otra nube rosa.

APLICACION DIDACTICA

LENGUA ESPAÑOLA: Poesía

Objetivos:

- Motivar a los alumnos hacia la poesía
- Conocer la versificación y rima
- Analisis de la estructura de un poema
- Crear poemas personales a partir de versos de otro poema.
- Manejar el Procesador de Textos para editar documentos, mover bloques, sacar copias por impresora, etc.

Con estas aplicaciones de textos en verso se pretende que los alumnos ejerciten de un modo diferente sus conocimientos sobre versificación y rima y también hagan una lectura más atenta y reflexiva de los poemas.

Aplicación primera

Se presenta a los alumnos en el ordenador un poema cuyos versos están desordenados y se les pide que los ordenen del modo que crean más apropiado. El profesor selecciona el poema de acuerdo con la edad y los objetivos del programa de Lengua de los alumnos. Si los alumnos están estudiando métrica y versificación se elegirán formas poéticas tales como sonetos, décimas, redondillas, églogas, odas, etc., en las que para reconstruir el poema, el alumno tenga que fijarse en la rima de los versos. Los poemas no deben ser muy largos.

También puede plantearse la actividad con poesía más libre en la que el alumno creará su propio poema a partir de los versos que se le dan, sin que tenga que coincidir exactamente con la versión del autor.

Una vez creadas las distintas versiones, los alumnos pueden comentarlas en pequeño grupo explicando las razones de su composición.

El profesor, después de este ejercicio con el que el alumno ha recreado en cierto modo el poema, y se ha obligado a una reflexión sobre el mismo, puede hacer un comentario de textos del modo tradicional.

En cursos más elementales pueden exponerse en el tablón de la clase las diferentes versiones que han surgido de un poema.

La práctica se plantea así a los alumnos:

1. Carga en el ordenador el documento "ALBERTI1" o "MIGUEL H.1" Los versos de los poemas están desordenados. Marcando y moviendo bloques ordénalos del modo que a tí te parezca más apropiado.

2. Saca una copia de tu versión por la impresora y coméntala con tus compañeros.

3. Carga el documento original ALBERTI2 o M.HDZ2 y compara los poemas de los autores con los tuyos.
   Los poemas se presentan desordenados del siguiente modo.

ALBERTI1 (MI CORZA)

Mi corza blanca
los lobos, buen amigo,
que huyeron por el rio
dentro del agua.
Los lobos la mataron
al pié del agua
mi corza buen amigo
los lobos la mataron

MiguelH1

Reiste tú junto al río,
niño solar.Y ese día
y el agua te sonreía.
el pez más viejo del río
Tan sombrío llegó a estar
de tanta sabiduría
(nada el agua le divierte)
que después de meditar,
brillantemente sombrío,
como amontonó,vivía
y el agua le sonreía.
El pez más viejo del río
tomó el camino del mar,
es decir, el de la muerte.
se quitó el aire sombrío

La versión original de los poemas sería así:

Alberti2 (Mi corza)

Mi corza buen amigo,
mi corza blanca
los lobos la mataron
al pié del agua.
Los lobos, buen amigo,
que huyeron por el río.
Los lobos la mataron
dentro del agua.

M.Hdz2

El pez más viejo del río
de tanta sabiduría
como amontonó, vivía
brillantemente sombrío,
y el agua le sonreía.
Tan sombrío llegó a estar
(nada el agua le divierte)
que después de meditar,
tomó el camino del mar,
es decir, el de la muerte.
Reiste tú junto al río,
niño solar. Y ese día
el pez más viejo del río
se quitó el aire sombrío
y el agua te sonreía.

Aplicacion segunda

Tiene el mismo planteamiento que la anterior pero aquí lo que se desordena son las estrofas de un poema, no de versos y el alumno tiene que ordenarlas moviendo los bloques.

Algunos poemas sencillos tipo fábulas ayudan al alumno a una lectura cuidadose, y serían apropiados para primera etapa de E.G.B. Otros más complejos necesitarían una mayor reflexión sobre el tema, el ritmo, etc..

Veamos como modelo la práctica con dos documentos de distinta dificultad.

1. Carga en el ordenador el documento FABULA1. Las estrofas están desordenadas. Ordénalas de modo que tengan sentido.

FABULA1

Los que por cuestiones
de poco momento
dejan lo que importa
llévense este ejemplo

Por entre unas matas,
seguido de perros,
no diré corría,
volaba un conejo

¿Qué ha de ser?-responde-.
Sin aliento llego;
dos pícaros galgos
me vienen siguiendo.

De su madriguera
salió un compañero,
y le dijo:-Tente,
amigo,¿Qué es esto?

-Si-replica el otro-,
por allá los veo,
pero no son galgos,
-¿pues qué son?-podencos.

-Son podencos ,vaya,
que no entiendes de eso.
-Son galgos,te digo
-digo que podencos

¿qué,podencos dices?
-sí, como mi abuelo.
Galgos y muy galgos;
bien visto lo tengo.

En esta disputa
llegando los perros,
pillan descuidados
a mis dos conejos .

(Iriarte)

2. Saca una copia de tu versión por la impresora.

3. Carga el original FABULA2 en tu ordenador y compárala con la versión que tú has hecho.

fabula2

Por entre unas matas,
seguido de perros,
no diré corría,
volaba un conejo

De su madriguera
salió un compañero,
y le dijo:-Tente,
amigo,¿Qué es esto?

¿Qué ha de ser?-responde-.
Sin aliento llego;
dos picaros galgos
me vienen siguiendo.

-Si-replica el otro-,
por allá los veo,
pero no son galgos,
-¿pues qué son?-podencos.

¿qué,podencos dices?
-sí, como mi abuelo.
Galgos y muy galgos;
bien visto lo tengo.

-Son podencos ,vaya,
que no entiendes de eso.
-Son galgos,te digo
-digo que podencos

En esta disputa
llegando los perros,
pillan descuidados
a mis dos conejos .

Los que por cuestiones
de poco momento
dejan lo que importa
llévense este ejemplo

(Iriarte)

Otro poema de mayor dificultad podría ser el siguiente en sus dos versiones:

DESORDENADO:

CARRETA1

Las Carretas (Juan Ramón Jiménez)

¡Cómo lloran las carretas,
camino de Pueblo Nuevo !

Son las carretas que pasan
estas tardes, al sol puesto,
las carretas que se llevan
del monte los troncos muertos.

Los bueyes vienen soñando,
a la luz de los luceros,
en el establo caliente
que sabe a madre y a heno.

Ya están ahí las carretas...
Lo han dicho el pinar y el viento,
lo ha dicho la luna de oro,
lo han dicho el humo y el eco...

¡Cómo lloran las carretas ,
camino de Pueblo Nuevo!

Y detrás de las carretas,
caminan los carreteros,
con la aijada sobre el hombro
y los ojos en el cielo.

Y cae el ángelus desde
la torre del pueblo viejo,
sobre los campos talados,
que huelen a cementerio.

En la paz del campo,van
dejando los troncos muertos
un olor fresco y honrado
a corazón descubierto.

¡Cómo lloran las carretas
camino de Pueblo Nuevo!

ORDENADO:

CARRETA2

Las Carretas (Juan Ramón Jiménez)

Ya están ahí las carretas...
Lo han dicho el pinar y el viento,
lo ha dicho la luna de oro,
lo han dicho el humo y el eco...

NUCLEO: PTEX
UTI: 0035
PAG.: 8

Son las carretas que pasan
estas tardes, al sol puesto,
las carretas que se llevan
del monte los troncos muertos.

¡Cómo lloran las carretas,
camino de Pueblo Nuevo !

Los bueyes vienen soñando,
a la luz de los luceros,
en el establo caliente
que sabe a madre y a heno.

Y detrás de las carretas,
caminan los carreteros,
con la aijada sobre el hombro
y los ojos en el cielo.

¡Cómo lloran las carretas ,
camino de Pueblo Nuevo!

En la paz del campo,van
dejando los troncos muertos
un olor fresco y honrado
a corazón descubierto.

Y cae el ángelus desde
la torre del pueblo viejo,
sobre los campos talados,
que huelen a cementerio.

¡Cómo lloran las carretas
camino de Pueblo Nuevo!

Aplicación tercera

Otra posibilidad es mezclar dos poemas diferentes bien del mismo tipo o de estilos muy distintos, en uno solo y pedir a los alumnos que separen los dos poemas.

Veamos un ejemplo con los sonetos de Góngora y Garcilaso.

1. El alumno carga en el ordenador el documento SONESOS1 que es el siguiente:

SONETOS1

El siguiente documento es una fusión de dos sonetos.Empleando el procedimiento de mover bloques separa los dos cuartetos y dos tercetos de cada uno de ellos.

Mientras por competir por tu cabello,
oro bruñido al sol relumbra en vano ,
mientras con menosprecio en medio el llano
mira tu blanca frente el lilio bello;

y en tanto que el cabello,que en la vena
del oro se escogió, con vuelo presto
por el hermoso cuello blanco,enhiesto,
el viento mueve,esparce y desordena;

mientras a cada labio por cogello,
siguen más ojos que al clavel temprano,
y mientras triunfa con desdén lozano
del luciente cristal tu gentil cuello;

En tanto que de rosa y azucena
se muestra la color en vuestro gesto
y que vuestro mirar ardiente,honesto,
con clara luz la tempestad serena;

Marchitará la rosa el viento helado,
todo lo mudará la edad ligera
por no hacer mudanza en su costumbre

coged de vuestra alegre primavera
el dulce fruto antes que el tiempo airado
cubra de nieve la airosa cumbre.

goza cuello,cabello,labio y frente,
antes que lo que fué en tu edad dorada
oro,lilio,clavel,cristal luciente,

no solo en plata o viola troncada
se vuelva,mas tú y ello juntamente
en tierra,en humo,en polvo,ensombra,en nada.

2. Moviendo bloques el alumno separa los dos poemas y comenta sus decisiones con su compañero o compañeros.

3. Finalmente saca las versiones originales y se compara con el trabajo realizado por sus alumnos.

SONETOS2

SONETO de GONGORA

Mientras por competir por tu cabello,
oro bruñido al sol relumbra en vano ,
mientras con menosprecio en medio el llano
mira tu blanca frente el lilio bello;

mientras a cada labio por cogello,
siguen más ojos que al clavel temprano,
y mientras triunfa con desdén lozano
del luciente cristal tu gentil cuello;

goza cuello,cabello,labio y frente,
antes que lo que fué en tu edad dorada
oro,lilio,clavel,cristal luciente,

no solo en plata o viola troncada
se vuelva,mas tú y ello juntamente
en tierra,en humo,en polvo,ensombra,en nada.

SONETO de GARCILASO

En tanto que de rosa y azucena
se muestra la color en vuestro gesto
y que vuestro mirar ardiente,honesto,
con clara luz la tempestad serena;

y en tanto que el cabello,que en la vena
del oro se escogió, con vuelo presto
por el hermoso cuello blanco,enhiesto,
el viento mueve,esparce y desordena;

coged de vuestra alegre primavera
el dulce fruto antes que el tiempo airado
cubra de nieve la airosa cumbre.

Marchitará la rosa el viento helado,
todo lo mudará la edad ligera
por no hacer mudanza en su costumbre

PRACTICAS PROCESADOR DE TEXTOS 1

1. Formateo del disco de trabajo.

   a) Con el ordenador apagado introduce el disco DOS que tenga la opción de formatear en la unidad A. Cierra la unidad con el pestillo y enciende el ordenador y el monitor.

   b) Pulsa la tecla INTRO cuando te pide la fecha y la hora

   c) Introduce el disco de trabajo en la unidad B y cierra pestillo

   d) Escribe FORMAT B: Cuando te pregunte siquieres formatear otro disco escribe N y pulsa INTRO

   e) Saca los dos discos de las unidades

2. Puesta en marcha del Procesador de Textos

   a: Cuando aparezca el indicador A en pantalla, introduce el disco del Procesador de Textos en la unidad A y cargalo en el ordenador.

   b) Introduce tu disco de trabajo ya formateado en la unidad B y cierra el pestillo.

3. Creación de un texto

   a) Selecciona del Menu principal del Procesador de Textos la opcion de editar.

   b) Escribe un texto para familiarizarte con el teclado y las distintas teclas de función.

4. Guardar el Texto

   Antes de finalizar la sesión hay que guardar el texto en disco siguiendo las instrucciones del manual del procesador de textos que estás empleando.

PRACTICAS PROCESADORES DE TEXTOS 2

1. PUESTA EN MARCHA DEL PROCESADOR DE TEXTOS

a) Con el ordenador apagado introduce el disco DOS en la unidad A, cierra la unidad con el pestillo y enciende el ordenador y el monitor.

b) Pulsa la tecla INTRO o RETURN cuando te pide la fecha y la hora

c) Cuando aparezca el indicador A en la pantalla saca el disco DOS de la unidad A e introduce en ella el disco del procesador de textos y cárgalo en el ordenador.

d) Introduce en la unidad B tu disco de trabajo.

2. CREACION DE UN TEXTO

a) Selecciona la opción editar del menú principal.

b) Introduce en el ordenador el siguiente texto , tal y como aparece en este apartado. Después realizarás las modificaciones que se te indican en los puntos siguientes para conocer todas las posibilidades del procesador de textos.No corrijas en esta primera fase los errores conforme te van surgiendo.

"De tanto aquilatar las posibilidades aparentemente infinitas del ordenador podría uno llegar a creer que es capaz de llevar a cabo cualquier cosa. ¿Hasta dónde llegará? Hemos visto ya que no estábamos seguros de saberlo. Esta incertidumbre, tanto si suscita entusiasmo como angustia, induce rápidamente a sobreestimar esa extarordinaria máquina. Sin embargo, una reflexión detenida permitirá determinar lo que, probablemente, jamás podrá realizar."

3. CORRECCCION DE ERRORES

Practica el movimiento del cursor a lo largo del texto

para localizar las palabras que necesitan corrección.

Emplea la tecla DEL para borrar el carácter bajo el que se encuentra el cursor (observa que al borrar se elimina el hueco del carácter). Emplea la tecla ← para borrar el carácter inmediatamente anterior al del lugar del cursor.

También puedes corregir errores simplemente escribiendo encima de la palabra.

Pulsa las teclas HOME y END y observa como el cursor va al principio y al final de la línea respectivamente

Pulsa las teclas CTRL y HOME simultáneamente ¿Dónde va el cursor? ¿ y si pulsas CTRL y END?

5. MODIFICAR UN TEXTO. Realiza lo siguiente:

-Subraya la pregunta ¿Hasta donde llegará? y la última frase.

-Intensifica la frase : "**podria uno llegar a creer que es capaz de llevar a cabo cualquier cosa**" y las palabras "**entusiasmo**" y "**angustia**"

-Inserta el siguiente texto detrás de la palabra"saberlo"

"Tres tipos de razones que resulta importante no confundir, frenan el desarrollo de sus capacidades."

-Borra el bloque:

"Hemos visto ya que no estábamos seguros de saberlo"

6. GUARDAR EL TEXTO.

Sigue las instrucciones de la práctica 1.

PRACTICAS PROCESADOR DE TEXTOS 3

1. Pon en marcha el ordenador y carga el Procesador de Textos siguiendo las instrucciones del punto 1 de la práctica 2.

2. Coloca tu disco de trabajo en la unidad B .

3. Carga en tu ordenador el texto que tienes guardado en el disco de trabajo con el nombre LIBER1.

   a) Los cuatro párrafos de ese texto están desordenados. Siguiendo las instrucciones del procesador de textos para mover bloques mueve los 4 bloques (párrafos) de ese texto para colocarlos en el orden que crees los escribió el autor.

   b) Separa con un punto y aparte " En llegando bajo la sombra..."

      Del mismo modo separa con punto y aparte: "Había por allí un bebedero umbrio..."

   c) Inserta la frase: " animado por el espectáculo que allí se divisaba" entre las palabras "frondosa" y "batí palmas".

   d) Borra de una vez la linea: " La mañana era clara , pura,traspasada de azul..."

   e) Borra las siguientes palabras de una vez, es decir, no vayas letra a letra. Sigue para ello las instrucciones de tu procesador de texto y emplea la tecla de función oportuna. "peluda" "polícromo" "traidores" y " marero"

4. Guardar el texto y Finalizar la sesión.

   Sigue las instrucciones de la practica anterior y guarda el texto con el nombre LIBER2

PRACTICAS PROCESADOR DE TEXTOS 4

1. Pon en marcha el ordenador y carga el procesador de textos siguiendo las instrucciones del punto 1 de la práctica 2.

2. Coloca tu disco de trabajo en la unidad B.

3. Selecciona del menú principal la opción cargar y carga de tu disco el texto guardado como IDILIO2

4. BUSQUEDA Y SUSTITUCION

   a) Sustituye de modo automático la palabra "PLATERO" por "COCOLIN" cada vez que aparece en el texto.

   b) Sustituye la palabra "FLORES" por "FLORECILLAS" de modo manual.

5. DEFINIR PAGINA

   Vuelve al menú principal y cambia el formato de la página.

   Modifica los valores de modo que el texto tenga solo 30 caracteres por linea.

6. IMPRIMIR

   Volviendo al menú principal selecciona la opción de imprimir.

   Conecta la impresora al ordenador y coloca en ella el papel.

   Imprime el texto con los valores por omisión. Después modificalos y vuelve a hacer distintos formatos de impresión.

PRACTICAS PROCESADORES DE TEXTOS 5

APLICACION DIDACTICA: REDACCION

1. Pon en marcha el ordenador y carga el Procesador de textos siguiendo las instrucciones de prácticas anteriores.

2. Vais a escribir en el ordenador todas las ideas que se os ocurran sobre un tema que acordeis entre todos.
   Teneis que hacerlo en modo esquemático, sin pensar demasiado en la redacción o en el orden y anotar todo lo que se os ocurra en torno a ese tema . Emplead para ello diez minutos.

3. Organizad las ideas que se os han ocurrido de modo lógico. Fusionad las ideas que sean semejantes. Discutid las que no esteis de acuerdo. Quitad las que en ese momento no os parezcan oportunas. Añadid las que se os ocurran ahora . Emplead para todo este tipo de acciones las teclas de función del procesador de textos.

4. Guardad en el disco lo que habeis escrito hasta el momento con el nombre IDEAS

5. Discutid de nuevo sobre las ideas que teneis para agruparlas en tres apartados:

   INTRODUCCION
   DESARROLLO
   CONCLUSION

   Moved debajo de estos epigrafes las ideas que teneis ya escritas y que considerais deberian introducir el tema, las que corresponderían al desarrollo del mismo, y las que serian una conclusión.

   NO REESCRIBAIS las ideas hacedlo empleando las teclas de función que os permiten mover bloques y líneas.

   Si alguno de los tres apartados os ha quedado incompleto incluid ahora las ideas que se os ocurran.

6. Guardad lo que acabais de hacer en el disco de trabajo con el nombre GUION

7. Empezad ahora a desarrollar cada una de las ideas dándoles forma y redactándola de la manera más literaria posible. De modo que el guión se convierta en una redacción.

   Cuando termineis o se termine el tiempo de clase guardad lo que habeis escrito hasta el momento con el nombre REDAC.1

MINISTERIO DE EDUCACION Y CIENCIA - BIBLIOTECA -

PROCESADORES DE TEXTOS

- COLE, Bernard C. (1985) BeyondWord Processing. New York, McGranw-Hill.
- GEOFFRION , Leo D. and olga P. GOFFRION (1983) Computers and Reading Instruction . Reading, MA, Addison-Wesley.
- GREENE, W.T.,L.V. SADLER and E.W. SADLER (1985) "Diagrammatic Writing -- Using Word Processing: Computer-Assisted Composition for the Development of Writing Skills".The Computing teacher , Vol. 12, nº 7, pp. 62-64
- HUNTER, Beverly (1985) Mis alumnos usan ordenador. Barcelona, Martínez Roca.
- JONASSEN, David H. (1982) The Technology of Text. Englewoood Cliffs, NJ, Educational Technology Publicationes.
- MONTGOMERY, Jim (Jim (1984) " Cloze Procedure: A Computer Application". The Computing Teacher, Vol. 11, nº 9, pp. 17-18/20.
- NOBLE, D.F. and V. NOBLE (1984) Improve Your Writing With Word Processing. Indianapolis, Indiana, Que Corporation.
- O'BRIEN, Peggy (1984) "Using Microcomputers in the Writing Class".The Computing Teacher, Vol. 11, nº 9, pp. 20-21.
- PIAZZA, C.L. & J.C. DAWSON (1984) "Cloosing a Word Processor for Writing - Instruction". Computers Reading and Language Arts, Vol. II, nº 1, pp. 10-12
- POLLITT, Alice Hunt (1984) "Warning to the Word Processor". The Computing - Teacher, Vol. 11, nº 9, pp. 48-49
- RODRIGUES, Dawn (1985) "Using Computers in the Language Arts Curriculum" . Curriculum Review, Vol. 24, nº 5, pp. 25-28
- WLLER, Art (1984) "Creative Writing With Computers : What Do Elementary Students Have to Say ?". Computers, Reading and Language Arts, Vol. II, nº 1, pp. 39-42

INSTITUTO DE TECNICAS EDUCATIVAS
BASE DE DATOS

HACIA UNA SOCIEDAD DE LA INFORMACION

Realizando un breve repaso a las distintas sociedades que nos han precedido y analizando la situación actual de los paises más avanzados, podremos predecir, en parte, el tipo de sociedad que van a vivir nuestros alumnos.

La sociedades primitivas y la sociedad feudal son esencialmente sociedades agricolas y ganaderas, desde nuestra perspectiva, sólo hay un sector productivo, el primario. El único recurso utilizado es la materia que produce la naturaleza.

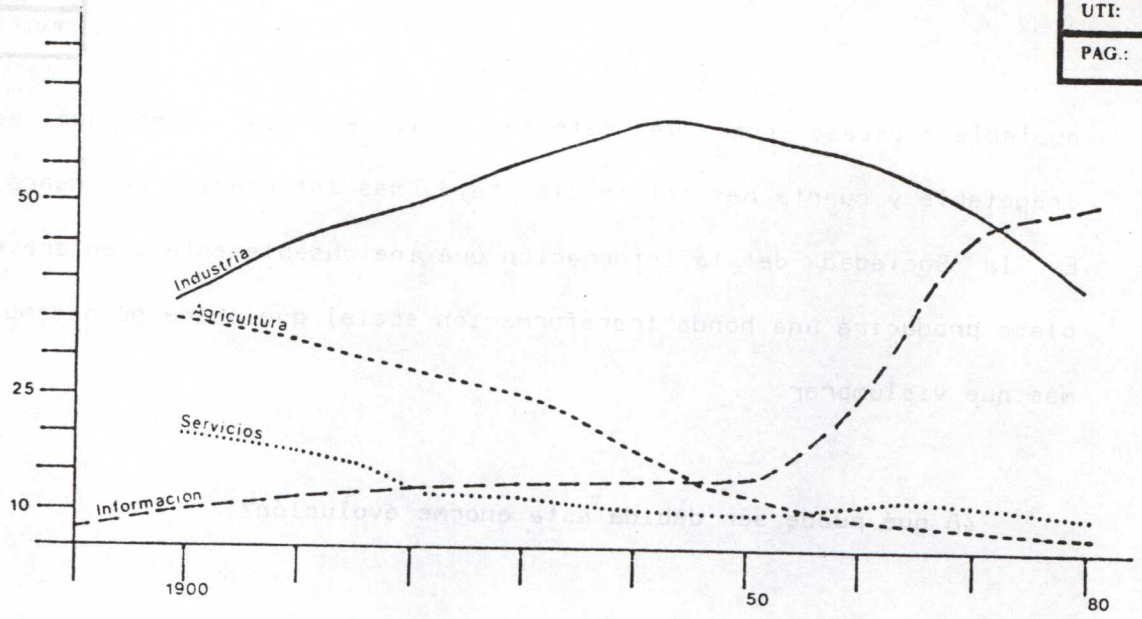
La etapa de los descubrimientos supondrá un crecimiento importante del comercio, iniciado en Europa siglos antes, comenzando a despuntar los otros sectores productivos. Los viajes, que es necesario realizar para los intercambios, constituyen un acicate para el progreso y desarrollo científico-técnico, favoreciendo la formación de una nueva sociedad, que sin embargo, produce aún pocas transformaciones en la materia.

Es en la primera revolución industrial, con el mejor aprovechamiento de la energia producida por los elementos naturales y como consecuencia la incorporación de las máquinas a la producción, cuando se efectúa una transformación social más

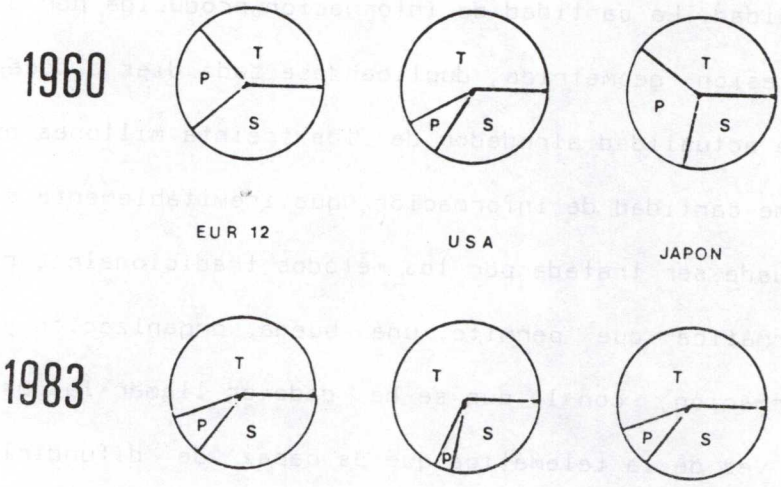
profunda, acrecentada en la segunda revolución industrial con el aprovechamiento de nuevas energias. La sociedad rural y agricola-ganadera se transforma en una sociedad industrial y urbana. Esta enorme transformación que se produce en menos de un siglo, no puede entenderse sin la mecanización de la imprenta, a través de la cual se consiguen difundir los conocimientos e investigaciones cientificas y técnicas en poco tiempo y en una gran cantidad de paises.

Hoy en dia se está produciendo una nueva transformación que se refleja en las actividades de la sociedad de los paises más desarrollados, como es la continuación del descenso del sector agricola-ganadero que comenzó con la revolución industrial, la disminución del sector industrial, sector más numeroso hace un par de décadas y, ésto es lo esencial, el predominio del sector servicios, destacando fundamentalmente dentro de este último, el número de personas dedicadas al tratamiento de la información , es decir, aquellas personas cuyo trabajo principal consiste, no en la transformación de la materia, como en el resto de la producción, sino en la transformación de símbolos.

En el siguiente gráfico, debido a M. Porat, se muestra la evolución de los distintos sectores en EE. UU. entre los años 1900 y 1980, donde se ha separado del sector servicios, el sector dedicado a la Información.



A continuación se muestra la relación entre los sectores en EE. UU. ,la CEE. y Japón en los años 1960 y 1983 según la revista Comunidad Europea de diciembre de 1985.



Nótese, sin embargo, que la producción industrial y agrícola de estos paises no ha disminuido, muy al contrario ha aumentado considerablemente, lo que indica la importancia que tiene la información para la mejor utilización de los recursos naturales, tanto materiales como energéticos. Nos dirigimos por tanto, hacia una nueva sociedad post-industrial en la que el nuevo "bien" no es

agotable o escaso como la materia o la energia, sino que es inagotable y cuanta más información haya, más información se creará. Es la Sociedad de la Información que inexcusablemente y en breve plazo producirá una honda transformación social que ahora no podemos más que vislumbrar.

¿A qué puede ser debida esta enorme evolución?

Entre otras razones, ocupa un lugar predominante el progreso científico y técnico que está teniendo lugar gracias a la difusión y acceso a la información que los mismos científicos y técnicos producen. En la actualidad viven el 90% de los científicos y pensadores que han existido a lo largo de la historia de la Humanidad. La cantidad de información producida por ellos crece en progresión geométrica, duplicándose cada diez o doce años, habiendo en la actualidad alrededor de los treinta millones de títulos. Esta enorme cantidad de información, que inevitablemente sigue creciendo, no puede ser tratada por los métodos tradicionales, necesita de la informática que permite una buena organización y acceso a dicha información, con lo que se ha dado en llamar las bases de datos, y a su vez de la telemática que es capaz de difundirla a cualquier lugar del planeta.

Debido a esta situación se ha creado la llamada industria de la información atendiendo a distintos aspectos de la recopilación y transmisión de los datos, siendo cada vez más numerosas las empresas dedicadas a esta actividad. En España, por ejemplo, la Red de Información Científica Automatizada (INCA) permite acceder a más de doscientas bases de datos del mundo, lo que supone más de cincuenta

millones de referencias sobre información científica y técnica.

En la actualidad, siguen incrementándose el número de entidades, que con carácter más o menos específico, crean sus propias bases de datos con difusión universal. También se está generalizando en todo tipo de empresas e instituciones que necesitan manejar grandes cantidades de información, como documentación, personal, almacén, contabilidad, producción, estadísticas, etc., la utilización de bases de datos específicas que permiten realizar una gestión más fiable y amplia que la que realizaban sin utilizar medios informáticos.

Estas facilidades de que disponen las bases de datos, pueden usarse en particular en los Centros Educativos , en los que no se utilizaban grades bloques de datos por la imposibilidad de un tratamiento adecuado.Es pues el momento de iniciar a nuestros alumnos en el tratamiento de la información, utilizando con imaginación los programas de gestión de bases de datos que en la actualidad están disponibles, de forma que, a la vez que aprenden los temas relativos a una o varias materias específicas, se familiaricen con las técnicas de organización y acceso a la información que inevitablemente deberán utilizar, tanto para su constante formación, como para el ejercicio de su futura profesión.

TERMINOLOGIA BASICA Y ESTRUCTURAS DE LAS BASES DE DATOS

No se pretende hacer un estudio exhaustivo de las distintas estructuras de la información, sino más bien, indicar de forma general las principalemente utilizadas por las distintas bases de datos, así como la terminología básica empleada en este campo.

El término "BASE DE DATOS" aparece por primera vez al comienzo de los años 60, para expresar un conjunto de datos relacionados entre sí, que están estructurados de forma que puede accederse a ellos automáticamente e independientes de los programas que gestionan dichos datos. Esta independencia se refiere a la posibilidad de modificar su estructura sin necesidad de modificar los programas que manipulan los datos.

Toda base de datos está formada por uno o varios FICHEROS, siendo éstos conjuntos conexos de información; por ejemplo, en una base de datos de una biblioteca, un fichero puede estar constituido por la información relativa a los libros, otro fichero, contendrá información sobre los lectores, etc.

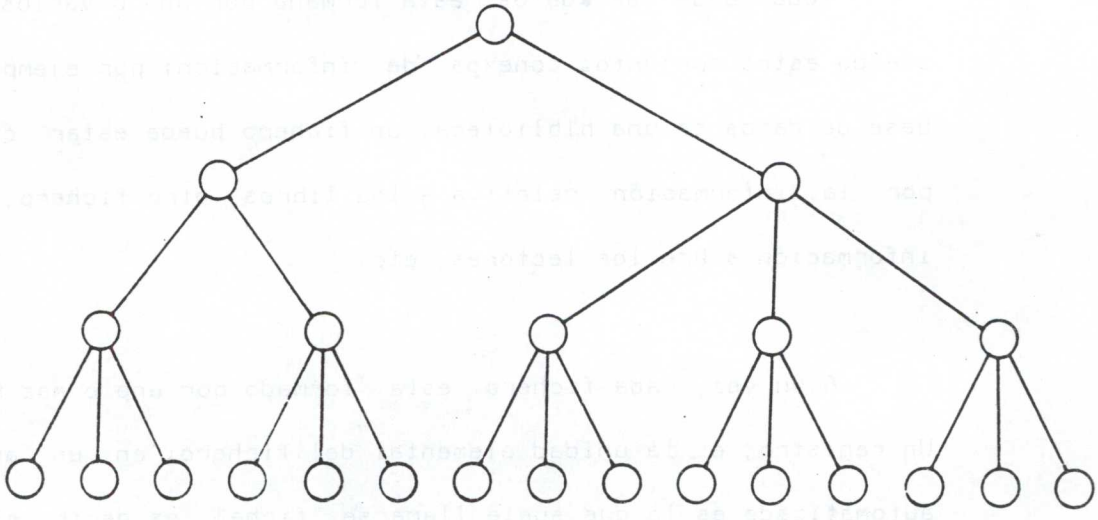
A su vez, cada fichero está formado por uno o más REGISTROS. Un registro, es la unidad elemental del fichero; en un archivo no automatizado es lo que suele llamarse "ficha", es decir, el conjunto de informaciones relativas a uno de los elementos. En el fichero de libros, un registro estaría constituido por la información correspondiente a un libro concreto con su título, autor, área, editorial, etc.

También el registro está formado por uno o más elementos llamados CAMPOS. Un campo es cada una de las informaciones que interesa almacenar en un registro. En el ejemplo anterior, un campo sería el título del libro, otro su autor, etc.

La forma de almacenar los ficheros, registros y campos es distinta dependiendo de las características de la información que se quiere tratar. Basicamente podemos distinguir tres tipos de estructura: Jerárquica, en Red y Relacional.

ORGANIZACION JERARQUICA

La organización Jerárquica que es la que cronológicamente es empleada en primer lugar, tiene una estructura arborescente, donde los nudos del mismo nivel corresponden a los campos y cada rama a un registro.

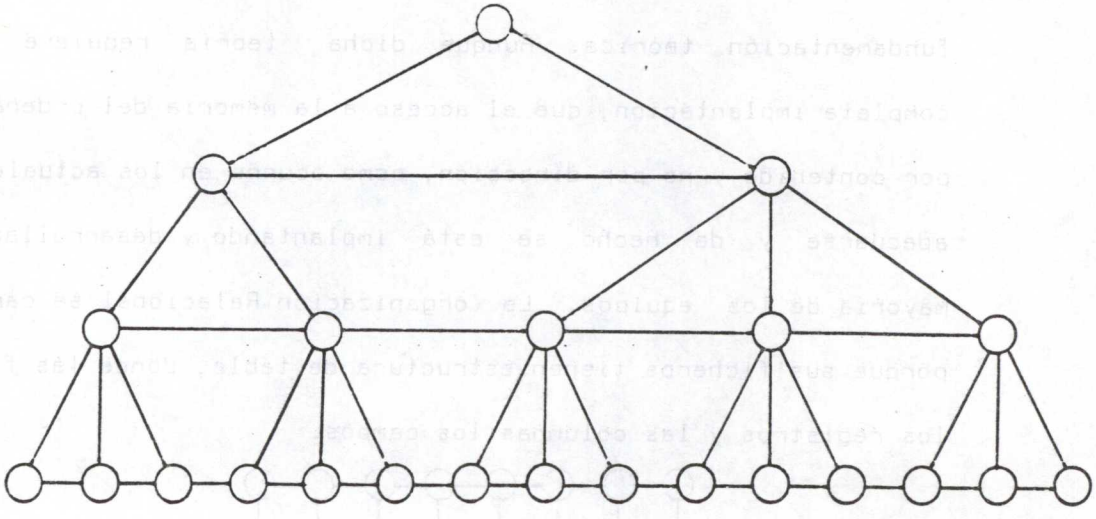


Esta forma de organización puede hacer lenta la obtención de determinadas informaciones, ya que para acceder a un nudo (campo)

hay que recorrer toda la rama, partiendo de la raiz,es decir, todos los campos precedentes en el registro. No obstante, existen estructuras arborescentes más sofisticadas que incluyen índices y que permiten acelerar el resultado de las consultas.

ORGANIZACION EN RED

Para resolver el problema de lentitud de la organización Jerárquica se utiliza la organización en Red que corresponde a una estructura de grafo, donde existen mas conexiones entre los nudos de forma que puedan recorrerse por distintos caminos sin necesidad de acudir cada vez a la raiz. Por ejemplo, si se crean conexiones entre nudos de igual nivel, el listado de titulos se obtendrá más rapidamente.



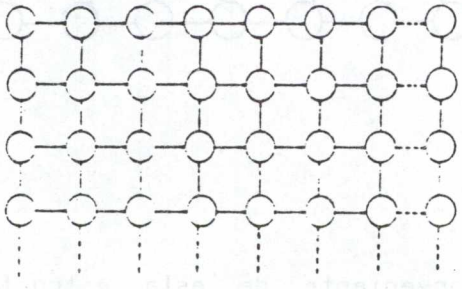
El inconveniente de esta estructura es la necesidad de

utilizar mucha más cantidad de memoria, al tener que almacenar en cada nudo las posiciónes de los campos siguientes.

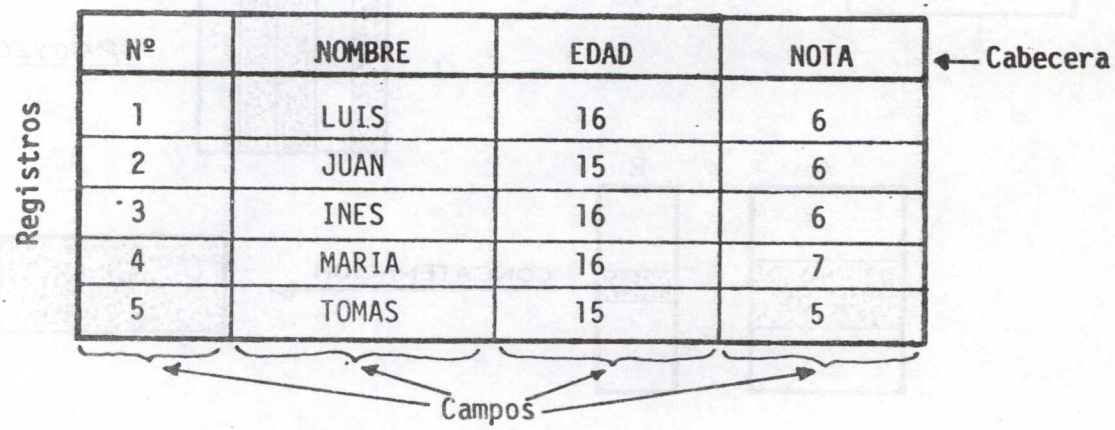
ORGANIZACION RELACIONAL

No obstante, el problema fundamental que suele plantearse al realizar una base de datos real, formada por varios ficheros, es la repetición de datos, lo cual va a dificultar su gestión, es decir la actualización, modificación, inserción, eliminación, consulta, etc. Para resolver estos problemas surgen los llamados Sistemas de Gestión de Bases de Datos (SGBD) y con ellos las llamadas Bases de Datos Relacionales que pueden resolver, mejor que otras organizaciones, las dificultades planteadas.

La teoría Relacional se basa en el concepto matemático de relación. Se debe a E. F. Codd, quien ha desarrollado una sólida fundamentación teórica. Aunque dicha teoría requiere para su completa implantación, que el acceso a la memoria del ordenador sea por contenido y no por dirección, como ocurre en los actuales, puede adecuarse y de hecho se está implantando y desarrollando en la mayoría de los equipos. La organización Relacional se caracteriza porque sus ficheros tienen estructura de tabla, donde las filas son los registros y las columnas los campos.



Las tablas son tratadas como conjuntos matemáticos, obtenidas como subconjuntos del producto cartesiano de los dominios en los que pueden tomar valores los distintos campos que la forman. Cada tabla dispone de una cabecera que es un registro especial donde figuran los identificadores de los campos (Atributos).



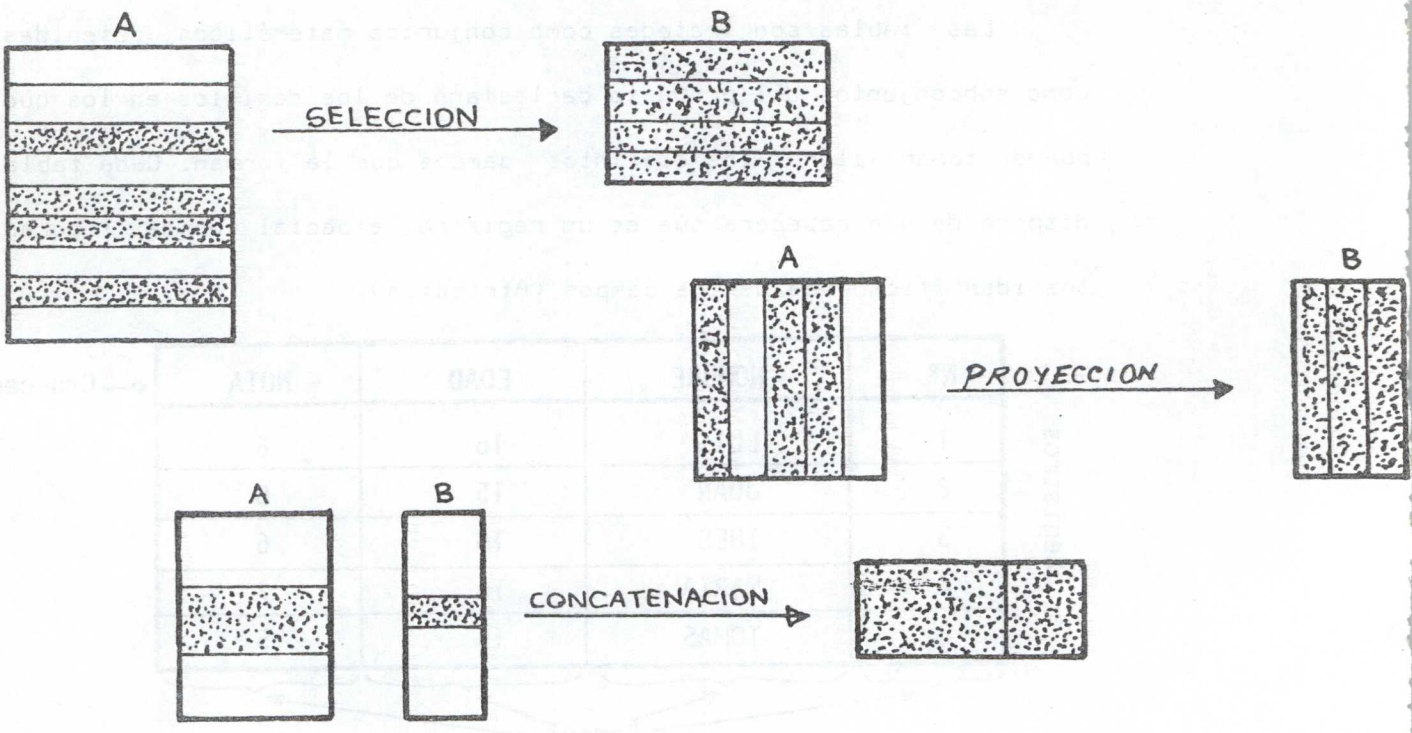
OPERACIONES RELACIONALES

Una base de datos relacional constará de varias tablas con las que se pueden efectuar tres operaciones fundamentales llamadas operaciones relacionales:

-SELECCION.- Consiste en la obtencion de una nueva tabla (fichero) formada por algunas de las filas (registros) de otra tabla ya existente.

-PROYECCION.- Da lugar auna nueva tabla formada por algunas columnas (campos) de otra tabla ya existente. Normalmentepueden efectuarse simultaneamente la selección y la proyección.

-CONCATENACION.- Une dos tablas para formar una nueva. Por lo general la unión de registros se efectúa si en ambas tablas coincide el contenodo de un campo prefijado.



CLAVES

Con esta organización es absolutamente necesaria la creación de indices que permitan acceder de forma rápida a los datos. El acceso a un dato concreto se realiza con tres valores: el nombre del fichero (tabla), el identificador del campo o atributo (columna) y una clave que determina el registro (fila). La clave es el código que permite identificar a cada registro de la tabla, teniendo distinto valor para cada uno de ellos. La clave estará formada por uno o mas atributos. Por ejemplo, en una base de datos de un Instituto de Bachillerato, una clave para el fichero de profesores sería su número de registro personal. En el ejemplo de la biblioteca, una clave para el fichero de libros podría ser el código universal asignado a cada libro, también sería clave el titulo, la edición y el número de ejemplar, siempre que no haya titulos iguales de distintos autores, en cuyo caso habria que incluir en la clave un cuarto atributo, el autor.

El desarrollo del hardware y software está permitiendo la creación de programas de aplicacción, para microordenadores, cada vez más completos, de los que se puede obtener un buen aprovechamiento en la educación. Actualmente están proliferando programas de usuario cada vez más fáciles de utilizar, que no requieren conocimientos específicos de programación y que, sin embargo, pueden aplicarse en una multitud de áreas. En este sentido han aparecido los llamados paquetes de programas que incluyen generalmente un gestor de base de datos, un procesador de textos, una hoja de cálculo y un programa de gráficos. De forma que con grán facilidad puede transladarse información de un módulo a otro, permitiendo una gestion muy ágil para la obtención de los resultados deseados. La mayoría de estos programas son de propósito general y utilizables,por tanto, en aplicaciones muy diversas.

BASES DE DATOS DOCUMENTALES

Merecen, no obstante, una mención especial los programas que gestionan bases de datos documentales. Estos programas permiten un tratamiento de la información muy sofisticado y pueden ser extremadamente útiles en determinadas áreas de la educación, donde los números y el cálculo no son elementos fundamentales. Las bases de datos documentales están muy desarrolladas debido a la necesidad de gestionar la información bibliográfica y textual, tanto por los medios de comunicación, como por las grandes bibliotecas, archivos y centros de documentación. Así, por ejemplo, todos los grandes diarios del mundo disponen de una base de datos documental en la que equipos de expertos introducen abstracts de las informaciones

diarias de los principales periódicos. El tiempo de respuesta a una consulta suele ser de unos pocos segundos.

Estas bases de datos se caracterizan porque sus registros, de longitud fija o variable según los casos, son textuales y aunque pueden contener campos fijos, la mayoria de los campos no lo son, de forma que las claves son todas las palabras que forman el texto o bien algunas palabras previamente señaladas (KWIC: key word in context), pero también pueden asignársele claves fuera del texto (KWOC: key word out of context). Por ejemplo, en una base de datos que incluyera una poesia de Antonio Machado, además de poder considerar clave todas o casi todas las palabras, podrian ser claves : Antonio Machado, poesia, el año en que fue compuesta, su generación, el tipo de verso, la rima, etc.

Las claves constituyen un diccionario, que permite la selección de registros a traves de las claves mediante restricciones que suelen ser los operadores lógicos ("Y", "O" y "NO"), obteniendo en muy breve tiempo el documento o los documentos que se soliciten. En algunos casos, al ser el diccionario un thesaurus, nos va guiando para encontrar el camino que permita localizar más rápidamente la información que se busca. En este sentido, se está investigando con bases de datos inteligentes, que permiten una búsqueda más rápida y eficaz.

Como puede verse las bases de datos documentales pueden ser muy prácticas en investigación y educación, tanto por su utilización para obtener conclusiones, como por el aprendizaje que supone su creación y constante actualización.

INTRODUCCION

Las bases de datos son potentes sistemas de tratamiento de ficheros.

Las informaciones contenidas en una base de datos, estan estructuradas en ficheros, registros y campos.

-Campo Es la unidad elemental de información.A los campos utilizados para localizar registros de forma directa, seleccionar contenidos, ó efectuar clasificaciones, se les conoce con el nombre de campos clave.

-Registro Es un conjunto de campos de una determinada entidad.Podemos decir que es el equivalente a la tipica Ficha, de un archivo clasico.

-Fichero Es el conjunto de registros de un mismo tipo.

Un módulo de Gestión de Base de Datos, es un conjunto de programas que permiten la creación y realización de las diferentes operaciones previstas, en la base de datos.

La Base de Datos del Open Access, es una base de datos relacional, ya que permite relacionar ficheros diferentes a traves de campos comunes.

Toma el nombre de relacional, del concepto matemático de relación o tabla bidimensional.Los registros se representan por filas y los campos por columnas (figura 1).



figura 1

La Base de Datos, puede considerarse el nucleo básico del Open Access, sobre el que se apoyan el resto de los módulos.Se indican a continuación las limitaciones de la misma:

Nº máximo de registros por fichero.....32000

Nº máximo de campos por registro.....55

Nº máximo de caracteres por registro.....1024

Nº máximo de campos clave por fichero.....15

Nº máximo de paginas por registro.....8

Nº máximo de campos DEBE CASAR.....4

Nº máximo de ficheros relacionados.....5

Longitud máxima de un campo de TEXTO.....58

Nº máximo de digitos de un campo DECIMAL....18

Nº máximo de cifras decimales.....15

Rango de un campo NUMERICO.....-32000 a +32000

La documentación desarrollada, a continuación, sobre el Gestor de Base de Datos del Open Access, esta agrupada en 3 grandes apartados:

-COMANDOS. Se incluyen todos los comandos, con una breve explicación de los mismos.Estan ordenados alfabeticamente, para una facil localización de los mismos.

-GUIA DE OPERACION. Se explican los pasos a seguir para realizar las operaciones fundamentales.

-APENDICES. Se da un chuletario sobre 3 aspectos basicos para el manejo del Gestor de Base de Datos.

-COMPORTAMIENTO DEL TECLADO.

-LENGUAJE DE INTERROGACION.

-ATRIBUTOS DE LOS CAMPOS.

COMANDOS

ACTUALIZAR. Actualiza el valor de un campo en todos los registros seleccionados.

AÑADIR. Transfiere datos entre ficheros.

CARTAS. Genera cartas o etiquetas personalizadas, utilizando el contenido de la Base de Datos y una plantilla con caracteres de sustitución, creada con un procesador de textos o editor.

CLASIFICAR. Clasifica los registros de acuerdo con los contenidos de los campos clave seleccionados, en orden ascendente o descendente.

CONTEXTO. Sale del Gestor de Base de Datos, exportando el fichero virtual, al módulo del Open Access que se seleccione.

CREAR. Crea un fichero y su máscara.

DEPURAR. Permite borrar los registros activos.

DISEÑAR. Cambia o crea una máscara de pantalla.

EXPORTAR. Crea un fichero SIF, que graba en el disco, para poder utilizar desde otros módulos.

EXTENDER. Amplia el fichero, añadiendo más registros vacios.

FICHEROS. Borra o cambia el nombre de los ficheros.

FORMATO. Define la estructura y apariencia de los informes.

IMPORTAR. Trae del disco un fichero SIF, que ha podido ser creado en otros módulos.

INFORME. Genera un informe, según la estructura definida en el comando FORMATO.

INTERROGAR. Busca los registros que satisfacen las condiciones y automaticamente accede al comando listar para mostrar los registros.Funciona como una clausula CUYO.

INTRODUCIR. Introduce datos en el fichero.

LISTAR. Muestra el contenido de los registros en forma tabulada.Mediante F6 podemos pasar a visualizar los datos a traves de una máscara, permitiendo modificar el contenido de los registros.

MODIFICAR. Crea un nuevo fichero, aprovechando para ello uno existente.El antiguo queda sin modificación y el nuevo vacio.

MOSTRAR. Permite examinar un fichero, pudiendo cambiar sus contenidos (F6), borrar registros (F5), realizar busquedas por contenidos de los campos (F4) e introducir nuevos registros (F9).

NUEVO. Desactiva el fichero virtual.

OPCIONES. Sale del Gestor de Base de Datos y presenta la ventana de opciones del Open Access.Si han habido modificaciones desde la ultima grabación, permite salir grabando (F10), o sin grabar (Esc).

VISUALIZAR. Visualiza los registros con su máscara de pantalla.Permite modificar la información, del registro visualizado, directamente.

GUIA DE OPERACION

El objeto de esta guía es iniciar el uso de la Base de Datos, siguiendo el desarrollo de un sencillo ejemplo, que permita ir introduciendo los comandos basicos y familiarizarnos con el modo de proceder de este módulo del Open Access.

Vamos a desarrollar los siguientes apartados:

1.Acceso al Gestor de la Base de Datos.

2.Crear un fichero.

3.Introducir datos.

4.Ver la información almacenada.

5.Modificar datos almacenados.

6.Variar el diseño de la máscara de pantalla.

7.Variar la estructura del fichero.

8.Copiar datos entre ficheros.

9.Ampliar la capacidad de un fichero.

10.Borrar y cambiar el nombre a un fichero.

1.ACCESO AL GESTOR DE LA BASE DE DATOS

a)Con dos unidades de diskette.

Desde ambiente del sistema operativo (A>), introducir el disco 1 del Open Access en la unidad A y teclear OA y <RET>.Aparecerá una pantalla como la de la figura 1.

Open Access E1.Ø1 COMPUTEC
(C) 1984 Software Products International Inc.

Fecha de hoy: 10/ 2/86

figura 1

Actualizar ó confirmar la fecha y pulsar <RET>, la pantalla presentará el Menú de OPCIONES del Open Access (figura 2).

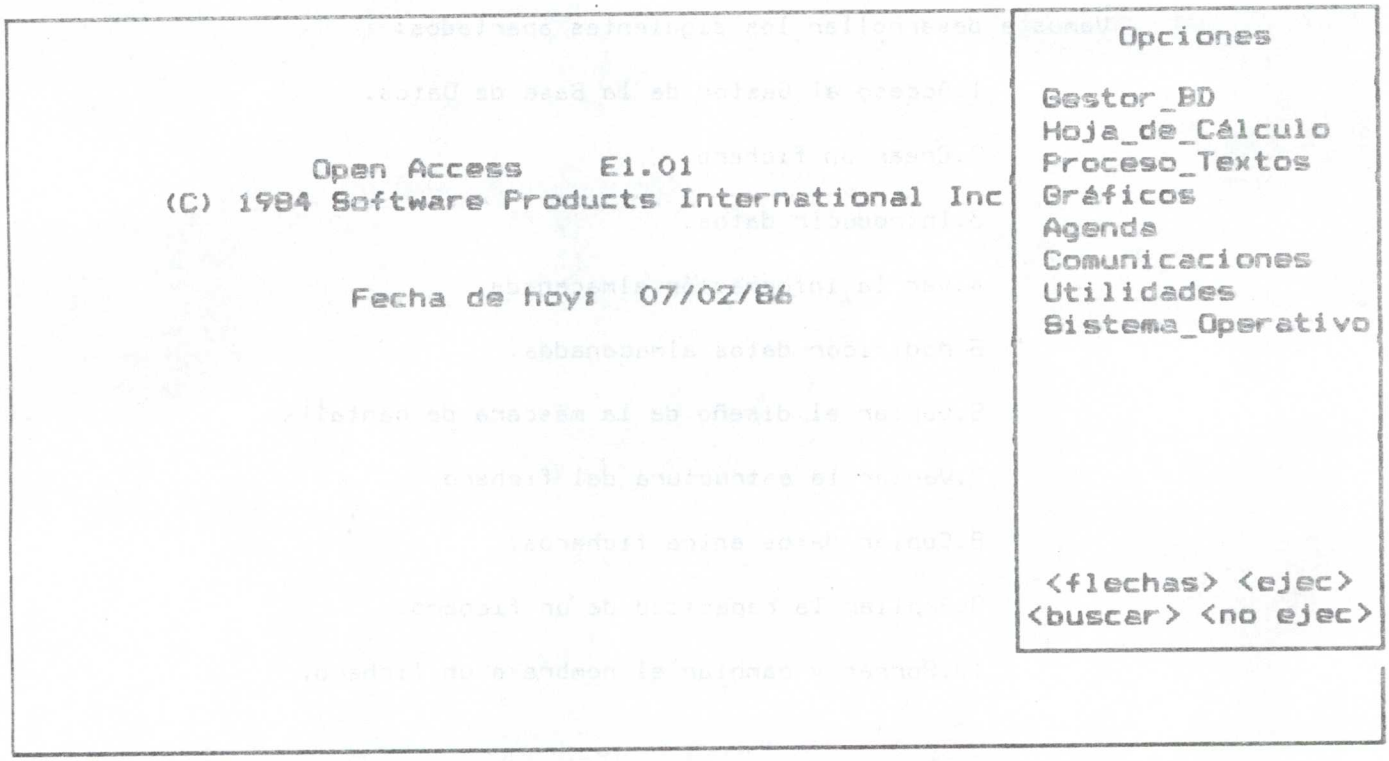


figura 2

Situar el cursor en Gestor_BD y pulsar <RET> (ó pulsar B).El Open Access pedirá el disco de Base de Datos (Codigo_3), que habrá que introducir en la unidad A, sustituyendo al de arranque, pulsando a continuación <ejec> F10.La pantalla presentará el **Menú de Operaciones de Base de Datos** (figura 3).

```
Open Access - Gestor de Base de Datos
(C) 1984 Software Products International Inc.

Menú de Operaciones de Base de Datos
- Ningún fichero activo -
Listar  Visualizar  Introducir  Actualizar  Clasificar  Mostrar
Añadir  Informe  Cartas  Depurar  Nuevo  Interrogar
<ejec>  <no ejec>  otro menú: <cambiar>
```

figura 3

Introducir el diskette de usuario en la unidad B, y ya nos encontramos en condiciones de trabajar con la Base de Datos .

Desde el **Menú de Operaciones de Base de Datos**, mediante la tecla **<cambiar> F6** se pasa al **Menú de Mantenimiento de ficheros de Base de Datos**(figura 4).

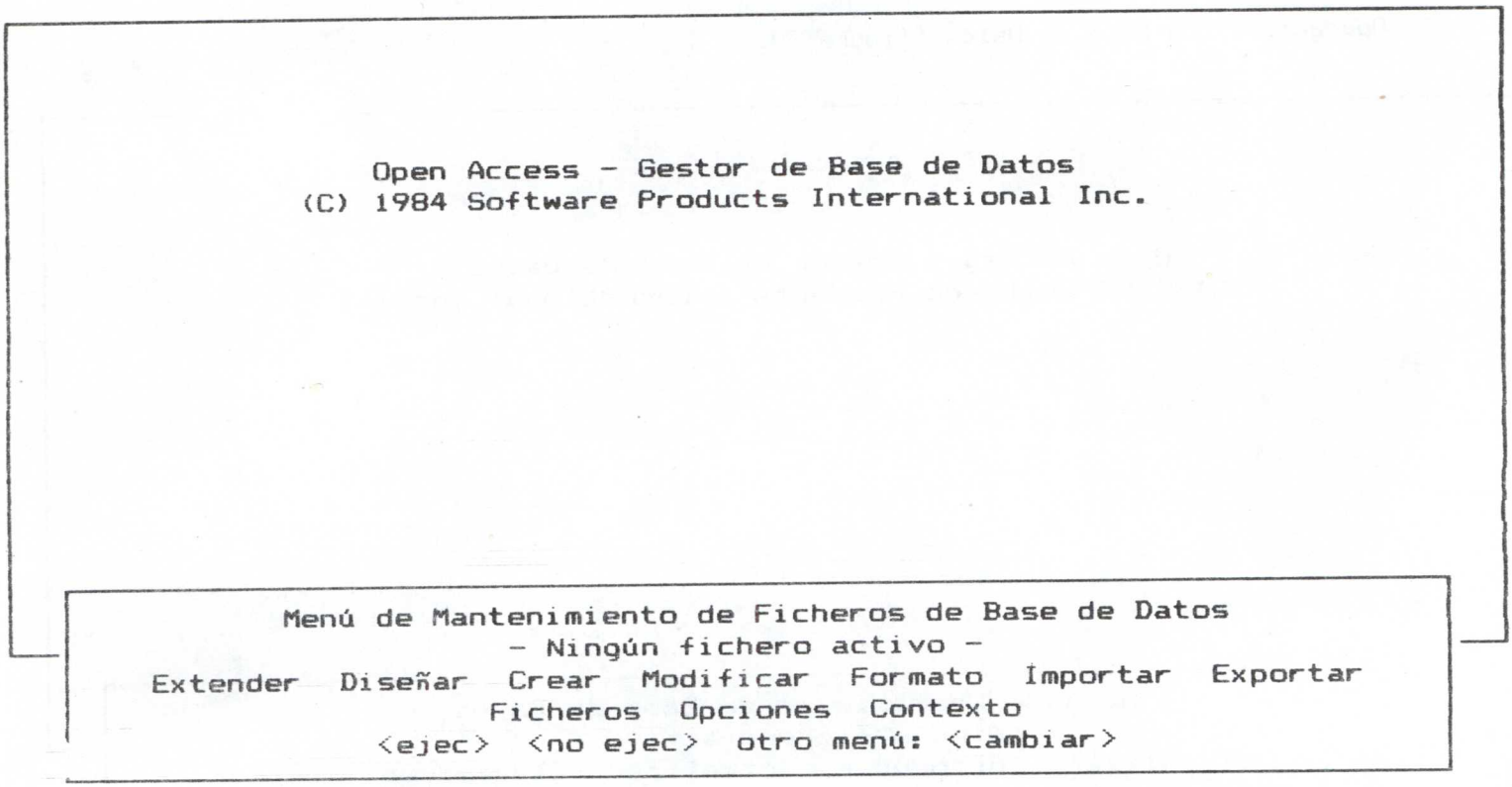


figura 4

Entre ambos menus se encuentran todos los comandos del Gestor de Base de Datos.La selección de cada uno de ellos, puede realizarse situando el cursor en el propio comando y pulsando **<ejec> F10** ó bien pulsando la inicial del comando, si bien en este ultimo caso, será necesario añadir la siguiente o siguientes letras cuando coincida la inicial en comandos distintos.

Tanto en esta posición, como durante la ejecución de cualquier comando,la tecla de <ayuda> F1 nos informa de lo que podemos hacer ó contestar en ese punto.Una nueva pulsación de dicha tecla, nos presenta una ventana con el comportamiento del teclado (figura 5).En ambas ventanas de ayuda, la flecha de movimiento del cursor sigue presentando la información, si no cabe.Para regresar al punto de trabajo, hay que pulsar la tecla Esc.

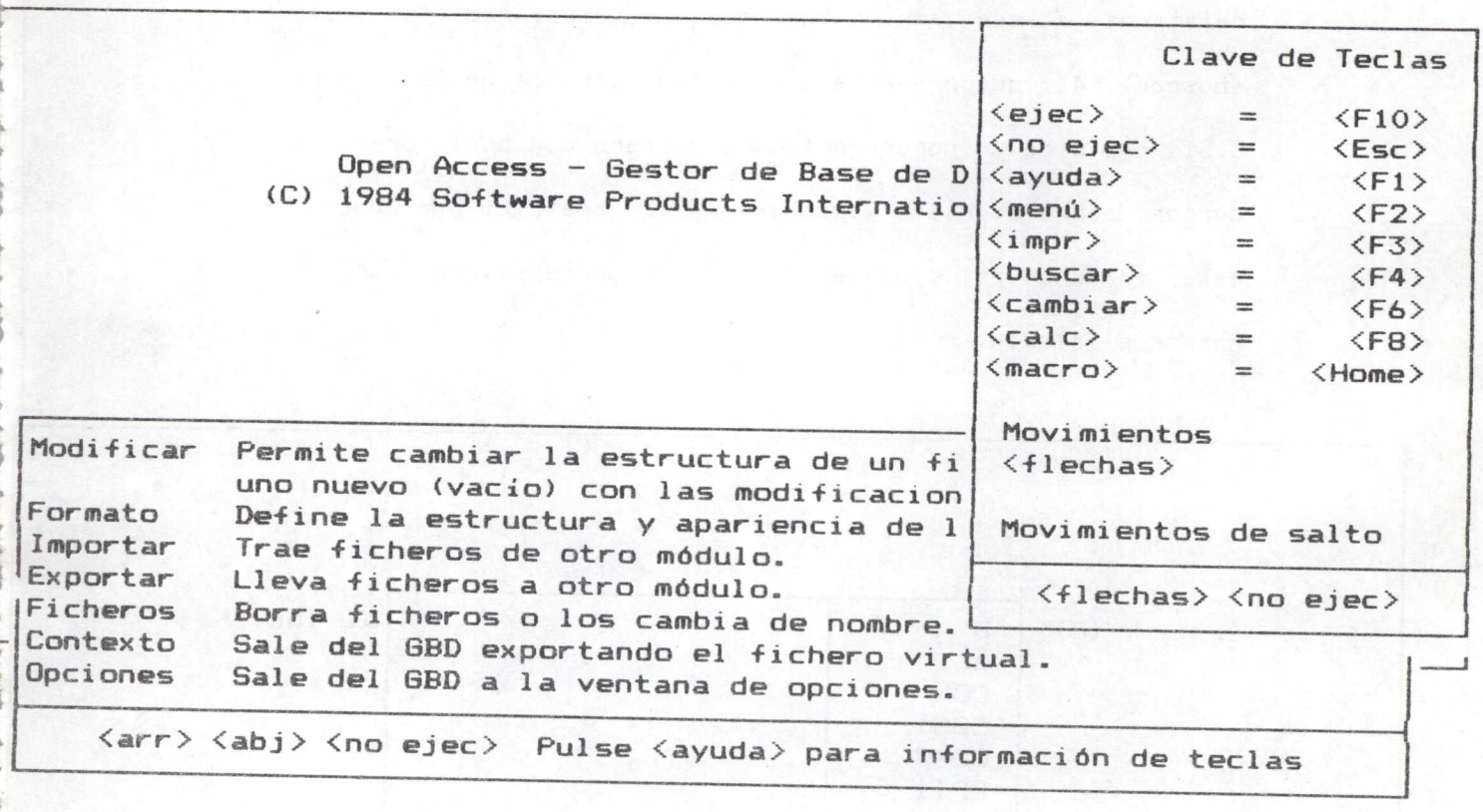


figura 5

b)Con unidad de disco duro.

El proceso es análogo, con las siguientes diferencias:

-El punto de partida es la situación en el directorio del Open Access.

-El unico diskette necesario es el de arranque, ya que los módulos del Open Access y el área de trabajo del usuario se encuentran en el disco duro.

2.CREAR UN FICHERO

Vamos a preparar un sencillo fichero de personal para nuestro Instituto.

Desde el **Menú de Mantenimiento de Ficheros**, seleccionamos el comando **Crear**.El Gestor de Base de Datos solicita el nombre del nuevo fichero, a cuya petición responderemos PERSONAL.

Cada vez que solicite un nombre de fichero, se puede pedir información sobre los ya existentes en el disco, relativos a ese ambito de trabajo, mediante la tecla **<buscar> F4**, que presenta una ventana como la de la figura 6.Si nos pide un nombre antiguo, bastará con posicionar el cursor sobre el mismo y pulsar <CR>.Si nos pide uno nuevo, esta ventana de ayuda evita errores de repetición de un nombre existente.

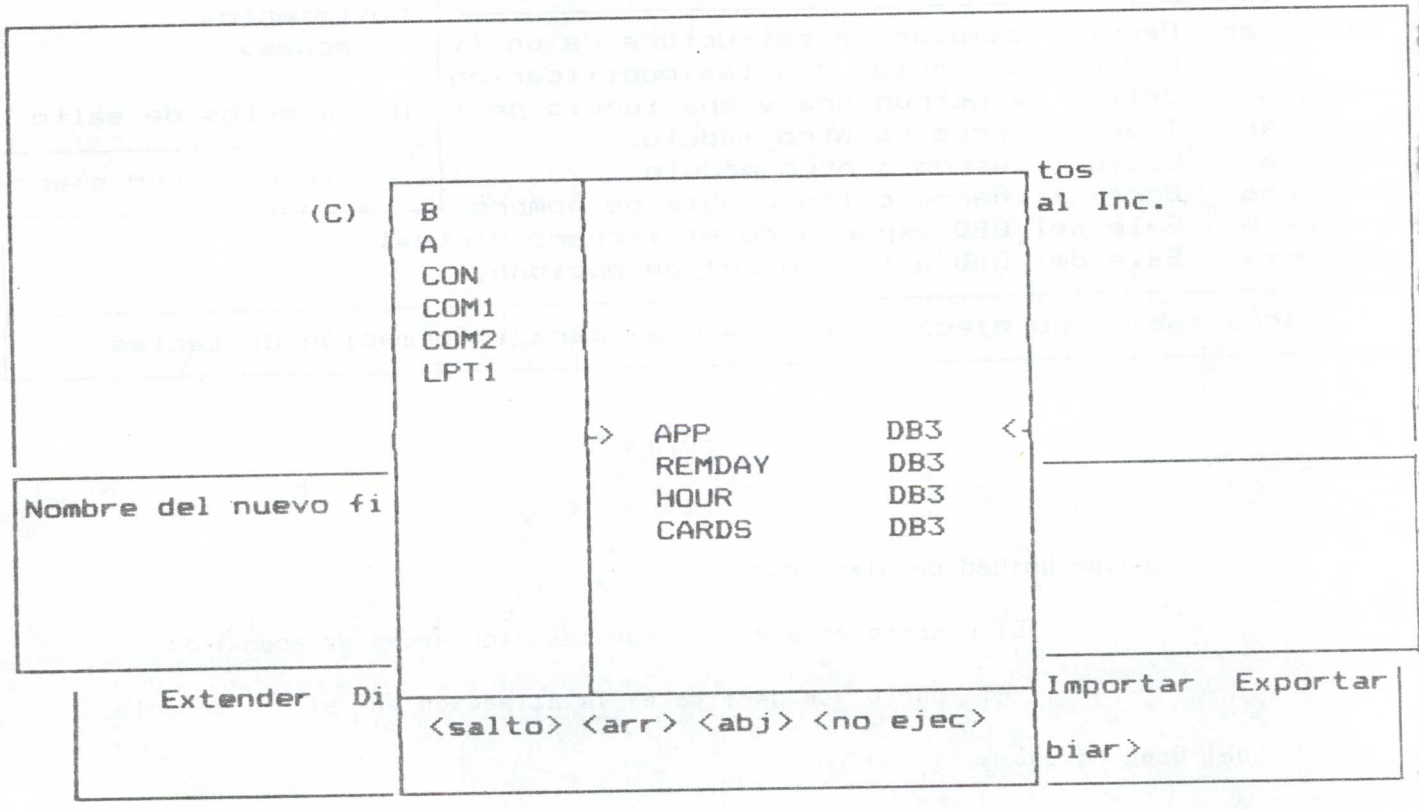


figura 6

El Gestor de Base de Datos nos lleva a una pantalla vacia, con un **Menú de Diseño de Fichero y Formato de Pantalla**, para permitirnos definir los atributos de los campos y el diseño de la mascara de pantalla.

Seleccionamos **Editar**, realizando , para cada campo, el siguiente proceso:

-Situar el cursor en la posición en que deseemos que aparezca el campo en la máscara de pantalla.

-Dar titulo al campo (si no incluye caracteres prohibidos, ni supera los 10 caracteres, puede coincidir con el nombre del campo).

-Definir el ancho del campo.Pulsando tantas veces F9 como espacios deseemos y <CR> para validarlo.

```
                    FICHERO DE PERSONAL
                    ===================

APELLIDOS

Nombre: APELLIDOS   Clase: No-Clave      Tipo: Texto          Tamaño: 30
 <tecl ed> <entrada> <mover ent> <borr ent> <cambiar> <menú> <buscar>

Menú de Diseño de Fichero y Formato de Pantalla
Salir  Editar  Línea_Cuyo  Nuevo  Tamaño
```

figura 7

-Modificar los atributos del campo.Si los asumidos por defecto (mostrados en la ultima linea de la máscara, figura 7) no son los deseados, tenemos que <cambiar> F6 a la ventana de Atributos de los Campos (figura 8), cuya descripción se encuentra en el Apendice B.

```
                    FICHERO DE PERSONAL
                    ===================

Nombre          : APELLIDOS
Clase           :  Clave-Unica  Clave  No-Clave
Tipo            :  Texto  Fecha  Número  Verdad/Falso  Decimal
Ajuste          :  Izquierda  Centrado  Derecha  Repetido
Evaluado        :  Normal  Autofecha  Automático  Salto
                   Dependiente  Autoincrementado  Rango
Modo Vídeo      :  Normal  Modo-1  Modo-2  Modo-3
Debe Llenarse:  Verdad  Falso
Ancho visible:     30
Duplicado       :  Verdad  Falso
Debe Casar      :  Verdad  Falso

                    <ejec> <no ejec> <arr> <abj>

Menú de Diseño de Fichero y Formato de Pantalla
    Salir  Editar  Línea_Cuyo  Nuevo  Tamaño
```

figura 8

Si el tipo seleccionado es Texto, pueden indicarse caracteristicas del mismo con <cambiar> F6, (figura 9).Si es decimal, la misma tecla permite definir el número de decimales.

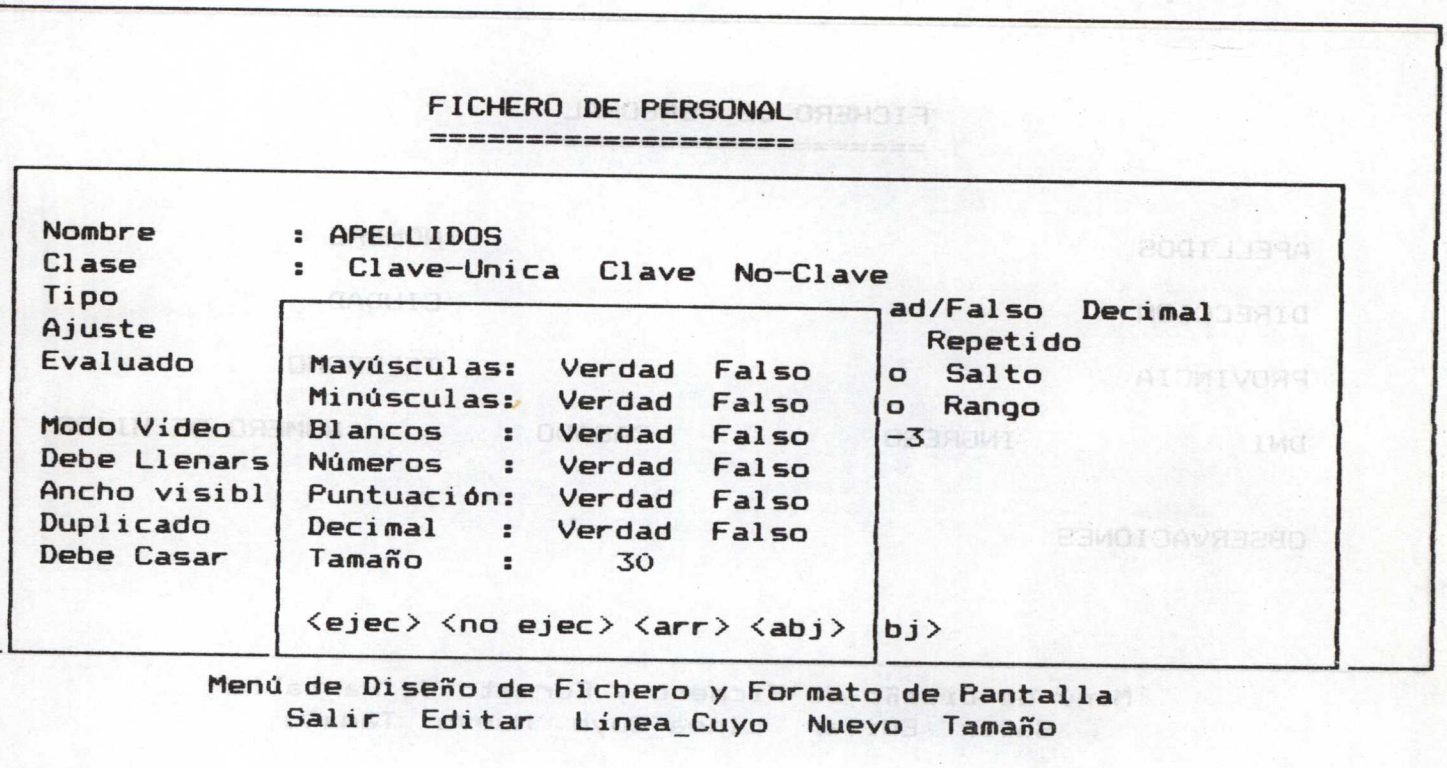


figura 9

Tras seleccionar las opciones deseadas, se validan con F10, que nos lleva de nuevo a la pantalla de diseño.Para pasar al campo siguiente <CR>.

-Grabación del diseño.Pulsamos la tecla <menú> F2, y ejecutamos Salir.

El Gestor de Base de Datos crea automaticamente 2 ficheros, uno con extensión .DB3 que ,contendrá la información, y otro .SMK para almacenar la máscara de pantalla asociada (figura 10).

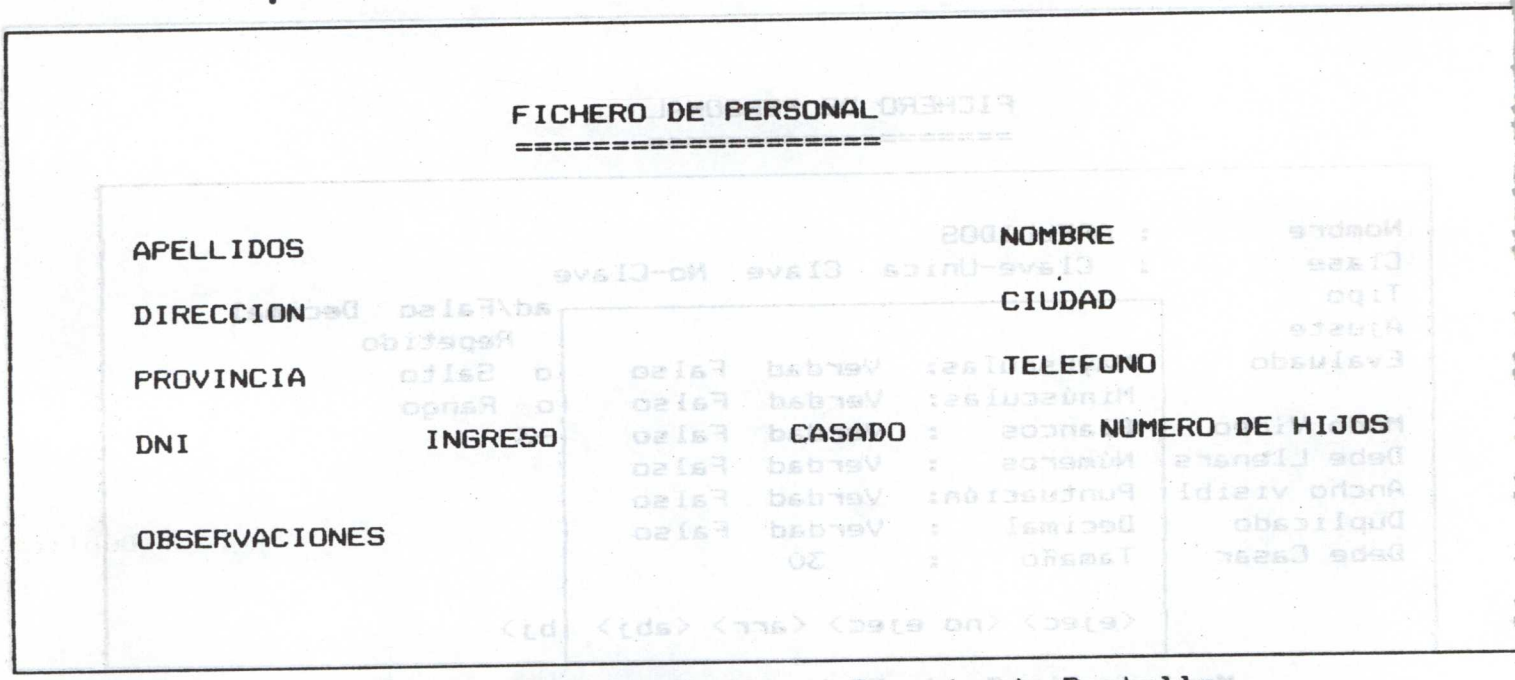


Figura 10

Inicialmente asume, por defecto, 25 registros para el fichero de Base de Datos, pero pide cofirmación ó modificación antes de realizar la reserva en el disco.

3.INTRODUCIR DATOS

Desde el **Menú de Operaciones de Base de Datos** seleccionar el comando **Introducir.**

Aparece la palabra **DE** (ver el Lenguaje de Interrogación en el Apendice C), que es la forma de pedirnos el nombre del fichero.Respondemos PERSONAL.

Presenta en pantalla la máscara del fichero.En la parte inferior izquierda, nos indica el tipo del campo en que nos encontramos.Observese, que los campos de Texto estan en blanco, los numéricos inicializados a 0, el de fecha con su formato 0/0/0 y el de Verdad/Falso con Falso.

Para validar la información de cada campo, pulsar **<CR>**.Para grabar el registro en el disco **F10**.Para finalizar el proceso de introducción de datos **F2** y el comando que vayamos a ejecutar, ó bien **Nuevo** para desctivar el fichero.

4.VER LA INFORMACION ALMACENADA

Para ver los datos, existen 3 comandos diferentes: **Listar, Mostrar** y **Visualizar.**

Listar presenta la información en forma listas por campos, con el nombre del campo como cabecera (figura 11).Para ver los campos que no aparecen en pantalla pulsar las teclas de movimiento del cursor.

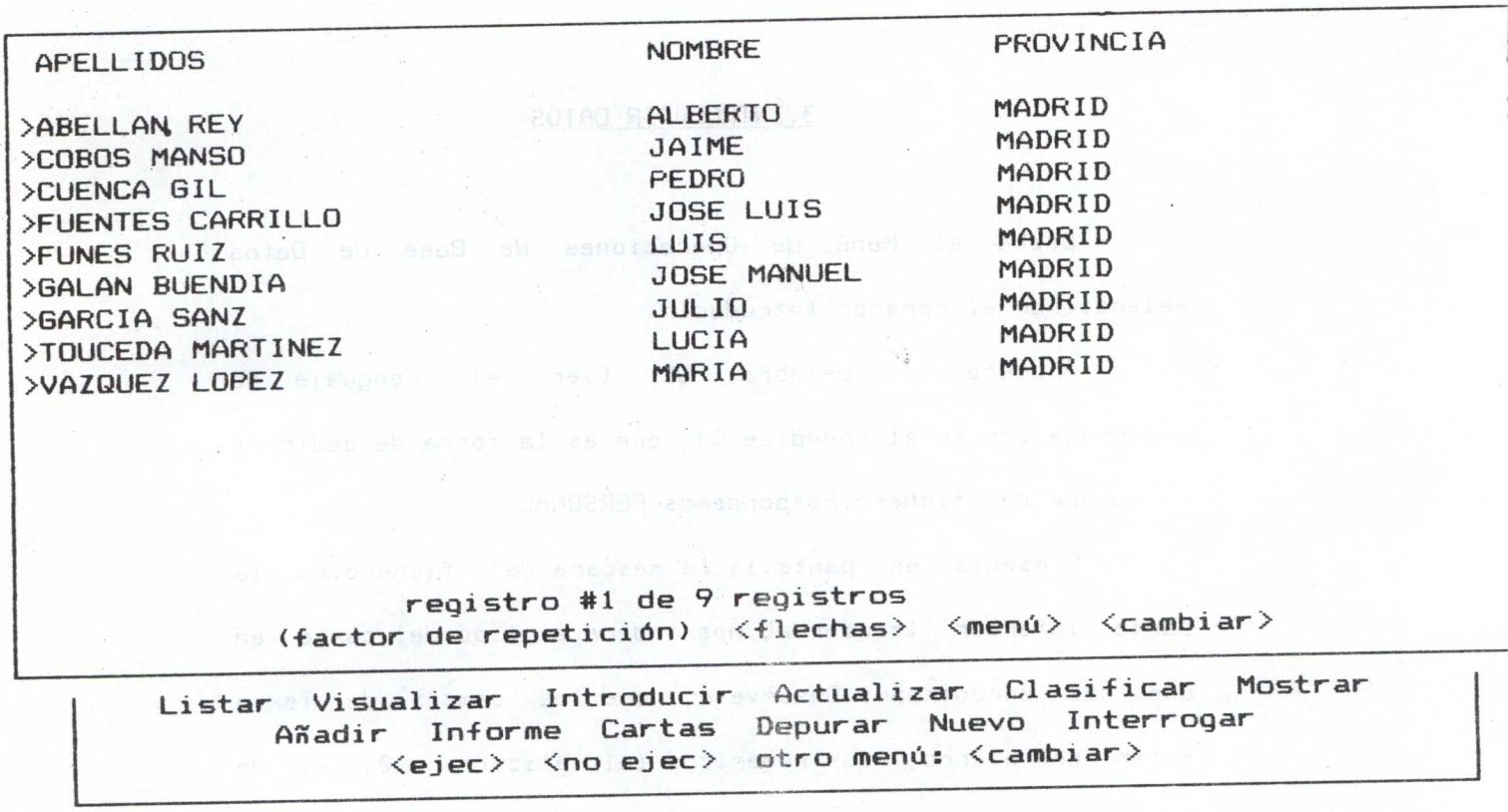


figura 11

Mostrar presenta la información a traves de 2 ventanas (figura 12).En la izquierda, aparecen los contenidos, de los diferentes registros, de un campo clave (el seleccionado en la primera columna de la ventana de la derecha).En la derecha, hay 2 columnas, la primera con los nombres de todos los campos del fichero y la segunda con los los contenidos del registro seleccionado.Para ver otras informaciones pulsar las teclas de movimiento del cursor.

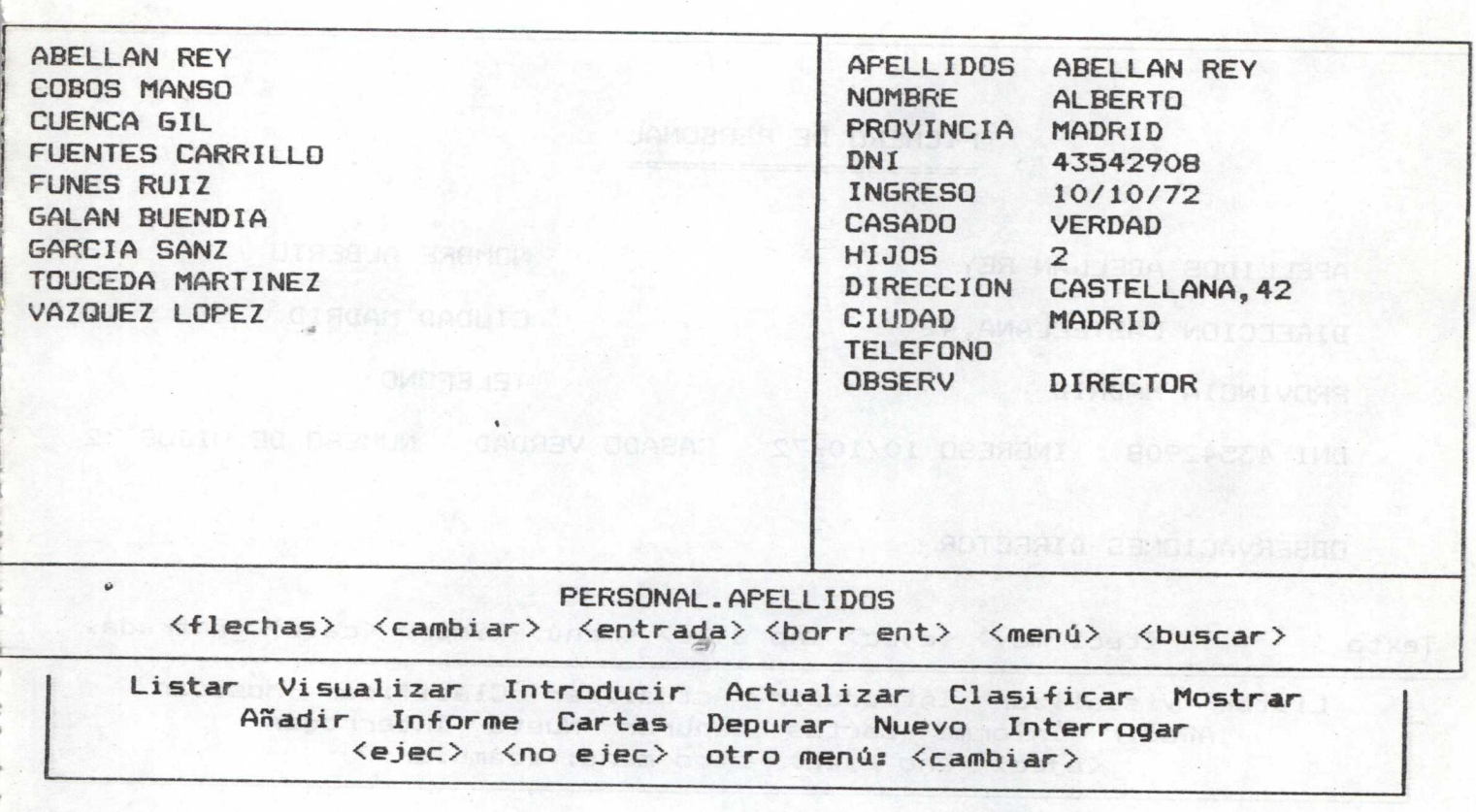


figura 12

Visualizar presenta la información con la máscara de pantalla (figura 13).Para ver el registro anterior ó siguiente, pulsar PgUp ó PgDn.

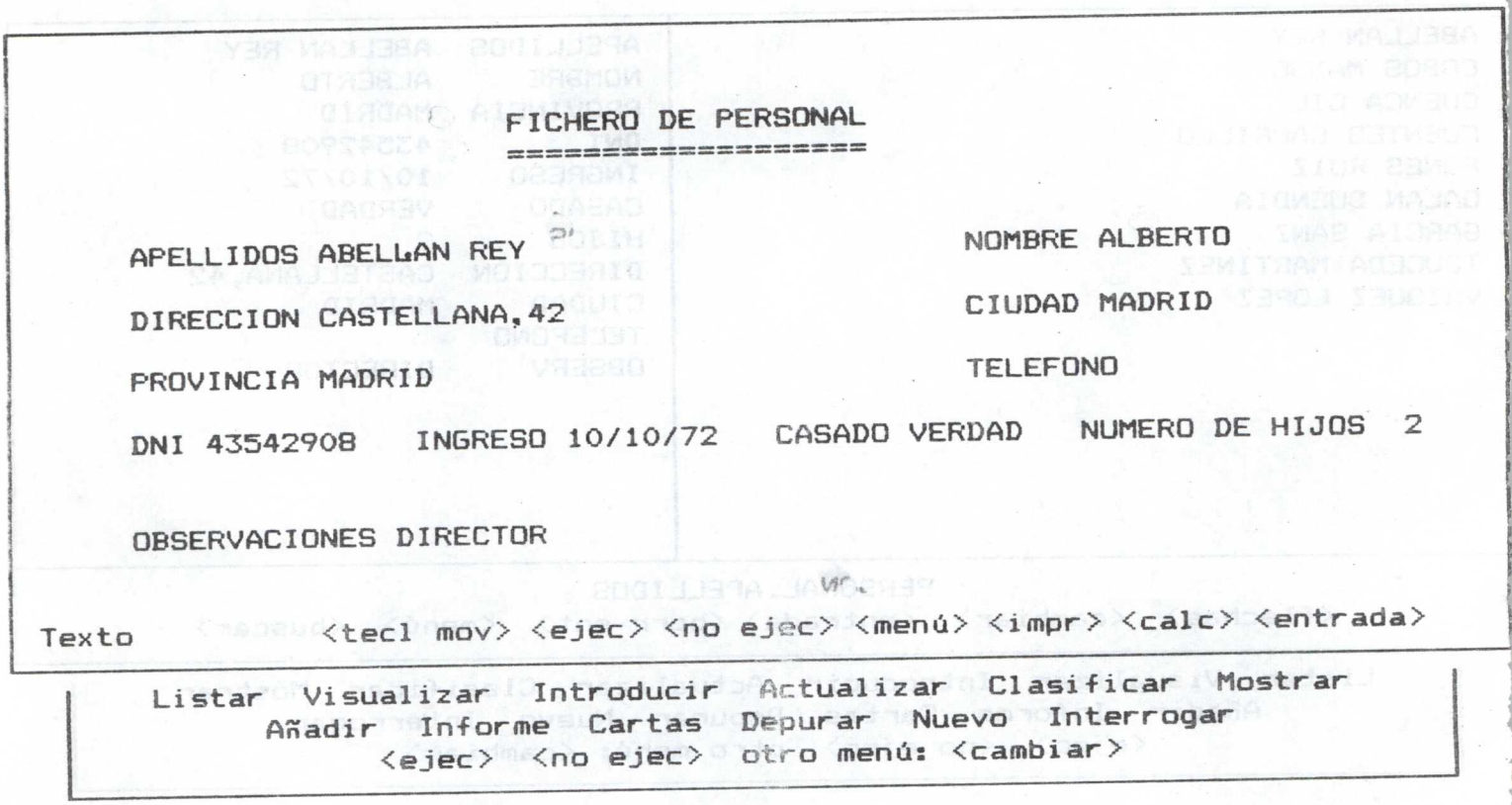


figura 13

5.MODIFICAR DATOS ALMACENADOS

Cuando se trata de modificar la información registro a registro los comandos utilizados son los mismos del apartado anterior, es decir, Listar, Mostrar y Visualizar.Con Visualizar, las correcciones se hacen directamente, sin más que situarnos en el campo deseado é introducir la nueva información.Con Listar y Mostrar, primero seleccionamos el registro y despues pulsamos <cambiar> F6, con lo que aparece su contenido con la máscara de pantalla y pasamos a una situación análoga a la de Visualizar.

Cuando queremos modificar el contenido de un campo en un conjunto de registros, es más potente el comando **Actualizar.**

6.VARIAR EL DISEÑO DE LA MASCARA DE PANTALLA

Seleccionar en el **Menú de Mantenimiento de Ficheros** el comando **Diseñar.**Indicado el nombre del fichero, obtenemos el mismo **Menú de Diseño de Fichero y Formato de Pantalla** del comando **Crear.**

Seleccionamos **Editar** y podemos variar la máscara de pantalla y ciertos atributos de los campos.No es posible cambiar Nombre, Clase y Tipo.

Ademas de realizar cambios en una máscara existente, tambien se pueden crear otras.Vamos a diseñar para nuestro fichero de personal una nueva máscara como la de la figura 14.Para ello indicamos el nombre PERSON2

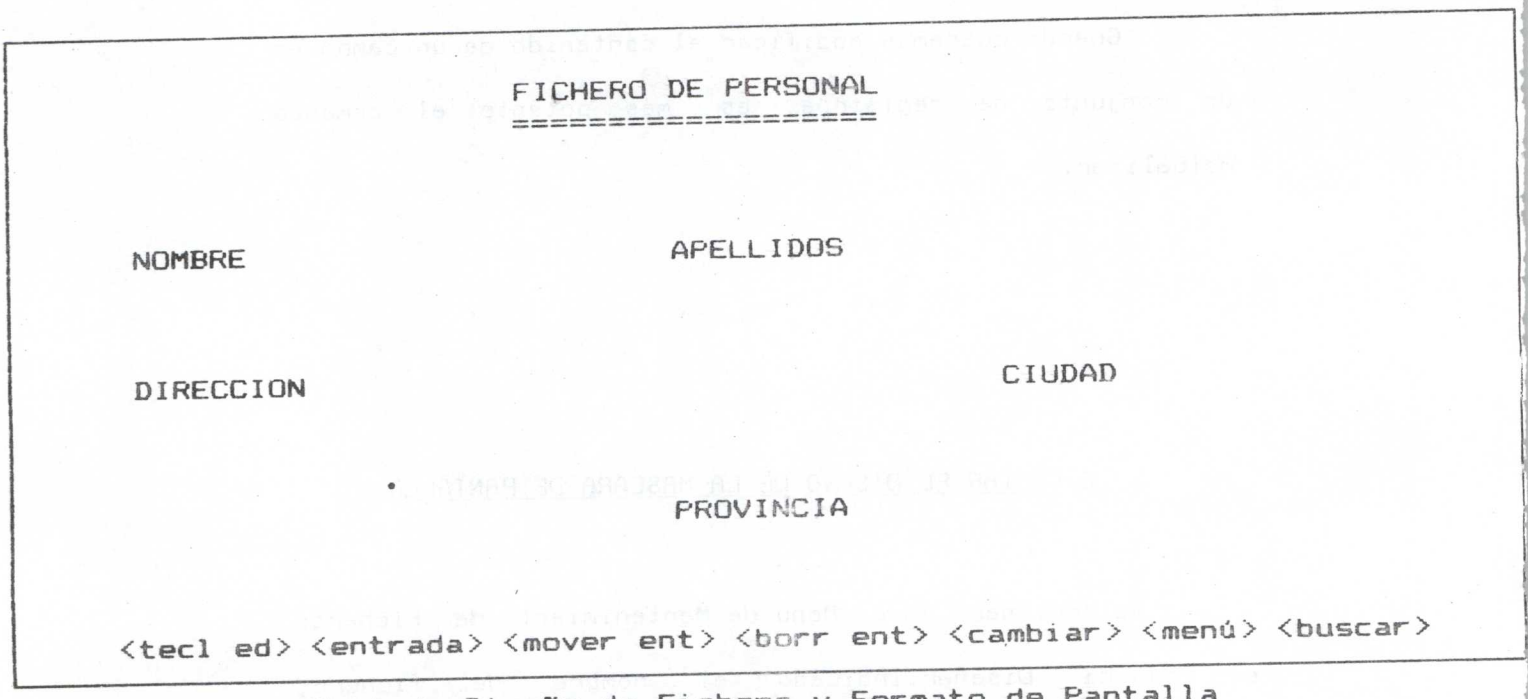


figura 14

7.VARIAR LA ESTRUCTURA DEL FICHERO

Seleccionar en el Menú de Mantenimiento de Ficheros el comando Modificar.Para modificar la estructura de un fichero es necesario crear otro, por lo que tras haber dado el nombre del fichero antiguo , nos pide el nombre del fichero de salida (modificado).

Vamos a suponer que necesitamos modificar el fichero creado (PERSONAL), para transformarlo en otro (PERSON3) cuyos campos y máscara de pantalla correspondan a los de la figura 15.

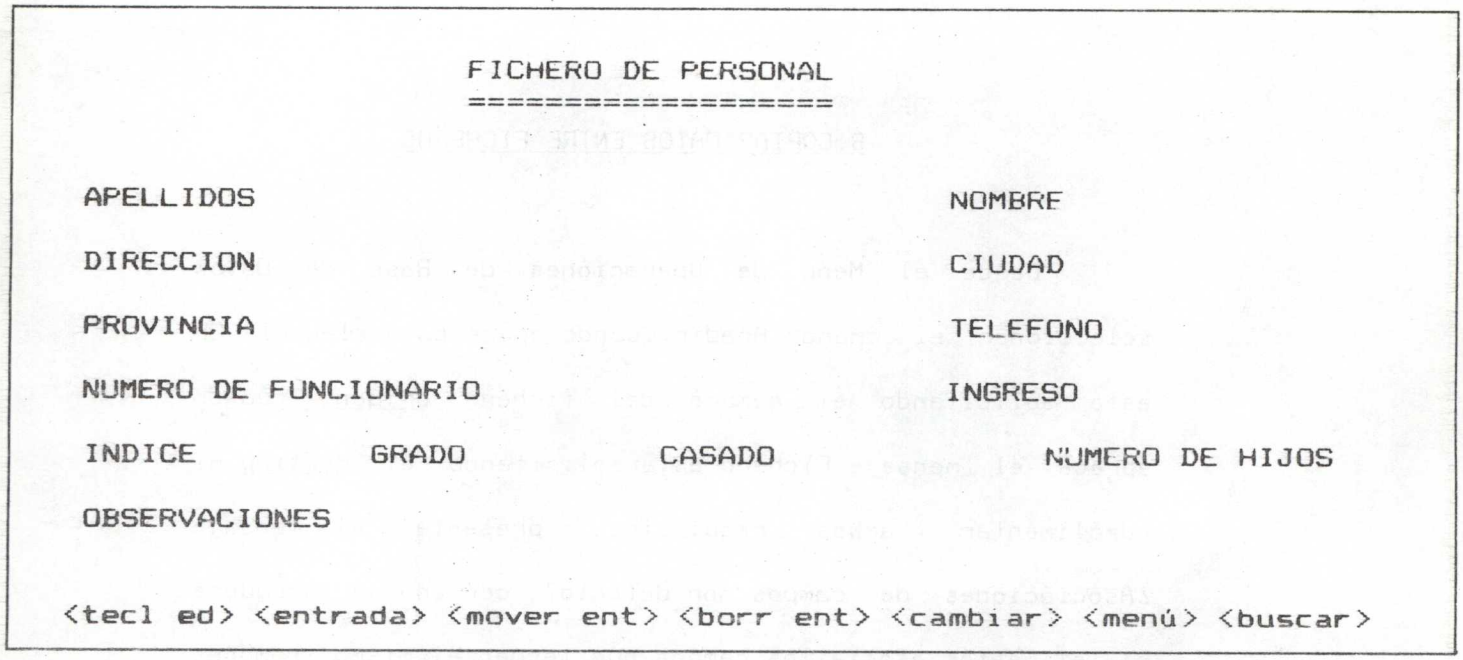


figura 15

Al igual que, en los comandos **Crear** y **Diseñar**, llegamos al **Menú de Diseño de Fichero y Formato de Pantalla** en el cual al seleccionar **Editar**, podemos relizar todas las operaciones de **Crear**.Una vez finalizado el proceso pulsamos **F2** y **Salir**, el Gestor de Base de Datos crea el fichero de extensión .DB3 y el de máscara de extensión .SMK.

El nuevo fichero de datos estará totalmente vacio, por lo que si suponemos, como sucederá en la practica,que el antiguo contiene información que es necesario transferir al nuevo, fichero, siempre seguirá, a este proceso, otro de transferencia de datos del antiguo al nuevo (ver apartado de Copiar datos entre Ficheros).

8.COPIAR DATOS ENTRE FICHEROS

Desde el **Menú de Operaciones de Base de Datos** seleccionar el comando **Añadir**.Cuando aparece la clausula **DE** esta solicitando el nombre del fichero **origen**, cuando aprece el mensaje **Fichero** esta solicitando el **destino**.Al cumplimentar ambos requisitos, presenta el mensaje **¿Asociaciones de campos por defecto?**, con lo que pregunta si el Gestor asocia los campos que tengan el mismo nombre, ó queremos confirmar una a una las asociaciones.Como en los diferentes ambientes del Open Access **F10** equivale al **Si** y **Esc** al **No.**

Los campos del fichero nuevo con asociación en el antiguo, quedan con la información transferida, los demás vacios, ó con los valores asumidos por defecto.

9.AMPLIAR LA CAPACIDAD DE UN FICHERO

Desde el **Menú de Mantenimiento de Ficheros** seleccionar el comando **Extender**.Tras indicarle el fichero a que nos referimos, nos incrementa en 25 el número de registros que ya tuviese el fichero, pudiendo confirmar ese valor, ó introducir los que sean necesarios.

10.BORRAR Y CAMBIAR EL NOMBRE A UN FICHERO

Las 2 operaciones tienen un desarrollo análogo.Desde el **Menú de Mantenimiento de Ficheros** seleccionar el comando **Ficheros**.Nos pide el nombre del fichero con el que queremos actuar, pudiendo tras responderle, decidir cual de las 2 operaciones vamos a realizar.

-Borrar un Fichero

Pulsar **<borrar lin>** **End-Del**, el Gestor pide confirmación (**F10**), para evitar posibles errores.Tras haber borrado el fichero, queda a la espera de que se le pueda dar otro nombre de fichero, para borrarlo ó cambiarle el nombre.Para salir de esta situación pulsar **Esc**.

-Cambiar el Nombre a un Fichero

Pulsar **<cambiar>** **F6**.Nos pide un nuevo nombre y realiza el cambio.

Cuando el fichero al que le cambiamos el nombre es de datos, nos pide tambien la **Máscara a Actualizar**, para asociar al menos un fichero .SMK con el nuevo nombre del

fichero .DB3.Tras cumplimentar esta petición vuelve a solicitarme una nueva máscara, por si queremos asociar varias al mismo fichero de datos.Para poder terminar con el proceso es necesario salirnos con **Esc.**(Para salir del comando **Ficheros** pulsar 2 veces **Esc**).

APENDICE B

ATRIBUTOS DE LOS CAMPOS

Los atributos de los campos pueden ser definidos (comando **Crear**), ó modificados (comandos **Diseñar** y **Modificar**), en el proceso de trabajo con la Base de Datos.Para poder realizarlo, hay que situarse en la ventana coorrespondiente (figura 1), pulsando **F6** desde la ventana de **Diseño de Fichero y Formato de Pantalla**, obtenida con alguno de los **3** comandos antes mencionados.

```
                    FICHERO DE PERSONAL
                    ===================

Nombre          : APELLIDOS
Clase           :  Clave-Unica  Clave  No-Clave
Tipo            :  Texto  Fecha  Número  Verdad/Falso  Decimal
Ajuste          :  Izquierda  Centrado  Derecha  Repetido
Evaluado        :  Normal  Autofecha  Automático  Salto
                   Dependiente  Autoincrementado  Rango
Modo Vídeo      :  Normal  Modo-1  Modo-2  Modo-3
Debe Llenarse:  Verdad  Falso
Ancho visible:     30
Duplicado       :  Verdad  Falso
Debe Casar      :  Verdad  Falso

                  <ejec> <no ejec> <arr> <abj>

Menú de Diseño de Fichero y Formato de Pantalla
    Salir  Editar  Línea_Cuyo  Nuevo  Tamaño
```

figura 1

Para seleccionar las diferentes opciones, hay que sobreiluminarlas utilizando las teclas de movimiento del cursor.Para confirmar los atributos seleccionados y regresar a la ventana de **Diseño de Fichero y Formato de pantalla**, pulsar **F10.**

Nombre Identifica al campo.Es el nombre real del campo en el fichero de datos del disco.Asume por defecto el del titulo dado en la máscara, salvo que contenga caracteres no validos ó supere los 10.En estos casos lo sustituye por el nombre del fichero, con un subindice que corresponde a la posición del campo.

Clase Permite dar al campo el caracter de clave.Admite hasta un máximo de 15 campos clave por registro.La clave-única da el carácter de clave-primaria, es decir, no permite introducir la misma información, en ese campo, en 2 registros.

Tipo Define el tipo de campo y su ocupación en Bytes.

TIPO	LONGITUD
Texto	1 Byte-caracter (máx 58)
Fecha	2 Bytes
Numérico	2 Bytes
Verdad/Falso	1 Byte
Decimal	10 Bytes

El tipo Texto dispone de un submenú (figura 2), al que se accede pulsando <cambiar> F6, que permite definir sus caracteristicas.

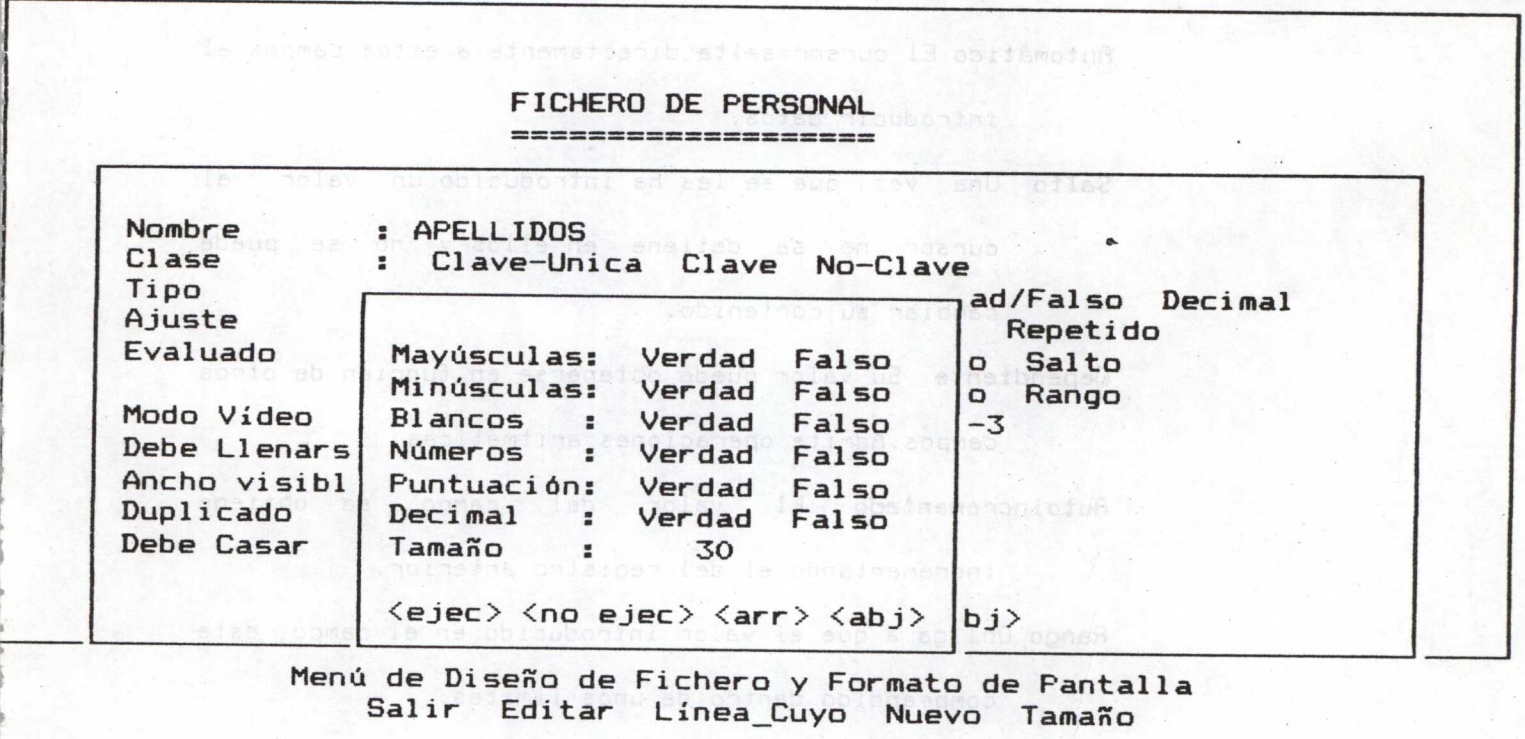


figura 2

El tipo Decimal permite, tambien con F6, definir el número de decimales, hasta un máximo de 15.

Ajuste Ajusta los datos a Izquierda, Centro, Derecha, ó hace que se repitan a lo largo de toda la máscara del campo.

Evaluado Caracteriza los datos del campo.

Autofecha Toma automaticamente la fecha del sistema operativo.

Automático El cursor salta directamente a estos campos al introducir datos.

Salto Una vez que se les ha introducido un valor, el cursor no se detiene en ellos y no se puede cambiar su contenido.

Dependiente Su valor puede obtenerse en función de otros campos.Admite operaciones aritméticas.

Autoincrementado El valor del campo se obtiene incrementando el del registro anterior.

Rango Obliga a que el valor introducido en el campo, este comprendido dentro de unos limites.

Modo Video Define la forma en que se verá el dato en pantalla.

Modo-1.....Video inverso

Modo-2.....Subrayado

Modo-3.....Color

Debe LLenarse Si se selecciona Verdad, indica que es obligatorio introducir un valor en este campo.No puede pulsarse <CR> en vacio.

Ancho Visible Determina la anchura de la máscara del campo en la pantalla.

Duplicado Toma el valor del registro anterior, hasta que se introduzca uno nuevo.

Debe Casar Su contenido debe coincidir con el de algún campo de otro fichero.

Solamente se puede definir este atributo, si ya existe el fichero que contiene el campo con el que debe casar.Debe indicarse (F6) el nombre del fichero y el del campo con el cual quiere conectarse con este atributo.

APENDICE C

LENGUAJE DE INTERROGACION

Es un lenguaje que permite definir los ficheros, registros y campos implicados en la operación a desarrollar por el comando de la Base de Datos.

Tiene una estructura similar al SQL de la casa IBM y utiliza las siguientes clausulas:

DE nombre de fichero [,nombre de fichero]...

- Selecciona el/los fichero/s especificado/s.

ELIGE nombre de campo [,nombre de campo]...

-Permite seleccionar unicamente los campos enumerados.

CUYO condición [Y|O condición]...

-Permite activar unicamente los registros que cumplan las condiciones especificadas.

ORDEN nombre de campo [,nombre de campo]...

-Permite indicar el/ los campos que define/n el criterio de ordenación.

La clausula **DE** es la unica obligatoria y aparece escrita cuando ejecutamos un comando, que utiliza el lenguaje de interrogación.Las demás son opcionales, y hay que introducirlas antes de cumplimentarlas.

PRACTICAS

AYUDADOS DE LA GUIA DE OPERACION:

1.- Crear el fichero de personal de un centro. Deberá tener, al menos, los campos especificados a continuacion, con las características básicas indicadas:

CAMPO	CARACTERISTICAS BASICAS
Apellidos	Clave, texto, debe llenarse
Nombre	Clave, texto, debe llenarse
Dirección	No-clave, texto
Ciudad	Clave, texto
Provincia	Clave, texto
D.N.I.	Clave unica, debe llenarse
Ingreso	Clave, fecha, debe llenarse
Casado	Clave, verdad/falso, debe llenarse
Hijos	Clave, numerico.

Y una máscara de pantalla análoga a la de la figura 10.

2.- Introducir datos en unos diez registros.

3.- Realizar diferentes pruebas con los comandos que permiten ver el contenido de los registros (listar, mostrar y visualizar). Para practicar el lenguaje de interrogación realizar:

- Un listado que solo incluya apellidos, nombre y fecha de ingreso de los que llevan mas de un trienio.
- Un listado de los casados, que indique apellidos, nombre, DNI. número de hijos, ordenado de mayor a menor número de hijos.

4.- Corregir datos de alguno de los registros. Por ejemplo un cambio de dirección. Comprobar las modificaciones realizadas.

5.- Diseñar una nueva mascara de pantalla (PERSON2.SMK), como la de la figura 14, que permita presentar ciertos campos del fichero de personal.

6.- Modificar la estructura del fichero, para obtener otro(PERSON3) con los campos especificados en la figura 15.
Decidir, razonadamente los atributos de los campos que no existen en el fichero de personal original.

7.- Transferir los datos del fichero original de personal al nuevo fichero introducir los datos que correspondan en los campos de nueva creación.

P R A C T I C A S

Diseñar libremente los ficheros de clientes, artículos y pedidos de la empresa FPS.A., efectuando inicialmente un boceto sobre papel y viendo las relaciones que puedan existir entre los campos de los tres ficheros.

Los campos que, como mínimo, deben tener estos ficheros son los siguientes:

CLIENTES	ARTICULOS	PEDIDOS
Codigo	Codigo	Codigo cliente
Nombre	Nombre	Codigo artículo
Dirección	PVP	Cantidad
Ciudad	Precio de Costo	Número de pedido
Provincia	Stock	Fecha pedido
Clase (1)		
	Unidades vendidas	
Ventas (2)		

(1) Se refiere a la categoría del cliente, para aplicarle diferentes descuentos.

(2) Para acumular las ventas a este cliente.

LAS BASES DE DATOS EN LAS AULAS

Uno de los grandes objetivos de la aplicación de la informática a la enseñanza es el de favorecer la capacidad de organización del saber, lo que representa una alternativa al puro conocimiento memorístico. Las Bases de Datos gestionadas por medios informáticos hacen posible un cambio curricular en los procesos de enseñanza-aprendizaje que se hace cada día más imprescindible en la actual sociedad postindustrial para adaptarse a las necesidades que impone la nueva cultura de la información.

Frecuentemente se suele esgrimir como argumento en contra de la utilización de programas de aplicación del tipo Procesadores de Textos, Bases de Datos, Hojas de Cálculo, etc., que son programas construidos con un sentido y finalidad empresarial. Esto es cierto para la mayoria de los existentes en la actualidad en España, no así en otros paises, como por ejemplo Inglaterra o Estados Unidos, donde existen adaptaciones, versiones y utilidades de tipo educativo. Por otro lado, es este mismo argumento, el que da un caracter particular al trabajo del alumno con dichos programas, acercándole al "trabajo de adulto".Y es con la utilización de las Bases de Datos,con lo que se consigue que los alumnos se inicien en esta forma de trabajo más real, de una forma natural, integrándo la informática en distintas disciplinas

del currículum y en muchos casos de una forma interdisciplinar.

La actividad del alumno que investiga en distintas fuentes, bibliotecas, periódicos, archivos..., en su propio entorno; que valora los datos recogidos, que los analiza y selecciona, que trabaja con un gran fichero, no con un pseudofichero de unos cuantos datos, en el que obtiene de una forma inmediata contestación a preguntas complejas, que necesita en muchos casos trabajar en grupo con sus compañeros; en definitiva, todas las actividades que se pueden plantear al utilizar una Base de Datos, no tiene parangón con la situación de ese mismo alumno, resolviendo en su mesa un problema totalmente teórico, lejano de su realidad y en demasiados casos completamente artificial.

El papel del profesor cambia radicalmente respecto a la situación tradicional y pasa de ser la persona que suministra toda la información necesaria para resolver el problema, a ser un elemento del grupo, que trabaja con los alumnos en la

planificación, búsqueda, análisis, selección, introducción en el ordenador y utilización de la información para la obtención de conclusiones nuevas.

Está claro que observando el desarrollo y crecimiento de la información en las últimas décadas, y las previsiones que de ello se derivan -nótese, por ejemplo, que cada diez o doce años, se duplica la cantidad de información científica creada por la Humanidad- nuestra concepción de la escuela debe cambiar. Cada vez es más difícil dominar todos los aspectos relativos incluso a una especialidad determinada. La formación, por tanto, se debe encaminar en dos vertientes que discurren paralelamente; por un lado, desarrollar las capacidades de resolución de problemas y por otro adquirir destrezas para el acceso y utilización de la información. Estas dos cualidades se obtienen, entre otras cosas, con la utilización de Bases de Datos, y el ordenador aquí es el instrumento indispensable que va a posibilitar al alumno este trabajo -resulta muy difícil pensar en jóvenes creando e incluso trabajando con archivos en soporte de papel, con el esfuerzo, paciencia y dedicación que ello supone-.

El trabajo con una Base de Datos tiene un carácter real

y no es algo abstracto y ajeno al alumno, que pertenece al mundo de los profesores o de la Física o la Geografía. Como ya se ha dicho, será en muchos casos interdisciplinar, lo que motivará la creación de grupos de trabajo de profesores especialistas en distintas áreas, formándose por tanto ellos mismos, en esa interdisciplinariedad que deben transmitir a sus alumnos.

La informática en la educación se suele contemplar desde tres puntos de vista: como E.A.O., como objeto de conocimiento en sí misma y como un útil de trabajo. Un caso particular de esta última modalidad sería la utilización de bases de datos con ordenador, que junto con los procesadores de textos, hojas de cálculo, diseñadores de gráficos y otros programas de usuario pueden ayudar a los estudiantes en muy diversas tareas. Pero la aportación especial de las bases de datos en las aulas es que proporcionan al alumno un elemento de conexión con lo que va a ser su vida futura, por lo que de "trabajo real y de adulto" tiene este tipo de actividad y que ya hemos mencionado anteriormente.

APLICACIONES DE LAS BASES DE DATOS

La forma más natural de iniciarse en el trabajo con Bases de Datos, es utilizar una que ya esté construida, siendo de gran importancia que el programa de gestión sea fácil de manejar para que no interfiera en el aprendizaje que se pretende realicen inicialmente los alumnos y que consiste en una familiarización con los datos que contiene, con la forma

en que esos datos están estructurados, con las posibilidades de selección de los datos y con su utilización para sacar conclusiones, resolver problemas y redactar informes. Con todo ello, se consigue por un lado, que obtengan un mejor conocimiento de la materia o materias de que trata la Base de Datos, y por otro, que reconozcan la terminología básica empleada en el tratamiento de la información, fichero, registro, campo, atributo, clave, etc., y además que sean capaces de plantearse y resolver sus propios prolemas.

La realización de las actividades planteadas en las siguientes cuatro etapas, lleva al alumno a un grado más que notable de conocimiento de las posibilidades y tratamiento de la información.

PRIMERA ETAPA: UTILIZACION DE UNA BASE DE DATOS

- Para comenzar y trás una breve explicación de la puesta en marcha del programa de gestión, los alumnos, por grupos, investigarán qué datos están recogidos, tomando notas de las características que más les llamen la atención, para después realizar una puesta en común en la que se llegue a definir la estructura de la base de datos. Este trabajo les permitirá conocer los ficheros que la componen, así como los campos que forman parte de los distintos registros y los atributos claves que identifican esos campos.

- Superada la fase anterior, los alumnos están en disposicion de referirse tanto a los ficheros, como a los campos por sus identificadores y ya podrán contestar preguntas, inicialmente elementales, que se resuelvan seleccionando en un fichero, un registro, un campo, o bien todos los registros de un fichero que cumplan una determinada condición, algunos campos específicos de todos los registros o varias cosas a la vez

- Es aconsejable que en esta fase sea el profesor quien plantee las consultas que deben realizarse y con posterioridad se resuelvan las propuestas por los alumnos. Con seguridad, algunas de estas últimas no podrán ser contestadas, éste es un buen momento para analizar las posibilidades y limitaciones del ordenador, del programa que gestiona la base de datos, así como de la propia organización de los datos.

- Conocida ya la estructura de la base de datos, la información que contiene y cómo extraerla del ordenador, el profesor puede proponer la resolución de problemas que requieran su utilización. Inicialmente los problemas serán elementales y el profesor guiará a los grupos que lo necesiten,para progresivamente ir aumentando el nivel de dificultad y la autonomía de los grupos que realizarán una puesta en común una vez obtenidos los resultados. Para los problemas más complejos, se debe orientar a los alumnos en la técnica de división del problema en otros

más simples y que pueden ser resueltos cada uno por un grupo distinto. Esto les obligará a adoptar acuerdos para el reparto del trabajo, así como a responsabilizarse de los resultados parciales que influyen en la solución global del problema.

- Se pretende en este momento, que los alumnos analicen los resultados y detecten los errores cometidos, interpretando los resultados contradictorios o absurdos que puedan obtenerse, bien por fallos en el proceso, o bien por defectos de la propia base de datos lo que les acostumbrará a contrastar los resultados con su propia experiencia.

- Seguramente se planteará la cuestión de intoducir los resultados obtenidos en nuevos campos, o bien la información contenida en los ficheros resulta insuficiente para resolver las cuestiones planteadas, con ello surge la necesidad de ampliar la base de datos. Se sugiere entonces a los alumnos que busquen información en sus casas, en bibliotecas o allí donde piensen que pueda haberla, para ampliar y mejorar la base de datos disponible. Con esta actividad, los alumnos se enfrentan al problema de localizar y acudir a las fuentes de información, recopilar ésta, seleccionarla, unificar unidades de medida, adaptar la información a una estructura preestablecida e introducir los datos en el ordenador.

- En este momento, los alumnos ya están en disposición de programar sus propios proyectos sobre la base de datos con la que han estado trabajando. El profesor puede sugerir algunos y encauzar los propuestos por los distintos grupos.

SEGUNDA ETAPA: CREACION DE UNA BASE DE DATOS

Si se han superado con éxito las distintas fases de la etapa anterior, los alumnos ya saben cómo se utiliza una base de datos que no fué creada por ellos. Se plantea ahora un objetivo de nivel superior: que el alumno cree su propia base de datos. Para ello, se propone la búsqueda, organización e introducción en el ordenador de la información necesaria para la creacion de una base de datos que ha sido diseñada por el profesor. EL camino a seguir es laborioso y formativo, pues los alumnos deben realizar muy variadas actividades que producirán el desarrollo de sendas capacidades cognitivas.

- En primer lugar, fijado el tema y los objetivos fundamentales que se persiguen con la base de datos que van a crear, deberán acordar los datos que se deben recoger y cuáles no, lo que les hará reflexionar sobre la importancia de determinados datos para la consecución de los objetivos propuestos, es decir, se fomentará en los alumnos una actitud de discriminación frente a los datos, distinguiendo lo fundamental de lo irrelevante.

- En segundo lugar, deberán analizar y planificar la búsqueda de los datos, así como localizar las fuentes de información. Esta actividad les permitirá conocer mejor el entorno en que se desenvuelven, así como conectar con distintas fuentes de información, tanto del tema que les ocupa, como de otros que en un futuro pueda interesarles.

- La recogida de datos, en tercer lugar, es una actividad que les pone en contacto con un medio no usual, con lo que de enriquecedor tiene el enfrentarse a nuevas situaciones. Permitirá interiorizar los problemas del área o asignatura de que se trate, al ponerse en contacto con datos reales, adquiridos de forma personal y con su propio esfuerzo.

- En cuarto lugar, deberán confrontar los datos recopilados, comprobar su exactitud y clasificarlos. Todo ello les hará adoptar una actitud más crítica hacia las fuentes de información, pues con seguridad encontrarán que de fuentes distintas se han recogido datos diferentes, o incluso contradictorios. La tarea de clasificación les dará una visión de conjunto del problema que no habían tenido con anterioridad.

- En quinto lugar, la introducción de datos en el ordenador es una tarea minuciosa que requiere mucha exactitud, una buena dosis de paciencia y un buen conocimiento de los recursos del programa de gestión para

corregir los errores o para optimizar el tiempo de entrada de datos.

- Por último, una vez creada la base de datos, se podrá comprobar si satisface o no todos los objetivos previstos y en cualquier caso, se dispone de una nueva información que puede ser utilizada por cualquier persona.

TERCERA ETAPA: DISEÑO DE UNA BASE DE DATOS

El nivel más avanzado en la utilización de un Programa de Gestión de Bases de Datos es su aplicación a un problema completo, donde el alumno, además de crear y utilizar la base de datos, debe previamente diseñarla.

Esta etapa requiere que el alumno profundice en el conocimiento del programa de gestión y disponga de algún modelo de diseño de bases de datos; para ello, el profesor puede exponer la estrategia seguida por él, en la creación y diseño de las bases de datos que han utilizado con anterioridad.

Según el grado de dificultad del programa de gestión, el profesor optará por proponer uno o varios ejemplos elementales, para que los alumnos diseñen e introduzcan en el ordenador sus ficheros, o por solicitar

NUCLEO: BDAT
UTI: 0025
PAG.: 11

de los alumnos que presenten sus propios proyectos, trabajando en grupos reducidos.

Las actividades que se realizan en esta etapa están orientadas al desarrollo de capacidades de un alto nivel cognitivo, como son: Planificación del proyecto, donde se deben fijar los límites del problema. Toma de decisiones correspondientes. Determinación de los objetivos que se pretenden alcanzar. Elección de la estrategia a seguir. Análisis del problema

CUARTA ETAPA: CREACION DEL PROGRAMA DE GESTION DE LA BASE DE DATOS

Existiría aún otra posibilidad, y es que los propios alumnos efectuaran un diseño en grupos, dirigido por el profesor de Informática, para realizar un programa que permita gestionar una base de datos simple.

Una vez que los alumnos han utilizado uno o más programas de gestión de bases de datos, siguiendo las tres etapas anteriores,han conseguido familiarizarse con la gestión y tratamiento de la información. Se trata ahora de que reconozcan y utilicen las distintas estructuras de almacenamiento: listas, tablas, árboles, etc., y que se planteen los problemas fundamentales de creación de los programas para el tratamiento de la información: almacenamiento consulta, eliminación, modificación, ampliación, selección, ordenación, etc.

Si los alumnos han trabajado con lenguaje LOGO, por ejemplo, podrán profundizar en el tratamiento de listas que efectúa este lenguaje, usando y creando bases de datos propias,incluso, en el caso de alumnos avanzados, diseñando un programa de gestión de la base de datos.

Las siguientes fases, orientadas al lenguaje LOGO, serían traducibles, con las matizaciones convenientes a otros lenguajes como PASCAL, etc.

- En primer lugar, se procede a la realización de los procedimientos para crear una base de datos muy simple, en la que todos los datos son del mismo tipo y están incluidos en una sola lista, por ejemplo, nombres de animales, elementos químicos, compuestos químicos, nombres de ciudades, nombres de reyes, etc. De esta forma, se pueden incluir todos los datos que se deseen. Enseguida surgirá la necesidad de poder obtener esos datos, lo cual puede resolverse mostrando en pantalla toda la lista. Se verá con ésto que el método es poco eficaz, sobre todo si la lista es grande, lo que obligará a realizar nuevos procedimientos que muestren uno o varios datos, a elección del usuario. Al visualizar la lista, probablemente, observarán que algún dato es incorrecto y por lo tanto, debe ser modificado; surge pues, la necesidad de crear un nuevo procedimiento que permita acceder a él y corregirlo.

Igualmente, a la hora de añadir nuevos datos se puede plantear el problema de que el procedimiento utilizado para la introducción inicial no es útil y habrá que modificarlo para permitir esta nueva faceta, o diseñar un procedimiento nuevo.

Con todo ésto, se habrá conseguido que los alumnos se inicien en la utilización de la estructura de lista en LOGO y que capten algunas necesidades del tratamiento automático de la información.

- La siguiente fase, consistirá en seguir los mismos pasos de la anterior, pero para una estructura más compleja como puede ser la estructura de tabla, donde todos los registros tienen el mismo número de campos. Por ejemplo:

[elemento-químico símbolo P.A. N.A. estado-natural]

Esta podría ser la estructura de uno de los registros.

Aquí, además de crear los procedimientos equivalentes a los de la fase anterior, deberán definir aquellos que permitan acceder a un registro o a un campo, así como seleccionar registros concretos que cumplan determinada condición y ordenar los registros por un campo. Dada la dificultad que puede entrañar la construcción de estos procedimientos, el profesor puede mostrar ejemplos de selección y ordenación, para que los alumnos puedan

utilizarlos, pues el objetivo que se persigue, no es tanto la programación de procedimientos complejos, como el advertir su necesidad y utilidad.

- Posteriormente y de forma análoga, se creará un fichero con estructura arborescente. Para ello nos puede servir de ejemplo cualquier clasificación de animales, plantas, paises, árboles genealógicos, etc. Al final de estas actividades, se puede mostrar como ejemplo, el programa "ANIMAL", con el que conseguirán, analizando sus procedimientos, alcanzar un mayor grado de conocimiento de las posibilidades del lenguaje LOGO.

- Para conseguir una mejor utilización de la memoria y una mayor velocidad de acceso a la información, se sugiere la creación de nuevas listas para que actúen como índices de los ficheros anteriores. Con esta actividad, el proceso se complica al tener que relacionar varios ficheros, uno de datos y otros de índices. Apreciarán también la necesidad de utilizar claves para la búsqueda y ordenación del fichero por un campo y entenderán mejor su significado. Como ejercicio, puede plantearse la obtención de listados ordenados por distintos campos, comparando el tiempo empleado utilizando índices y sin utilizarlos.

- La gestión simultánea de dos o más ficheros de una base de datos constituye la fase siguiente. Podrán utilizar ficheros con estructura de tabla, al estilo de las bases de datos relacionales, y definir los procedimientos que efectúan las operaciones de selección, proyección y concatenación.

- Por último, para aquellos alumnos que conozcan en profundidad las posibilidades del lenguaje, se propone la realización de un programa de propósito general que permita la construcción de algunas bases de datos.

APLICACION DE LAS BASES DE DATOS: GEOGRAFIA DE LA POBLACION

INTRODUCCION

El tema elegido para este trabajo ha sido la Demografía, que forma parte de la asignatura de Geografía Humana y Económica que se imparte en Segundo de bup. Dicho tema es muy adecuado para la creación de una base de datos, ya que para su estudio es necesario el constante manejo de series estadísticas, diagramas, etc.

El trabajo que nos hemos propuesto desde el punto de vista del currículum, permite determinar el grado de desarrollo de un pais a través del estudio de su población, estudio que suscita en el alumno el interés y, muchas veces, la perplejidad ante la comprobación de ciertos datos demográficos y todos los problemas que de ellos se derivan, haciéndole conocer mejor el mundo en que vive en la actualidad e, incluso, en el futuro.

OBJETIVOS

Capacitar para:

- Seleccionar los datos relevantes para resolver un problema.
- Analizar datos, extrayendo conclusiones de los mismos.
- Organizar datos de modo sistemático.
- Utilizar bases de datos.
- Resolver problemas a partir de una base de datos (comprobar hipótesis, relacionar datos, extraer conclusiones,...)
- Conocer las características del mundo en el que viven.
- Comprobar el crecimiento de la población y los problemas que puede acarrear.
- Caracterizar, desde el punto de vista demográfico, los paises desarrollados y subdesarrollados:
  a) En su crecimiento natural: tasas de natalidad, mortalidad y crecimiento vegetativo.
  b) En la estructura de su población.
  c) En su porcentaje de población urbana y rural.
- Conocer la situación en que se encuentran otros seres humanos.
- Captar las diferencias, cada vez más profundas, entre los paises desarrollados y subdesarrollados.
- Conseguir que el alumno, indirectamente, se familiarice con la localización geográfica de los distintos paises.
- Acostumbrar al alumno al manejo de series estadísticas con las que compruebe sus conocimientos teóricos.
- Desarrollar su capacidad de relación en el análisis y conclusiones de los diversos datos en los que trabaje.

GEOGRAFIA DE LA POBLACION

INTRODUCCION

Vamos a trabajar con una base de datos de la población del mundo; es decir, se ha introducido en el ordenador la información correspondiente a la población de todos los paises del mundo, clasificándola y organizándola de tal forma que podamos acceder a ella fácilmente y podamos relacionar y seleccionar los datos que nos interesen. El programa que nos permitirá realizar esta gestión y que hemos utilizado para introducir los datos es el Gestor de Bases de Datos del OPEN ACCESS. Este programa permite realizar las operaciones, eligiendo entre las opciones del menú que se presenta en la pantalla y que, a lo largo de la aplicación, iremos reconociendo.

PUESTA EN MARCHA

Para comenzar a trabajar seguiremos los siguientes pasos:

1) Introducir en el buzón superior (A) el disco con el Sistema Operativo (S.O. DOS) y encender el monitor y el ordenador.

2) Introducir la fecha y la hora cuando la pida o pulsar [↵]

3) Cuando aparezca A> , sacar el disco del DOS e introducir el del OPEN ACCESS #1 en A.

4) Pulsar oa y la tecla [↵], volver a pulsar [↵] cuando se detenga pidiendo el disco #1.

5) Pulsar [↵] cuando aparezca la fecha.

6) Pulsar [F10] cuando aparezca encendido el GESTOR DE BASE DE DATOS en la ventana de OPCIONES que se muestra a la derecha. Saldrá un mensaje solicitando el disco #3.

7) Sacar de A el disco #1, e introducir el #2, también en A. El disco con los datos se introduce en el buzón inferior (B). Pulsar [↵] ó [F10] .

8) Aparecerá en la pantalla el MENU de operaciones de la Base de Datos y sobreiluminada, la opción LISTAR.

Estas operaciones hay que realizarlas siempre que queramos trabajar con la base de datos.

EXPLORACION DE LA BASE DE DATOS

La información incluida en esta base de datos está organizada en varios bloques que se llaman FICHEROS. Uno de ellos es el fichero MUNDO, con el que se va a trabajar en primer lugar.

ACTIVIDADES

I) Consulta de registro y campos

Pulsar [F10] cuando esté sobreiluminada la opción LISTAR. Aparecerá una línea destacada en la que se escribirá el nombre del fichero que queremos consultar, en este caso MUNDO.

Al pulsar [F10] aparecerá en pantalla una lista alfabética de paises con sus datos respectivos. Al final de la lista se indica el número de fila en el que está el cursor centelleante y el número de filas que tiene el fichero. A cada fila se le llama REGISTRO.

Con las teclas [↓] y [↑] se puede pasar el cursor de un registro al siguiente o al anterior.

En la parte superior aparece una cabecera con el identificador de la información para cada columna. Cada columna del fichero se denomina CAMPO.

Suele ocurrir que no todos los registros caben en la pantalla. Para poder ver los siguientes se pulsarán simultáneamente las teclas [END] [↓] (salto abajo) y aparecerán los registros siguientes. Para retroceder a los anteriores, pulsar [END] [↑] (salto arriba).

También es frecuente que no aparezcan todos los campos en la pantalla. Para ver los que no aparezcan se utilizan las teclas [←] [→].

Ejercicios:

- Consultar y anotar toda la información que hace referencia a España, Camerún, India, Brasil y Australia.

- Busca en el Atlas la localización de diez paises del listado que no sepas situar y anota sus nombres.

- Anota los nombres de los campos de la base de datos que aparecen en la cabecera de la lista.

NOTA: Si durante el desarrollo de éste o cualquier otro ejercicio observas algún dato con el que no estés de acuerdo o que te sorprenda, anótalo y coméntalo con el profesor.

II) Utilización de máscaras para los registros.

Sitúa el cursos en el registro relativo a España, anota los dos o tres paises anteriores y posteriores. Pulsa [F6] para cambiar a otra forma de presentación. El programa preguntará si la máscara es MUNDO.SMK, pulsar [F10] para confirmarlo; aparecerán los datos del registro correspondiente a España con un formato ó máscara previamente diseñado.

Pulsando la tecla [PgDn] se va obteniendo la información de los registros posteriores y con [PgUp] la de los registros anteriores. De esta forma se puede recorrer el fichero registro a registro.

Si alguno de los campos de un registro está vacío o es erróneo, puedes modificarlo; bastará con situar el cursos en dicho campo, usando las teclas [↓] ó [↑] , escribir el contenido correcto y pulsar [F10] para que se guarde esa información. Si no quieres modificarlo, pulsa [Esc] .

Para pasar de nuevo al listado general, pulsa la tecla [F2].

Para finalizar con este listado y volver a las opciones del Menú, pulsar [F2] de nuevo.

Ejercicios:

- Sitúa el cursor en Portugal y escribe en el campo Area: EUROPA DEL SUR, y en el campo Zona Climática: TEMPLADA.

  Comprueba que los datos han sido almacenados pasando al siguiente registro y retrocediendo.

- Introduce la Zona Climática de ISLANDIA.

- Busca en el Atlas los datos de las Islas Cook, Islas del Canal, Islas del Pacífico, Islas Feroes, Islas Salomon e Islas Vírgenes e introduce los datos que faltan.

  Comprueba que los datos han sido almacenados.

III) Consultas en otros ficheros.

Vamos ahora a consultar otro de los ficheros de la Base de Datos en el que hay datos de población de los años 1984, 2000 y 2025; se llama POBLAC84

Si hemos estado trabajando con el fichero MUNDO, en la parte superior del menú figura (B:MUNDO.SMK) para indicar que ese fichero está activo (abierto) y que las operaciones que efectuemos se harán sobre él. Si ahora queremos usar otro fichero hay que desactivar (cerrar) el fichero MUNDO. Esto se hace pulsando la tecla [F2] para acceder al menú y [N] de [Nuevo] que es una de las opciones del menú. Aparecerá en la parte superior del menú: -ningún fichero activo-. Pulsando [L] de [Listar] volverá a aparecer la línea destacada en la que escribiremos POBLAC y [F10] para ejecutar. Aparece el listado de este fichero.

Ejercicios:

- Recorrer el fichero, por registros y campos.
- Anotar los nombres de los campos de este fichero.
- Anotar los datos relativos a España.
- Con la máscara POBLAC84.SMK, consultar y anotar los datos referentes a DINAMARCA, CHINA, CONGO, NICARAGUA y ESTADOS UNIDOS DE AMERICA. Escribe lo que te sugiere la comparación de los distintos datos.

Hay un tercer fichero en esta base de datos que contiene información sobre distintas tasas relativas a la población. Se llama TASAS84 y su máscara TASAS84.SMK.

Ejercicios:

- Recorrer el fichero TASAS84 por registros y campos.
- Anotar los nombres de los campos.
- Anotar los datos relativos a España utilizando la máscara.
- Buscar en el libro de texto el significado de cada uno de los campos que no conozcas.
- Anotar los datos relativos a SUIZA, PORTUGAL, ARGENTINA, CANADA, GABON, CUBA. Analiza los datos, comparando unos y otros, y escribe tus conclusiones.

Existe aún otro fichero con porcentajes de población según distintos sectores, del año 1982, que se denomina SECTORES. Accede a él y realiza los mismos ejercicios que con el fichero anterior.

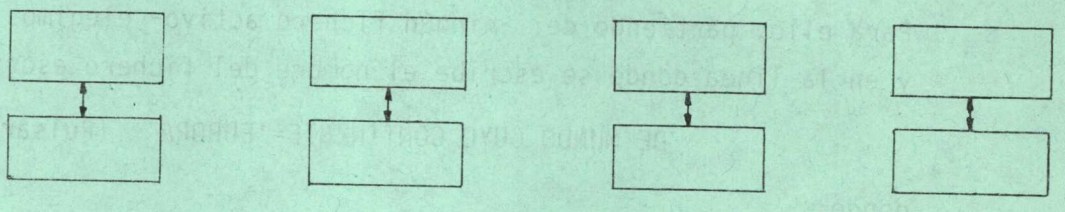
Estos cuatro ficheros componen la parte fundamental de la base de datos. No obstante, hay otros ficheros que también se utilizarán en la aplicación y que se indicarán en el momento oportuno.

IV) Definición en gran grupo de la estructura de la Base de Datos.

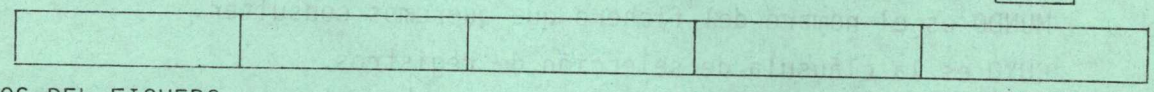
- Ficheros de datos utilizados y sus nombres.
- Ficheros de máscara y sus nombres.
- Campos de cada uno de los ficheros: número, nombre, significado y unidad de representación.
- Número de registros de cada fichero.
- ¿Has reparado que los listados se han obtenido por orden alfabético?

FICHEROS

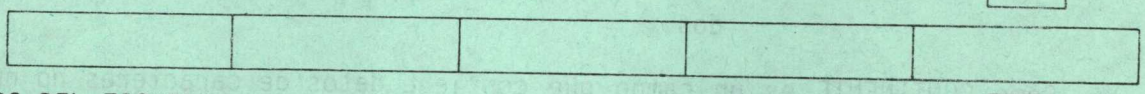
MASCARAS



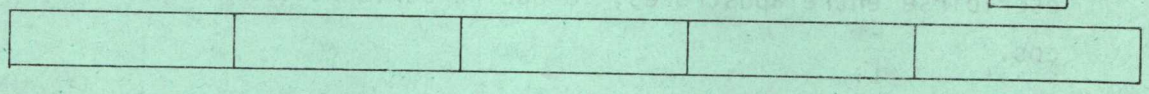
CAMPOS DEL FICHERO: Nº de registros:



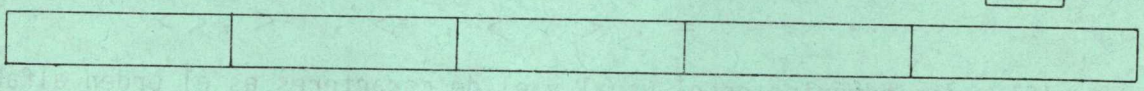
CAMPOS DEL FICHERO: Nº de registros:



CAMPOS DEL FICHERO: Nº de registros:



CAMPOS DEL FICHERO: Nº de registros:



V) Selección de registros en un fichero.

A lo largo de ésta y las siguientes actividades, se utilizará una de las características fundamentales de las Bases de Datos, sin la cual la recopilación de información carecería de utilidad práctica. Es la posibilidad de elegir de entre todos los datos aquéllos que cumplen alguna condición específica y poderlos destacar del resto. Veremos en primer lugar la selección de registros.

En el fichero MUNDO vamos a seleccionar los registros de los países de Europa. Para ello, partiendo de: -ningún fichero activo- elegimos la opción [Listar] y en la línea donde se escribe el nombre del fichero escribimos:

DE MUNDO CUYO CONTINENTE='EUROPA' (Pulsar [F10])

donde:

DE es la cláusula que precede al nombre del fichero.
MUNDO es el nombre del fichero que queremos consultar.
CUYO es la cláusula de selección de registros.
CONTINENTE='EUROPA' es la condición que deben cumplir los registros seleccionados.

Como CONTINENTE es un campo que contiene datos de caracteres no numéricos debe escribirse entre apóstrofes, lo que no sería necesario en caso de datos numéricos.

Las relaciones que pueden utilizarse son:

= , < , > , => , =< , <>

(El orden numérico es el usual y el de caracteres es el orden alfabético).

Para seleccionar otros registros, hay que cerrar el fichero actual con la opción [Nuevo] y [Listar] la nueva selección.

Ejercicios:

- Anota los cinco países con mayor y menor superficie de Europa.
- Anota los paises del área del CARIBE.
- Anota los paises del mundo con superficie mayor de 500.000 Km^2.
- Anota los paises con población mayor de cien millones de habitantes.
- Anota los paises con tasa de crecimiento negativa.

Pueden seleccionarse también combinando dos o más condiciones con las cláusulas

Y para que se seleccionen los registros que cumplan las dos condiciones simultáneamente.

O para que se seleccionen aquéllos que cumplan alguna de las dos.

NO para que se seleccionen los que no cumplan la condición.

Ejemplos:

. DE MUNDO CUYO CONTINENTE='ASIA' Y SUPERFICIE > 500.000 [F10]
Dará como resultado la lista de todos aquellos paises de Asia con superficie mayor que 500.000 Km^2.

. DE MUNDO CUYO CONTINENTE='EUROPA' O CONTINENTE='AMERICA' [F10]
Dará la lista de todos los registros de los paises europeos o americanos.

. DE MUNDO CUYO NO (CONTINENTE='EUROPA') [F10]
Dará la lista de todos los registros de los paises no europeos.

Ejercicios:

- Anotar todos los paises con población en el año 1984 menor que 40.000.000 y en el año 2000 mayor que 40.000.000.
- Anotar todos los paises con densidad menor que 50 hab/Km^2 y población mayor de 50.000.000.
- Anotar todos los paises con densidad menor que 50 hab/Km^2 , ó que tengan población superior a 50 millones de habitantes.
- Anotar todos los paises de ASIA que no pertenezcan al SURESTE ASIATICO.
- Anotar todos los paises de AMERICA con superficie menor de 100.000 Km^2.
- Consulta libre.

VI) Selección de campos de un fichero.

Con frecuencia no se necesita recoger toda la información del registro, sino la de uno o algunos campos. Para seleccionar los campos PAIS y SUPERFICIE, escribiremos:

DE MUNDO ELIGE PAIS,SUPERFICIE

Al ejecutarlo obtendremos la lista de todos los paises y sus superficies.

Naturalmente, pueden seleccionarse a la vez registro y campos. Así, por ejemplo al ejecutar:

DE MUNDO ELIGE PAIS,SUPERFICIE (Pulsar [↵])
CUYO CONTINENTE='AMERICA' Y SUPERFICIE <100.000 (Pulsar [↵])
Y CLIMA ='TROPICAL' (Pulsar [F10])

obtenemos todos los paises de América de la zona climática tropical con superficie menor que 100.000 km^2.

Ejercicios:

- Seleccionar y anotar todos los paises y su superficie que sean de un clima templado y tengan superficie menor que 80.000 km^2 .
- Seleccionar los paises y su tasa de mortalidad infantil cuyo crecimiento sea menor que el 2%.
- ¿Qué lugar ocupa España en cuanto a superficie dentro de los paises europeos?
- ¿Qué lugar ocupa España en cuanto a superficie dentro de los paises de la Comunidad Económica Europea?
- ¿Qué paises tienen menor índice de mortalidad infantil que España?

VII) Ordenación y clasificación de registros por campo.

Las últimas preguntas del ejercicio anterior pueden contestarse más rápidamente si a las cláusulas de selección se añade la cláusula de ordenación:

ORDEN: Cláusula que indica que los registros se ordenarán por el campo o campos especificados. Caso de no especificarse esta cláusula, el orden se hace por el primer campo, como se ha visto hasta ahora. Los listados se obtenían por orden alfabético de paises.

Ejemplos:

```
DE MUNDO ELIGE PAIS,SUPERFICIE
CUYO CONTINENTE='EUROPA'
ORDEN SUPERFICIE
```

dará una lista de los paises de Europa ordenados por su superficie de mayor a menor.

```
DE MUNDO
ORDEN CONTINENTE,AREA,PAIS
```

dará una lista de los registros ordenados por continentes, dentro de los que estén en un continente por áreas y dentro de los que pertenezcan a una misma área, por orden alfabético de paises.

Ejercicios:

- Responder a las tres últimas preguntas del ejercicio anterior utilizando la cláusula ORDEN.
- Anotar los diez paises con mayor población en el año 1984.
- Anotar los diez paises con mayor población prevista en el año 2000.
- Analizar los resultados.

VIII) Consultas en varios ficheros concatenados.

La gestión de los ficheros no sería útil si no pudieran relacionarse campos de distintos ficheros. Con lo visto hasta ahora, sería muy difícil responder, por ejemplo, a la siguiente pregunta:

"¿Qué paises de Asia o Africa tienen una población mayor que cuarenta millones y tasa de crecimiento superior al 3%?

Ya que requiere hacer consultas a tres ficheros, pues el campo CONTINENTE está en el fichero MUNDO, el campo POBLACION en POBLAC y el campo CRECIMIENTO en el fichero TASAS84.

La pregunta se puede responder ya que pueden utilizarse todos los ficheros simultáneamente, pero debe incluirse en la cláusula de selección tanto los nombres de los ficheros como de los campos que relacionan a un fichero con otro.

La respuesta se obtendría escribiendo:

```
DE MUNDO,POBLAC84,TASAS84
ELIGE PAIS,POBLACION,CRECIMIENT
CUYO MUNDO.PAIS=POBLAC84.PAIS Y POBLAC84.PAIS=TASAS84.PAIS
Y POBLACION>40.000.000 Y CRECIMIENT>3 Y (CONTINENTE='ASIA' O
CONTINENTE='AFRICA')
```

en menos de un minuto se obtiene la lista de los 9 registros seleccionados: Bangladesh, Corea, República Democrática Popular, Egipto, Filipinas, Irán, Nigeria, Pakistán, Thailandia, Turquía.

Ejercicios:

- Obtener una lista ordenada por continentes, áreas y paises de aquellos paises cuya población actual sea mayor que 30 millones, cuyo índice de fecundidad sea mayor que 4 y su población rural menor del 50%.
- Obtener una lista ordenada por continente y tasa de natalidad de todos los paises que no estén en la relación anterior.

CRECIMIENTO Y DISTRIBUCION DE LA POBLACION

La población del mundo está creciendo de forma desbordante. Vamos a comprobarlo a través de los datos recogidos en el ordenador.

También comprobamos la desigualdad de este crecimiento y las consecuencias que estos hechos producen en la Tierra.

ACTIVIDADES

I) Con los datos de la población mundial (fichero PMUNDIAL), prepara un gráfico e indica su ritmo de crecimiento hasta el año 2000.

Explica cuáles son las causas del diferente ritmo de crecimiento y sus consecuencias.

Compara tu gráfico con el obtenido en el ordenador, ejecutando con la tecla HOME el programa PROCESO desde el menú principal de la base de datos.

II) Lista por continentes y áreas los paises y su población en el año 1984.

Suma los datos primero por áreas y después por continentes.

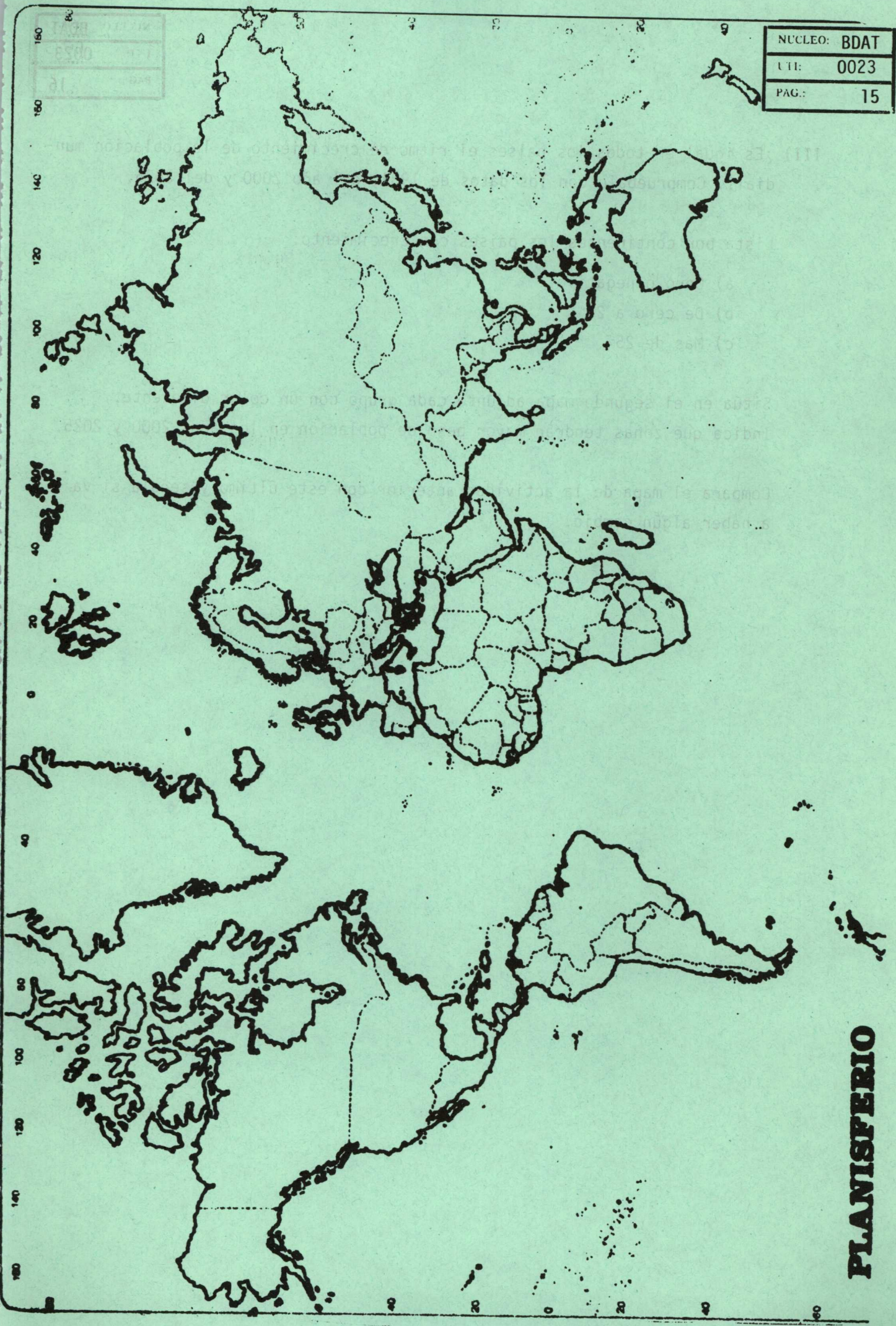
Colorea en el mapa adjunto las distintas áreas según su mayor o menor población.

Utiliza un tono progresivamente más intenso según vaya aumentando la población.

Anota las áreas y su población en orden decreciente de población.

NOTA.- Ejecutar PROCESO2 después de pulsar [HOME] desde el menú principal de la base de datos.

NUCLEO: BDAT
UTI: 0023
PAG.: 15



PLANISFERIO

III) ¿Es igual en todos los paises el ritmo de crecimiento de la población mundial?. Compruébalo con los datos de 1984, del año 2000 y del 2025.

Lista por continentes los paises con crecimiento:

a) Cero o negativo.
b) De cero a 25%
c) Más de 25%

Sitúa en el segundo mapa adjunto cada grupo con un color diferente. Indica qué zonas tendrán mayor peso de población en los años 2000 y 2025.

Compara el mapa de la actividad anterior con este último y señala si va a haber algún cambio.



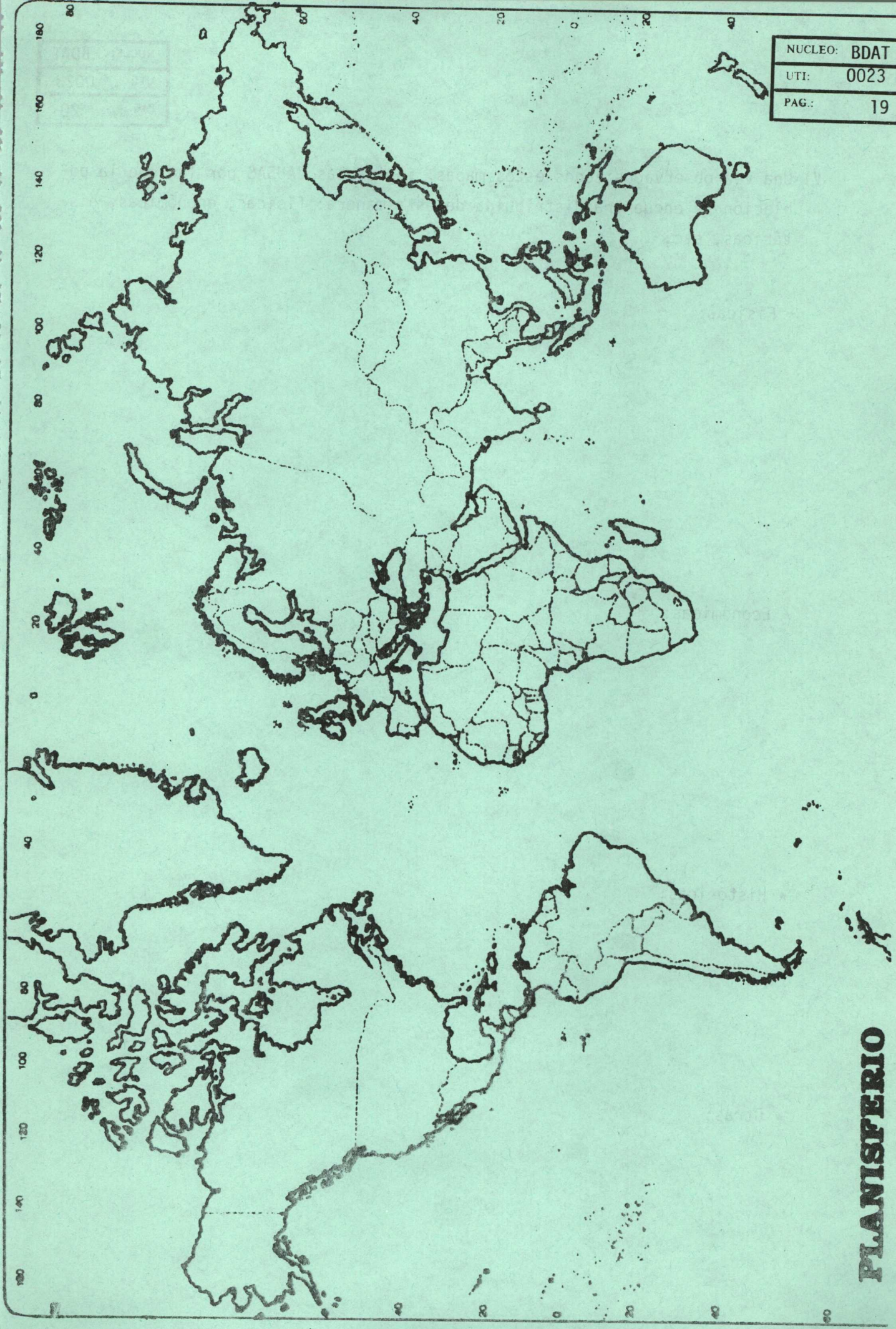
PLANISFERIO

NUCLEO: BDAT
UTI: 0023
PAG.: 17

IV) Haz una lista por continentes y áreas de los paises y su densidad relativa en el año 1984.

Colorea en el tercer mapa las distintas áreas según la mayor o menor densidad y compáralo con el mapa de población del año 1984 (actividad II).

Escribe tus conclusiones.



PLANISFERIO

V) Una vez observados todos estos mapas, indica las CAUSAS por las que la población se encuentra distribuida de esta manera: físicas, económicas, históricas, etc.

* Físicas:

* Económicas:

* Históricas:

* Otras:

MOVIMIENTOS NATURALES DE LA POBLACION

Hemos estudiado en los datos anteriores la variación que existe en las cifras de población. Vamos a analizar los elementos que determinan esa variación:

- Naturales: Natalidad y Mortalidad.
- Migraciones.

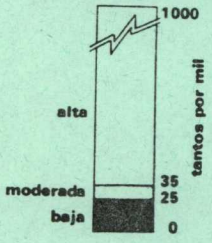
Trataremos, en este trabajo, de los movimientos naturales.

Como las cifras absolutas son de difícil comparación, utilizaremos una serie de tasas: Tasa de Natalidad, Tasa de Mortalidad, Crecimiento Vegetativo, Tasa de Fecundidad y Tasa de Mortalidad Infantil.

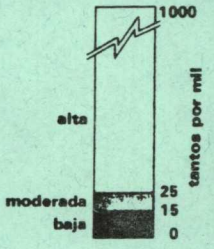
ACTIVIDADES

I) Observa las siguientes escalas:

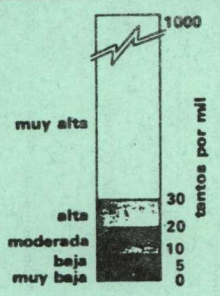
ESCALA DE VALORES TN



ESCALA DE VALORES T.M



TASAS CONSIDERADAS C.V.



Busca cuatro paises de cada una de las siguientes características:

- TN y TM altas.
- TN y TM bajas.
- TN alta y TM baja.
- TN baja y TM alta.
- TN media y TM baja.
- TN alta y TM media

Rellena las cuatro primeras columnas de la tabla de la página siguiente.

	PAIS	TN	TM	CU	GRADO DE DESARROLLO	REGIMEN DEMOGRAFICO
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

Lee con atención los siguientes textos:

a)

La principal causa del incremento masivo de la población reside en la disminución de los índices de mortalidad (...). Fue durante la II Guerra Mundial cuando la «medicina sin eficacia» fue reemplazada bruscamente por técnicas médicas incomparablemente más eficaces. La «revolución sanitaria» sobrevenida a mediados del siglo XX se caracterizó por el empleo en masa de productos químicos poco costosos, medicamentos o sustancias obtenidas por industrias altamente evolucionadas (...).

Los países del Tercer Mundo han sido, sobre todo, objeto de campañas sanitarias masivas, esencialmente orientadas contra las endemias infecciosas perceptibles de propagarse rápidamente a lo lejos y capaces, por tanto, de amenazar la salud de otros países (...).

En Europa occidental, la baja de la mortalidad, mucho más precoz que en los países del Tercer Mundo, fue en parte el resultado de mejoras de las condiciones sanitarias y, sobre todo, consecuencia de la elevación del nivel de vida (mejor alimentación, mejores viviendas). En el Tercer Mundo, la disminución de la mortalidad no es una señal de mejora de las condiciones de existencia, sino la prueba de la notable eficacia de los medios médicos modernos que actúan a despecho de condiciones económicas y sociales desastrosas (...).

Con unos índices de mortalidad inferiores al 8-9 por 1.000, numerosos países subdesarrollados presentan una mortalidad inferior a la de algunos países desarrollados (...). Sin embargo, esta sorprendente comparación está sensiblemente falseada por el hecho de que los países del Tercer Mundo cuentan con muchísimos jóvenes (más del 50 por 100) y muy pocos ancianos, mientras que los países europeos tiene menos jóvenes y muchas más personas de edad avanzada.

LACOSTE, V.: Geografía del Subdesarrollo. Pags. 110-114. Recogido en Akal: "Trabajos prácticos de Geografía 2º de BUP", pag. 18.

b)

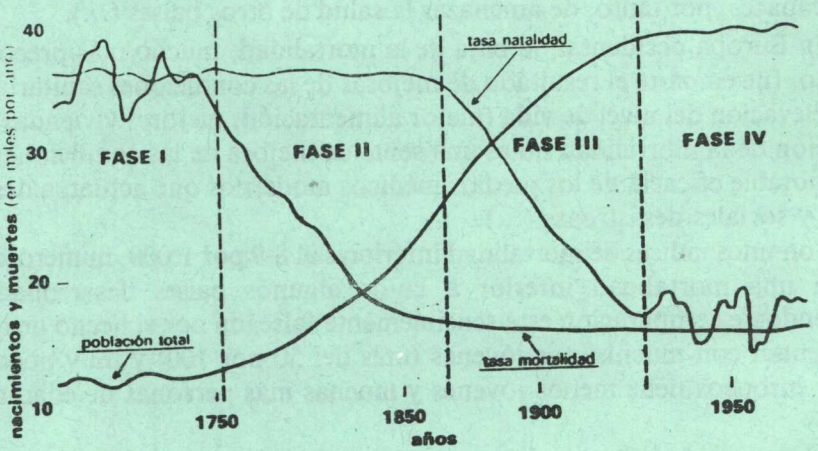
Otro método (de prevención de nacimientos) consiste en modificar previamente las condiciones económicas y culturales de las familias, particularmente las de las capas populares. El nivel cultural parece aún más importante que el económico (...). Cuando una mujer recibe instrucción sobre los males que amenazan a su hijo, se ve instintivamente forzada a procurarle todos los cuidados a su alcance; por esto mismo, comenzará a temer más que antes la llegada de un nuevo hijo más joven y que obligaría, en cierto modo, a sacrificar al precedente (...). De ahí que los conocimientos en puericultura tengan un efecto favorable suplementario, pues da a los padres una consciencia del valor de su hijo. La palabra «consciencia» no es, quizá la más apropiada, pues debemos reconocer que, por lo general, estos procesos se desarrollan de modo inconsciente. Mas esta toma en consideración de lo que es y vale una vida humana, esta importancia concedida a un ser que disfrutaba anteriormente de tan poca, contribuye al necesario cambio de mentalidad.

SAUVY: El hambre, la guerra y el control de natalidad. Pags. 174-175. Recogido en Akal: "Trabajos prácticos de Geografía 2º de BUP", pag. 23.

Indica las causas de las diferentes tasas de Natalidad y Mortalidad que presentan estos paises. Escribe a continuación, en la tabla, el grado de desarrollo de estos paises (pais desarrollado, pais en vías de desarrollo y pais subdesarrollado).

Observa ahora el siguiente modelo de la evolución demográfica del occidente europeo:

FASE I: Régimen demográfico antiguo.
FASE II y III: Revolución demográfica.
FASE IV: Régimen demográfico moderno.



Conforme a este modelo, escribe en la tabla el régimen demográfico en que se encuentran estos paises.

¿Un bajo crecimiento vegetativo corresponde siempre a un pais desarrollado? Explica por qué.

Según lo observado anteriormente, explica la relación que existe entre el régimen demográfico y el grado de desarrollo.

II) Estudia, junto con tu grupo, la tasa de Mortalidad Infantil del continente que os corresponda. Relaciona la tasa de Mortalidad Infantil con la tasa de Mortalidad general de ese continente y saca conclusiones.

Compara con el resto de los grupos los resultados obtenidos y saca conclusiones.

¿Crees que la tasa de Mortalidad Infantil podría ser un índice significativo del grado de desarrollo? ¿Por qué?

NUCLEO: BDAT
UTI: 0023
PAG.: 26

III) Analiza el índice de Fecundidad por continentes y relaciónalo con el resto de las tasas estudiadas hasta ahora.

Explica, a partir de este análisis, qué tipo de población predominará:

- población joven (0-15 años)
- población adulta (15-64 años)
- población vieja (+ de 64 años)

ESTRUCTURA DE LA POBLACION POR EDAD Y SEXO Y POBLACION ACTIVA

El estudio de la estructura de la población nos ayuda a completar y matizar las conclusiones a las que hemos llegado tras el análisis del crecimiento y distribución de la población.

La estructura de la población se puede estudiar desde distintos puntos de vista. Los más utilizados son los que comprenden la división de la población por edad y sexo y según la población activa.

ACTIVIDADES

I) Recoge e introduce en el ordenador los datos relativos a la distribución de la población por edad y sexo de los distintos paises.

Realiza distintas listas de los paises ordenados según los distintos grupos de edad y sexo.

Haz los siguientes ejercicios:

a) Anota los diez primeros y los diez últimos de cada lista.
b) Sitúalos por continentes.
c) Relaciónalo con el crecimiento vegetativo, indicando su grado de desarrollo.

¿Qué problemas se les plantearán a estos paises.

¿Son iguales para los distintos tipos según su nivel de desarrollo?

II) Escribe la población activa de los paises recogidos en la tabla utilizada anteriormente.

Plantea una regla general a partir de esos datos relacionando el nivel de desarrollo con los sectores predominantes. Justifica la contestación.

GRADO DE DESARROLLO	SECTORES PREDOMINANTES
Pais muy desarrollado	Sector primario
Pais desarrollado	Sector secundario
Pais en vías de desarrollo	Sector Terciario
Pais subdesarrollado	

NUCLEO: BDAT
UTI: 0023
PAG.: 29

III) Comprueba con los datos del ordenador si es cierta la siguiente afirmación:

"Todos los paises con un sector terciario mayor que el secundario son paises muy desarrollados"

Explica las conclusiones que obtengas ayudándote del siguiente texto:

Si analizamos la población activa de la mayoría de países de América Latina, podemos observar: 1) incremento del sector terciario; 2) débil incremento del secundario; 3) fuerte corriente migratoria campo-ciudad. Hay que señalar que el elevado porcentaje de población dedicada al sector terciario no presupone una demanda real de empleo, sino que una gran parte de este sector corresponde a actividades no especificadas: vendedores ambulantes, revendedores, limpiabotas, servicios domésticos, etcétera.

Recogido en Akal: "Trabajos prácticos de Geografía 2º de BUP", pág. 168.

NUCLEO: BDAT
UTI: 0023
PAG.: 30

IV) Haz una lista de paises según su sector Terciario en orden decreciente.

¿A qué se debe el alto porcentaje del sector terciario en algunos paises?

Cita profesiones que incluirías en este sector.

GRADO DE URBANIZACION

El crecimiento de las ciudades está relacionado con el desarrollo económico. Sin embargo, el fenómeno urbano en la actualidad no presenta esta característica.

Vamos a comprobarlo a partir de los datos de población urbana.

ACTIVIDADES

I) Relaciona el grado de urbanización con los porcentajes de población activa:

SECTOR PREDOMINANTE	GRADO DE URBANIZACION
Primario	
Secundario	
Terciario	

Realiza gráficos con paises característicos.

II) Añadir los campos CAPITAL y ciudades importantes al fichero MUNDO y crear un nuevo fichero de dichas ciudades en distintos años, así como el índice de crecimiento y estimaciones para años venideros.

Utilizando este fichero, indica a qué tipo de paises pertenecen las ciudades más pobladas del mundo en el pasado, en la actualidad y en el futuro.

Realiza un gráfico con las 15 ciudades más habitadas en los años 1984, 2000 y 2025.

Con estos datos y el siguiente texto contesta a las preguntas propuestas a continuación.

A este respecto, en las naciones no industriales —que constituyen todavía la mayoría de las naciones del mundo— no hay miedo de que se repita la historia. En el siglo XIX y principios del XX, el crecimiento de las ciudades se debió al progreso económico, y contribuyó, a su vez, a dicho progreso. Las ciudades absorbieron la mano de obra sobrante del campo, utilizándola en la producción de mercancías y en los servicios, producción que, a su vez, redundó en beneficio de la modernización de la agricultura. Pero en la actualidad, en los países subdesarrollados, como en las naciones avanzadas, el crecimiento de las ciudades está desligado del desarrollo económico y, por ende, de la migración campo-ciudad. Ello deriva, en grado considerable, del crecimiento de la población en general; crecimiento que no tiene precedentes en la historia de los países no industriales y que se debe a las modernas técnicas sanitarias combinadas con los elevados coeficientes de natalidad.

La aceleración del crecimiento de la población mundial duplica hoy la del período anterior a 1940, y los países más avanzados han tenido que ceder el puesto a los más retrasados en lo que respecta a la rapidez de ese crecimiento. En estos últimos países el crecimiento es tan acelerado que resulta virtualmente imposible el ir creando a igual ritmo los servicios urbanos necesarios para el cuidado de las ingentes e interminables legiones de niños, así como de los innumerables campesinos que van acrecentando sin cesar las masas urbanas. Y es todavía más arduo el problema del desarrollo de la expansión de las tierras de labor con la suficiente celeridad y de la movilización, a la vez, del capital necesario para dar alojamiento a la sin cesar creciente población campesina. No se trata aquí de un problema de urbanización ni de migración campo-ciudad, sino de un problema de multiplicación humana. Problema nuevo, tanto en lo que refiere a su envergadura como en lo tocante a su ajuste. Con dolorosa expresión, podríamos calificarlo de crecimiento urbano desbocado.

A medida que la población humana va aumentando,

NUCLEO: BDAT
UTI: 0023
PAG.: 33

van creciendo también las ciudades, independientemente de si aumenta o de si disminuye la urbanización. Lo cual quiere decir que algunas ciudades llegarán a alcanzar un perímetro tan vasto que, a su lado, las metrópolis del siglo XIX se nos antojarán villorrios. Si el perímetro urbanizado de Nueva York continuara ensanchándose sólo al ritmo del crecimiento de la población de la nación (de acuerdo con las proyecciones medias de esta última elaboradas por la Oficina del Censo), para 1985 alcanzaría la cifra de 21 millones de habitantes y de 30 millones para el 2010. Yo he calculado que si la población de la India creciera de acuerdo con lo indicado por las proyecciones de las Naciones Unidas, su mayor ciudad tendría en el año 2000 de 36 a 66 millones de habitantes.

DAVIS, K: La urbanización de la población humana", pags. 33-34.

¿Qué relación hay entre población urbana y el grado de desarrollo de los paises a lo largo del tiempo?

¿Van a encontrar problemas los paises subdesarrollados ante el aumento de la población de sus ciudades?

Razona tus respuestas.

NÚCLEO: BDAT
UFI: 0023
PAG.: 34

CONCLUSIONES

1) Con arreglo a los diversos aspectos estudiados, caracteriza un pais desarrollado, un pais subdesarrollado y un pais en vías de desarrollo.

   Señala claramente las diferencias entre ellos.

NUCLEO: BDAT
UTE: 0023
PAG.: 35

2) Analiza España desde los distintos puntos de vista que hemos ido tratando.

NÚCLEO: BDAT
UTE: 0023
PAG 36

3) Sobre esta misma base de datos, plantea nuevos proyectos. Por ejemplo: Relación entre la zona climática y otros índices como crecimiento vegetativo, grado de urbanización, etc.

CREACION Y DISENO DE BASES DE DATOS

Se citan a continuación varias posibles aplicaciones que podrían realizarse dentro de la asignatura de Geografía.

1) Sobre la población de Alcalá de Henares, recogiendo datos sobre:

   a) Población total desde 1900.
   b) Número de inmigrantes desde 1900.
   c) Población del censo de 1981 por edades y sexo.
   d) Distribución de la población por distritos.
   e) Evolución de las Tasas de Natalidad, Mortalidad y Crecimiento Vegetativo.
   f) Análisis de la población escolar.
   g) Distribución de la población por sectores productivos.
   h) Cifras de parados.
   ...

2) Demografía española:

   a) Población total española de los siglo XVIII, XIX y XX.
   b) Evolución de las Tasas de Natalidad, Mortalidad y Crecimiento Vegetativo en estos mismos siglos.
   c) Momento de comienzo en España de los distintos ciclos demográficos.
   d) Análisis de población por edad y sexo de los tres últimos censos.
   e) Análisis de la evolución de la población activa y de su distribución por sectores.
   f) Evolución de la población urbana y rural en los últimos dos siglos y previsiones.
   g) Distribución de la población española por comunidades.

3) Ampliación de la base de datos de Demografía con los índices económicos:

   a) Producción de acero.
   b) Consumo de energía.
   c) Producción de ácido sulfúrico.
   d) Producción y consumo de petróleo.
   e) Coches, teléfonos, televisores, etc.

NUCLEO: BDAT
UTI: 0023
PAG: 38

4) Climatología de los distintos paises y/o ciudades del mundo con la construcción gráfica de los correspondientes climodiagramas, que incluyen temperaturas y precipitaciones en los distintos meses del año.

5) Geografía Física de España o de distintos paises

BASE DE DATOS Y GRAFICOS EN EL OPEN ACCESS

Una de las características fundamentales de un paquete integrado como OPEN ACCES es la conexión entre las distintas opciones, en este caso vamos a centrarnos fundamentalmente en la transmisión de datos del Gestor para generar gráficos.

Una vez que se ha accedido al **GESTOR DE BASE DE DATOS**, ejecutando la orden

DE...ELIGE...CUYO...ORDEN

se dispone de un fichero virtual formado por los ficheros, registros y campos seleccionados. Para realizar un gráfico que corresponda a esos datos, se pasa el menú principal y se ejecuta **CONTEXTO** que transfiere el fichero virtual a la opción elegida. Optando por **GRAFICOS** el fichero es leído y tras un breve espacio de tiempo en que muestra sucesiva y automáticamente:

PAIS	PRIMARIO	SECUNDARIO	TERCIARIO	RURAL
YEMEN,REP. ARABE DEL	75.00	11.00	14.00	81.00
BOTSWANA	4.93	32.14	62.94	82.00
AFGANISTAN	78.00	8.00	13.00	82.00
MOZAMBIQUE	66.00	18.00	16.00	82.00
KENYA	78.00	10.00	12.00	84.00
NIGER	91.00	3.00	6.00	84.00
ETIOPIA	80.00	7.00	13.00	84.00
TAILANDIA	76.00	9.00	15.00	85.00
LAOS	75.00	6.00	19.00	85.00
PAPUA Y NUEVA GUINEA	82.00	8.00	10.00	86.00
TANZANIA	83.00	6.00	11.00	86.00

Moviendo 21 registros
1>.....

Extender Diseñar Crear Modificar Formato Importar Exportar
Ficheros Opciones Contexto
<ejec> <no ejec> otro menú: <cambiar>

y el menú de la opción gráficos mientras trabaja:

```
                    Open Access  -  Gráficos
          (C) 1984 Software Products International Inc.


                              Trabajando

        Menú de Operaciones de Gráficos
Cargar Almacenar Niveles Importar Vista Impr/Diaposit Gráfico Opciones
<ejec> <no ejec> <flechas> <cambiar> <ayuda> <calc>
```

termina presentando la **PANTALLA PRINCIPAL DE GRAFICOS**

```
Nombre
Tipo              Superpuesto   Ventanas   Tres-D   Sencillo
Total Niveles      <1..30>20
Total Posiciones   <1..30> 3
Número Nivel Actual          1
 ombre Nivel Actual     ISLANDIA
Tipo Nivel Actual       Barras   Líneas   Pastel
                Títulos en el Gráfico                    Color          Fondo
Arriba                                                                    0
Lado                                                                      0
Abajo                                                                     0
Máximo deseado                                    Dato Máximo    61212000,0000
Mínimo deseado                                    Dato Mínimo      240000,0000
División de Ejes   <1..10> 10

Paleta de color    <0..2>   2
Color de Pantalla  <0..15>  0

 <ejec> <no ejec> <flechas> <tecl mov> <menú> <pintar> <ret> <ayuda> <calc>
```

Desde esta pantalla que se presenta inicializada por los valores del fichero transferido, se puede obtener un gráfico pulsando F7

Esta pantalla asume por defecto ciertas opciones, las cuales pueden modificarse desde teclado para obtener, con los mismos datos, gráficos variados.

PANTALLA PRINCIPAL DE GRAFICOS

LOS DATOS.

TOTAL NIVELES: Indica el número de registros transferidos (1....30)

TOTAL POSICIONES: Indica el número de campos numéricos transferidos (1...30)

NUMERO NIVEL ACTUAL: Puntero que indica el registro a dibujar.

NOMBRE NIVEL ACTUAL: Contenido de un campo alfabético del mismo registro y que haya sido transferido.

EL FORMATO

Pueden incluirse textos breves en la cabecera, en el margen izquierdo y al pie de pantalla.

MAXIMO Y MINIMO DESEADO: Amplía o restringe el dominio de representación en pantalla. Por defecto toma el dato máximo redondeado y cero como mínimo.

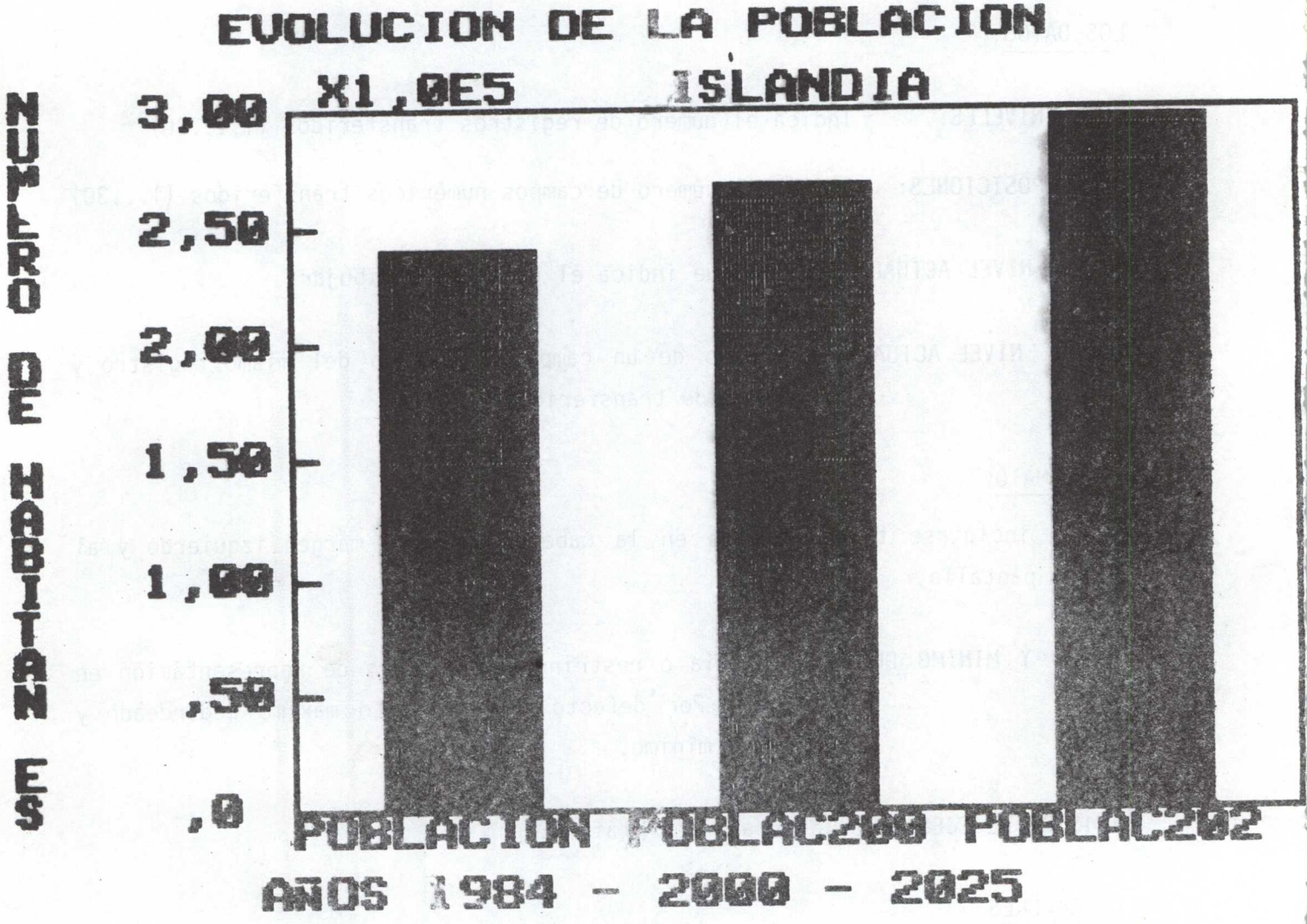
DIVISION DE EJES: Fija la escala del gráfico.

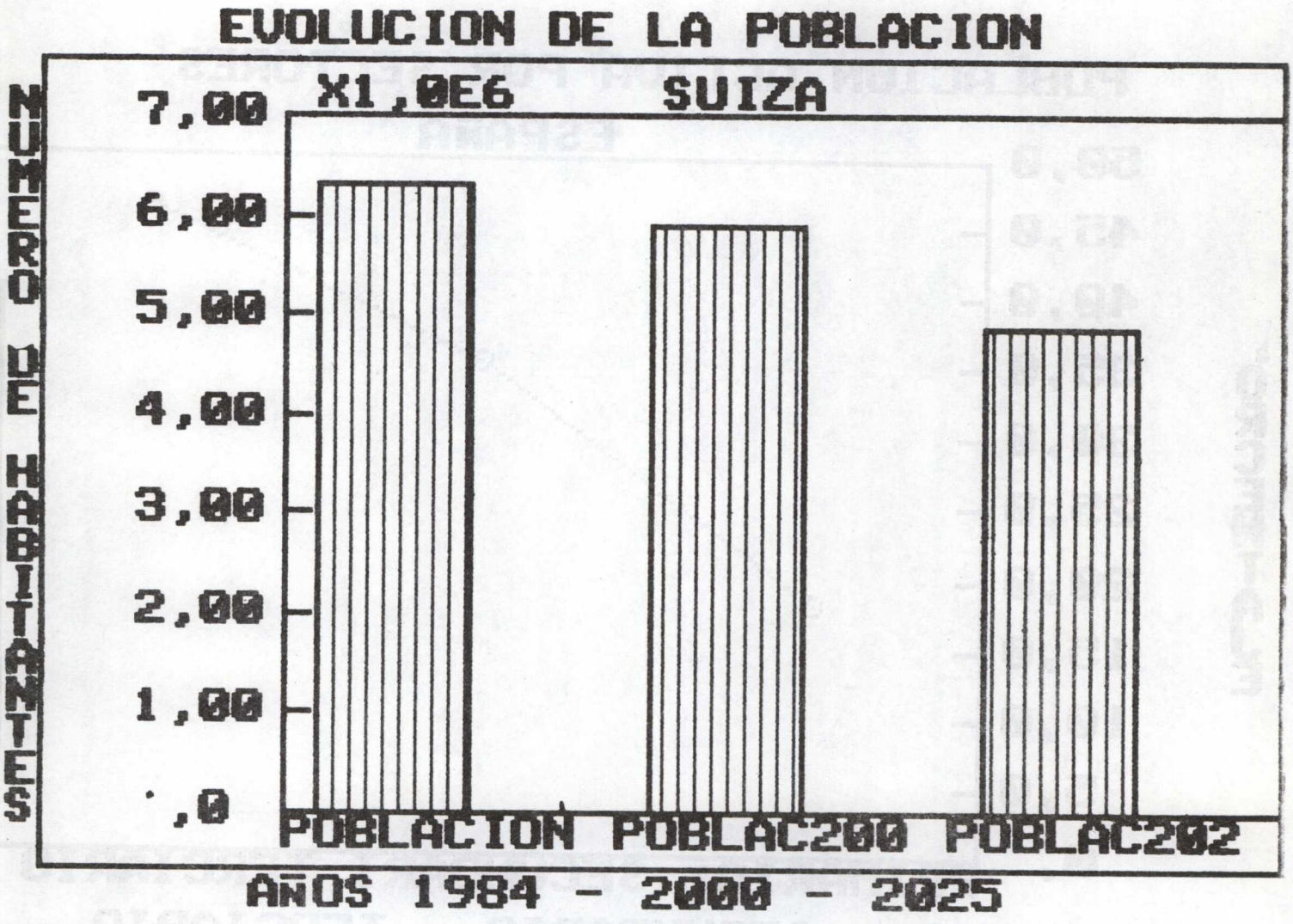
LOS COLORES

Dispone de opciones para modificar los colores de los textos de los márgenes, del gráfico, del fondo, etc.

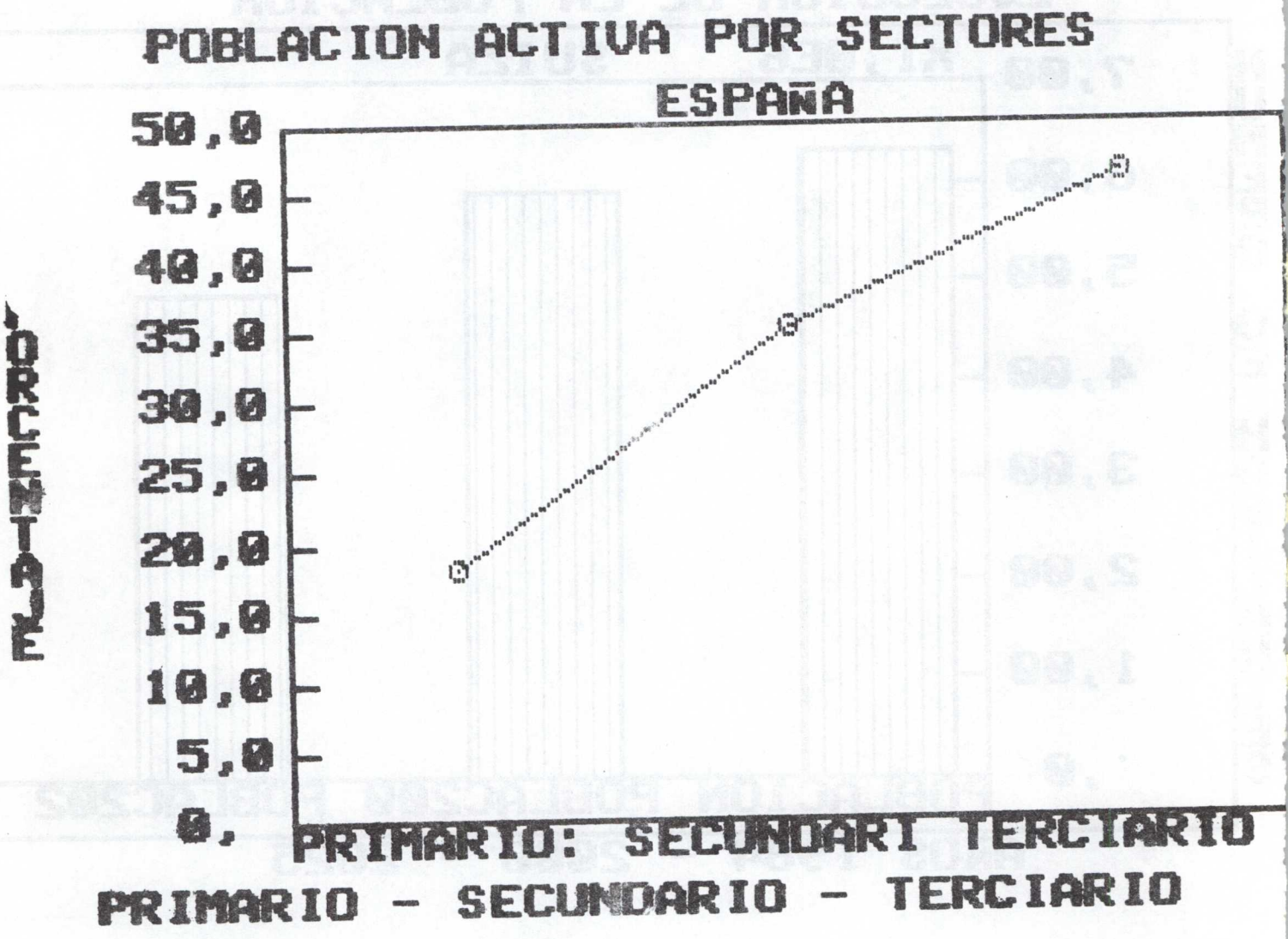
TIPOS DE GRAFICOS

BARRAS



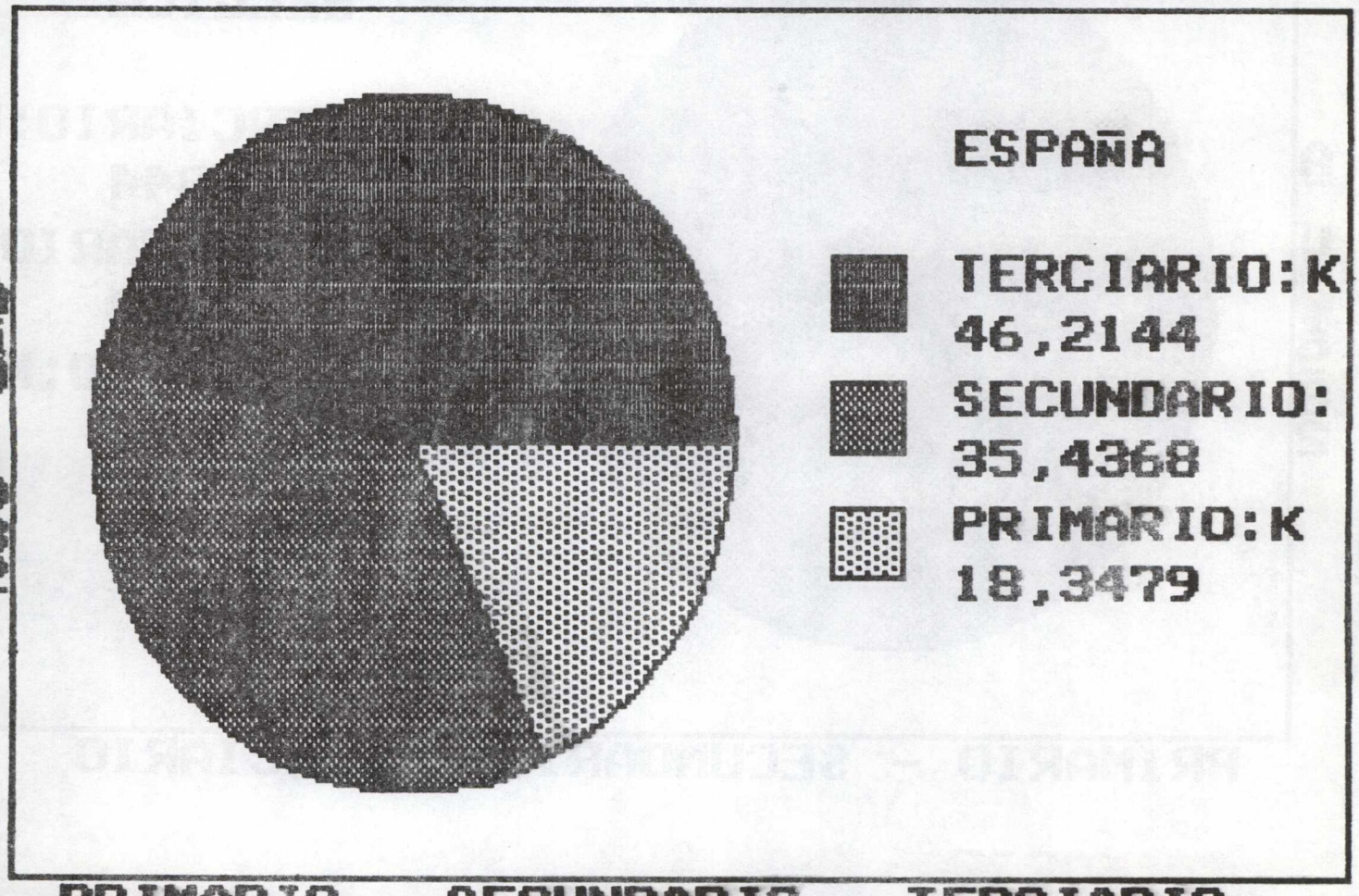

EVOLUCION DE LA POBLACION
X1,0E6
SUIZA
7,00
6,00
5,00
4,00
3,00
2,00
1,00
,0
NUMERO DE HABITANTES
POBLACION
POBLAC200
POBLAC202
AÑOS 1984 - 2000 - 2025

LINEAS


POBLACION ACTIVA POR SECTORES
ESPAÑA
50,0
45,0
40,0
35,0
30,0
25,0
20,0
15,0
10,0
5,0
0,
PRIMARIO: SECUNDARI TERCIARIO
PRIMARIO - SECUNDARIO - TERCIARIO

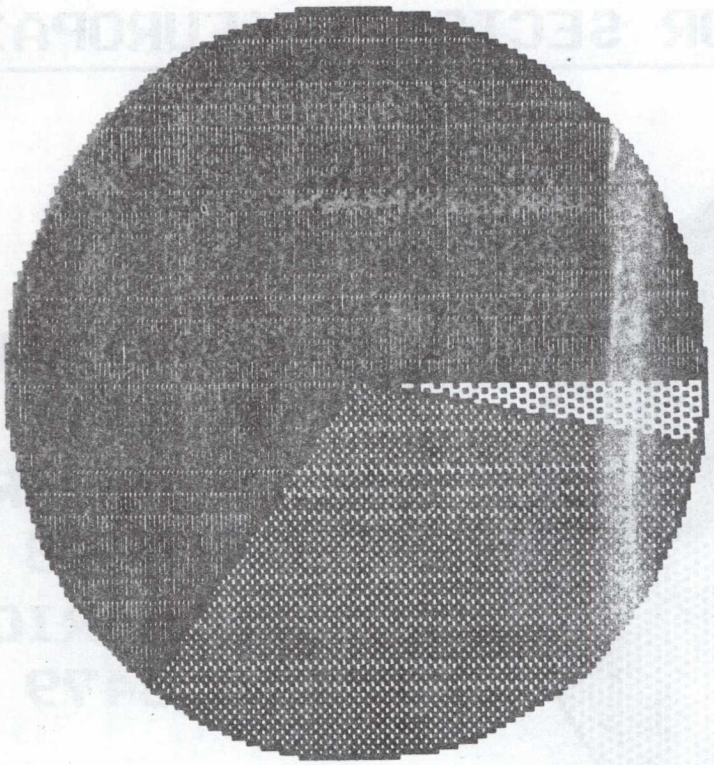
PASTEL

DISTRIBUCION POR SECTORES (EUROPA)



PRIMARIO - SECUNDARIO - TERCIARIO

DISTRIBUCION POR SECTORES (EUROPA)



BELGICA

TERCIARIO:K 64,6844

SECUNDARIO: 32,3588

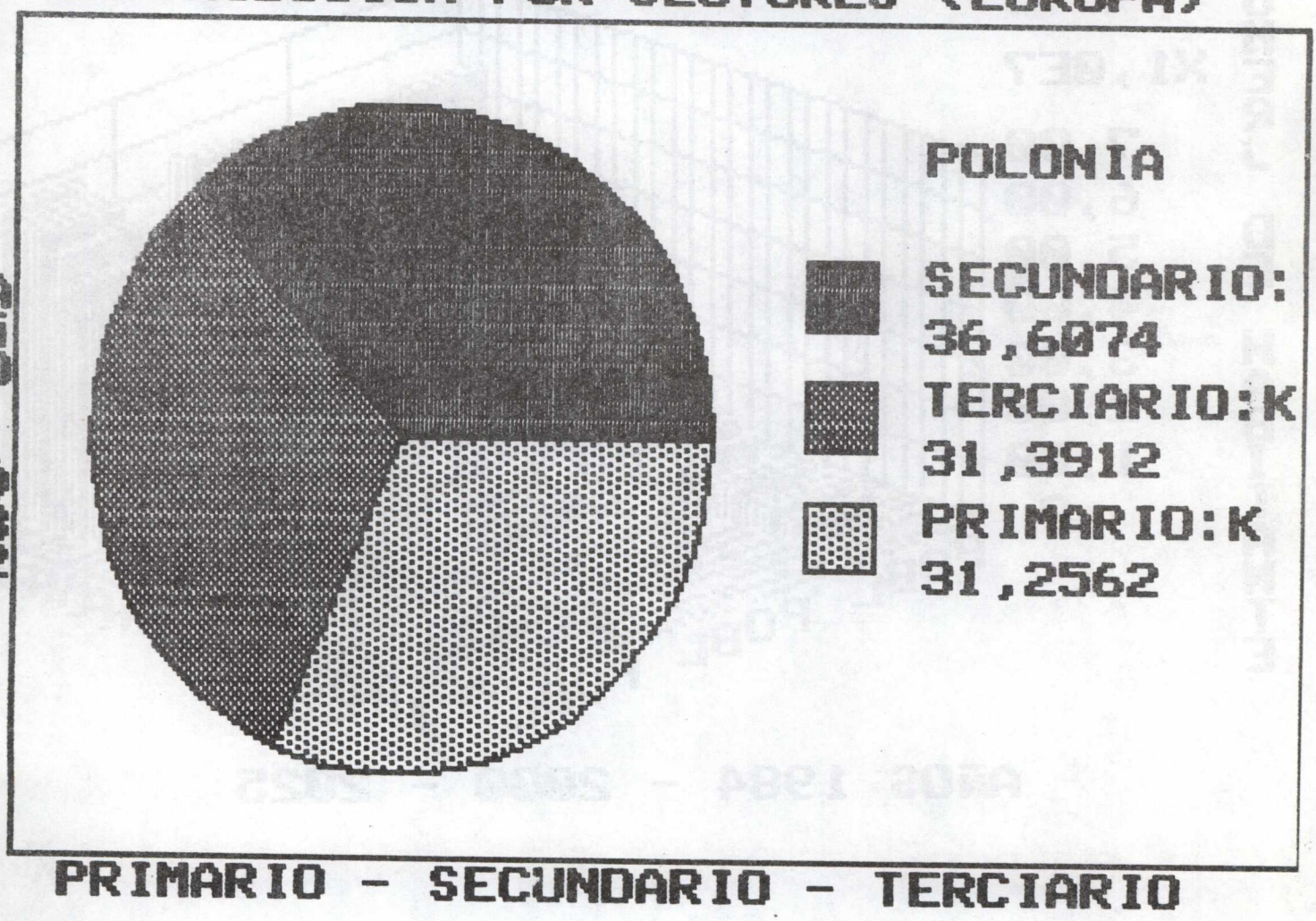
PRIMARIO:K 2,9568

UNIDAD 1982

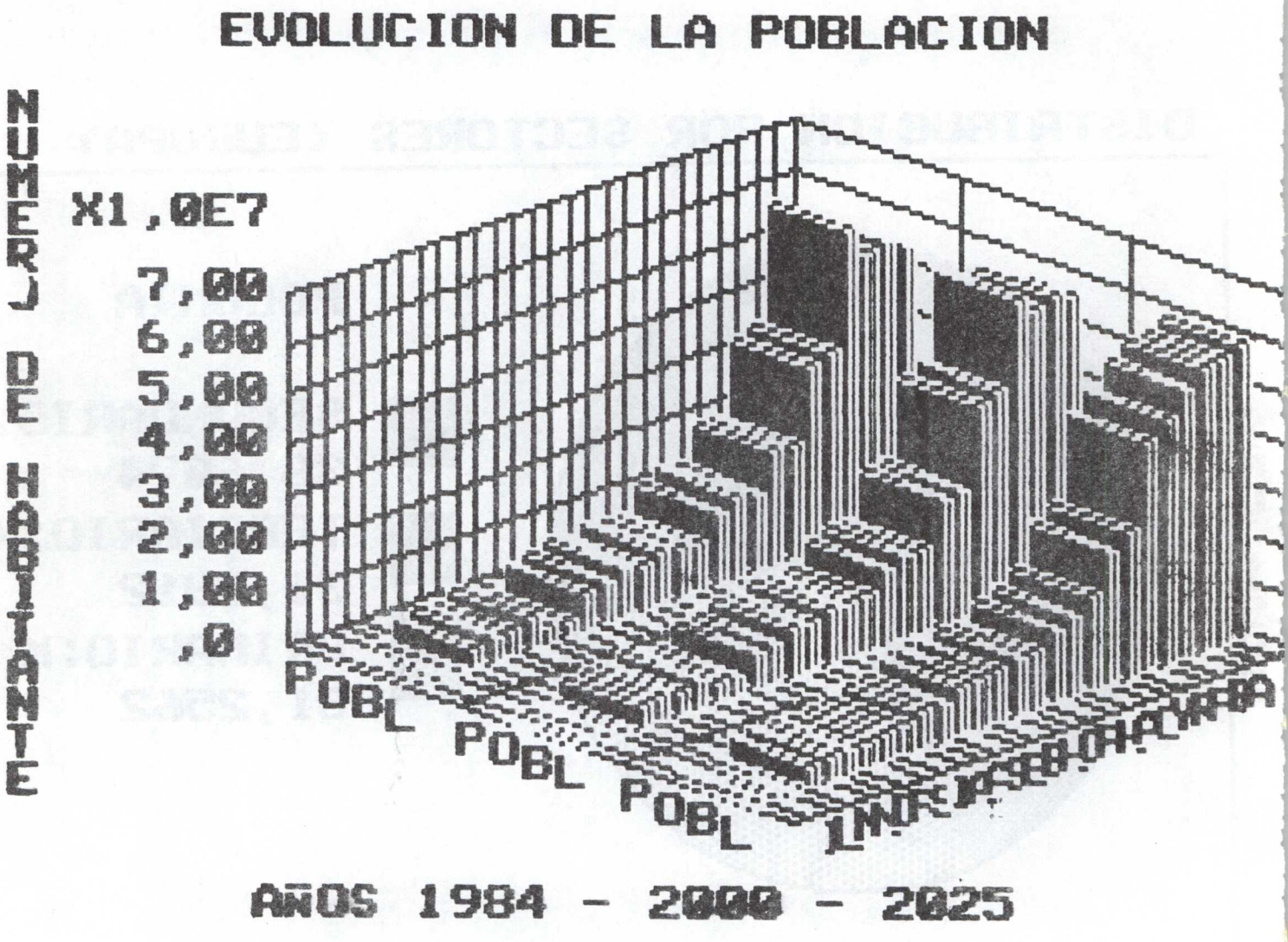
PRIMARIO - SECUNDARIO - TERCIARIO

MODOS DE PRESENTACION

SENCILLO: Muestra el gráfico que corresponde al nivel (registro) actual. Admite todos los tipos: Barras, líneas, pastel.

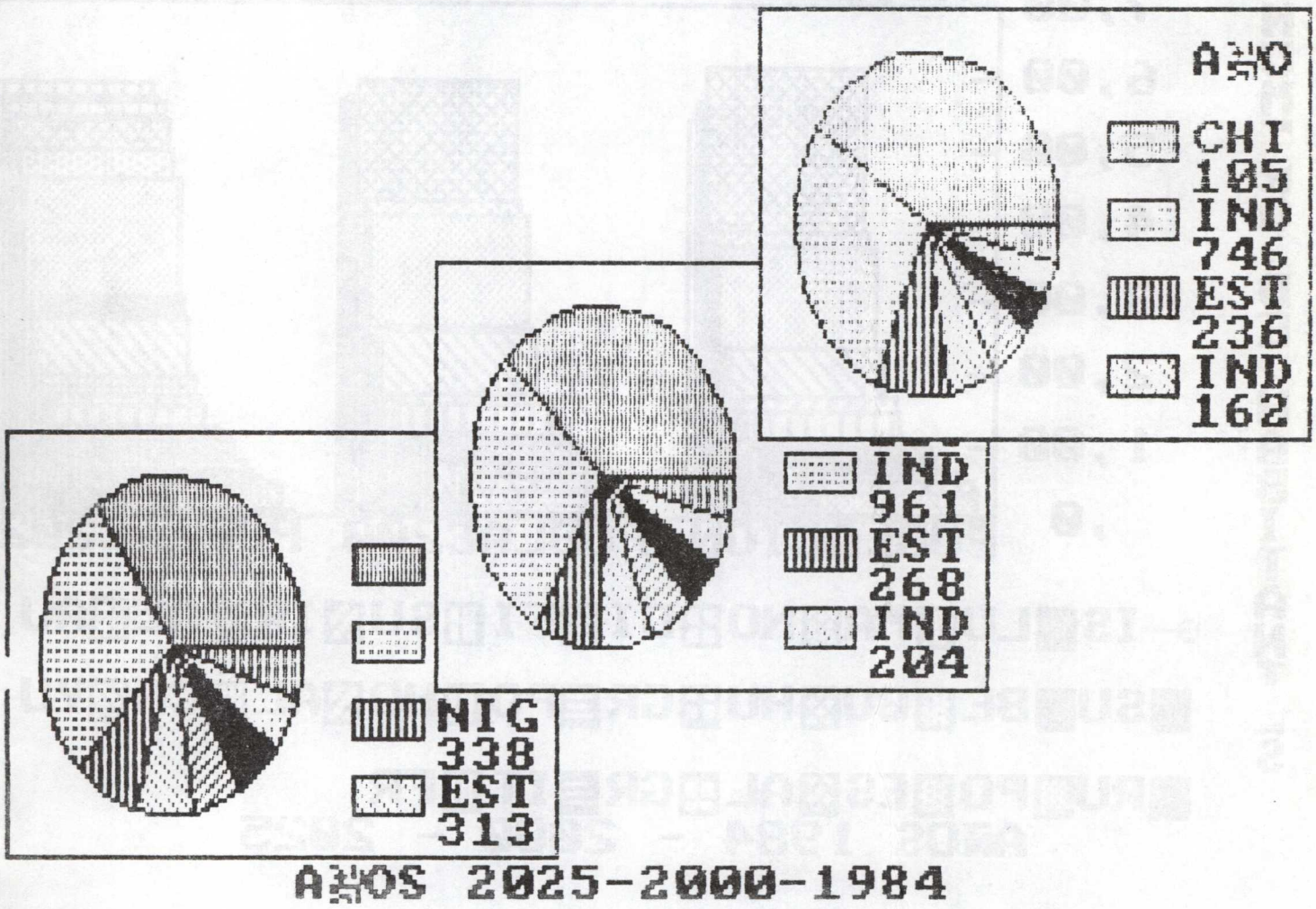


TRES-D: Muestra el gráfico de información de todos l[illegible]IVELES (registros) y POSICIONES. (Solo admite opción Barras)



VENTANAS

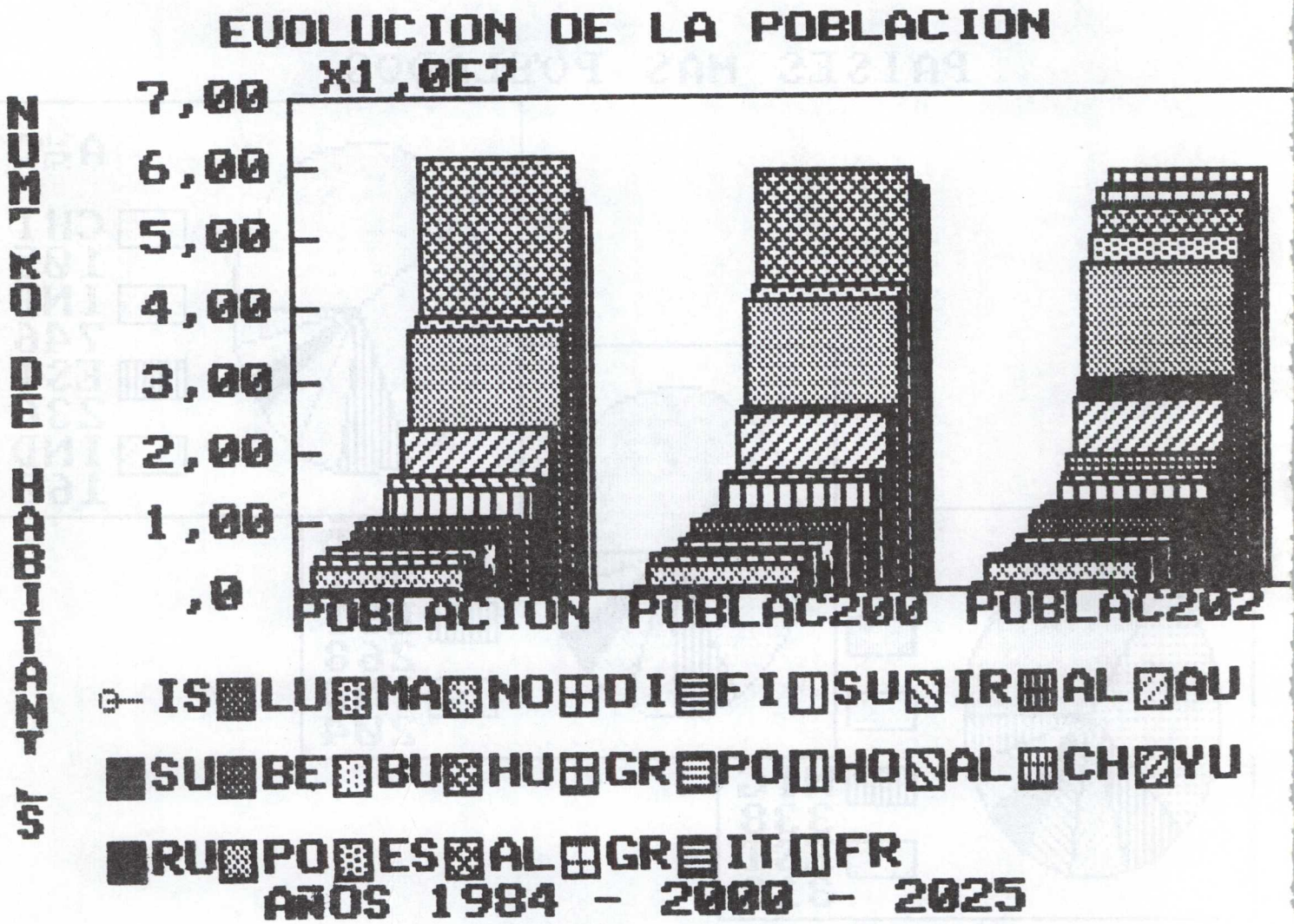
Muestra ee l gráfico del nivel (registro) actual y al pulsar ↵ aparece el gráfico del siguiente nivel. Admite todos los tipos.

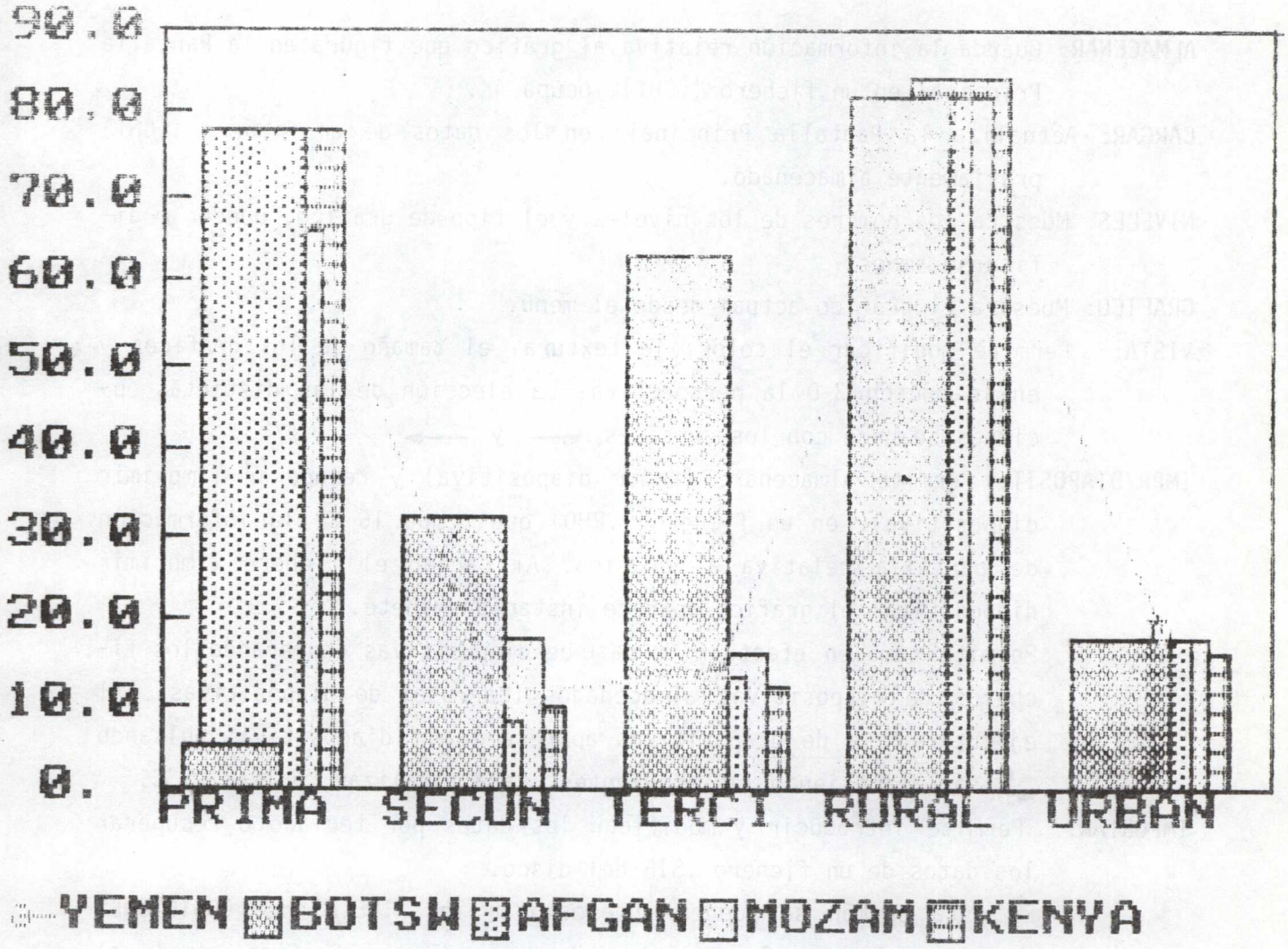


En modo VENTANAS la opción VISTA permite definir la posición y el tamaño de las sucesivas ventanas (Usar PgDn y PgUp).

SUPERPUESTO

Muestra todos los gráficos de todos los niveles en la misma pantalla. Admite los tipos Barras y Líneas.





desde la Pantalla Principal de Gráficos con F7 dibuja el gráfico elegido.

MENU DE GRAFICOS

ALMACENAR: Guarda la información relativa al gráfico que figura en la Pantalla Principal en un fichero (.CHT), ocupa 4K.

CARGAR: Actualiza la Pantalla Principal con los datos de un fichero (.CHT) previamente almacenado.

NIVELES: Muestra los nombres de los niveles y el tipo de gráfico, pueden modificarse ambos.

GRAFICO: Muestra el gráfico actual desde el menú.

VISTA: Permite modificar el color, la textura, el tamaño de los gráficos y en la opción 3-D la perspectiva. La elección de las distintas opciones se hace con los cursores ← y →

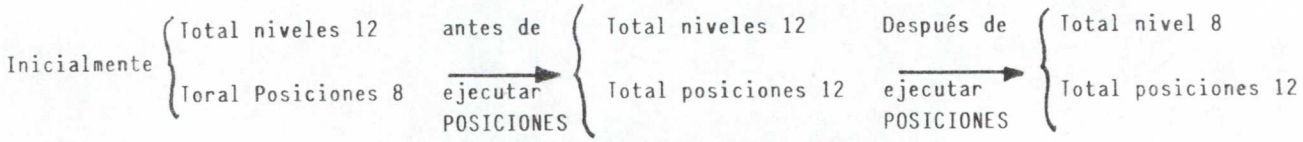
IMPR/DIAPOSIT: Permite almacenar, (tomar diapositiva) y recuperar (Imprimir diapositiva), en un fichero (.PHO) que ocupa 16 K, la información de pantalla relativa al gráfico. Al cargar el fichero (Imprimir diapositiva) el gráfico aparece instantánemanete.

Permite también efectuar un pase de diapositivas enumerando los ficheros de diapositivas almacenados (carrusel de diapositivas). Al ejecutar Pase de diapositivas aparece la 1ª diapositiva, pulsando ↵ van apareciendo las siguientes. Para finalizar, pulsar **ESC.**

IMPORTAR: Permite introducir y modificar los datos por teclado o recuperar los datos de un fichero .SIF del disco.

En esta opción se pueden intercambiar los NIVELES por POSICIONES para hacerlo en el caso de haber accedido a Gráficos desde el Gestor de Base de Datos se elige la opción **RENOVAR-CONTEXTO** con lo que aparece una nueva ventana en la que se ejecuta **POSICIONES.** Previamente debe de haberse modificado el número de posiciones o de niveles para que ambos tengan el mismo número, que debe ser el valor que inicialmente fuese mayor, si no se hace así, puede perderse parte de la información o impedir la ejecución de la última acción. Una vez realizada se disminuye el número de posiciones o niveles para que los números queden intercambiados.

Ejemplo:



Observaciones.

F7 dibuja el gráfico desde la pantalla principal.

F6 intercambia la pantalla principal y menú.

ESC vuelve siempre a la pantalla principal.

F3 saca una diapositiva en carrusel por impresora.

BASE DE DATOS: TABLA DE ALIMENTOS

Introducción

El gestor de Bases de Datos es un programa del Open Access que permite crear bases de datos y manejarlas posteriormente: listar, clasificar, ordenar, etc. El ejemplo que se presenta es una base de datos de 132 alimentos principales con sus características nutritivas. Los campos que forman esta base de datos son los siguientes:

	NOMBRE	CLASE	TIPO	ANCHURA
1.	ALIMENTO	clave única	texto	25
2.	PLATO	clave	texto	8
3.	CALORIAS	clave	numérico	3
4.	PROTEINAS	clave	decimal	4
5.	GRASAS	clave	decimal	4
6.	HIERRO	clave	decimal	4
7.	CALCIO	clave	numérico	3
8.	VITA	clave	numérico	5
9.	VITB1	clave	decimal	4
10.	RIBOFLAVIN	clave	decimal	4
11.	NIACINA	clave	decimal	4
12.	VITC	clave	numérico	3

Objetivos

- Utilizar bases de datos.
- Aprender algunos comandos del gestor de bases de datos.
- Resolver problemas a partir de una base de datos.
- Clasificar y ordenar los alimentos en función de las distintas características de los mismos.
- Analizar los datos, extrayendo conclusiones de los mismos.

Creación de la base de datos

La creación de la base de datos se realiza por medio del comando CREAR. Llamamos ALIMENTO al fichero. La máscara de pantalla que debemos diseñar puede ser la siguiente:

```
ALIMENTO                                        Plato


   Calorías   (gr)
   Proteinas  (gr)
   Grasa      (gr)

                  Minerales:
                              Calcio (mg)
                              Hierro (mg)
                  Vitaminas:
                                       A  (UI)
                                       B1 (mg)
                             Riboflavina (mg)
                             Niacina     (mg)
                                       C  (mg)
```

Menú de Diseño de Fichero y Formato de Pantalla
Salir Editar Línea_Cuyo Nuevo Tamaño

Para introducir registros usamos el comando INTRODUCIR y damos entrada a los siguientes alimentos cuyos datos están calculados para una ración de 100 gr:

TABLA DE LOS ALIMENTOS

ALIMENTO	PLATO	CAL	PROT	GRAS	Minerales Fe	Minerales Ca	Vitaminas A	Vitaminas B1	Vitaminas RIBO	Vitaminas NIAC	Vitaminas C
Aceite	grasas	884	0.0	100.	0.0	0	0	0.00	0.00	0.0	0
Aceitunas	fruta	135	1.0	14.0	2.0	100	300	0.03	0.08	0.5	0
Acelgas	verdura	22	2.0	0.3	2.5	100	2800	0.05	0.06	0.4	35
Ajos	verdura	100	4.5	0.2	2.3	20	0	0.21	0.08	0.6	9
Albaricoque	fruta	54	1.0	0.2	0.5	20	3000	0.04	0.06	0.7	10
Alcachofas	verdura	50	3.0	0.2	1.5	50	280	0.20	0.01	0.8	5
Almejas	pescado	78	13.0	1.4	17.0	142	250	0.10	0.16	1.4	0
Almendras	fruta	480	26.0	40.0	4.0	250	0	0.03	0.60	4.5	0
Alubias rojas, blanc	legumbre	330	20.0	2.5	7.0	130	30	0.35	0.20	2.0	0
Apio	verdura	20	1.1	0.2	0.5	50	30	0.05	0.04	0.4	7
Arenques frescos	pescado	160	19.0	8.0	1.1	100	0	0.05	0.15	3.5	0
Arroz	cereal	360	7.0	0.8	1.1	10	0	0.08	0.03	1.7	0
Atún enlatado aceite	pescado	300	23.0	22.0	1.2	42	100	0.05	0.20	10.0	0
Atún fresco	pescado	180	20.0	10.0	1.2	38	100	0.10	0.20	2.5	0
Avellanas	fruta	540	16.0	50.0	3.0	250	0	0.30	0.50	5.0	0
Azúcar	azúcar	400	0.0	0.0	0.0	0	0	0.00	0.00	0.0	0
Bacalao fresco	pescado	75	17.0	0.5	0.6	20	0	0.06	0.08	2.2	0
Bacalao salado	pescado	310	62.0	5.0	2.5	93	0	0.12	0.25	6.0	0
Batatas	tubércul	115	11.3	0.5	1.0	35	7000	0.10	0.10	0.6	30
Berenjena	verdura	27	1.0	0.2	0.8	20	30	0.04	0.04	0.8	5
Besugo fresco	pescado	100	17.0	3.6	0.8	30	0	0.06	0.08	2.2	0
Bollo suizo	cereal	315	8.0	7.0	1.0	40	0	0.07	0.05	1.0	0
Bonito fresco	pescado	150	21.0	5.0	1.0	35	0	0.05	0.10	2.5	0
Boquerones	pescado	170	20.0	10.0	1.0	500	100	0.08	0.20	2.7	0
Caballa fresca	pescado	175	20.0	10.0	1.2	40	100	0.08	0.20	2.7	0
Cacahuete	fruta	560	29.0	45.0	3.0	50	0	0.25	0.30	0.5	0
Calabaza	verdura	15	0.8	0.1	2.3	18	0	0.03	0.03	0.4	17
Calamares	pescado	80	14.0	1.0	1.7	144	250	0.07	0.16	1.4	0
Cangrejos	pescado	100	17.0	2.0	1.8	110	1000	0.10	1.00	2.8	0
Castañas frescas	fruta	170	2.0	1.6	1.0	50	0	0.25	0.30	0.5	0
Cebollas	verdura	40	1.4	0.2	1.0	35	50	0.03	0.04	0.2	8
Cerdo (menos grasa)	carne	280	15.0	25.0	1.7	8	0	0.60	0.20	3.0	0
Cerdo (muy grasa)	carne	375	13.0	35.0	2.0	6	0	0.30	0.15	2.5	0
Cerezas	fruta	60	1.1	0.4	0.4	20	650	0.05	0.06	0.4	10
Chirimoya	fruta	80	1.0	0.2	0.6	34	0	0.09	0.13	0.9	16
Chorizo	carne	210	24.0	12.0	3.5	30	0	0.20	0.15	3.0	0
Ciruelas	fruta	60	0.9	0.2	0.5	20	400	0.06	0.04	0.5	5
Coco	fruta	300	3.5	27.0	1.8	13	0	0.04	0.03	0.6	4
Coles de Bruselas	verdura	47	5.0	0.3	1.5	40	200	0.16	0.16	0.9	90
Coliflor	verdura	30	3.0	0.3	1.0	25	100	0.15	0.10	0.6	75
Conejo	carne	160	20.0	10.0	2.4	16	0	0.05	0.18	8.0	0
Cordero	carne	250	18.0	20.0	2.5	8	0	0.07	0.15	2.5	0
Escarola	verdura	20	1.7	0.2	1.7	80	2300	0.07	0.12	0.4	11
Esparragos	verdura	20	2.0	0.2	1.0	20	1100	0.15	0.18	1.0	8
Espinacas	verdura	25	2.3	0.3	3.0	80	10000	0.10	0.20	1.0	50

NUCLEO: BDAT
UTI: 0038
PAG.: 4

TABLA DE LOS ALIMENTOS

ALIMENTO	PLATO	CAL	PROT	GRAS	Minerales Fe	Ca	Vitaminas A	B1	RIBO	NIAC	C
Fresas	fruta	40	0.8	0.6	0.8	28	60	0.03	0.07	0.3	60
Fresón	fruta	40	0.9	0.5	0.7	30	100	0.03	0.07	0.3	90
Galletas	cereal	380	7.0	7.0	1.2	45	0	0.07	0.05	1.0	0
Gallos	pescado	85	18.0	1.3	0.8	30	0	0.07	0.08	2.0	0
Gambas	pescado	100	18.0	3.0	1.8	110	0	0.08	0.15	2.4	0
Garbanzos	legumbre	360	20.0	6.5	8.0	130	150	0.45	0.18	1.6	0
Guisantes secos	legumbre	346	22.0	2.0	5.0	60	100	0.55	0.15	2.5	0
Guisantes verdes	verdura	85	6.6	0.4	2.0	25	600	0.35	0.20	2.0	25
Habas frescas	verdura	85	6.6	0.4	2.0	25	600	0.35	0.20	2.0	25
Habas secas	legumbre	330	25.0	2.0	5.0	100	100	0.50	0.30	2.3	0
Harina de trigo	cereal	360	7.0	0.8	1.1	10	0	0.08	0.03	1.7	0
Higado de vaca	carne	130	20.0	4.0	14.0	10	7000	0.40	4.50	15.0	10
Higos	fruta	65	1.0	0.4	0.6	53	100	0.06	0.05	0.5	2
Higos secos	fruta	280	3.0	0.8	3.0	90	70	0.10	0.10	1.5	2
Huevo gallina	huevo	160	12.0	12.0	3.0	60	1000	0.15	0.30	0.1	0
Jamón magro	carne	170	33.0	4.4	1.4	48	0	0.15	0.15	4.3	0
Judias verdes	verdura	39	2.4	0.3	1.0	56	500	0.08	0.10	0.5	15
Jurel fresco	pescado	170	20.0	10.0	1.0	20	100	0.08	0.20	2.7	0
Langostinos	pescado	115	18.0	4.3	1.7	190	0	0.08	0.15	2.4	0
Leche condensada	lácteo	325	8.1	8.4	0.4	280	400	0.07	0.40	0.2	1
Leche vaca completa	lácteo	65	3.3	3.0	0.1	120	120	0.04	0.20	0.1	2
Lechuga	verdura	16	1.3	0.2	0.8	30	2000	0.04	0.08	0.2	18
Lenguado	pescado	100	19.0	2.5	0.8	22	0	0.07	0.08	2.0	0
Lentejas	legumbre	320	22.0	2.0	7.0	60	100	0.40	0.20	2.0	0
Limón	fruta	35	0.8	0.3	0.6	40	0	0.04	0.02	0.1	40
Macarrones,fideos,et	cereal	360	10.0	0.6	1.0	20	0	0.12	0.05	1.0	0
Mandarina	fruta	43	0.8	0.2	0.4	33	200	0.08	0.03	0.2	35
Manteca de cerdo	grasas	825	0.0	99.0	0.0	0	0	0.00	0.00	0.0	0
'Mantequilla	grasas	720	0.6	82.0	0.1	17	3200	0.00	0.00	0.0	0
Manzana	fruta	55	0.4	0.4	0.3	6	100	0.04	0.03	0.2	0
Margarina	grasas	720	0.6	81.0	0.3	3	0	0.00	0.00	0.0	0
Melocotón	fruta	55	0.8	0.2	0.6	10	1000	0.02	0.05	0.9	0
Melón	fruta	25	0.7	0.2	0.5	20	1200	0.05	0.04	0.6	3
Membrillo (carne de)	fruta	75	0.4	0.1	0.4	5	450	0.01	0.02	0.7	0
Merluza	pescado	80	19.0	0.5	0.8	30	0	0.05	0.10	3.0	0
Mermelada	fruta	300	1.0	0.3	0.3	12	0	0.02	0.02	0.2	0
Mero	pescado	90	19.0	0.7	1.5	30	0	0.10	0.05	3.0	0
Miel	azúcar	300	0.3	0.0	0.8	5	0	0.04	0.14	1.2	0
Morcilla	carne	160	15.0	10.0	40.0	15	0	0.05	0.10	3.0	0
Naranja	fruta	42	1.0	0.2	0.4	33	200	0.08	0.20	0.2	5
Nueces	fruta	600	13.0	60.0	3.0	100	0	0.30	0.50	3.0	0
Palometa	pescado	125	20.0	5.0	0.7	25	0	0.05	0.08	2.2	0
Pan de trigo	cereal	280	8.0	0.8	1.4	30	0	0.11	0.06	1.1	0
Pasas	fruta	280	3.0	0.8	3.0	80	80	0.12	0.11	1.6	0

TABLA DE LOS ALIMENTOS

ALIMENTO	PLATO	CAL	PROT	GRAS	Minerales Fe	Ca	Vitaminas A	B1	RIBO	NIAC	C
Patatas	tubércul	85	2.0	0.1	0.6	10	0	0.10	0.03	1.5	20
Pavo	carne	260	20.0	20.0	4.0	21	0	0.09	0.15	8.0	0
Pepino	verdura	13	0.8	0.1	0.3	15	20	0.04	0.05	0.2	20
Pera	fruta	60	0.6	0.3	0.3	10	20	0.02	0.04	0.1	5
Perejil	verdura	43	3.2	0.6	3.1	190	6000	0.12	0.24	1.0	140
Pescadilla	pescado	75	17.0	0.5	0.8	28	0	0.06	0.08	2.2	0
Pimento verde	verdura	30	1.4	0.3	0.7	8	300	0.07	0.08	1.0	100
Pimiento rojo	verdura	30	1.4	0.3	0.7	8	1000	0.07	0.08	1.0	100
Plátano	fruta	100	1.3	0.3	0.5	10	100	0.05	0.04	0.6	8
Pollo	carne	200	18.0	15.0	1.5	12	0	0.10	0.16	8.0	0
Pomelo	fruta	30	0.6	0.2	0.5	25	0	0.04	0.02	0.2	35
Puerros	verdura	50	1.8	0.2	1.3	60	50	0.09	0.06	0.5	18
Pulpo	pescado	60	13.0	0.3	2.5	40	0	0.02	0.07	1.3	0
Queso de bola	lácteo	352	26.2	27.5	1.0	900	310	0.01	0.45	0.1	0
Queso de Burgos	lácteo	215	19.0	15.0	0.3	210	40	0.02	0.30	0.1	0
Queso de Cabrales	lácteo	385	20.5	32.5	1.0	700	310	0.01	0.45	0.1	0
Queso de Camembert	lácteo	305	18.0	26.0	0.5	162	240	0.05	0.47	0.4	0
Queso Gruyère	lácteo	420	30.0	33.0	1.0	700	400	0.01	0.45	0.1	0
Queso manchego	lácteo	310	24.1	23.5	1.0	400	300	0.05	0.47	0.4	0
Queso Roquefort	lácteo	364	22.4	30.5	0.5	700	300	0.03	0.45	0.4	0
Rábanos	verdura	20	1.0	0.1	1.2	30	30	0.03	0.03	0.3	24
Rape	pescado	86	19.0	1.1	1.5	30	0	0.10	0.05	3.0	0
Remolacha	verdura	42	2.0	0.1	1.0	25	20	0.03	0.06	0.4	10
Repollo	verdura	25	1.6	0.2	0.4	50	100	0.07	0.05	0.3	50
Requesón	lácteo	70	16.0	0.8	0.3	100	50	0.02	0.30	0.1	0
Riñones	carne	130	17.0	7.0	5.3	20	300	0.50	2.00	6.0	0
Salchichas	carne	400	13.0	35.0	2.0	10	0	0.30	0.10	2.5	0
Salchichón	carne	420	25.0	35.0	3.6	10	0	0.24	0.21	3.0	0
Salmonete	pescado	100	18.0	3.1	0.7	30	0	0.05	0.07	2.0	0
Sandía	fruta	22	0.5	0.1	0.2	6	200	0.02	0.03	0.2	5
Sardina fresca	pescado	160	26.0	6.5	3.0	100	100	0.08	0.21	3.0	0
Sardinas en aceite	pescado	300	25.0	22.0	2.2	340	100	0.05	0.38	7.0	0
Sémola	cereal	360	9.0	1.0	1.5	16	0	0.15	0.08	1.0	0
Sesos	carne	130	12.0	6.0	3.0	12	175	0.13	0.20	3.0	0
Ternera magra	carne	156	19.5	8.0	2.4	11	20	0.14	0.25	6.3	0
Ternera semigrasa	carne	190	19.0	12.0	21.0	10	40	0.06	0.16	3.6	0
Tocino	grasas	760	3.0	82.0	1.0	5	0	0.15	0.04	0.9	0
Tomate	verdura	20	1.1	0.1	0.6	11	1000	0.07	0.04	0.5	20
Trucha	pescado	162	18.0	10.0	1.0	30	0	0.05	0.05	2.8	0
Uvas	fruta	65	0.7	0.4	0.6	19	80	0.06	0.04	0.2	5
Vaca grasa	carne	300	17.0	25.0	2.5	10	50	0.06	0.15	3.3	0
Vísceras en general	carne	140	16.0	7.0	2.0	12	0	0.10	0.05	3.0	0
Zanahoria	verdura	40	1.5	0.2	0.7	40	10000	0.06	0.04	0.7	5

Utilización de la base de datos

1. LISTAR

Accedemos al Gestor de Bases de Datos y pulsamos 'L' para listar. Al preguntar el programa sobre el fichero escribimos ALIMENTO y pulsamos <F10>. Al cabo de unos segundos aparece en la pantalla la lista de los 15 primeros alimentos del fichero ordenados alfabéticamente. En la lista aparecen todos los campos. Para poder visualizar los que no aparecen en la pantalla pulsamos las teclas de movimiento del cursor a la izquierda o derecha. El cursor está posicionado en el primer registro del fichero. Podemos ver los restantes registros, pulsando las teclas <End> y <cursor abajo> simultáneamente. La pantalla se borra y aparece la lista con los siguientes 15 registros. Con estos movimientos podemos ver el contenido de todos los registros del fichero.

2. VISUALIZAR

Si deseamos ver todo el contenido de un determinado registro, colocamos el cursor en el mismo y pulsamos <F6> (cambiar). El programa pregunta si el fichero de máscara de pantalla que se desea utilizar es B:ALIMENTO.SMK. Pulsamos <F10> para confirmarlo. Entonces aparece en pantalla el contenido del registro. Con <F6> además, es posible modificar los valores de los campos de este registro. Pulsando <PgUp> o <PgDn> se puede visualizar el registro anterior y posterior respectivamente. Esta operación se puede realizar también desde el menú accediendo al comando VISUALIZAR. Este acceso se realiza bien utilizando las teclas de moyimiento del cursor hasta que se ilumine el

comando o bien, escribiendo la o las iniciales del mismo.

3. CLASIFICAR

Para listar o visualizar los alimentos ordenados por platos, por ejemplo, hay que utilizar el comando CLASIFICAR. Pulsamos <F2> para volver al menú y accedemos a este comando. El programa pregunta el orden por el que se desea clasificar. Escribimos PLATO, ALIMENTO para realizarlo por tipos de plato y, dentro del mismo tipo, por orden alfabético del alimento. Una vez que la máquina termina este proceso (el comando cambia de color), debemos acceder al comando LISTAR o VISUALIZAR para observar esta clasificación.

Si se desea confeccionar la lista de los alimentos ordenada por su aportación calórica, se debe volver a clasificar en el orden -CALORIAS para que aparezca la lista en orden decreciente (de ahí el signo -) de las calorias proporcionadas por cada alimento.

Estas listas también se pueden realizar con el comando LISTAR pero es necesario cerrar el fichero activo con el comando NUEVO. Al pulsar el comando LISTAR escribimos:

DE ALIMENTOS ORDEN PLATO,ALIMENTO

o bien

DE ALIMENTOS ORDEN -CALORIAS.

4. SELECCIONAR

Deseamos confeccionar la lista de las verduras que aporten vitamina A ordenadas por dicha aportación en orden decreciente. En dicha lista debe aparecer el nombre del alimento, su plato, la vitamina A y la vitamina C, por ejemplo.

Pulsamos <F2> para volver al menú y accedemos al comando NUEVO para cerrar el fichero activo. Pulsamos L para listar y cuando aparezca la cláusula DE escribimos:

```
DE ALIMENTO
ELIGE ALIMENTO,VITA,VITC
CUYO PLATO='VERDURA' Y VITA>0
ORDEN -VITA
```

Al cabo de unos segundos nos aparece la lista:

ALIMENTO	VITA	VITC
Espinacas	10000	50
Zanahoria	10000	5
Perejil	6000	140
Acelgas	2800	35
Escarola	2300	11
Lechuga	2000	18
Esparragos	1100	8
Tomate	1000	20
Pimiento rojo	1000	100
Habas frescas	600	25
Guisantes verdes	600	25
Judias verdes	500	15
Pimento verde	300	100
Alcachofas	280	5
Coles de Bruselas	200	90
Repollo	100	50
Coliflor	100	75
Puerros	50	18
Cebollas	50	8
Apio	30	7
Rábanos	30	24
Berenjena	30	5
Remolacha	20	10
Pepino	20	20

5. FORMATO e INFORME

La lista de los alimentos se puede confeccionar también con el comando INFORME. Para ello hay que crear previamente un fichero de informe. Esto se hace con el comando FORMATO. Accedemos a él para crear el fichero ALIMENTO.PMK. Dentro del menú de dicho comando debemos diseñar la cabecera, los registros y el pie del informe. Ayudándonos de los movimientos del cursor, definimos

La cabecera:

```
Página(P).....................TABLA DE LOS ALIMENTOS.....................
.....................................................................Minerales.Vitaminas
ALIMENTO              .PLATO    .CAL.PROT.GRAS.Fe...Ca..    A..B1...RIBO.NIAC.C..
```

los registros:

```
(ALIMENTO             ).(PLATO ).(C).(PR).(GR).(HI).(C).(VIT).(VI).(RI).(NI).(V)
```

y dejamos el pie y los totales en blanco.

Una vez definidas las características del formato de informe, lo grabamos en el disco con la opción SALIR del menú de FORMATO. Ahora accedemos al comando INFORME y nos aparecerá la lista.

Ejercicios

1. Localizar las aportaciones nutritivas de algunos alimentos que faltan en el fichero (yogur, leche desnatada, calabacín, hígado de cerdo, arroz integral, chocolate, bebidas alcohólicas, otros) y añadirlos al fichero. (Comando INTRODUCIR)

2. Confeccionar las listas de los alimentos por platos incluyendo todas sus características.

3. Elaborar las listas de los alimentos ordenadas por las distintas características nutritivas anotando los diez alimentos más ricos y los diez alimentos más pobres en cada una.

4. Los aportes diarios recomendables son los siguientes:

	CALO	PRO	CALC	HIE	VI A	VB1	RIB	NIA	V C
HOMBRES (45 años)	3000	70	800	10	5000	1.5	1.8	20	75
MUJERES (45 años)	2200	58	800	12	5000	1.1	1.5	17	70
MUCHACHOS (16-19)	3600	100	1400	15	5000	1.8	2.5	25	100
MUCHACHAS (16-19)	2400	75	1300	15	5000	1.2	1.9	16	80

Confeccionar distintos menús diarios que se ajusten a los aportes anteriores.

BASES DE DATOS

BERENGUER, José M. (Ed.)(1985) **Bases de datos. Retos, oportunidades y esperanzas.** Madrid, FUINCA.

Catálogo de bases de datos españolas (1985). Madrid, FUINCA.

DAINES, D. (1985) **Las bases de datos en la Educación Básica.** Barcelona, Gustavo Gili.

GARCIA-RAMOS, L.A. y F. RUIZ I TARRAGO (1985) **Informática y Educación. Panorama, aplicaciones y perspectivas.** Barcelona, Luis A. García-Ramos.

HUNTER, Beverly (1985) "Problem Solving with Data Bases". **The Computing Teacher**, vol. 12, no. 8, pp. 20-27.

KRUGLINSKI, David (1983) **Sistemas de administración de bases de datos.** New York, Osborne/McGraw-Hill.

LEWIS, T.G. y M.Z. SMITH (1985) **Estructuras de datos. Programación y aplicaciones.** Madrid, Paraninfo.

MARTIN, James (1985) **Organización de las bases de datos.** Englewood Cliffs, NJ, PHH Prentice Hall.

MARTIN, K. and T. RIORDON (1984-85) "Beyond Turtle Graphics: Data Bases with Logo". **The Computing Teacher**, vol. 12, no. 4, pp. 26-27

MAYNE, A. y M.B. WOOD (1983) **Introducción a las bases de datos relacionales.** Madrid, Ediciones Díaz de Santos.

SERRANO DE ENTRAMBASAGUAS, Guillermo (1976) **Bases de datos.(Una concepción de Sistemas de Información).** Madrid, Servicio de Publicaciones del MEC.

SMITH, Henry C. (1985) "Database Design: Composing Fully Normalized Tables from a Rigorous Dependency Diagram". **Communications of the ACM**, vol. 28, no. 8, pp. 826-838.

WIEDERHOLD, Gio (1985) **Diseño de bases de datos.** New York, Osborne/McGraw-Hill.

DANZIN, André y otros (1984) "**Informática y evolución de la sociedad**". Madrid. Banco de Bilbao. Instituto de Ciencias del Hombre.

COLL-VINENT, Roberto (1982) "**Bancos de Datos: Teoría de la Teledocumentación**". Barcelona. A.T.E.

FUINCA (1981) "**Bases de datos del mundo**". Madrid. Alhambra.

LOS LENGUAJES DE PROGRAMACION EN LA ENSEÑANZA

LENGUAJES DE PROGRAMACION PARA LA ENSEÑANZA GENERAL

El empleo de la informática en la enseñanza general, hasta ahora, se ha reducido a la enseñanza de los lenguajes de programación, y la enseñanza de la programación, se ha asociado con la capacidad de algoritmizar, tener una mente algorítmica, que consiste en una mentalidad secuencial, lo que es un algoritmo.
Esto está cada vez más cuestionado.
La enseñanza de un lenguaje de programación, no tiene que hacerse sólo como un fin en sí mismo, sino como algo más; puesto que no se acepta la necesidad de que los/las alumnos/as tengan que ser programadores/as.
La cuestión está en analizar las utilidades de los lenguajes de programación en la enseñanza general, que no es una enseñanza especializada.

Estas utilidades serían:

1.- DESARROLLAR CAPACIDADES:

- lógicas.
- de resolución de problemas, favoreciendo distintas estrategias.
- de autoaprendizaje: ya que hay que especificarle al ordenador una serie de mensajes concretos: instrucciones, en qué secuencia, etc.
- expresivas: existen infinitas posibilidades de combinaciones para trabajar con gráficos, sonidos.

2.- APRENDER POR MEDIO DEL LENGUAJE DE PROGRAMACION:

esto es mucho más que el desarrollo de la capacidad algorítmica , puede servir para aprender a través de él otras cosas.
Es el caso del lenguaje LOGO y también el de otros lenguajes, cuando se hace una simulación.

3.- LA ALFABETIZACION INFORMATICA:

este es un concepto cambiante. Hay que asociarlo con la capacidad de expresarse y escribir adecuadamente, en la acepción natural de la palabra alfabetizar.

La persona analfabeta lo es porque no es capaz de expresar mínimamente sus ideas para que las entienda el ordenador.
Un lenguaje de programación del ordenador también puede colaborar a esa alfabetización.
Dentro de esa alfabetización habrá que conocer lo que es un programa, lo que es un funcionamiento en modo directo y una instrucción ejecutada en medio de un programa, lo que son estructuras básicas de programación, etc.

4.- USO INSTRUMENTAL:

ejemplos:
- en matemáticas: para calcular valores de una función, se puede hacer un pequeño programa para resolverlo, y ésto hará reflexionar sobre el proceso de cómo la función se aplica a ese valor.
- en lenguaje: para mejorar el estilo de redacción con un procesador de textos, aunque éste sería otro tipo de lenguaje, entendido más ampliamente.

LENGUAJES DE PROGRAMACION PARA LA ENSEÑANZA DE LA INFORMATICA

¿Para qué puede servir saber programar?.
Dentro de una asignatura específica de Informática, el saber programar, sin olvidar los fines descritos en el apartado anterior, sirve como:

1. INTRODUCCION A LOS ORDENADORES

Se trata de conocer lo que es un ordenador, y nada mejor para conocerlo que programarlo, sobretodo si esto se hace con un lenguaje de bajo nivel.
El mismo lenguaje LOGO, sin ser un lenguaje de bajo nivel, puede servir para estos fines.

2. CONOCIMIENTO INTERNO DE LOS ORDENADORES

Utilizando lenguajes de muy bajo nivel: código máquina o un lenguaje ensamblador, se puede obtener un claro conocimiento de cómo funciona un ordenador internamente.

3. ANALISIS Y RESOLUCION DE PROBLEMAS

Hay técnicas didácticas muy extendidas que consisten en relacionar un problema de la vida real con su solución informática.

Lenguajes que se pueden utilizar, según el tipo de problemas, son los que permiten dividir un problema en módulos más simples.

4. CONOCER TECNICAS DE PROGRAMACION

Esto sería muy posterior al análisis y resolución de problemas. Habrá que conocer lo que es un organigrama, cómo se codifica en un lenguaje, saber utilizar un lenguaje descriptivo, programación modular, programación funcional.

5. TRABAJAR SOBRE ESTRUCTURAS DE DATOS

Como podrían ser: las variables de BASIC, los registros dinámicos de PASCAL, la estructura general de lista de los lenguajes LOGO y LISP, estructuras de tipo matricial.

6. ALMACENAMIENTO Y RECUPERACION DE INFORMACION

Conviene utilizar estructuras de ficheros. Sirven todos los lenguajes que tengan manejo de ficheros, incluso LOGO, que al ser modular, es más instructivo.

7. INICIARSE EN LAS TECNICAS DE LA INTELIGENCIA ARTIFICIAL

Utilizando un tipo de programación declarativa, como en PROLOG, o en lenguajes ya clásicos de I.A. como LISP. Como iniciación a estas técnicas puede servir el propio lenguaje LOGO, ya que es un heredero del LISP, pero más fácil de uso, y, por tanto, permite manejar de manera modular y más simple, los conceptos básicos de la I.A.

NUCLEO ININ
VOL 0031
PAG 3

ANALISIS COMPARATIVO ENTRE LOS DIFERENTES LENGUAJES DE PROGRAMACION DESDE UN PUNTO DE VISTA EDUCATIVO

Terminología básica para la clasificación de los lenguajes:

Declarativo/Imperativo

Un lenguaje declarativo representa un enfoque de programación que trata de "declarar" en lenguaje natural el problema sin pensar en el funcionamiento del ordenador. Se pone a una concepción"imperativa" de la programación en la que hay que realizar los programas de acuerdo con los procedimientos que el ordenador va a realizar para la resolución del problema.

Procedimental:

Un lenguaje procedimiental está basado en una estructura de procedimientos simples que permiten modularizar los problemas y ampliar el lenguaje definiendo nuevos procedimientos el propio usuario.

Alto Nivel/Bajo Nivel

Los lenguajes de alto nivel en los que cada instrucción representa un número elevado de instrucciones de máquina, por lo que son más cómodos para el usuario, se despreocupa de cómo funciona internamente la máquina.

Los lenguajes de bajo nivel se adaptan más a la estructura de la máquina, y por tanto dan lugar a programas largos y más difíciles de realizar por no expertos, si bien optimiza el aprovechamiento de la máquina.

Interactivo

Los lenguajes interactivos permiten modificar cualquier instrucción mediante el teclado e inmediatamente se puede ejecutar el programa modificado.

En los lenguajes no interactivos, cuando se efectúa una modificación hay que volver a traducir todo el programa (compilación) a código es más rápida que en los lenguajes interactivos.

Recursivo

Permite definir procedimientos que se llaman a sí mismos, dando lugar a programas con un nivel de abstracción y formalización bueno.

CARACTERISTICAS DE LOS LENGUAJES MAS UTILIZADOS COMERCIALMENTE

BASIC

Es interactivo, no estructurado y de alto nivel.
Es el lenguaje más difundido en los microcomputadores, ya que su interactividad le hace cómodo para el usuario. Da lugar a programas poco comprensibles, si bien, se puede evitar ésto a base de una disciplina de programación.

Es un lenguaje potente y que tiene muchas variantes, siendo la más extendida la de la casa Microsoft (BASIC MICROSOFT). Cada nueva versión que aparece, tiene más posibilidades. Es un lenguaje que nació pensado en facilitar la tarea de programación al usuario y pensando en el mundo educativo,. ya que se desarrolló en un centro de enseñanza (Darmouth College) para hacer más asequible la programación científica de FORTRAN a los alumnos. Tiene buenas posibilidades gráficas, normalmente realizadas en los microordenadores, y manejo de cadenas de caracteres y ficheros.

No admite recursividad.

LOGO

Es un lenguaje de alto nivel, interactivo y estructurado en procedimientos, con muy buenas capacidades gráficas (basadas en un punto orientado a "tortuga"). Pensado únicamente para el mundo educativo y desde la perspectiva de la psicología genética. Es un lenguaje que crece con el usuario y que permite desarrollar sus capacidades intelectuales y ser utilizado como instrumento de aprendizaje.

Posee una gran capacidad de manejo de listas y su estructura favorece el empleo de la recursividad. Logo es un derivado del lenguaje de Inteligencia Artificial LISP y cumple el mismo papel respecto de éste que BASIC tenía res-

pecto de FORTRAN.

Hay muchas variantes de Logo, unas presentan manejo de "tortugas" múltiples, "tortugas" móviles, programación de choques...
Otras hacen más hincapié en manejo de ficheros y listas más potentes.

Se puede decir que si bien el manejo de las listas es muy potente, tal vez sea algo abstracto al forzar el empleo de la recursividad, lo que hace que la parte de listas, pese a todo, sea bastante menos utilizada en el mundo educativo que la posibilidad gráfica y la inexistencia de variables con índices lleva a veces a programas algo "retorcidos" para algunas operaciones conceptualmente simples.

PASCAL

Lenguaje de alto nivel no interactivo, estructurado, permite la recursividad, posee estructuras de datos muy potentes como el registro, y es necesario definir previamente los tipos de variables y de datos a utilizar.

Permite la programación estructurada y la recursión, es muy potente, siendo el más utilizado en el mundo universitario.

Hay versiones, como el UCSD-PASCAL, y el TURBO PASCAL que es una de las más usadas en microordenadores, que tiene incorporados gráficos de "tortuga"

El hecho de no ser interactivo hace más trabajoso el contacto con la máquina. Es más duro aprender, al ser más abstracto que BASIC.

PROLOG

Lenguaje de alto nivel, descriptivo y de Inteligencia Artificial, su nombre viene de Programming in Logic y como su propio nombre indica, realiza las ideas de programación lógica. Es declarativo y muy potente para manejo de listas y bases de datos relacionales, con lo que se emplea en el mundo educativo para la enseñanza de la lógica como un lenguaje de Ordenador, sistemas expertos, sintaxis y semántica del lenguaje natural y como ayuda a la

enseñanza de la historia, geografía, etc. a través de las bases de datos lógicas.
LLeva internamente realizada la búsqueda en estructuras de árbol con vuelta hacia atrás. La estructura lógica es a veces poco adecuada para la descripción de algunos problemas, que da lugar a programas no muy claros, al ser preciso definir los procedimientos como claúsulas lógicas que son siempre ciertas.

Es el lenguaje en el que se están basando las investigaciones y desarrollos de la denominada quinta generación, la de la Inteligencia Artificial, especialmente en Japón.

Lenguaje de AUTOR

Son lenguajes específicamente pensados para el desarrollo de programas de E.A.O. en distintas materias. Permiten encadenamiento de preguntas y respuestas, módulos de aprendizaje, análisis de respuestas, etc.

Son adecuados para el desarrollo cómodo de programas de E.A.O., pero al ser lenguajes requieren conocimientos de programación para desarrollar lecciones. Los sistemas de autor permiten trabajar definiendo directamente pantallas, lo que elimina la necesidad de conocimientos de programación, si bien son más rápidos.

- Lenguaje PILOT, que lleva incorporadas gráficas de "tortuga". Es el más utilizado en microordenadores. Es similar al BASIC pero con algunas facilidades extras para análisis de respuestas.

- Lenguaje TUTOR de plato (Control Data). Pantalla de plasma y respuesta táctil.

- Private Tutor (Ordenador personal de IBM) Es un sistema de autor simple de uso pero limitado.

Otros lenguajes

C : Es un lenguaje de, relativamente, bajo nivel y estructurado. Es muy utilizado en conexión con el sistema operativo UNIX por programadores profesionales en lugar del ensamblador.

FORTH : Lenguaje muy ingenioso, que permite crecer con instrucciones definidas por el usuario. Es de nivel relativamente bajo. Utiliza notación polaca inversa. Su mayor campo de aplicación ha sido los juegos de ordenador.

LISP: Lenguaje de Inteligencia Artificial muy bien construído y con grandes posibilidades de manejo de listas (su nombre deriva de List Processing). De este lenguaje derivó el Logo. Tiene grandes posibilidades de manipulación de estructuras complejas de símbolos, pero tiene como inconveniente el excesivo número de paréntesis que utiliza.

FORTRAN: Lenguaje de alto nivel apropiado para el cálculo científico. De uso muy extendido en este área (Formula Translation)

COBOL: Lenguaje de alto nivel orientado a las tareas administrativas de uso muy extendido en el mundo empresarial.
(Common Business Oriented Languaje)
En el mundo educativo donde tiene su mayor sentido es en la formación profesional administrativa.

PL-1: Lenguaje orientado a las tareas administrativas, mejor estructurado que el COBOL, está menos extendido que éste. Fue desarrollado por IBM.

APL: Lenguaje interactivo de alto nivel orientado al cálculo matemático. Utilizado en algunas universidades para la enseñanza de las matemáticas.

COMAL: Lenguaje de alto nivel y estructurado, es algo más fácil que PASCAL obliga a una metodología de programación y evita los inconvenientes de BASIC. Su uso está difundido en algunos sistemas educativos.

ENSAMBLADOR: Lenguaje de bajo nivel, y por tanto muy dependiente de la máquina con la que se trabaja. Muy útil para programación de juegos, control de dispositivos. Tiene como ventaja en la enseñanza el que su uso permite conocer el verdadero funcionamiento interno del ordenador.

FILOSOFIA DEL LENGUAJE LOGO

ORIGENES DEL LENGUAJE.

Es conveniente, para llegar a conocer bien las características, posibilidades y la metodología de trabajo más adecuada del lenguaje LOGO, tener en cuenta sus antecedentes inmediatos y sus orígenes y, por tanto, saber que en la fecha de su creación (aparece a finales de los 60) los lenguajes de programación existentes no tenían como finalidad primordial el ser utilizados con fines educativos. Las aplicaciones del ordenador en este campo se limitaban a utilizarlo como herramienta de trabajo, para cálculos laboriosos, almacenamiento de datos, etc., o con los primeros programas de EAO (Enseñanza Asistida por Ordenador) a trasladar la enseñanza programada de los libros a las nuevas máquinas, dotando a este tipo de enseñanza de más posibilidades y mayor flexibilidad, pero sin llegar a representar cambios sustanciales ni obtener un buen aprovechamiento de este nuevo medio.

LOGO es un lenguaje de programación que nace en el laboratorio de Inteligencia Artificial del MIT (Instituto Tecnológico de Massachusetts). Uno de sus creadores, Seymour Papert, es un

matemático que ha trabajado con anterioridad (1959-1964) con Piaget en el Centro Internacional de Epistemología Genética de Ginebra interesándose por la psicología del aprendizaje y la construcción del conocimiento. En su libro Desafío a la mente, Papert hace observar la gran cantidad de conocimientos que los niños son capaces de incorporar desde que nacen, mediante lo que podemos llamar un proceso de "aprendizaje piagetiano", es decir, sin una enseñanza impuesta y pasiva, sino de una forma espontánea y natural. Así aprenden a hablar, aprenden geometría intuitiva para sus desplazamientos y lógica elemental para manejar a sus padres, entonces se pregunta por qué ciertos aprendizajes se demoran muchos años o no se llegan a producir en absoluto sin ayuda de una instrucción especial. A esta incógnita, une una gran preocupación, como matemático que es, por el gran fracaso que se produce en el aprendizaje de las Matemáticas y por lo que él llama "matemafobia" endémica que existe en nuestra cultura. Como Piaget, considera al niño constructor de sus propias estructuras intelectuales, pero disiente de él en cuanto que considera que la causa del problema no se encuentra tanto en la formalidad o complejidad de los conceptos, sino en la pobreza de materiales, modelos o metáforas que nuestra cultura ofrece para que dicha construcción sea posible.

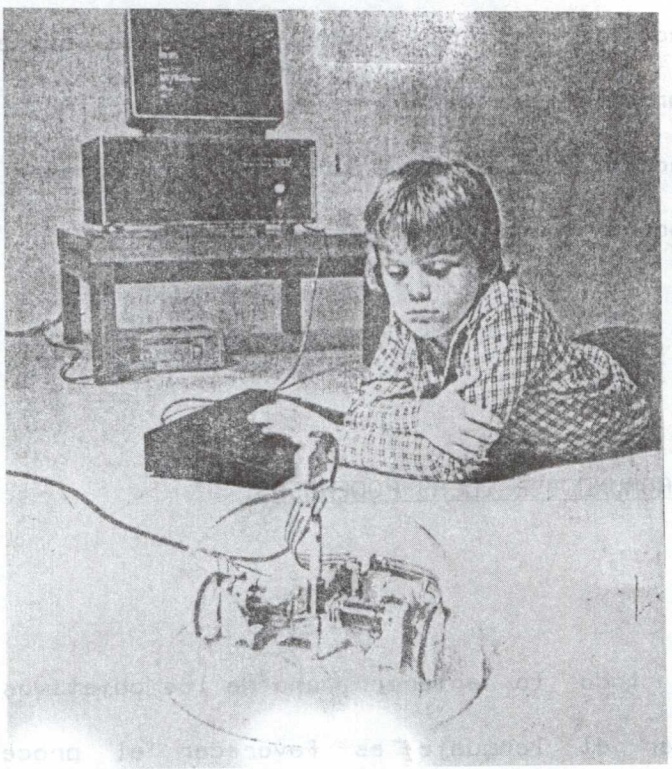
Al contrario de las teorías del aprendizaje basadas en la psicología conductista, en la psicología cognitiva se pone de relieve la participación del sujeto en el proceso de aprendizaje. Este proceso se realiza a base de incorporar informaciones nuevas

que entran en conflicto o producen cierto desequilibrio respecto a las informaciones asimiladas previamente. El progreso se produce a través del proceso que el sujeto sigue para superar el desfase y restablecer el equilibrio, pero ésto sólo es posible si entre el objeto nuevo de conocimiento y los marcos asimiladores del individuo, existe un desfase óptimo, es decir, si la situación es totalmente ajena a los esquemas previos, el individuo no tendrá posibilidad ninguna de incorporar el nuevo conocimiento. La gradación que se puede seguir con LOGO, avanzando cada alumno a su propio ritmo, y la no necesidad de conocimientos previos para trabajar con este lenguaje conduce a producir dichos desfases óptimos.

MICROMUNDOS E IDEAS PODEROSAS.

Por todo lo anterior, uno de los objetivos que se pretenden al crear el lenguaje es favorecer el proceso de aprendizaje, buscando elementos que sirvan para la concreción y construcción del pensamiento formal. Y así aparece la tortuga como un objeto transicional que vincula el conocimiento formal con el corporal, con los esquemas sensorio-motores y produce una visión antropomórfica de los problemas. La tortuga LOGO es uno de los elementos "con los cuales pensar" que nos ofrece este lenguaje, es el más asequible a los niños de corta edad y que además puede

aportar esa componente afectiva que Papert considera necesaria o por lo menos favorecedora del aprendizaje (capítulo: Los engranajes de mi infancia del libro anteriormente citado). En los primeros contactos con el ordenador y su programación, el "hacer de tortuga", es decir, el identificarse con ella y que los niños realicen previamente, real o mentalmente, los movimientos que quieren que efectúe la tortuga mecánica por el suelo o ver realizados en la pantalla del ordenador, puede facilitar la tarea.



En la mayor parte de las versiones del lenguaje, la tortuga LOGO es un elemento luminoso, con forma de triángulo, una silueta de tortuga o un pequeño angulo en algunas otras. Puede ser controlado desde el teclado y, entre otras acciones, puede realizar desplazamientos con las órdenes AVANZA y RETROCEDE, orientarse según un rumbo dado (PONRUMBO) y efectuar cambios de dirección (GIRADERECHA y GIRAIZQUIERDA).

En sus desplazamientos, si lleva el lápiz preparado (orden BAJALAPIZ), la tortuga puede ir dejando un rastro, con el que realiza los gráficos, o bien moverse sin dejar huella (orden BAJALAPIZ).

Pero la tortuga no es el único elemento motivador y de ayuda del lenguaje, todo en él está orientado a posibilitar la creación de otros objetos con los cuales pensar, éstos son los llamados "micromundos" en la terminología LOGO y que se pueden definir como subconjuntos de la realidad o de una realidad construida, cuya estructura es acorde con un mecanismo cognitivo determinado y que puede suministrar un entorno donde explorar y descubrir. Así pues, con LOGO se pueden crear micromundos para la exploración de elementos geométricos en el plano y en el espacio tridimensional, o en el campo de la física, la música, o la gramática por citar algunos. Por ejemplo, uno de los más citados es el micromundo de las tortugas dinámicas o "dinatortugas", para explorar y utilizar como introducción a la mecánica newtoniana. Con este micromundo parece que se alcanza un fin importante pues se cubre una parcela del programa oficial de la física escolar, pero éste no es el objetivo más valorado por los creadores y usuarios de LOGO. Más importante es que trabajando con éste u otro micromundo los alumnos realizan un aprendizaje totalmente activo de una forma creativa y descubriendo por sí mismos nuevas leyes y propiedades; exploran, cambian parámetros y observan los resultados contrastándolos con sus predicciones.

Al trabajar con la tortuga, se puede comprobar, por ejemplo, que al caminar alrededor de un objeto totalizando su giro a medida que avanza, sumando los giros a la derecha y restando los giros a la izquierda, el resultado es de 360 grados no importa cual sea la forma del objeto. Esta propiedad que en los ambientes LOGO se suele denominar "teorema de la vuelta o del viaje completo de la tortuga", es lo que Papert llama una "idea poderosa" pues facilita la obtención de otros conocimientos, por ejemplo, cómo deben ser los giros para dibujar una poligonal cerrada o calcular el ángulo interior de un polígono regular.

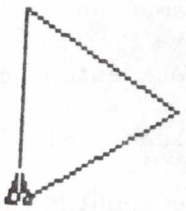
Desde luego que la creación de micromundos o la aparición de ideas poderosas es posible utilizando otros medios ajenos al lenguaje o al mismo ordenador, pero LOGO está diseñado para que ésto sea posible con una cierta facilidad.

Estrategias de resolución de problemas.

Un aspecto importante que se deriva de la programación en LOGO es que se favorece el uso de distintas estrategias de resolución de problemas. Por una parte, al ser un lenguaje interactivo, se puede trabajar en modo directo, lo que se suele llamar "pilotando la tortuga", y con lo cual se obtiene una respuesta inmediata a las órdenes dadas. Se puede trabajar también creando los procedimientos que "enseñen" a la tortuga lo que deberá hacer con posterioridad para realizar nuestros proyectos.

```
?AVANZA 40
?GIRADERECHA 60
?AVANZA 40
?GIRADERECHA 60
?AVANZA 40
?GIRADERECHA 60
```

```
PARA TRIANGULO
REPITE 3 [AV 40 GD 60]
FIN
```



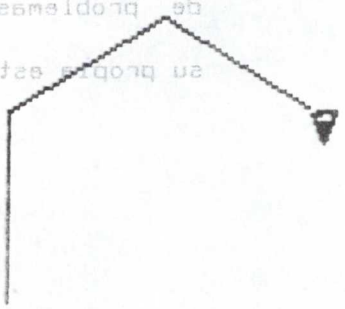
A la hora de planificar un trabajo también se pueden seguir distintas estrategias: Análisis descendente del problema (top down) subdividiéndo el problema total en otros más simples (modularización). Análisis ascendente (botton up) resolviendo las partes elementales y componiendo los resultados. Y lo que suele ser la técnica más habitual, una mezcla de las anteriores o integrar un procedimiento ya definido en otro más complejo, proponer finalidades intermedias para acercarse a la solución final a partir de versiones más sencillas del problema, como puede ser conseguir el dibujo de una "circunferencia" a partir de polígonos regulares de número de lados creciente. Así pues, con LOGO se pueden seguir las sugerencias de Polya para la resolución de problemas, pero lo más importante es que cada sujeto utiliza su propia estrategia.

Hay que tener en cuenta que Logo se presenta como un instrumento didáctico, pero no deja de ser un lenguaje de programación y a los muchos inconvenientes que se puede suponer que se derivan de ello, por ejemplo fomentar un tipo de pensamiento mecánico, hay que oponer las ventajas que se pueden obtener. Un programador es una persona diestra en aislar y corregir errores y de pensamiento sistemático. Al programar en un lenguaje se pierde la tendencia a pensar que las cosas están mal o bien hechas simplemente. Teniendo que rectificar el programa sucesivas veces para conseguir el efecto perseguido se alcanzan varios objetivos.

El papel del error

En la búsqueda del error hay que reflexionar mucho sobre el propio pensamiento y el sujeto que realiza esta acción se acostumbra a buscar el posible fallo de su razonamiento, siendo ésto algo que realiza por sí mismo e inmediatamente que se produce puesto que el ordenador nos presenta el resultado y los posibles errores sin demora. Esta práctica, por tanto, puede ayudar a evitar esa situación, que todos los docentes conocen muy bien, de alumnos que dan respuestas disparatadas con la mayor naturalidad y sin ninguna comprobación ni rectificación posterior.

```
PARA TRIANGULO.EQUILATERO :LADO
REPITE 3 [AVANZA :LADO GIRADERECHA 60]
FIN
```



La experiencia continua de productos no perfectamente acabados puede eliminar el temor a equivocarse tan fuertemente arraigado en los alumnos y en los no alumnos, puesto que la corrección de un error en informática se entiende más como una depuración del programa, utilizando el sentido de la palabra inglesa "bug" que no tiene las connotaciones negativas que tiene en castellano la palabra error. De este modo, el error no es tomado ya tanto como un fracaso sino como un elemento enriquecedor, pues se pueden producir muchos resultados que no eran los esperados pero que dan lugar a nuevas experiencias y nuevos conocimientos.

Todo esto que acabamos de decir puede ser válido para cualquier lenguaje de programación, pero lo que aporta LOGO es la posibilidad de detectar más facilmente los fallos por la propia estructura del lenguaje, modularizable en procedimientos cortos que se pueden probar de forma independiente.

Al ser LOGO un lenguaje que se estructura en procedimientos, creando nuevas órdenes que se incorporan al lenguaje como primitivas, es un lenguaje que crece con el usuario conforme se van definiendo nuevos procedimientos. Parte de elementos tan sencillos como son las órdenes de la geometría de la tortuga, por lo que suele ser considerado como un lenguaje para niños exclusivamente, pero a su vez con una estructura de fundamental de datos tan sencilla como es la lista y con elementos de control tan potentes como la recursión (capacidad de un procedimiento de llamarse a sí mismo o de definirse en función de sí mismo) hace que sea un lenguaje de posibilidades casi ilimitadas, capaz de

resolver con técnicas simples cualquier tipo de problemas, hasta llegar a introducirse en el campo de la inteligencia artificial pues no en vano se puede considerar a LOGO hijo de LISP, aunque, eso si, mucho más sencillo que éste).

La tarea del educador no debe consistir en enseñar al alumno a programar, sino en aportar los elementos que el alumno necesite para poder seguir avanzando y en caso de observar que utiliza siempre la misma estrategia, sugerir la existencia de otras posibles; aunque esta situación se suele resolver sola en los ambientes de aprendizaje LOGO con el diálogo de los propios alumnos que se comunican sus experiencias y realizaciones.

Por todo lo dicho anteriormente, se puede considerar que LOGO es un instrumento de ayuda al desarrollo de capacidades cognitivas, que trata de llenar el vacío que hay en nuestra cultura de modelos de pensamiento sistemático, pero no está en la mente de sus creadores el ser la panacea que resuelva todos los problemas de la educación.

A la mayor parte de los profesores que utilizan el lenguaje LOGO les resulta difícil encontrar verdaderas aplicaciones dentro de sus respectivas áreas, sobre todo si son del área humanista o si pretenden utilizar LOGO como medio para enseñar una parte

concreta de su materia, por ejemplo: ¿Cómo enseñar a conjugar verbos con LOGO? Quizá el problema esté en que utilizan este lenguaje como E.A.O., al estilo tradicional y no de forma consecuente con su concepción. Quizá debiéramos comenzar con una revisión cuidadosa de los conocimientos que se deben adquirir, en función de la situación actual y los nuevos medios existentes. Los contenidos y los métodos no deben ser los mismos si tenemos sólo lápiz y papel o disponemos también de nuevas tecnologías.

Lo trivial es efectuar una simple aplicación de estas tecnologías a viejos métodos, pero igual que el cine nace como una adición de teatro y fotografía y sólo posteriormente se convierte en un arte distinto, debemos intentar que las tecnologías y en particular el ordenador se integren en la educación produciendo una nueva forma de aprendizaje.

Caracteristicas de Logo

Analizaremos detenidamente las caracteristicas generales mas importantes del lenguaje.

Graficos de Tortuga

Los graficos en logo se realizan mediante la "huella" que deja en sus desplazamientos un punto orientado, denominado "tortuga". La forma de dirigir la tortuga es mediante ordenes simples, que permiten hacerla avanzar un numero de pasos (AVANZA o AV), retroceder un numero de pasos (RETROCEDE o RE), girar a la derecha un angulo (GIRADERECHA o GD), idem a la izquierda (GIRAIZQUIERDA o GI), colocarse en una posicion de coordenadas dadas (PONPOS), orientarse segun un rumbo dado (PONRUMBO), bajar o subir el "lápiz" ficticio con el que dibuja (BAJALAPIZ o SUBELAPIZ). Esta idea tan simple origina muchas posibilidades graficas y geometricas.

Los graficos realizados por el metodo descrito forman, lo que se denomina en terminologia Logo, un "micromundo" de grandes posibilidades tanto geometricas (geometria de la tortuga) como estéticas y como instrumento para el desarrollo del pensamiento de los niños, y una ayuda para la concreción de conceptos abstractos, como los de variable, que pasaran a ser elementos útiles, ya que darán poder al alumno para realizar, por ejemplo, cuadrados de tamaño variable.

La importancia de esta idea ha llegado a influir sobre otros

lenguajes, haciendo que algunas versiones de otros leguajes, como es el caso de PASCAL, las incorporen.

Un ejemplo de uso de graficos de tortuga para dibujar un hexagono de lado 40 seria:

```
AVANZA 40  GIRADERECHA 60
AVANZA 40  GIRADERECHA 60
AVANZA 40  GIRADERECHA 60
AVANZA 40  GIRADERECHA 60
AVANZA 40  GIRADERECHA 60
AVANZA 40  GIRADERECHA 60
```

Es decir, una secuencia de seis despazamientos de 40 pasos seguidos cada uno de ellos de un giro de 60 grados. Si utilizamos la orden REPITE, que permite repetir una lista de instrucciones el numero de veces deseado, quedaria:

```
REPITE 6 [ AVANZA 40 GIRADERECHA 60]
```

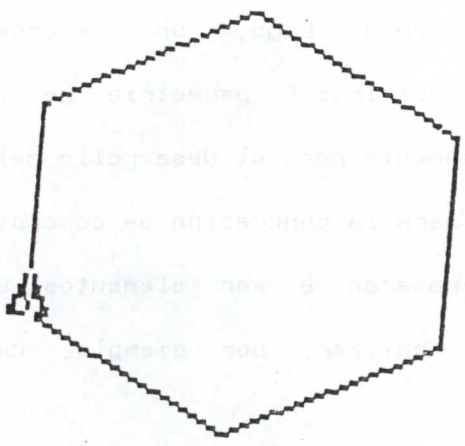


figura HEXAGONO

Interactivo

Logo es un lenguaje que permite el desarrollo interactivo de programas, ya que las ordenes dadas pueden ser ejecutadas directamente. Ya se vió en el ejemplo anterior como se ejecutaban las órdenes de forma inmediata, asi al ordenar AVANZA 40 y pulsar INTRO, la tortuga avanzaba 40 pasos.

Esta interaccion es muy importante cuando se trata de usuarios poco experimentados. Es ademas el hecho de poseer un intérprete lo que dió tanta popularidad a BASIC.

Procedimental

Una caracteristica que comparte Logo con otros lenguajes de ordenador, y en contraposicion con BASIC, es que es procedimental. Esto significa que programar en Logo consiste en enseñar procedimientos al ordenador, y que,una vez que los ha aprendido, el ordenador los entiende como si fueran nuevas primitivas del mismo. Por ejemplo, si volvemos al hexagono anterior, podriamos haberlo definido de la forma:

```
PARA HEXAGONO
REPITE 6 [AVANZA 40  GIRADERECHA 60]
FIN
```

Se puede decir que un procedimiento es una serie de ordenes que con un nombre dado enseñan al ordenador una nueva instrucción. Esta se define mediante PARA nombre seguido de la lista de ordenes que forman el cuerpo del procedimiento y acabado con un FIN. Si se desea introducir parametros de entrada al procedimiento, estos han de ir a continuacion del nombre precedidos del signo ":".Por ejemplo, si en el caso anterior deseamos dibujar hexagonos de distintos lados, definiriamos HEXAGONO de la forma:

```
PARA HEXAGONO :LADO
REPITE 6 [AVANZA :LADO  GIRADERECHA 60]
FIN
```

Y si ejecutamos para diferentes valores de "LADO:

```
HEXAGONO 20
HEXAGONO 30
HEXAGONO 40
```

obtendriamos los tres hexagonos de la figura

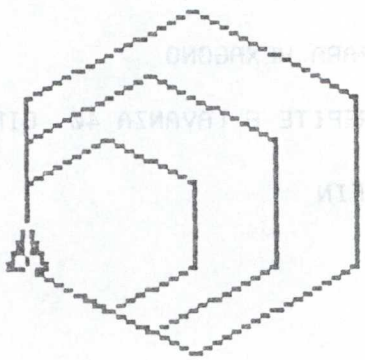


fig tres hexagonos

A su vez el procedimiento HEXAGONO se podria utilizar para definir otro procedimiento, por ejemplo, si definimos SUPERHEXAGONO como el grafico formado por seis hexagonos girados 60 grados entre si

```
PARA SUPERHEXAGONO :LADO
REPITE 6 [HEXAGONO :LADO  GIRADERECHA 60]
FIN
```

dando lugar al ejecutarlo al diseño de la figura

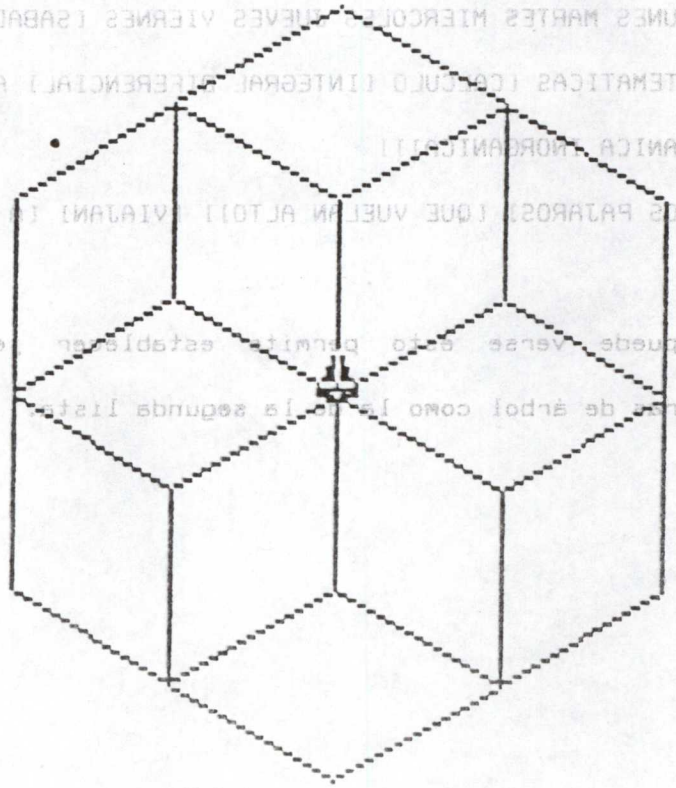


fig SUPERHEXAGONO

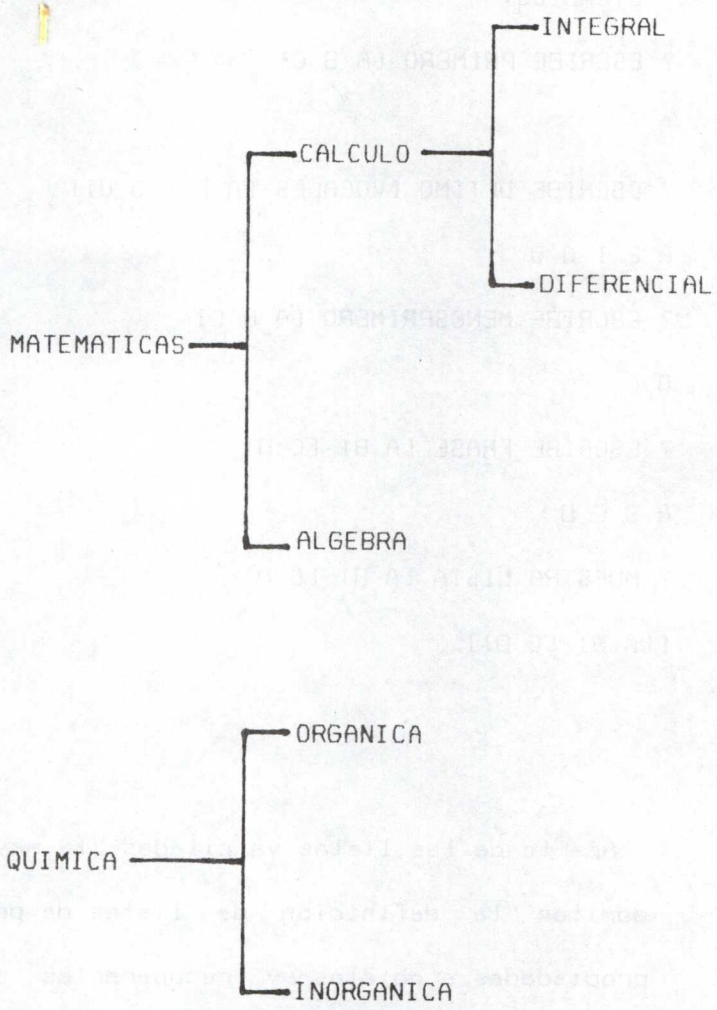
A su vez este procedimiento podria utilizarse para definir otro y asi sucesivamente. Esta estructura procedimental permite, como veremos mas adelante, realizar una programacion modular.

Tratamiento de Listas

La estructura básica en la que se basa Logo es la lista , no en vano es un heredero del lenguaje LISP (LISt Processing). En realidad los propios procedimientos son tratados como listas por el intèrprete.Esta estructura de lista es muy general, ya que sus elementos pueden ser a su vez listas. Algunos ejemplos de listas son:

```
[LUNES MARTES MIERCOLES JUEVES VIERNES [SABADO DOMINGO]]
[[MATEMATICAS [CALCULO [INTEGRAL DIFERENCIAL] ALGEBRA] [QUIMICA
[ORGANICA INORGANICA]]]
[[[LOS PAJAROS] [QUE VUELAN ALTO]] [VIAJAN] [A TIERRAS LEJANAS]]]
```

Como puede verse ésto permite establecer jerarquías de datos, estructuras de árbol como la de la segunda lista.



Estas estructuras de listas básicas en los lenguajes de Inteligencia Artificial,permiten realizar funciones inteligentes como búsquedas en estructuras de árbol, análisis sintáctico, etc.

Las órdenes básicas para para manejo de listas son: extracción del primer elemento (PRIMERO), idem del último (ULTIMO), o de un elemento cualquiera (ITEM), eliminación del primer elemento de una lista (MENOSPRIMERO), idem del último (MENOSULTIMO), número de elementos de una lista (CUENTA), añadir un elemento al final de una lista (PONULTIMO) o al principio de ella (PONPRIMERO), concatenación de dos listas (FRASE) y creación de una lista cuyos elementos son sus entradas (LISTA).

Ejemplos:

```
? ESCRIBE PRIMERO [A B C]

A

? ESCRIBE ULTIMO [VOCALES [A E I O U]]

A E I O U

? ESCRIBE MENOSPRIMERO [A B C]

B C

? ESCRIBE FRASE [A B] [C D]

A B C D

? MUESTRA LISTA [A B] [C D]

[[A B] [C D]]
```

Aparte de las listas ya citadas, la mayoria de las versiones de Logo admiten la definición de listas de propiedades que permite asignar propiedades a objetos y recuperarlas después, siendo este tipo de listas de gran poder para establecimiento de bases de hechos.

Recursivo.

Una caracteristica importante de Logo es que es recursivo, y no sólo lo es, sino que favorece el uso de la recursión como técnica de programación, lo que da una gran potencia al lenguaje. Pero veamos en que consiste la recursión.

Un procedimiento se dice que es recursivo cuando se define en función de si mismo.

Veremos más adelante cómoo se pueden generar gráficos utilizando la recursión y cómo el manejo recursivo de listas es una de las técnicas de programación que hace de Logo un lenguaje de alto poder para procesar estructuras de datos complejas, ya que la definición de un problema en función de si mismo permite reducir un problema a una versión reducida de sí mismo, llegando a la solución por sucesivas reducciones hasta llegar a un problema trivial.

Extensible

La posibilidad de definir nuevos procedimientos, ampliando así el vocabulario que conoce el ordenador, hace que Logo sea un lenguaje extensible que crece conforme a nuestras necesidades, pudiendo definir lenguajes a la medida del usuario.

Uso fácil y agradable

Su interactividad, junto con un editor de pantalla cómodo y unos mensajes de error claros hacen de Logo un lenguaje de uso sencillo y agradable. Ejemplos de algunos de estos mensajes de error son :

NO SE LO QUE HACER CON ..... EN .....

NO SE LO QUE ES ..... EN .....

Programación Modular

La estructura de Logo basada, como ya hemos visto, en los procedimientos, permite realizar programas de forma modular, es decir, descomponiéndolos en partes que pueden resolverse separadamente y más simples que el problema inicial. Esta modularización puede hacerse a varios niveles, ya que cada módulo se puede, a su vez, descomponer en partes bien sencillas. Este proceso puede dar una respuesta a los planteamientos de la programación estructurada. En contraposición a otros lenguajes, que como BASIC, dan lugar a programas poco entendibles, Logo permite y estimula la realizacion de programas claros, en los que cada procedimiento tiene una misión definida. En el caso de listas, de la misma forma, cada procedimiento realiza una función definida, que da una salida a otros procedimientos, permitiendo, con un estilo similar al de LISP una técnica de programación orientada a las funciones.

En el caso de gráficos, la modularización resulta clara, ya que cada módulo representa una parte del gráfico. Por ejemplo, para dibujar la CIUDAD de la figura los módulos con los que se puede construir son procedimientos que dibujan triángulos y rectángulos de dimensiones variables, y que combinados adecuadamente dan lugar al procedimiento CASA, y al ejecutarlo con parámetros diferentes dibuja casas de diferente tamaño, y siguiendo el mismo proceso de modularización se obtiene la ciudad deseada.

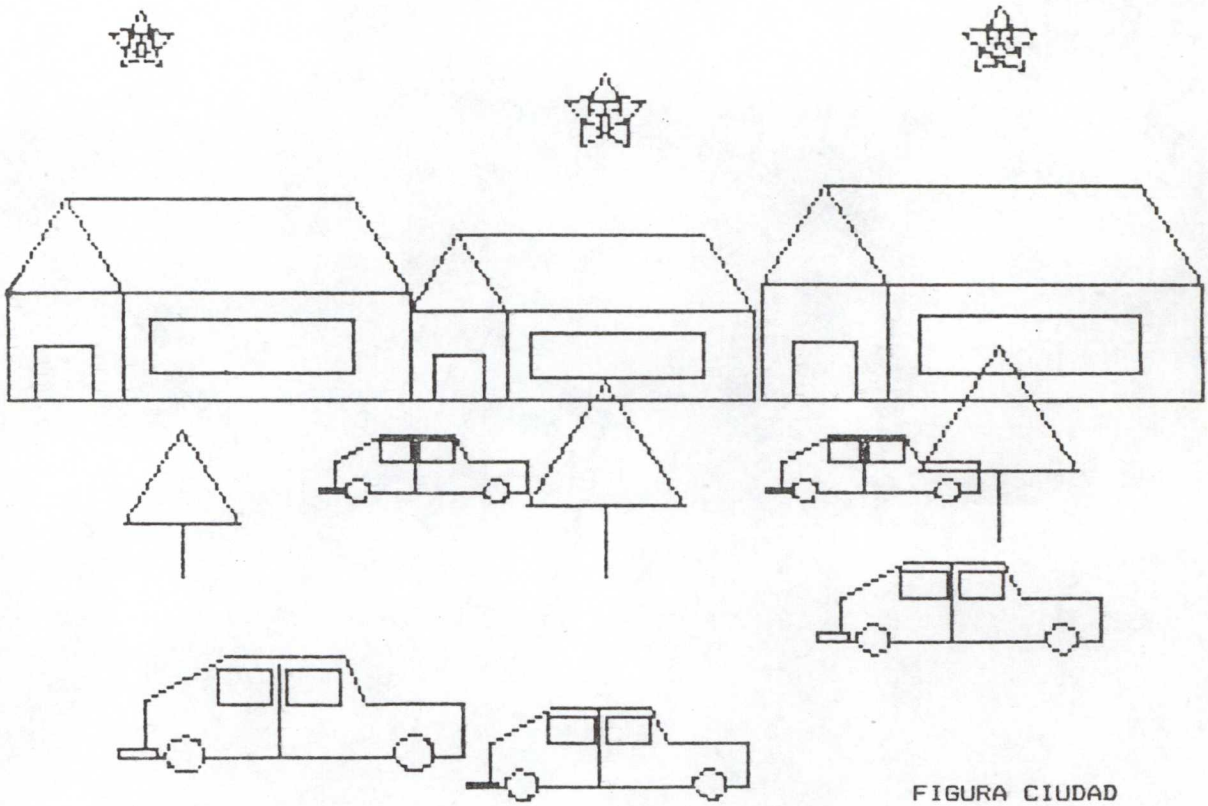


FIGURA CIUDAD



APLICACIONES DIDACTICAS DEL LENGUAJE LOGO

Aparte de las aplicaciones generales descritas de desarrollo de la capacidad de resolución de problemas, de "aprender a aprender", de instrumento para el desarrollo del pensamiento formal, trataremos de dar una visión más concreta de las posibilidades de Logo, y para ello revisaremos las aplicaciones que se han desarrollado en Logo para las distintas áreas.

Logo y la informática

Ya se ha comentado a lo largo del articulo, que Logo es un lenguaje muy adecuado para iniciarse en la Informática. En efecto, es muy fácil perder el miedo a los ordenadores y realizar pequeños programas que funcionen desde el principio y eviten frustraciones al principiante, y que más tarde le permita avanzar progresivamente. Para ello, este lenguaje es ideal. Pero, ¿qué aporta Logo al aprendizaje de la Informática?

Permite aprender a manejar el teclado, los distintos modos de trabajo como edición y ejecución, el concepto de algoritmo, de procedimiento, de técnica modular de programación. La realimentación que se produce al contrastar un diseño con su realización mediante la programación del algoritmo correspondiente es inmediata, al tratarse de un lenguaje interactivo.

Logo da una respuesta a los planteamientos de la programación estructurada al favorecer la construcción modular de los programas, lo que permite el aprendizaje de una técnica de programación fundamental.

Logo permite también manejar las estructuras básicas de control, como el condicional, y comprender la recursión y sus posibilidades como técnica de programación así como las variables y las estructuras de listas.

Las posibilidades del lenguaje permiten, incluso, una iniciación a la Inteligencia Artificial, pudiendo desarrollar microsistemas expertos de modo análogo a como se hace en LISP.

Logo y las matemáticas

Ya se comentó cómo el origen de Logo se encuentra en las preocupaciones de Papert por las dificultades que, en aprendizaje de las matemáticas, encuentran gran cantidad de alumnos. Por tanto, este es el área, aparte de la propia Informática, en la que es más evidente su aplicación.

La idea de Papert consistió en dotar al alumno de un lenguaje sencillo, pero que permitiera ampliar la capacidad de manipulación de simbolos de los alumnos y, por tanto, capaz de expresar ideas complejas y que a la vez dotara de poder al

alumno; es decir, que en lugar de que, como ocurre en la enseñanza de las matemáticas habituales, no se tratara de manejar conceptos abstractos, al margen del mundo del niño, sino de "hacerle matemático", dándole posibilidades de realizar cosas concretas con las ideas abstractas. Por otra parte, Logo, al ser un lenguaje altamente formalizado, es en sí mismo una estructura matemática, y expresándose en este lenguaje, el alumno adquiere un modo matemático de pensamiento.

Bajando a niveles más concretos, podemos distinguir dos tipos de aprendizaje, el que se refiere al curriculum "tradicional" de matemáticas y el propio de un nuevo enfoque del curriculum de matemáticas, que Logo propicia.

El nuevo enfoque curricular tiene relación con la Teoria General de Sistemas. En efecto, el trabajo en Logo hace que se parta del concepto de "estado" de la tortuga, definido por su posición y orientación, y todo se obtenga mediante variaciones efectuadas respecto al estado anterior, obteniéndose, por ejemplo, un circulo mediante pequeños y constantes desplazamientos seguidos de giros, en lugar de la más cartesiana definición de puntos que equidistan de un lugar llamado centro. En este aspecto, es de destacar el libro de Abelson y DiSessa, "Turtle Geometry" (Geometria de la tortuga), considerado por Papert como el primer libro de las nuevas matemáticas. La denominada "geometria de la tortuga" queda conectada con la cartesiana y la vectorial mediante las órdenes que, como PONX, PONY, PONPOS, PONRUMBO, que

permiten situar la tortuga en el punto de coordenadas y rumbo dado, y las funciones COORX, COORY, POS y RUMBO, que nos permiten conocer las coordenadas cartesianas y la orientación de la tortuga.

Otros ejemplos más "clásicos" de aplicaciones en este área que permite Logo serían: cambiar de base entre sistemas de numeración, operaciones con vectores, simetrías, operaciones con conjuntos manejando las estructuras de lista, representación de funciones, y muchas más que, sin duda, los profesores de matemáticas no tardarán en descubrir al conocer las posibilidades del lenguaje.

La idea que subyace en el aprendizaje a través de este lenguaje es la de concreción del pensamiento formal, la de posibilitar un entorno manipulativo y activo con el que el alumno experimenta y desarrolla estrategias de pensamiento y de resolución de problemas

Logo y las ciencias

Son muy numerosas las aplicaciones del lenguaje Logo a las Ciencias, la gran mayoría basadas en simulaciones de fenómenos físicos, químicos o biológicos. Estas simulaciones comprenden las clásicas a las que estamos acostumbrados a ver realizadas en otros lenguajes, tales como caída de cuerpos, parábolas de tiro,

movimiento de fluidos, etc, ya que, naturalmente, son realizables en Logo. Péro, además de las descritas, destacaremos algunas que representan un enfoque diferente consistente en estar hechas "desde" la perspectiva de Logo; es decir, tratando de crear ambientes de aprendizaje interactivo, micromundos sobre los que el alumno puede operar.

Simulación del movimiento de los animales

Uno de los trabajos de mayor interés en el campo de las ciencias es el desarrollado por Abelson y Goldenberg, que desarrollan modelos computacionales del comportamiento animal. La metodología consiste en que el alumno realiza un modelo de un fenómeno biológico, lo cual es, sin duda, una actividad intelectual altamente elaborada e interdisciplinaria, ya que incluye aspectos de biología, física, matemáticas y ordenadores. La realización de este modelo se va efectuando a partir de lo más simple, movimiento al azar, a lo más complejo, seguimiento olfativo o visual con pieza estática o móvil (persecuciones).

En la figura se muestra el gráfico resultante cuando la presa realiza un desplazamiento en circulo ajeno a las intenciones del perseguidor, que la sigue por la vista.

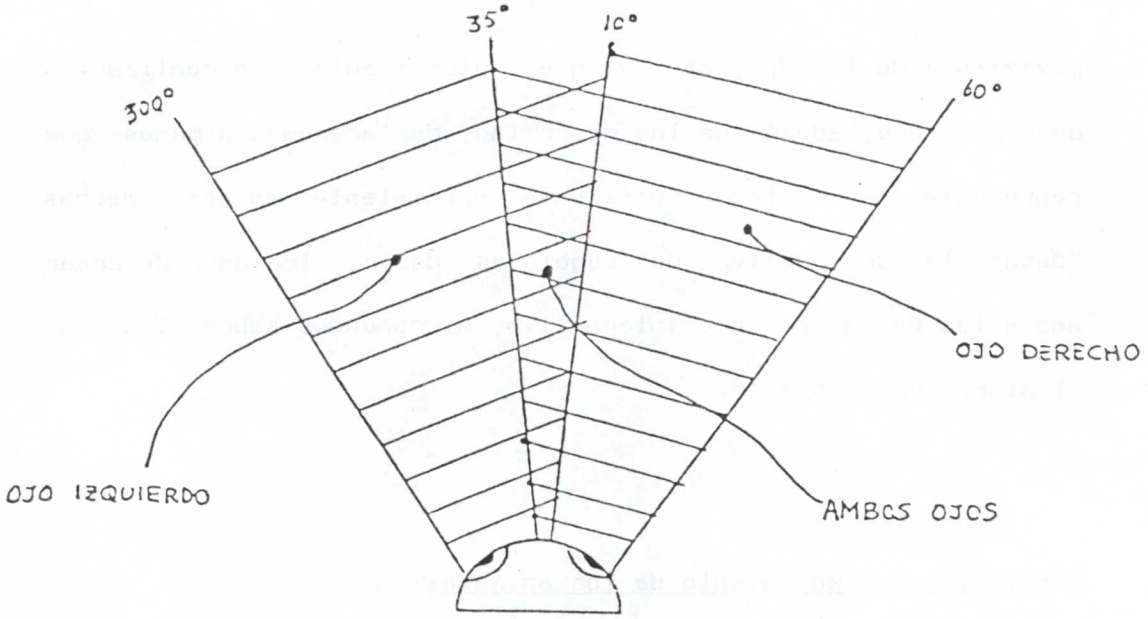


FIG. Modelo de Vision con dos Ojos

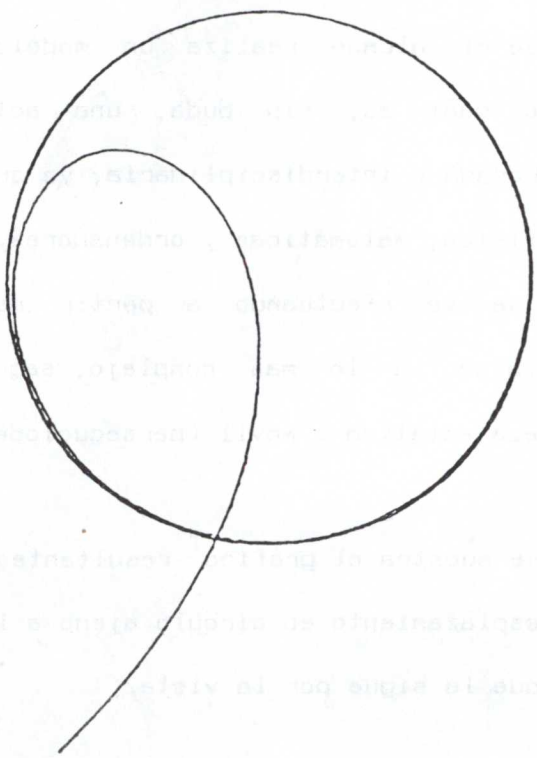


FIG. Persecucion

El modelo utilizado puede ser el correspondiente a la visión con dos ojos que indica la figura, y una estrategia consistente en girar en el sentido adecuado cuando se sale del campo visual.

Alas para la tortuga

Existen distintos desarrollos de micromundos que se basan en dotar a las tortugas de propiedades físicas, de forma que se convierten en objetos manipulables con los cuales experimentar y aprender física de forma activa. Uno de los ejemplos más clásicos es el de las dinatortugas, o tortugas dinámicas, que son objetos a los que se les hace cumplir las dos primeras leyes de Newton, de manera que los alumnos pueden aprender comceptos como la inercia o la fuerza centrífuga. En efecto, un alumno que pretende dirigir a una tortuga dinámica dándole un empuje en una dirección determinada, comprueba de manera inmediata que el movimiento que se produce es realmente el resultante de la composición de dos fuerzas, la que él ha suministrado y la de inercia. De este modo, puede ir desarrollando estrategias que le permitan pilotar estas tortugas de acuerdo con sus deseos, por ejemplo si desean que realicen un movimiento circular han de seguir una estrategia cuya justificación última se encuentra en el concepto de fuerza centrífuga.

Logo y la lengua

Las aplicaciones del lenguaje Logo a la lengua pueden ser de dos tipos: Por un lado, las más tradicionales de listas diccionario, construcción de juegos de palabras, etc., tradicionalmente realizadas en todos los lenguajes de programación. Por otro lado, las que son más frecuentemente realizadas en entornos Logo, como la generación de escritura aleatoria o la generación de historias. La escritura al azar consiste en generar, combinando listas de sustantivos, verbos, etc., frases o rimas. Estas actividades, con un caracter de juego, permiten a los alumnos conocer de forma divertida los distintos tipos de combinaciones que se pueden hacer con las palabras y realizar trabajos sobre la sintaxis y la concordancia. Los estudios gramaticales que se pueden efectuar permiten generar gramáticas o efectuar análisis sintácticos de oraciones que se adapten a una determinada gramática definida por unas leyes.

Otra aplicación que aparece frecuentemente es la de la escritura "creativa" producida mediante un "generador de historias", en las que el alumno puede ir creando sus propias historias en las cuales, utilizando un método de selección en árbol, va realizando las distintas posibilidades o combinaciones de sucesos, recurriendo al árbol en situaciones correspondientes que vaya eligiendo al contestar los mensajes que le plantea el ordenador. Este tipo de escritura hace reflexionar mediante una actividad de juego sobre el proceso de construcción de una historia y sus múltiples posibilidades.

Logo y musica

Los trabajos realizados por Jeanne Bamberger han revolucionado la enseñanza de la música mediante el ordenador. Sus trabajos se basan en acoplar a un ordenador una caja de música o sintetizador musical conectado a un amplificador estéreo. El sistema trabaja con la base del lenguaje Logo, y para la generación de sonidos dispone de seis voces, con un rango de 6 octavas, 8 parametros que permiten modificar la envolvente, sonidos de percusión, etc.

La orientación del sistema posee el espiritu de Logo; es decir, se ha diseñado para trabajar de modo intuitivo y sencillo, pero permite también realizar un trabajo profesional para la composición interactiva. Hay una diferencia fundamental entre la Música-Logo y los métodos tradicionales de música por ordenador, y es el enfoque cognitivo que representa al desarrollar un entorno que permite el pensamiento musical intuitivo y su modo de trabajo verdaderamente interactivo.

El enfoque procedimental permite a los alumnos trabajar a diferentes niveles de estructura musical. Se pueden diseñar micromundos de forma que el alumno puede modificar los parámetros básicos: secuencias, relaciones de intervalo, etc.

Asi, la misma estructura de un procedimiento que permite dibujar espirales, cambiando las órdenes ADELANTE por otras musicales como BOOM o PLAY en el sistema musical descrito, tienen una función distinta, pero ambas comparten la misma estructura profunda:

```
PARA SUBIRTONO :TONO :INCREMENTO
SI :TONO >12 [ALTO]
PLAY :TONO 6
SUBIRTONO :TONO + :INCREMENTO :INCREMENTO
FIN
```

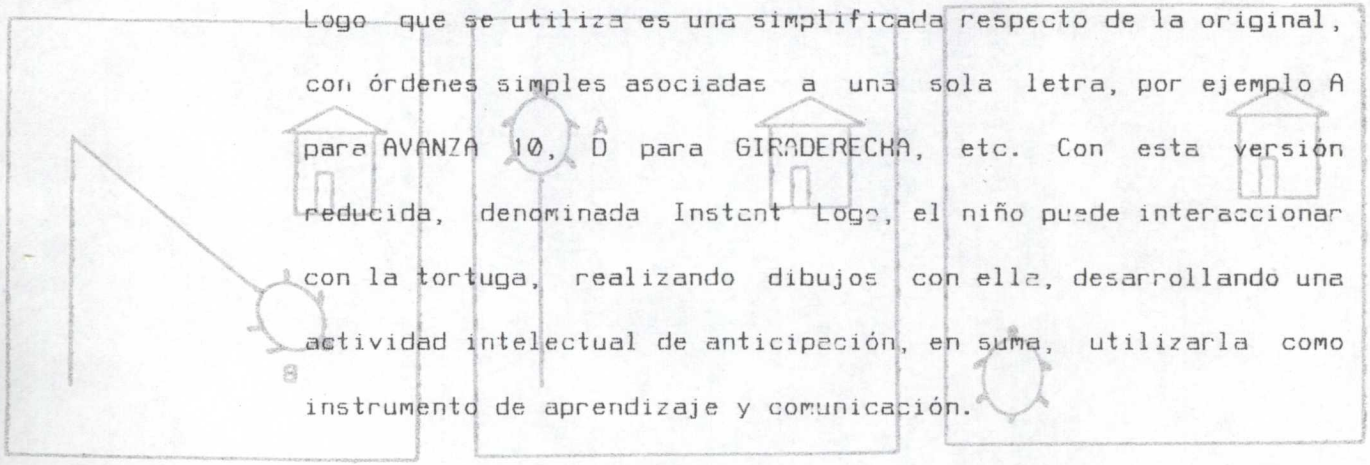
Logo y educación especial

Se han realizado numerosas experiencias en el terreno de la Educación Especial utilizando el lenguaje Logo. Estas investigaciones, de las que destacamos la llevada a cabo en el M.I.T. por Sylvia Weir, han puesto de manifiesto las grandes posibilidades de los ordenadores para la educación de niños disminuidos.

El ordenador, dada su gran flexibilidad, posee un gran potencial educativo en la enseñanza de niños especiales, tanto en caso de disminuidos físicos, como psiquicos o con dificultades de comunicación.

Logo produce un entorno de aprendizaje que por su interactividad, la naturaleza estructurada de las actividades, la realimentación inmediata, sus posibilidades de concretar ideas abstractas, es muy adecuado para este tipo de niños, a los que el ordenador puede introducir en mundos con los que construir su pensamiento a los que, de otro modo, por dificultades de comunicación, no tendrían acceso. De acuerdo con lo anterior, se han logrado resultados interesantes con niños autistas, disléxicos y paralíticos cerebrales, entre otros.

En los casos de niños con dificultades motoras, se suelen utilizar teclados especiales que, con grandes pulsadores y muy separados, permiten una comunicación más simple. La versión de Logo que se utiliza es una simplificada respecto de la original, con órdenes simples asociadas a una sola letra, por ejemplo A para AVANZA 10, D para GIRADERECHA, etc. Con esta versión reducida, denominada Instant Logo, el niño puede interaccionar con la tortuga, realizando dibujos con ella, desarrollando una actividad intelectual de anticipación, en suma, utilizarla como instrumento de aprendizaje y comunicación.



La flexibilidad del ordenador permite una educación a la medida. Por ejemplo, un profesor puede realizar pequeños procedimientos que efectúen una serie de dibujos de casas, árboles, etc., y programar la máquina de forma que asocie una tecla a cada procedimiento, y así, basta con el niño teclee "C" para que la tortuga dibuje una casa, o "A" para un árbol. Más adelante puede

hacer que el niño escriba el nombre completo de la palabra para que aparezca el correspondiente dibujo, y así progresivamente, de acuerdo con el nivel que vayan alcanzando los niños.

En la figura se muestra la interacción de un niño paralítico cerebral con un programa que dibuja una casa y con la tortuga en un punto al azar, y el niño debe dirigirla con órdenes sencillas, de forma que al pulsar A avanza, D gira a la derecha e I gira a la izquierda. En la segunda viñeta se le dió una indicación mediante la pregunta: ¿Estás seguro de que vas a la casa? En la tercera viñeta, trata de rectificar infructuosamente. Estos primeros pasos pueden ir dando paso, poco a poco, a actividades más inteligentes, de forma que el niño va ganando autocontrol.

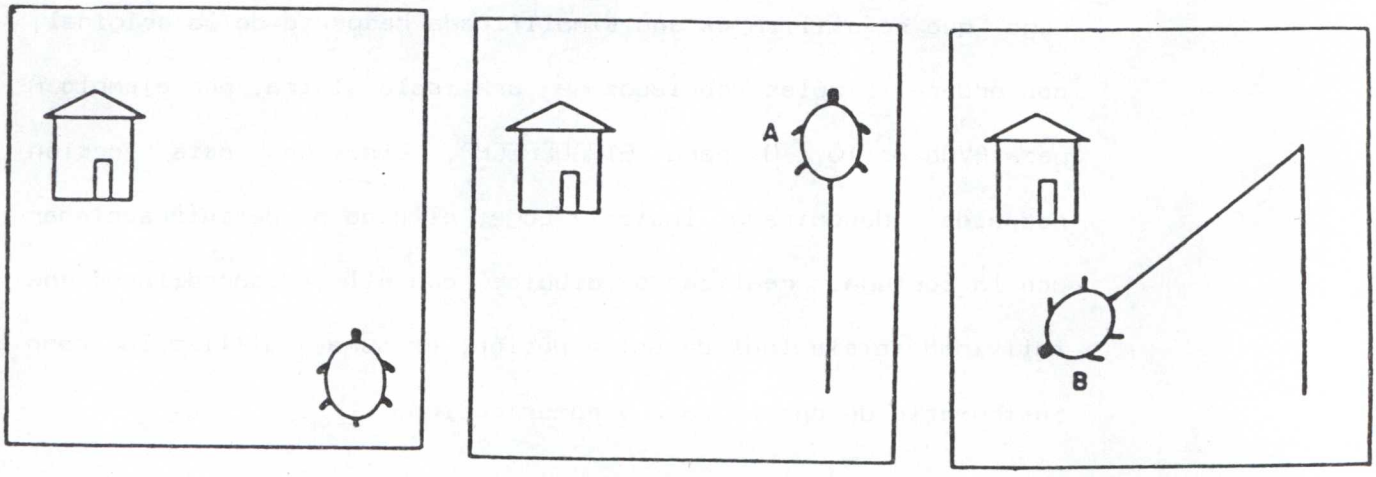


FIG. Interacción con el ordenador de un niño paralítico cerebral

Logo y la inteligencia artificial

En lenguaje Logo, dada su estructura de listas similar a la de los lenguajes de Inteligencia Artificial, se pueden realizar programas "inteligentes" de resolución de problemas, representación de conocimientos o sistemas expertos.

RESOLUCION DE PROBLEMAS

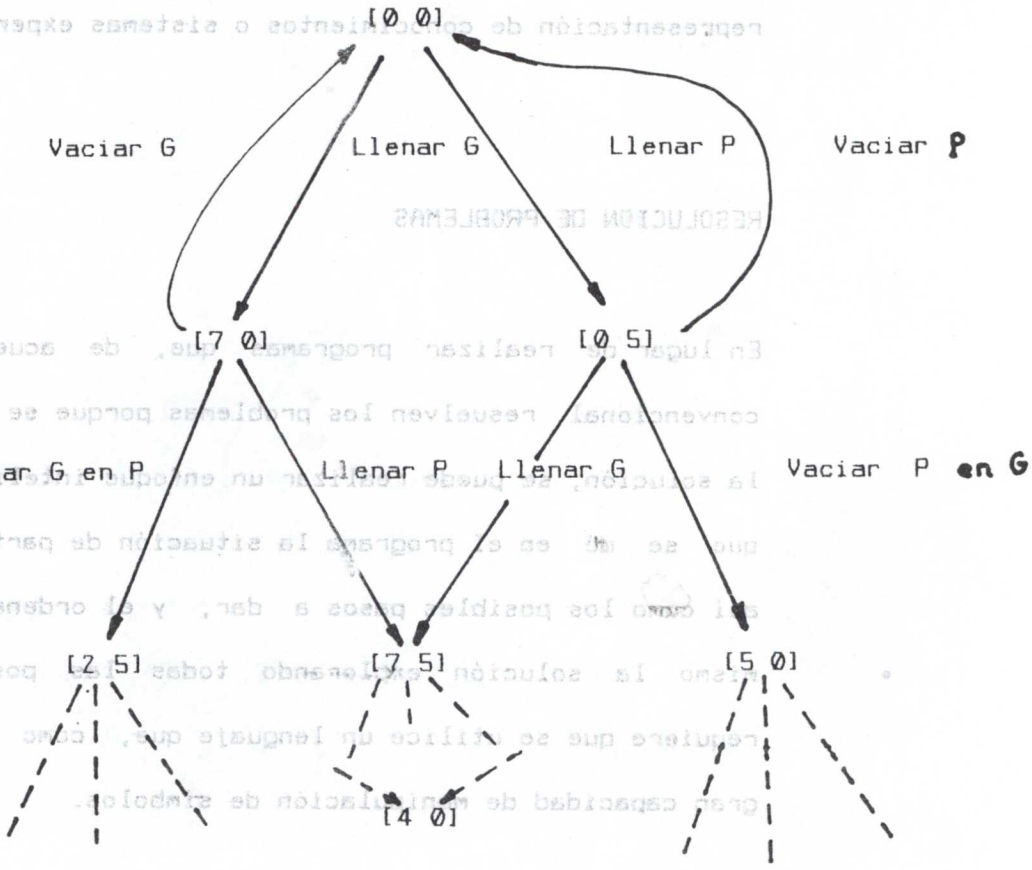
En lugar de realizar programas que, de acuerdo con el método convencional, resuelven los problemas porque se les ha programado la solución, se puede realizar un enfoque inteligente, de forma que se dé en el programa la situación de partida y la deseada, así como los posibles pasos a dar, y el ordenador busque por sí mismo la solución explorando todas las posibilidades. Esto requiere que se utilice un lenguaje que, como Logo, tenga una gran capacidad de manipulación de símbolos.

Ejemplos de este tipo de problemas se describen en lo que sigue:

Problema de los depósitos:

"Se tienen dos depósitos de 7 litros y 5 litros de capacidad respectivamente. Se desea realizar una medida exacta de 4 litros, para lo cual se dispone de una fuente de agua sin limitación, pudiéndose realizar cualquier operación: llenar un depósito, vaciarlo, vaciar uno en otro, etc.".

A partir del dato de partida, ambos depósitos vacíos ([0 0]), el programa debe ir recorriendo en el árbol de posibilidades hasta llegar a la situación deseada, en la que uno de los depósitos tiene 4 litros. En la figura se muestra el esquema de las posibilidades que ha de ir analizando el programa.

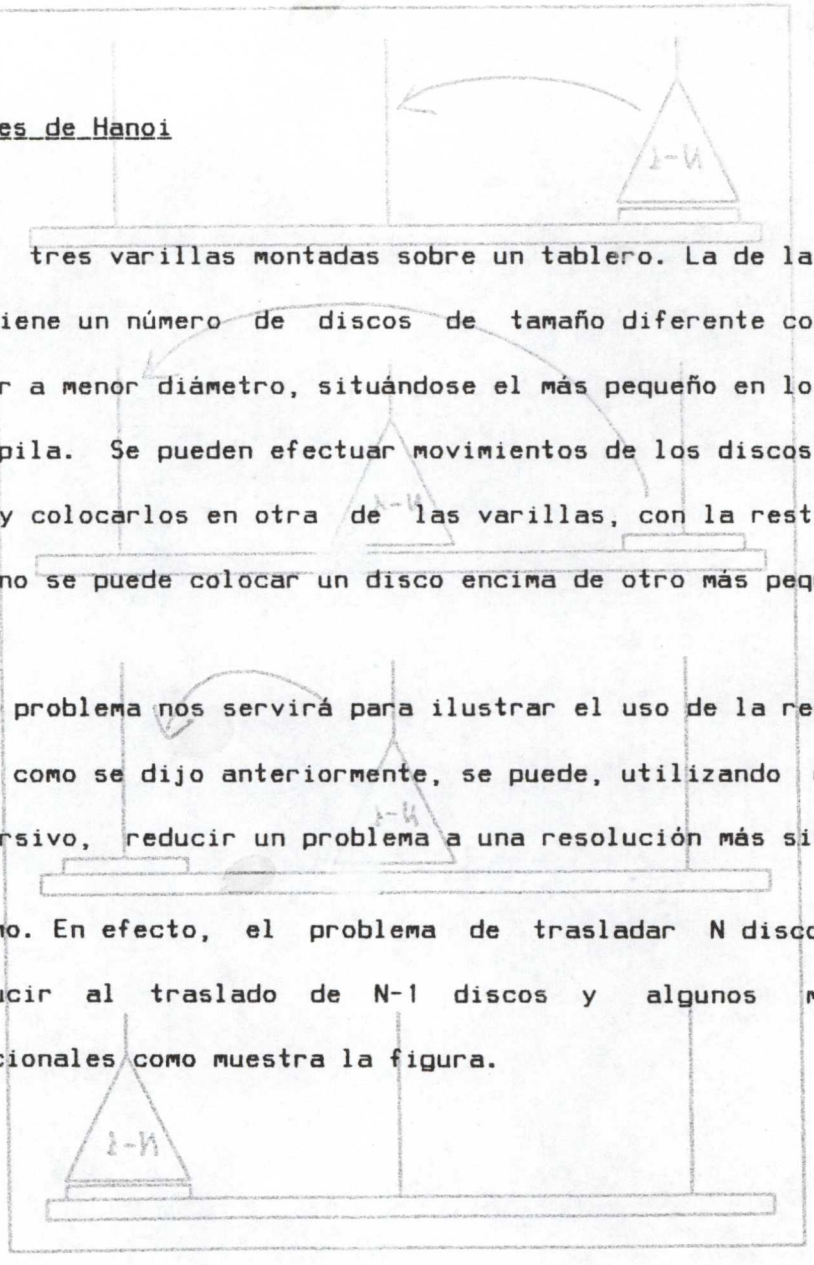


Problemas como el descrito son fácilmente programables en Logo, utilizando las estructuras de listas y la recursión, realizando un proceso de búsqueda combinatoria siguiendo un recorrido por el árbol y dando marcha atrás si se llega a un punto por el que ya se ha pasado o no tenga salida.

Torres de Hanoi

"Hay tres varillas montadas sobre un tablero. La de la izquierda sostiene un número de discos de tamaño diferente colocados de mayor a menor diámetro, situándose el más pequeño en lo alto de la pila. Se pueden efectuar movimientos de los discos de uno en uno y colocarlos en otra de las varillas, con la restricción de que no se puede colocar un disco encima de otro más pequeño".

Este problema nos servirá para ilustrar el uso de la recursión ya que, como se dijo anteriormente, se puede, utilizando un método recursivo, reducir un problema a una resolución más simple de sí mismo. En efecto, el problema de trasladar N discos se puede reducir al traslado de N-1 discos y algunos movimientos adicionales como muestra la figura.



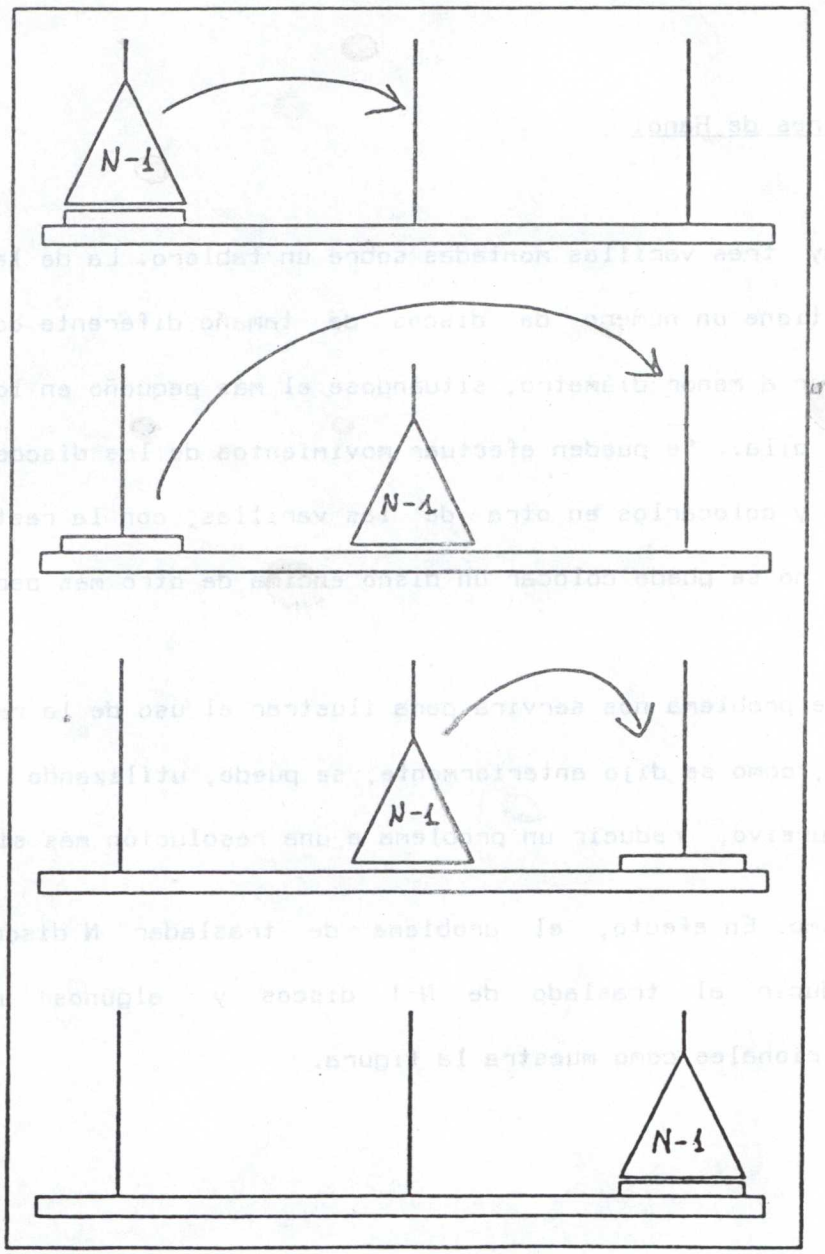


FIG. TORRES DE HANOI

De acuerdo con la idea anterior, el procedimiento Logo sería:

```
PARA HANOI :NUMERO :INICI :INTER :META
SI :NUMERO=0 [ALTO]
HANOI :NUMERO-1 :INICI :META :INTER
MOVERDISCO :INICI :META
HANOI :NUMERO-1 :INTER :INICI :META
FIN
```

El uso de técnicas como las descritas: búsquedas en árboles, recursión, etc., permiten desarrollar programas para representar conocimientos, o sistemas expertos en los que, a partir de una base de conocimientos adecuadamente representados y de unas reglas, se infieren nuevos conocimientos. Es, por tanto, Logo un lenguaje que, por su sencillez, permite iniciarse en el estudio de la Inteligencia Artificial.

METODOLOGIA DE TRABAJO EN LOGO

El método seguido en la utilización del lenguaje en las aulas es de la mayor importancia, ya que la simple enseñanza de Logo como lenguaje de programación limitaría considerablemente sus posibilidades educativas. Coexisten en la actualidad enfoques muy diferentes, desde el aprendizaje por descubrimiento puro, hasta la enseñanza dirigida, la utilización de micromundos cerrados para el aprendizaje de conceptos concretos o las aplicaciones abiertas que persiguen un desarrollo más creativo.

Cabe destacar como posibilidad a desarrollar un enfoque de integración en el curriculum, de modo que se creen diversos subconjuntos del lenguaje con unos objetivos concretos: aprendizaje geométrico, de conceptos generales (función, variable, conjuntos, etc.), entre otros. Este enfoque permite subordinar el lenguaje a unos objetivos específicos, por una parte, y lograr, por otra parte, objetivos generales de resolución de problemas y desarrollo de capacidades lógico-matemáticas. La metodología de este tipo de aplicaciones

también puede ser diversa: hojas de trabajo de alumno que orienten la secuencia de pasos a seguir o trabajo más libre basado en la construcción con las "piezas de mecano" que se proporciona a los alumnos para que elabore sus propios proyectos. Un ejemplo de este tipo de entornos de aprendizaje puede ser un conjunto de procedimientos para el trabajo con vectores, que el alumno puede combinar para realizar un aprendizaje activo de la geometria vectorial. En este libro se muestran algunos ejemplos de aplicación.

CONSIDERACIONES FINALES

Casi nadie pone en duda las grandes posibilidades de Logo como un instrumento educativo poderoso y sencillo a la vez, y el lector de este trabajo habrá podido hacerse una idea precisa de sus posibilidades, tanto desde el punto de vista informático como eductivo. Sin embargo, es preciso realizar algunas puntualizaciones y plantear algunos interrogantes que, si somos sinceros hemos de reconocerlo así, aún no están resueltos en paises con una mayor tradición en el uso de este lenguaje.

La programación en Logo realizada por niños pequeños presenta algunas dificultades ajenas a la propia estructura del lenguaje, como es el caso de los mensajes de error. Un mensaje del tipo "NO SE QUE ES AV50", no expresa claramente el verdadero error que ha cometido el niño y que es simplemente que se le ha olvidado poner un espacio en blanco entre la orden AV y el número 50. Otro fallo

que puede tener una incidencia negativa en el proceso de aprendizaje estriba en el hecho de que el niño, una vez que aprende a manejar algunos "trucos" del lenguaje para realizar gráficos, las típicas espirales de todo tipo, se dedica a repetirlas jugando con ellas, sin entender con precisión los mecanismos por los que se realizan estos dibujos. A este respecto es muy importante que el profesor haga explícito lo que se pretende realmente, es decir, que si mediante un programa para dibujar cuadrados de diferente tamaño lo que realmente se desea es que el alumno capte el concepto de variable y lo maneje, lo diga claramente, ya que así hará que el alumno capte la verdadera dimensión del trabajo que está realizando.

Otro punto a tener en cuenta es que el manejo recursivo de listas, siendo muy potente y elegante formalmente, puede, que por su nivel de abstracción, sea difícilmente entendible por el alumno, pudiendo crearle frustraciones. A este respecto es importante tener en consideración que es importante realizar programas y enseñar técnicas de programación que sean una ayuda, desde el punto de vista psicológico, antes que elegante desde un punto de vista formal.

La mayor dificultad del manejo de listas en Logo ha hecho que la mayoría de los profesores se hayan limitado en sus aulas a utilizar las gráficas de tortuga únicamente, con lo cual quedan disminuidas las posibilidades del lenguaje.

Puede ocurrir, al introducir el lenguaje Logo en el curriculum, un proceso similar al ocurrido con la matemática moderna, y es que una herramienta poderosa sin una adecuada preparación previa de los profesores, tanto en los aspectos de conocimientos técnicos como de la filosofía subyacente, devalúe el verdadero interés que Logo posee como instrumento de aprendizaje.

Como consideración final, resaltaré la importancia del uso del lenguaje materno en Logo, y por tanto, es imprescindible que las máquinas en las que se trabaje posean una versión en la lenguaje habitual de los alumnos para no añadir a las dificultades del lenguaje informático las derivadas del uso de una lengua extranjera.

PRIMITIVAS Y PROCEDIMIENTOS

GRAMATICA DEL LOGO

El LOGO es un lenguaje construido con bloques que se pueden combinar de acuerdo con ciertas reglas que constituyen su gramática. Hay que aprender estas reglas para dar al ordenador las instrucciones adecuadas.

PROCEDIMENTOS INCORPORADOS

Los bloques constitutivos del lenguaje LOGO son los procedimientos. A los procedimientos que ya se encuentran en el lenguaje LOGO se les denomina primitivas, no hay que definirlas, el ordenador las "entiende" directamente. Por ejemplo, si se teclea:

```
?BORRAPANTALLA .  abreviadamente  BP
```

Se ejecuta la acción pedida, porque ya está definido **BORRAPANTALLA** ; es una palabra reservada y se borran todos los gráficos de la pantalla.

Igualmente, las palabras ESCRIBE o **AVANZA** son igualmente reservadas, de modo que si se escribe:

```
? ESCRIBE  "HOLA
```

el ordenador responde imprimiendo

```
HOLA
```

O si se le ordena:

```
?AVANZA  50
```

hace que la tortuga avance 50 pasos en la dirección en la que se encuentre.

ENTRADAS A LOS PROCEDIMIENTOS

Algunos procedimientos no requieren entradas, como es el caso de BORRAPANTALLA BAJALAPIZ, SUBELAPIZ, GOMA, .,...; otros, por el contrario, requieren entradas, como por ejemplo ESCRIBE, y si se escribe ESCRIBE sin darle una entrada, nos aparecerá en pantalla un mensaje de error.

Serían correctos, por tanto:

```
? ESCRIBE "HOLA
? ESCRIBE [HOLA BUENOS DIAS]
? AVANZA 50
? RETROCEDE 20
? GIRADERECHA 45
? PONPOS [50 60]
? SUMA 20 42
```

Otros procedimientos requieren más de una entrada, como por ejemplo SUMA que con dos entradas, por ejemplo 20 y 42 sumaría dichos valores:

```
? SUMA 20  42
```

Este procedimiento SUMA así como otros, también admite tres, cuatro o más entradas.

DIFERENCIA ENTRE COMANDOS Y OPERACIONES

Hay dos tipos de procedimientos en LOGO: los que producen como salida un valor, a los que llamaremos operaciones y los que no producen un valor de salida, a los que llamaremos comandos.

Ejemplos de comandos: ESCRIBE, AVANZA, BORRAPANTALLA, PONPOS
Ejemplos de operaciones: SUMA,AZAR

Así, SUMA3 5 genera una salida (8), y dicha salida ha de ser entrada a otro procedimiento, de forma que globalmente, una línea de lenguaje LOGO debe ser un comando. Por ejemplo, si intentamos ejecutar:

```
?SUMA 3 5
```

nos saldrá un mensaje de error. Por el contrario, sería válido:

```
?ESCRIBE SUMA 3 5
```

Unas operaciones, como vemos, pueden dar salidas a comandos o a operaciones, pero en definitiva, el primer elemento de la línea ha de ser un comando, como hemos visto en el ejemplo anterior.

La operación AZAR genera un número al azar comprendido entre 0 y el número dado menos 1. Por ejemplo: AZAR 5 produce una salida que es un número aleatorios entre 0 y 4. Así, por ejemplo:

```
?ESCRIBE AZAR 5
2
```

o bien AZAR puede dar salida a la operación SUMA. El siguiente ejemplo realiza la suma de dos números generados al azar, uno de ellos menor de 5 y otro menor de 10, e imprime el resultado.

```
?ESCRIBE SUMAAZAR 5 AZAR 10
9
```

En el ejemplo siguiente se imprime "PREMIO si el número generado al azar es un 0.

```
?SI AZAR 2=0 [ESCRIBE "PREMIO]
```

En este caso la proposición AZAR 2=0 da una salida que puede ser verdadera o falsa que es la entrada al procedimiento SI, que es ya un comando que ordena ejecutar el comando ESCRIBE "PREMIO, caso de que la condición sea verdadera.

En el ejemplo siguiente se imprime siempre el mensaje BIEN.

```
?SI2<3 [ESCRIBE "BIEN]
```

VARIABLES

Las variables en LOGO son como cajas con etiquetas con el nombre asociado en el exterior y un contenido en su interior.

El signo : indica en LOGO el contenido de esa caja. Así: :A, :ARBOL, :LADO representa el valor de las variables (contenido de las cajas) cuyos nombres (etiquetas) son A , ARBOL y LADO respectivamente.

Para asignar un valor a una variable (introducir algo en la caja correspondiente) se utiliza el procedimiento HAZ como muestran los siguientes ejemplos:

```
? HAZ "A 20
? HAZ "LADO 50
? ESCRIBE :LADO
50
```

Si queremos dar a una variable el valor que tiene otra, por ejemplo, a la variable LADO le damos el valor que tiene la variable A, sería:

```
? HAZ "LADO :A
? ESCRIBE :LADO
20
```

Igualmente se pueden asignar palabras o listas. Por ejemplo:

```
? HAZ "NOMBRE "JUAN
```

crea la variable NOMBRE y le asigna JUAN.

O bien asignar una lista

```
? HAZ "DIAS [LUNES MARTES MIERCOLES JUEVES VIERNES SABADO]
```

Obsérvese la diferencia existente entre el nombre de una variable ("LADO) su contenido (:LADO), si se escribiera LADO, el ordenador lo interpretaría como un procedimiento.

DIFERENCIA ENTRE VARIABLES GLOBALES Y LOCALES

Cuando se trabaja en LOGO creando una variable con el comando HAZ dicha variable permanecerá en el área de trabajo hasta que se borre. Se llaman variables globales. Hay, por el contrario, otras variables que se definen como entradas de un procedimiento y que permanecen en el área de trabajo sólo mientras se está ejecutando el procedimiento. Así, en el procedimiento SALUDA que se define en el siguiente apartado, la variable "NOMBRE sólo permanece en memoria mientras se ejecuta SALUDA, por tanto la variable "NOMBRE sería local. Por el contrario, si se ejecuta:

```
?. HAZ "NOMBRE "JUAN
```

la variable "NOMBRE sería global.

DEFINICION DE PROCEDIMIENTOS NUEVOS

Si el usuario desea definir un procedimiento, puede hacerlo escribiendo **PARA** seguido del nombre del procedimiento y acabar escribiendo **FIN** en la última línea. Por ejemplo, el procedimiento SALUDA tiene como entrada un nombre y la salida es la impresión de un mensaje nominal como HOLA BUENOS DIAS.

```
PARA SALUDA :NOMBRE
ESCRIBE "HOLA
ESCRIBE [BUENOS DIAS]
FIN
```

y al ejecutarlo:

```
?SALUDA "LUIS
 HOLA
 LUIS
 BUENOS DIAS
```

Si se desea modificar el procedimiento o también para crearlo desde el principio, se puede hacer llamando al EDITOR con el comando EDITA seguido del nombre del procedimiento:

```
?EDITA "SALUDA
```

y seguidamente manejando el teclado realizar las modificaciones o inserciones pertinentes.

Ejemplo:

El siguiente procedimiento (MONEDA) imprime CARA o CRUZ simulando la acción de echar una moneda al aire, generando un número al azar (0 ó 1).

```
PARA MONEDA
HAZ "M AZAR 2
SI :M = 0 [ESCRIBE CARA]
SI :M = 1 [ESCRIBE CRUZ]
FIN
```

Al ejecutar tres veces el procedimiento MONEDA, que es un comando:

```
?MONEDA
CARA
?MONEDA
CARA
?MONEDA
CRUZ
```

Se podría haber mandado ejecutar MONEDA, por ejemplo 5 veces, utilizando el procedimiento REPITE:

```
?REPITE 5 [MONEDA]
CARA
CRUZ
CRUZ
CARA
CARA
```

Ejemplo:

El siguiente procedimiento dibuja un cuadrado de lado 50 en la pantalla:

```
PARA CUADRADO
REPITE 4 [AVANZA 50 GIRADERECHA 90]
FIN
```

Si se desea que el procedimiento dibuje un cuadrado de lado variable según la entrada al mismo, podríamos haber definido el procedimiento CUADRADO de la forma:

```
PARA CUADRADO :LADO
REPITE 4 [AVANZA :LADO GIRADERECHA 90]
FIN
```

Obsérvese cómo las entradas al procedimiento han de estar en la primera línea de la definición y al ser variables tienen el signo : delante.

Así, al ejecutar

```
? CUADRADO 20
```

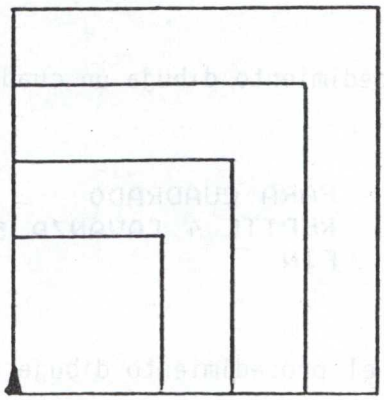
en la pantalla dibuja un cuadrado de lado 20.



Si se ejecuta:

```
? CUADRADO 20
? CUADRADO 30
? CUADRADO 40
? CUADRADO 50
```

La figura que aparece es:



Por tanto, ejecutar CUADRADO 20 tiene el mismo efecto que ejecutar:

```
?REPITE 4 [AVANZA  20 GIRADERECHA 90
```

y sin más que cambiar el valor asociado al comando **AVANZA** nos daría cuadrados de diferente tamaño, o bien se podría hacer simplemente, aplicando el concepto de variable:

```
? HAZ "LADO 20
? REPITE 4 [AVANZA :LADO GIRADERECHA 90]
```

El procedimiento cuadrado es por tanto un comando nuevo, y una vez definido se le puede llamar y se puede considerar como perteneciente al lenguaje.

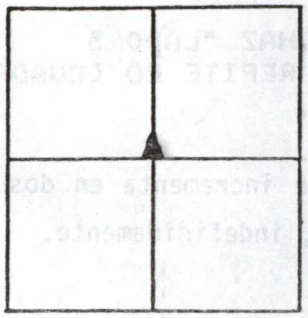
Un procedimiento puede llamar a otro procedimiento, y así por ejemplo, podemos definir el procedimiento SUPERCUADRADO que llama al procedimiento CUADRADO.

```
PARA SUPERCUADRADO :LADO
REPITE 4 [CUADRADO :LADO GIRADERECHA 90]
FIN
```

Y al ejecutar

```
SUPERCUADRADO 40
```

nos dibuja 4 cuadrados de lado 40 como muestra la figura.



Nótese cómo al definir un procedimiento es conveniente hacerlo de tal forma que al ejecutarlo una vez la tortuga quede en la misma posición que tenía al iniciar dicho procedimiento.

Esto equivaldría a ejecutar directamente:

Cuadrado

```
?REPITE 4 [REPITE 4 [AVANZA 40 GIRADERECHA 90] GIRADERECHA 90]
```

Un procedimiento puede llamarse a sí mismo. Así, podemos definir el procedimiento CUADRADOSINFIN

```
PARA CUADRADOSINFIN
REPITE 4 [AVANZA 50 GIRADERECHA 90]
CUADRADOSINFIN
FIN
```

Con lo cual la tortuga recorrería permanentemente el cuadrado de lado 50.

Lo anterior equivale a:

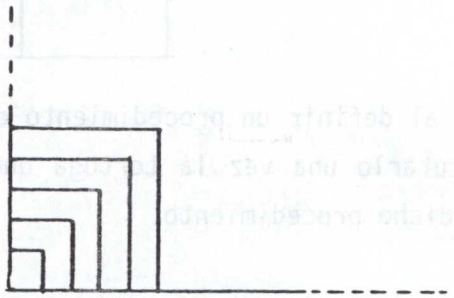
```
PARA CUADRADOSINFIN
  CUADRADO 50
  CUADRADOSINFIN
  FIN
```

Vemos que en este caso hay una llamada al procedimiento CUADRADO y otra a CUADRADOSINFIN, es decir, que equivale a ejecutar: siempre [CUADRADO 50]

Si queremos que el cuadrado aumente de tamaño en cada paso podríamos hacer:

```
? HAZ "LADO 5
? REPITE 20 [CUADRADO :LADO HAZ "LADO :LADO + 2]
```

Con lo cual :LADO se incrementa en dos unidades y dibujaría cuadrados de lado 5, 7, 9, 11,..., indefinidamente.



Si deseamos realizar el procedimiento anterior para valores diferentes de lado inicial e incremento, podemos definir dos variables ("LADO e "INCREMENTO). Lo haremos definiendo un procedimiento con dos variables:

```
PARA CUADRADOSINFIN :LADO :INCREMENTO
CUADRADO :LADO
CUADRADOSINFIN :LADO + :INCREMENTO :INCREMENTO
FIN
```

Con lo cual el procedimiento CUADRADOSINFIN se llama a sí mismo con un "LADO aumentado en el valor de la variable "INCREMENTO. Por tanto, equivale a:

```
PARA CUADRADOSINFIN :LADO :INCREMENTO
CUADRADO :LADO
HAZ "LADO :LADO + :INCREMENTO
CUADRADOSINFIN :LADO :INCREMENTO
FIN
```

Así, al ejecutar:

```
? CUADRADOSINFIN 5 2
```

nos daría la figura anterior.

COMANDOS DE "TORTUGA"

Los comandos más importantes para realizar gráficos de tortuga en LOGO permiten a la tortuga avanzar, retroceder girar a derecha e izquierda, cambiar de color, de forma.

Los más básicos son:

```
AVANZA, en abreviatura AV
RETROCEDE        RE
GIRADERECHA      GD
GIRAIZQUIERDA    GI
```

que nos permiten ya realizar, como vimos antes, gráficos todo lo complejos que queramos.

Ejemplo:

Veamos cómo generaremos un comando que dibuje en la pantalla un hexágono de lado 40.

Basta con hacer avanzar a la tortuga 40 pasos y hacerla girar 60 grados a la derecha. Esto puede realizarse mediante la serie de comandos:

```
? AVANZA 40
? GIRADERECHA 60
        :
        :
        :
? AVANZA 40
? GIRADERECHA 60
```

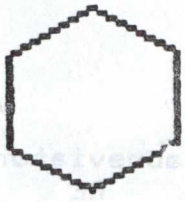
O lo que sería más fácil, utilizando el comando REPITE cuyo formato general es:

? REPITE veces [lista de instrucciones]

En nuestro caso:

```
? REPITE 6 [AVANZA 40 GIRADERECHA 60]
```

En ambos casos se obtiene un hexágono de lado 40.

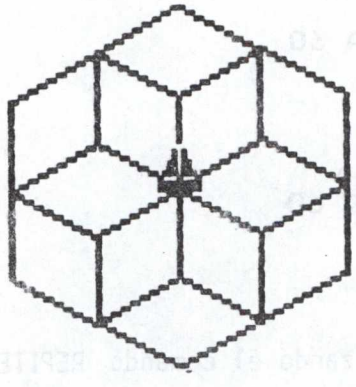


Obsérvese que la tortuga queda en la misma situación de partida (posición y orientación iguales).

Podemos construir una figura compuesta de 6 hexágonos. Basta con repetir 6 veces esta figura y girar 60 grados a la derecha cada vez que dibuja un hexágono:

```
? REPITE 6 [REPITE 6 [AVANZA 40 GIRADERECHA 60] GIRADERECHA 60]
           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        Hexágono
```

Obteniéndose la figura:



Se podría haber llegado al mismo resultado, definiendo un procedimiento al que llamaremos HEXAGONO, de la forma siguiente:

```
PARA HEXAGONO
REPITE 6 [AVANZA 40 GIRADERECHA 60]
FIN
```

Y después haber ejecutado:

```
? REPITE 6 [HEXAGONO GIRADERECHA 60]
```

o haber definido el procedimiento SUPERHEXAGONO y ejecutarlo a continuación:

```
PARA SUPERHEXAGONO
REPITE 6 [HEXAGONO GIRADERECHA 60]
FIN
```

y ejecutarlo con el comando

```
?SUPERHEXAGONO
```

Ejemplo:

Podemos dibujar un círculo simplemente con la siguiente instrucción:

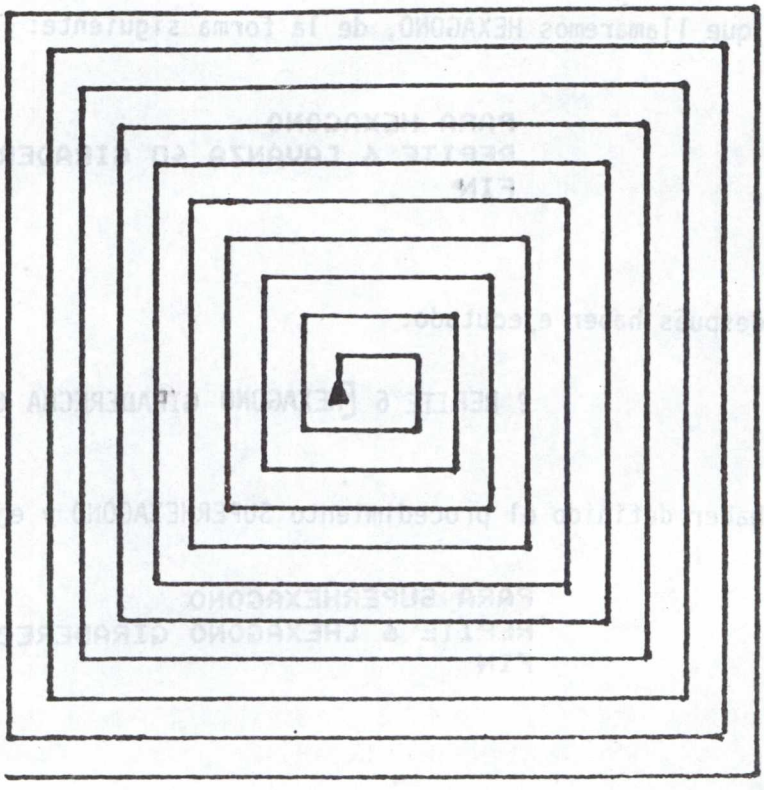
```
? REPITE 360 [AVANZA 1 GIRADERECHA 1]
```



Ejemplo:

Dibujaremos ahora una espiral cuadrada. Para ello, basta con hacer avanzar la tortuga cada vez un paso más largo y girando 90 grados.

```
? HAZ "LADO 5
? REPITE 40 [AVANZA :LADO GIRADERECHA 90 HAZ "LADO :LADO + 1
```



El número total de líneas es 40, la primera de longitud 5 y la última de longitud 45.

LEVANTANDO Y BAJANDO LA PLUMILLA

Se puede conseguir con las órdenes adecuadas que la tortuga dibuje o no en su recorrido por la pantalla. Estas órdenes son:

SUBELAPIZ (SL)
BAJALAPIZ (BL)

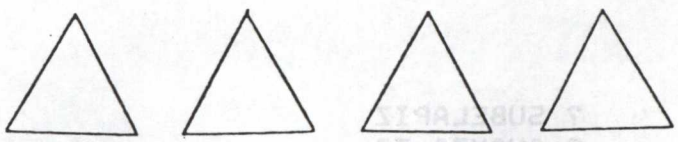
Ejemplo:
Trazaremos cuatro triángulos de lado 40 en tres posiciones diferentes.
Definimos, por comodidad, el procedimiento TRIANGULO

```
PARA TRIANGULO
REPITE 3 [AVANZA 40 GIRADERECHA 120]
FIN
```

El proceso sería entonces:

```
? SUBELAPIZ
? GIRAIZQUIERDA 90
? AVANZA 100
? GIRADERECHA 120
? BAJALAPIZ
? TRIANGULO
? SUBELAPIZ
? GIRADERECHA 60
? AVANZA 50
? GIRAIZQUIERDA 60
? BAJALAPIZ
? TRIANGULO
```

Obteniéndose la figura:



Obsérvese cómo las instrucciones entre cada orden de SUBELAPIZ y la siguiente de BAJALAPIZ los movimientos de la tortuga se realizan sin dejar "huella".

Además, se pueden borrar los trazos realizados por la tortuga poniendo la tortuga en el modo borrado (activando el "borrador") con el comando:

GOMA

También se puede poner la tortuga en el modo invertido, y lo que hace es borrar aquellos trazos ya dibujados y dibujar cuando pasa por trazos no dibujados, con el comando:

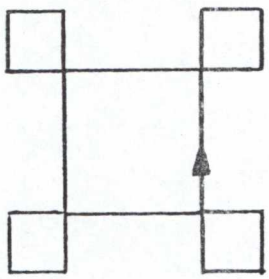
INVIERTELAPIZ (IL)

Ejemplo:
Realizaremos un programa que dibuje cuatro cuadrados pequeños unidos por un cuadrado central.

Supuesto ya definido el procedimiento CUADRADO como sigue:

```
PARA CUADRADO :LADO
REPITE 4 [AVANZA :LADO GIRADERECHA 120]
FIN
```

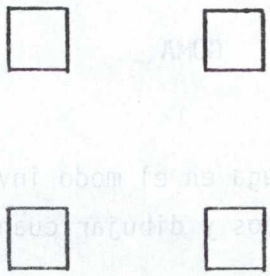
Un programa que nos dará la figura:



es el:

```
? SUBELAPIZ
? AVANZA 30
? BAJALAPIZ
? REPITE 4 [CUADRADO 20 GIRAIZQUIERDA 90 AVANZA 50]
```

y después haremos que borre el cuadrado central quedando la nueva figura:

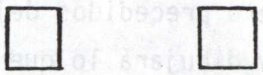


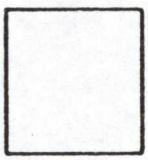
con el programa:

```
? SUBELAPIZ
? AVANZA 30
? BAJALAPIZ
? REPITE 4 [CUADRADO 20 GIRAIZQUIERDA 90 AVANZA 50]
? GOMA REPITE 4 [ [GIRAIZQUIERDA 90 AVANZA 50]
```

Si repetimos indefinidamente el proceso anterior, utilizando además el comando INVIERTELAPIZ obtendremos alternativamente ambas figuras:



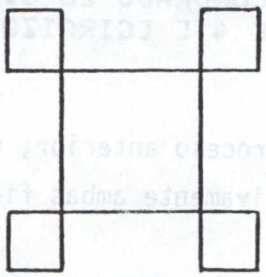




y el programa sería:

```
? SUBELAPIZ
? AVANZA 50
? BAJALAPIZ
? REPITE 4 [CUADRADO 20 GIRAIZQUIERDA 90 AVANZA 50]
? GOMA REPITE 4 [GIRAIZQUIERDA 90 AVANZA 50]
? REPITE 100 [INVIERTELAPIZ REPITE 4 [CUADRADO 20
GIRAIZQUIERDA 90 AVANZA 50]
```

Es decir, el programa primero crea la figura:



Después borra el cuadrado central y posteriormente volvemos a pasar por toda la figura creada inicialmente, pero precedidos del comando **INVIERTELAPIZ** que borrará lo que encuentre dibujado y dibujará lo que encuentre borrado 100 veces.

SACANDO Y ESCONDIENDO LA TORTUGA

Con las ordenes **OCULTATORTUGA** y **MUESTRATORTUGA**, se hace desaparecer la tortuga y hacerla reaparecer respectivamente. Esto se efectúa únicamente a efectos de visibilidad pero no de las acciones que realiza. Así, si ejecutamos un programa con una orden previa de OCULTATORTUGA ., el dibujo se realiza igualmente, como en el ejemplo siguiente:

```
? OCULTATORTUGA
? CUADRADO 20
```



Sus abreviaturas son OT y MT

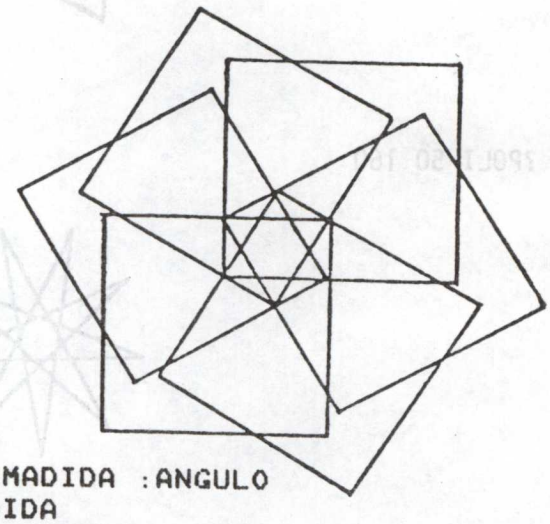
EJEMPLOS DE GRAFICOS DE TORTUGA

Efectuaremos algunos diseños basados en las instrucciones estudiadas hasta el momento.

```
PARA DIBUJO
REPITE 6 [ AVANZA 20 GIRADERECHA 60 CUADRADO 75]
FIN
```

Al ejecutar aparece el dibujo:

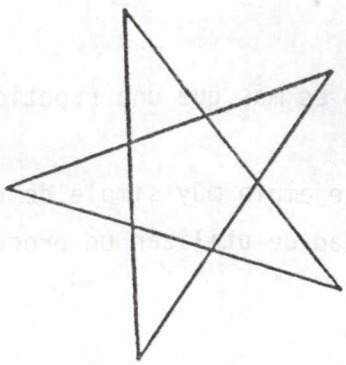
?DIBUJO



```
PARA POLI :MADIDA :ANGULO
AVANZA :MADIDA
GIRADERECHA :ANGULO
POLI :MEDIDA :ANGULO
FIN
```

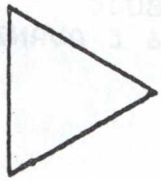
Al ejecutar el procedimiento POLI, se obtiene la figura:

?POLI 80 144

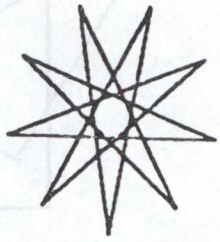


Para otros valores, las ejecuciones son:

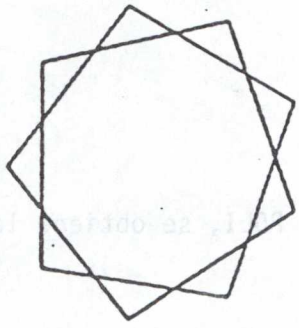
?POLI 50 120



?POLI 50 160



?POLI 60 80



El ejemplo anterior no es más que una repetición de un ciclo indefinido.

Esto corresponde a un ejemplo muy simple de recursión. En general, se llama recursión a la capacidad de utilizar un procedimiento como parte de su propia definición.

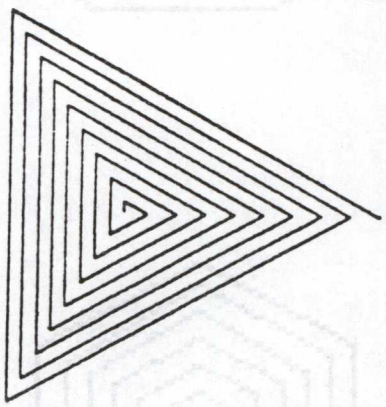
El ejemplo siguiente muestra otro tipo de recursión, esta vez es algo más que una simple repetición: El procedimiento POLIESPI, que se define en función de

sí mismo, pero con valores diferentes:

```
PARA POLIESPI :MEDIDA :ANGULO
AVANZA :LADO
GIRADERECHA :ANGULO
POLIESPI :LADO + 3 :ANGULO
FIN
```

y al ejecutarlo:

```
?POLIESPI 1 120
```



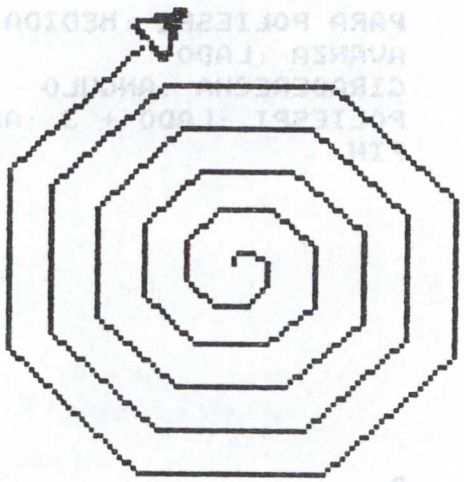
produce una espiral triangular, ya que cada lado es tres unidades mayor que el anterior, pues equivale a como si se le hubiera hecho ejecutar la secuencia siguiente:

```
? AVANZA 1
? GIRADERECHA 120
? AVANZA 4
? GIRADERECHA 120
? AVANZA 7
? GIRADERECHA 120
?      :
       :
```

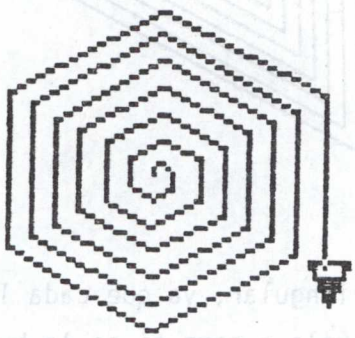
y así indefinidamente.

Otras ejecuciones darían:

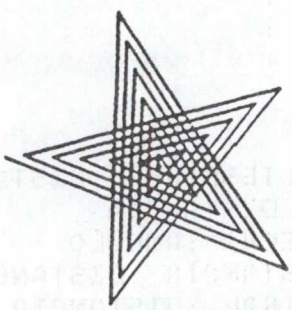
?POLIESPI 1 45



?POLIESPI 1 60



?POLIESPI 1 144

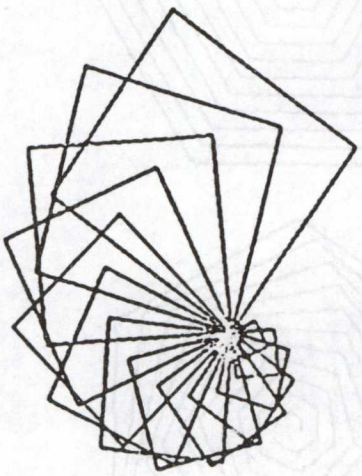


Otro ejemplo sería el procedimiento ESPIRALRECTA.

```
PARA ESPIRALRECTA :LADO :ANGULO
CUADRADO :LADO
GIRADERECHA :ANGULO
ESPIRALRECTA :LADO + 3 :ANGULO
FIN
```

Y al ejecutarse con los valores indicados:

```
?ESPIRALRECTA 1 10
```



Obsérvese que esto equivale a ejecutar el procedimiento CUADRADO con un lado incrementado en tres unidades y girado un ángulo de 10° cada vez, obteniéndose por tanto el conjunto de cuadrados de la figura.

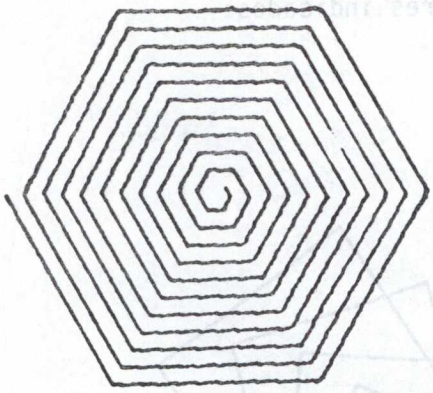
Se puede añadir al procedimiento anterior otra variable (CAMBIO) obteniendo el POLIESPIRAL, como sigue:

```
PARA POLIESPIRAL :DISTANCIA :ANGULO :CAMBIO
AVANZA :DISTANCIA
GIRADERECHA :ANGULO
HAZ "DISTANCIA :DISTANCIA + :CAMBIO
POLIESPIRAL :DISTANCIA :ANGULO :CAMBIO
FIN
```

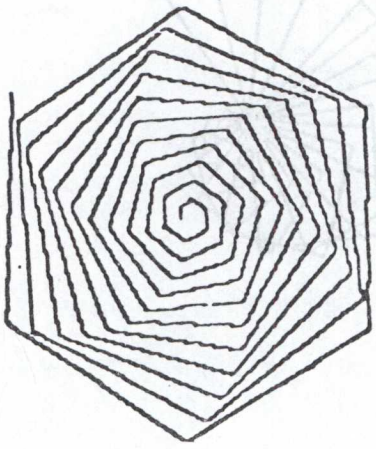
Como puede deducirse de la definición, el procedimiento realiza un dibujo avanzando "DISTANCIA, girando "ANGULO, incrementando "DISTANCIA en "CAMBIO, volviendo hacia adelante, girando, incrementando, etc.

En las siguientes gráficas se muestran diversas ejecuciones del procedimiento para diferentes valores de las variables.

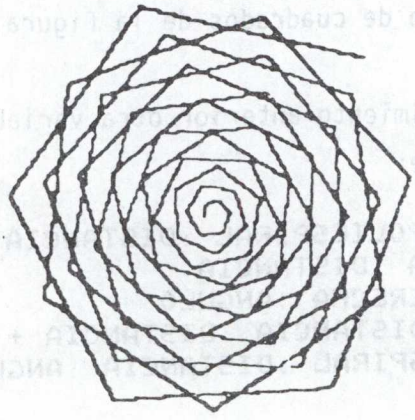
?POLIESPIRAL 10 60 3



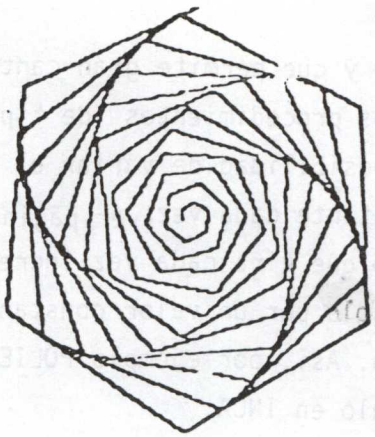
?POLIESPIRAL 10 61 3



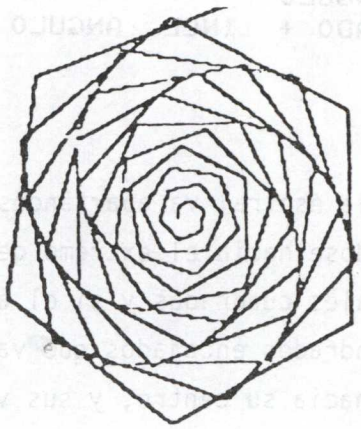
? POLIESPIRAL 10 65 3



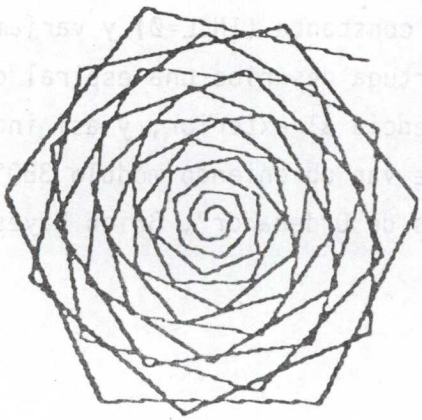
? POLIESPIRAL 10 62 3



? POLIESPIRAL 10 63 3



? POLIESPIRAL 10 64 3



ESPIRALES MULTIPLES

Un procedimiento sencillo y que permite gran cantidad de variaciones y exploración geométrica, son los procedimientos de tipo poligonal (procedimiento POLI) si le añadimos la posibilidad de variar el lado incrementándolo o multiplicándolo por una constante cada vez(ver página número 23) y si además variamos el ángulo que gira cada vez incrementándolo en una cantidad constante, o multiplicándolo por un valor constante, se aumentan las posibilidades del procedimiento. Así, por ejemplo POLIESPIRAL incrementa cada vez el lado en INCL y el ángulo en INCA.

```
PARA POLIESPIRAL :LADO :ANGULO :INCL :INCA
AVANZA :LADO
GIRADERECHA :ANGULO
POLIESPIRAL :LADO + :INCL :ANGULO + :INCA :INCL :INCA
FIN
```

Al ser INCL una constante, la espiral va apartándose de su centro aumentando cada vez el lado, expandiéndose hacia el extremo de la pantalla; si el ángulo es de 90° se obtienen espirales cuadradas y si el ángulo es próximo a los 90° se obtienen una serie de cuadrados encajados que van retorciéndose y haciendo un movimiento de procesión hacia su centro, y sus vértices realizan espirales secundarias.

Si los ángulos son pequeños, se consiguen espirales casi circulares.

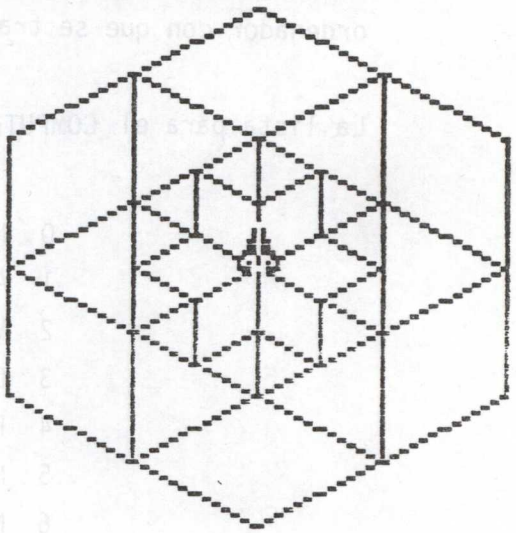
Si el lado lo mantenemos constante (INCL=Ø) y variamos cada vez el ángulo (INCA≠Ø), entonces la tortuga describe una espiral que tiende hacia dentro, seguida de otra con tendencia al exterior, y así indefinidamente. Esto es debido a que los ángulos se van obteniendo módulo 360° (para una explicación más detallada ver "Juegos de Ordenador", Brian Hayes en "Investigación y Ciencia, abril 1984).

FIGURAS QUE UTILIZAN OTRAS FIGURAS

Algunas de las gráficas obtenidas en anteriores apartados hacían uso de otras figuras ya definidas como procedimientos. Este era el caso de ESPIRALRECTA que utilizaba CUADRADO, o el de SUPERHEXAGONO que utilizaba el procedimiento HEXAGONO. Este proceso se puede ampliar y definir ahora el procedimiento HEXAGONO.ESTRELLA basándonos en los anteriores.

```
PARA HEXAGONO.ESTRELLA :LADO
SUPERHEXAGONO :LADO
SUPERHEXAGONO :LADO / 2
FIN
```

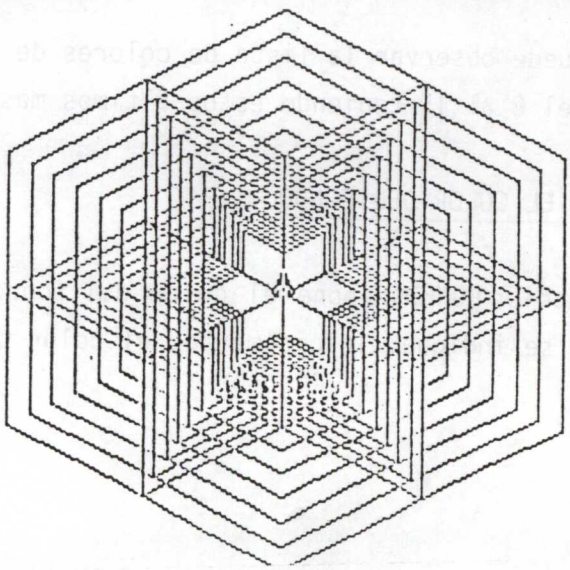
que da lugar a la figura



Y aún más, ahora se puede definir un procedimiento que utilice el anteriormente definido.

```
PARA ESTRELLAS :LADO
REPITE 6 [HEXAGONO.ESTRELLA :LADO HAZ "LADO :LADO - 5]
FIN
```

que como su definición indica repite 6 veces la figura HEXAGONO.ESTRELLA disminuyendo cada vez el lado en 5 unidades, dando lugar al gráfico de la figura.



COLORES DE FONDO

LOGO permite variar además los colores, tanto del fondo como de los dibujos de la tortuga.

La lista de colores del fondo se relaciona con un número, dependiendo del ordenador con que se trabaje.

La lista para el **COMPUTEC** es la siguiente:

0 Negro
1 Azul
2 Verde
3 Cyan
4 Rojo
5 Magenta
6 Marrón
7 Blanco
8 Gris
9 Azul brillante
10 Verde brillante
11 cyan brillante
12 Rojo brillante
13 Magenta brillante
14 Amarillo
15 Blanco intenso

Como se puede observar la lista de colores de 0 a 7 se corresponde con los colores del 8 al 15 teniendo estos últimos más intensidad.

CAMBIANDO EL COLOR DEL FONDO

La instrucción **PONFONDO** pone el color del fondo de acuerdo con el número de color que se indique. Inicialmente, el color de origen es el negro (número 0)

Así, para cambiar a verde claro se diría:

```
? PONFONDO 2
```

La función FONDO nos da como salida el número de color actual del fondo.
Así, si ejecutamos ahora

```
? ESCRIBE FONDO
  2
```

La abreviatura de BACKGROUND es BG

Ejemplo:

Definiremos un procedimiento que cambia el color del fondo cíclicamente cada dos segundos e imprime cada vez el número del color en cada momento.

```
PARA CAMBIARCOLOR
SI FONDO = 15 [PONFONDO 0 ESPERA 30]
ESCRIBE FONDO
PONFONDO 1 + FONDO
ESPERA 30
CAMBIACOLOR
FIN
```

Se utiliza la orden ESPERA que lo que hace es detener el proceso durante poco menos de 2 segundos.

Al ejecutar el procedimiento CAMBIARCOLOR se cumplirá lo ya dicho.

Obsérvese cómo cuando el color del fondo es 15 se pasa a 0 (negro) y en caso contrario se incrementa en 1 el número de color.

Este procedimiento es indefinido, y para detenerlo hay que pulsar la tecla F-2.

COLOR DE LA TORTUGA Y DE LOS DIBUJOS

El color de la tortuga y por tanto de los dibujos que va realizando con su rastro, depende también del ordenador que se esté utilizando.

El COMPUTEC tiene seis colores dispuestos en dos paletas llamadas PALETA 0 y PALETA 1 numerados de la manera siguiente:.

PALETA 0	PALETA 1
1 verde	1 cyan
2 rojo	2 magenta
3 marrón	3 blanco

Inicialmente la tortuga tiene el color 3 de la paleta 1, es decir, es blanca y deja un rastro blanco.

Combinando las instrucciones SETPAL y SETPC seguidas de un número, obtenemos los seis colores posibles. Así por ejemplo:

```
? PONPALETA 0
? PONCL 1
```

hacen que la tortuga quede de color verde y si después ordenamos

```
? CUADRADO 20
```

dibujará un cuadrado del mismo color.

A partir del procedimiento TRIANGULO

```
PARA TRIANGULO :LADO
REPITE 3 [AV :LADO GD 120]
FIN
```

podríamos definir un procedimiento que dibujara triángulos de distinto color.

```
PARA TRIANGULOS.COLOR
PONPALETA 0 PONCL 1 TRIANGULO 30 GD 120
PONCL 2 TRIANGULO 30 GD 120
PONCL 3 TRIANGULO 30
FIN
```

Como se ve, utilizamos solamente los colores de la paleta 0. Este procedimiento dibuja primero un triángulo verde, después uno rojo y por último uno marrón.

Si con este dibujo ordenamos

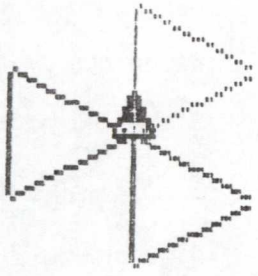
```
? PONPALETA 1
```

podemos comprobar que los triángulos cambian de color, es decir, el verde pasa a cyan, el triángulo rojo queda color magenta y el marrón, blanco.

Con las funciones COLORLAPIZ (CL) y PALETA obtenemos el color y la paleta que se están utilizando.

Así ejecutando:

```
? PONPALETA 1
? PANCL 2
? ESCRIBE CL
2
ESCRIBE PALETA
1
```



CONSIDERACIONES PRÁCTICAS

En este apartado vamos a desarrollar una serie de detalles prácticos, imprescindibles para poder trabajar con LOGO desde el primer momento.

Control de la pantalla

Para imprimir un mensaje en la pantalla se utiliza el comando ESCRIBE Se debe distinguir entre variables palabras y listas, como ilustra el siguiente ejemplo:

```
? HAZ "LADO 5
? ESCRIBE :LADO
5
? ESCRIBE "LADO
LADO
```

Para situar el cursor en un lugar determinado de la pantalla se utiliza la orden

PONCURSOR posición

donde "posición" es una lista que consiste en el número de la fila y el de la columna de la pantalla.

Para borrar la pantalla se utiliza el comando

BORRAPANTALLA (abreviado BP)

Para eliminar el texto de la pantalla se utiliza

BORRATEXTO (abreviado BT)

con lo que el cursor queda situado en el borde superior izquierdo de la pantalla.

Para borrar los gráficos de la pantalla sin afectar el estado de la tortuga, es decir, sin cambiar su posición ni su orientación, se utiliza

LIMPIA

La pantalla tiene tres estados diferentes:

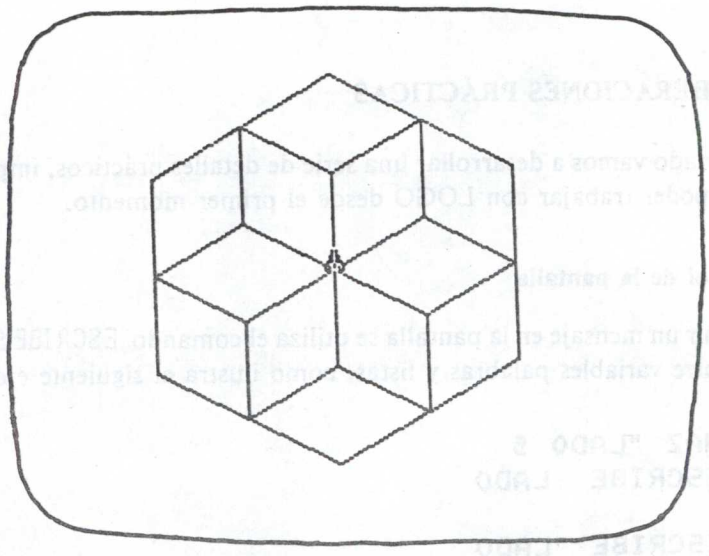
—Modo texto, en el que sólo aparece el texto escrito, no los gráficos. Este modo se alcanza con el comando

PANTALLATEXTO (abreviado PT)

que hace desaparecer momentáneamente los gráficos.

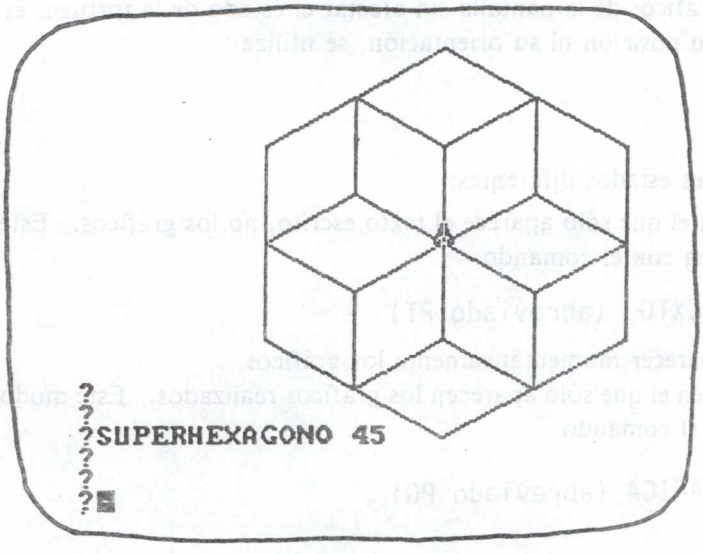
—Modo gráfico, en el que sólo aparecen los gráficos realizados. Este modo se alcanza con el comando

PANTALLAGRAFICA (abreviado PG)



que hace desaparecer momentáneamente el texto.
Modo mixto, en el que la pantalla queda dividida en dos ventanas: una para gráficos, la superior, y la otra para texto. Este modo se alcanza con el comando

```
PANTALLAMIXTA .(abreviado PG)
```



En algunos sistemas el comando es PANTALLAMIXTA(abreviado PM) o "pantalla parcial". Al utilizar este comando desaparecen momentáneamente las líneas de texto salvo las últimas.

Manejo del editor

Ya hemos descrito el método que permite crear procedimientos nuevos. Veremos a continuación algunas cuestiones prácticas relacionadas con la creación y edición de prodecimientos.

El intérprete de LOGO, al recibir una orden de la forma PARA nombre-del-procedimiento', entra automáticamente en modo de "definición" (a no ser que el procedimiento cuyo nombre se especifica esté ya definido), lo que se indica por el hecho de que en la pantalla aparece el signo '>'. Este modo se abandona al acabar la definición. La secuencia de definición se ilustra con el ejemplo del procedimiento SALUDA:

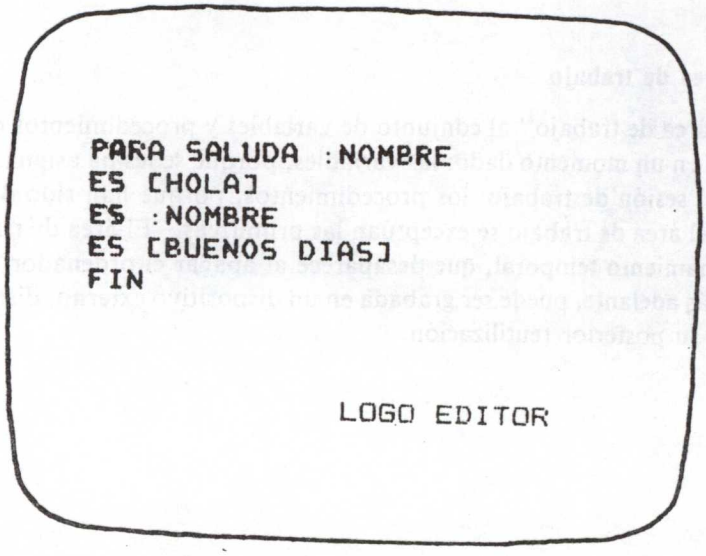
```
? PARA SALUDA :NOMBRE
> ES [HOLA]
> ES :NOMBRE
> ES [BUENOS DIAS]
> FIN
```

Si intentamos ahora definir otra vez SALUDA, con intención, por ejemplo, de modificarlo, provocaremos un mensaje de error y el intérprete no entrará en el modo de definición:

```
?PARA SALUDA :NOMBRE
YA EXISTE EL PROCEDIMENTO SALUDA
?
```

Para modificar el procedimiento (o también para crearlo desde el primer momento), se puede llamar al "*editor*":

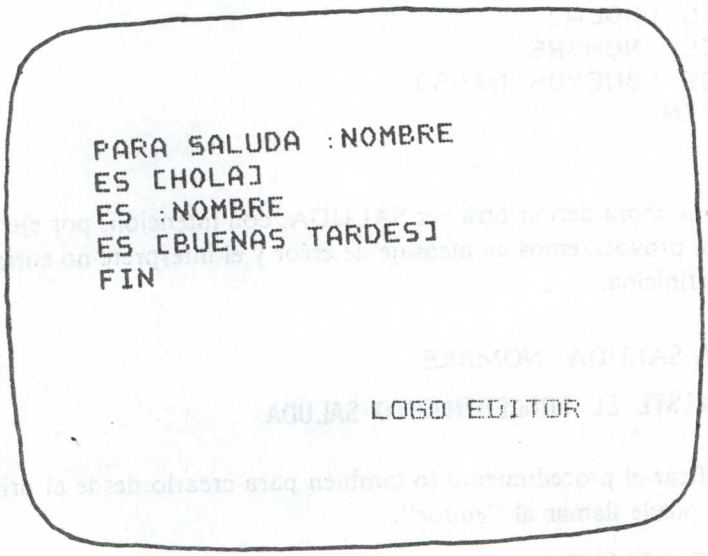
```
? ED "SALUDA
```



Cuando aparece en la pantalla lo que se muestra en la figura, se ha entrado en el modo de edición; entonces podemos insertar, borrar, etc. El editor de LOGO es lo que se denomina un "editor de pantalla", ya que permite desplazar el cursor a lo largo y al lo ancho de la pantalla, pulsando las teclas correspondientes (↑ ↓ ← →), cuyo significado es obvio. Todo lo que se teclee aparecerá en la posición del cursor, inserto en el punto en que se encuentre éste. Para borrar caracteres se utiliza una tecla específica (DEL).

No vamos a entrar en la descripción detallada del funcionamiento del editor, ya que puede variar de un ordenador a otro, por lo que es conveniente, para familiarizarse con el que se vaya a utilizar, consultar el manual del lenguaje.

Siguiendo las instrucciones anteriores, podemos modificar SALUDA hasta dejarlo de la siguiente forma:



El área de trabajo

Se llama "área de trabajo" al conjunto de variables y procedimientos que LOGO conoce en un momento dado: las variables, porque se les ha asignado valor en la actual sesión de trabajo; los procedimientos, porque han sido definidos en ella. Del área de trabajo se exceptúan las primitivas. El área de trabajo es un almacenamiento temporal, que desaparece al apagar el ordenador. Como veremos más adelante, puede ser grabada en un dispositivo externo, disco o cassette, para su posterior reutilización.

Primitivas con las que se opera sobre el área de trabajo

Hay primitivas que permiten listar las variables o los procedimientos, borrar procedimientos o variables globales, etc. Para poder ofrecer ejemplos del uso de estas primitivas vamos a suponer que en la sesión actual de trabajo, es decir, desde el momento en que encendimos el ordenador, hemos definido los procedimientos y ejecutado las instrucciones que se indican:

```
? PARA CUADRADO :LADO
> REPITE 4 [AV :LADO GD 60]
> FIN
QUEDA DEFINIDO CUADRADO
> PARA SALUDA :NOMBRE
> ES [HOLA]
> ES :NOMBRE
> ES [BUENOS DIAS]
FIN
QUERDA DEFINIDO SALUDA
? PARA PASEAR
> AV AZAR 5
> GD AZAR 91
> HAZ "PASOS :PASOS + 1
> PASEAR
> FIN
QUEDA DEFINIDO PASEAR
? CUADRADO 40
? SALUDA "ANTONIO
HOLA
ANTONIO
BUENOS DIAS
? HAZ "PASOS 0
? PASEAR
? ES :PASOS
60
? HAZ "LISTA [A B C]
? HAZ "NÚMERO 20
```

El ordenador emite un mensaje que indica que da por terminada la definición de CUADRADO

Dibuja en la pantalla un cuadrado de lado 40

La tortuga empieza a moverse aleatoriamente por la pantalla, hasta que interrumpimos el proceso desde el teclado

La actividad descrita hace que en el área de trabajo haya varios procedimientos y variables. Si ahora deseamos conocer el estado del área de trabajo podemos utilizar las siguientes primitivas:

- IM (imprime) para listar la definición de un procedimiento o de una lista de procedimientos:

  ? IM "CUADRADO

```
PARA CUADRADO :LADO
REPITE 4 [AVANZA :LADO GD 90]
FIN
```

- IMPS (imprime procedimientos) para imprimir el listado de todas las definiciones de procedimientos.
- IMTS (imprime títulos) para imprimir los títulos de todas las definiciones:

```
? IMTS
CUADRADO :LADO
SALUDA :NOMBRE
PASEAR
```

- IMVS (imprime variables) para imprimir los nombres y valores de las variables:

```
? IMVS
LISTA CONTIENE [A B C]
NUMERO CONTIENE 20
PASOS CONTIENE 60
```

- IMTODO (imprime todo) para imprimir un listado de ambas cosas, procedimientos y variables.

- NODOS para imprimir el número de nodos libres, lo que nos da una idea del espacio de memoria libre (1 nodo equivale a 5 bytes).

Se pueden borrar variables y procedimientos mediante las primitivas siguientes:

— BORRA (BO) seguida del nombre del procedimiento. Por ejemplo,

```
BORRA "PASEAR
```

— BOVAR (borra variable) borra la variable o lista de variables que se escriben como entrada. Por ejemplo,

```
BOVAR [LISTA NUMERO]   Borra las variables "LISTA y "NUMERO
BOVAR "PASOS           Borra la variable PASOS
```

— BOVAR (borra variables); borra todas las variables del área de trabajo.

— BOPROCS (borra procedimientos); borra todos los procedimientos del área de trabajo.

— BETODO (borra todo); borra el área de trabajo.

Almacenamiento y recuperación del área de trabajo

Para almacenar el área de trabajo completa en disco se debe escribir

```
GUARDA "nombre del fichero
```

En algunos sistemas hay que especificar la unidad, por ejemplo

```
GUARDA "D:nombre del fichero   para grabar en disco
GUARDA "C:nombre del fichero   para grabar en cassette
```

Para recuperar el fichero que está grabado en el disco B con el nombre DIBUJOS se escribe

```
CARGA "B:DIBUJOS
```

Al almacenar en un fichero se graban las variables junto con sus valores, salvo que hayamos borrado las variables con BOVARS

PRACTICAS (PRIMITIVAS Y PROCEDIMIENTOS)

1) Comprobar el funcionamiento de los procedimientos HEXAGONO y SUPERHEXAGONO ya vistos y de manera análoga definir DODECAGONO y SUPERDODECAGONO.

2) Definir un procedimiento que dibuje rectángulos de distinntas dimensiones.

3) Definir procedimientos para dibujar cuadrados y hexágonos centrados en la pantalla.

4) Dibujar un árbol de navidad y a partir de él hacer un procedimiento que dibuje un bosque de abetos. (Se pueden introducir primitivas que lo coloreen adecuadamente)

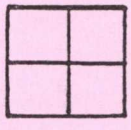
5) Definir un procedimiento equivalente al procedimiento POLI, ya visto, que no sea un proceso indefinido. Utilizar la primitiva REPITE y ejecutarlo con distintas entradas.

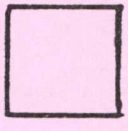
6) Idem con POLIESPIRAL



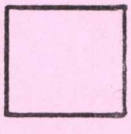
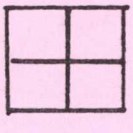
7) Crear un procedimiento que dibuje un polígono regular cualquiera, a partir de la longitud del lado y del número de lados.

8) Definir un procedimiento que haga:

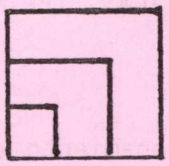
a) el dibujo 

b) utilizando el "borrador" quede 

c) utilizando el comando de "inversión" quede alternativamente

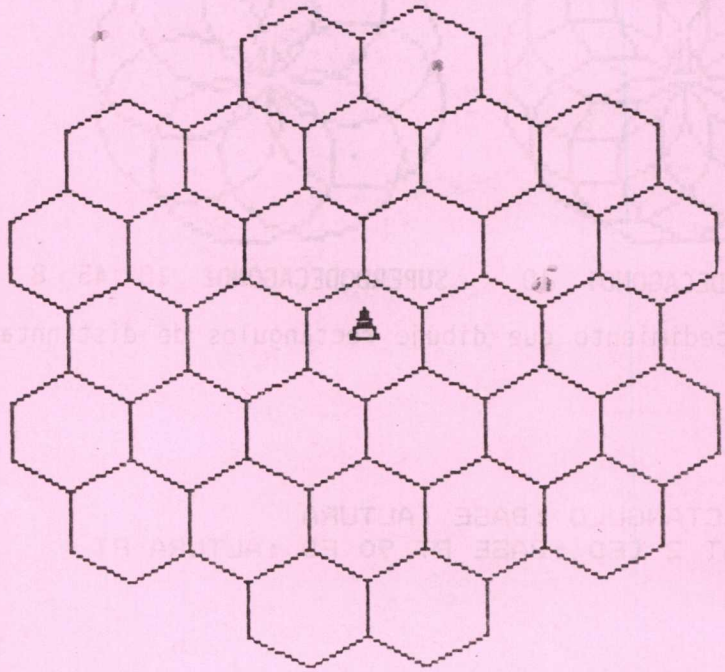
 

9) Dibujar tres cuadrados como en la figura



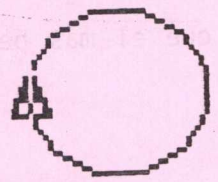
de forma que el más pequeño sea marrón, el siguiente verde y el mayor rojo.

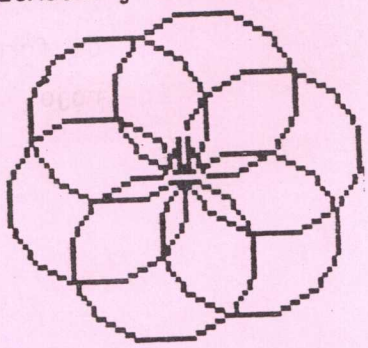
10) Definir un procedimiento que dibuje una colmena como en la figura:



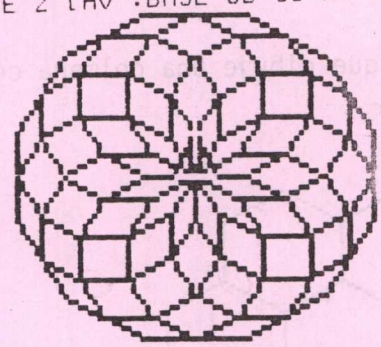
PRACTICAS (PRIMITIVAS Y PROCEDIMIENTOS)

1) Comprobar el funcionamiento de los procedimientos HEXAGONO y SUPERHEXAGONO ya vistos y de manera análoga definir DODECAGONO y SUPERDODECAGONO.

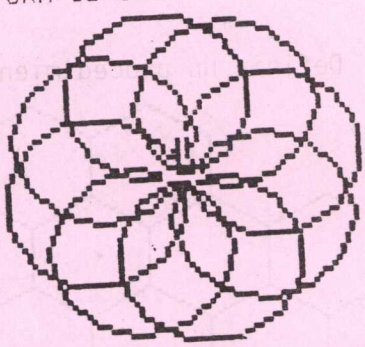




```
PARA DODECAGONO :LADO
REPITE 12 [AV :LADO GD 30]
FIN
PARA SUPERDODECAGONO :LADO
REPITE 6 [DODECAGONO :LADO GD 60]
FIN
PARA SUPERDODECAGONO1 :LADO
REPITE 12 [DODECAGONO :LADO GD 150]
FIN
PARA SUPERDODECAGONO2 :LADO :ANGULO :N
REPITE :N [DODECAGONO :LADO GD :ANGULO]
FIN
PARA RECTANGULO :BASE :ALTURA
REPITE 2 [AV :BASE GD 90 AV :ALTURA GD 90]
FIN
```



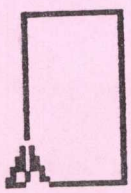
SUPERDODECAGONO1 10



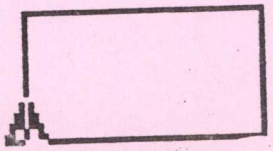
SUPERDODECAGONO2 10 45 8

2) Definir un procedimiento que dibuje rectángulos de distinntas dimensiones.

```
TO RECTANGULO :BASE :ALTURA
REPEAT 2 [FD :BASE RT 90 FD :ALTURA RT
90]
END
```



RECTANGULO 40 20

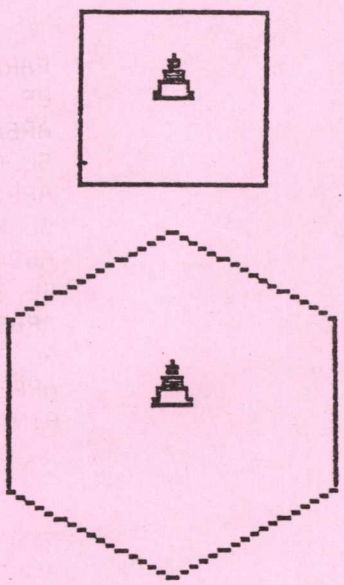


RECTANGULO 30 50

3) Definir procedimientos para dibujar cuadrados y hexágonos centrados en la pantalla.

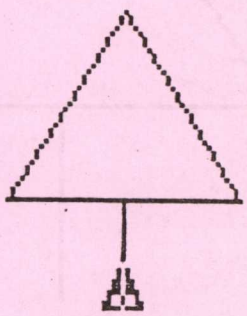
```
PARA CUADCENT :LADO
SL RE :LADO/2 GI 90
AV :LADO/2 GD 90 BL
REPITE 4 [AV :LADO GD 90]
SL CENTRO BL
FIN

PARA HEXAGCENT :LADO
SL RE :LADO
GI 60 BL
REPITE 6[AV :LADO GD 60]
SL CENTRO BL
FIN
```

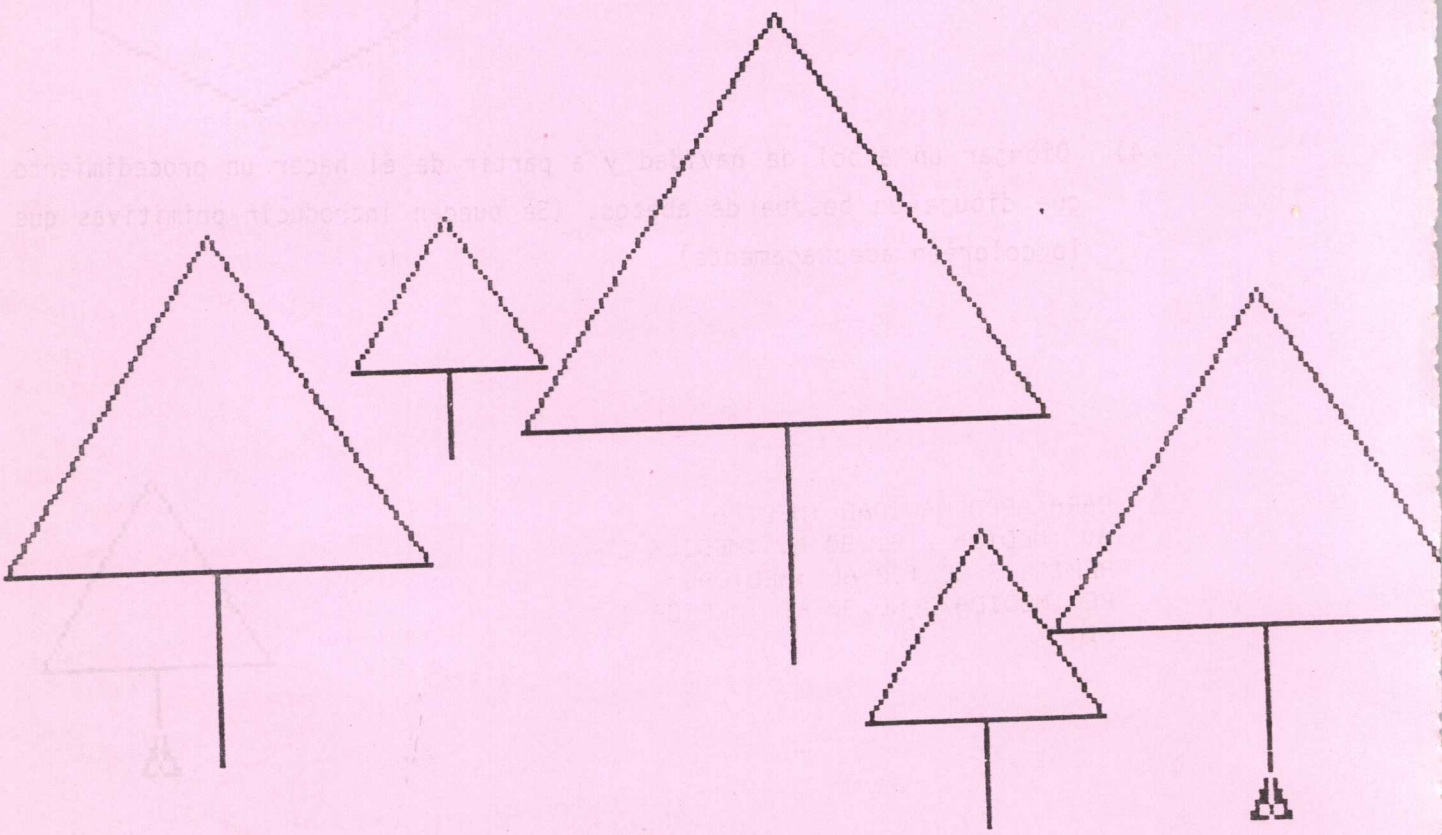


4) Dibujar un árbol de navidad y a partir de él hacer un procedimiento que dibuje un bosque de abetos. (Se pueden introducir primitivas que lo coloreen adecuadamente)

```
PARA ARBOLNAVIDAD :MEDIDA
AV :MEDIDA/2 GI 90 AV :MEDIDA/2
REPITE 3[GD 120 AV :MEDIDA]
RE :MEDIDA/2 GD 90 RE :MEDIDA/2
FIN
```

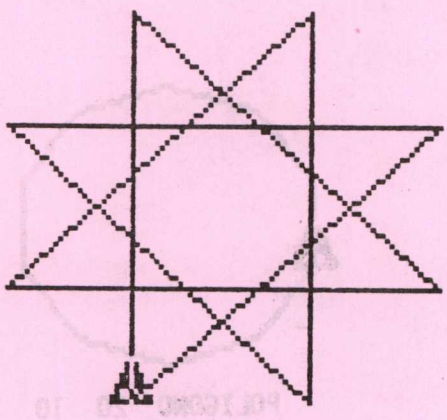


```
PARA BOSQUE
BP SL RE 90 GI 90 AV 110 GD 90 BL
ARBOLNAVIDAD 90
SL GD 90 AV 50 GI 90 AV 79 BL
ARBOLNAVIDAD 40
SL RE 50 GD 90 AV 70 GI 90 BL
ARBOLNAVIDAD 110
SL GD 90 AV 40 GI 90 AV 40 BL
ARBOLNAVIDAD 50
SL GD 90 AV 60 GI 90 BL
ARBOLNAVIDAD 60
FIN
```

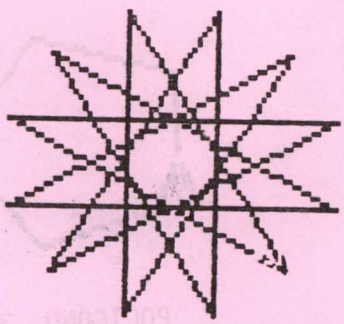


5) Definir un procedimiento equivalente al procedimiento POLI, ya visto, que no sea un proceso indefinido. Utilizar la primitiva REPEAT y ejecutarlo con distintas entradas.

```
PARA POLI2 :MEDIDA :ANGULO :N
REPITE :N [AV :MEDIDA GD :ANGULO]
FIN
```



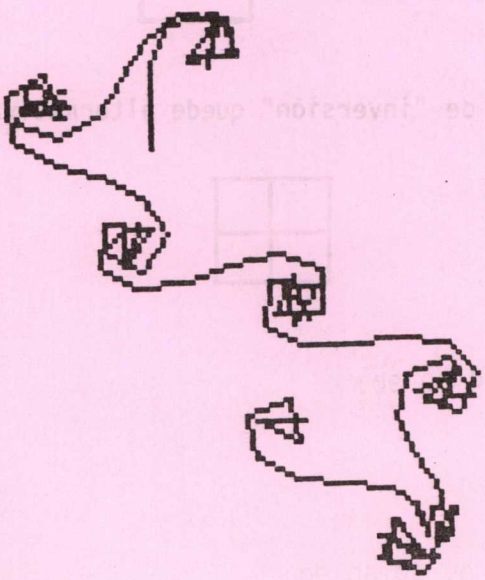
POLI2 90 135 8



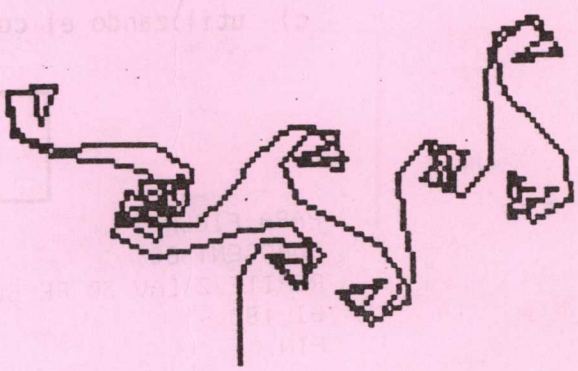
POLI2 70 150 12

6) Idem con POLIESPIRAL

```
PARA POLIESPIRAL2 :LADO :ANGULO :INCL :INCA
REPITE 100 [AV :LADO GD :ANGULO HAZ "LADO
:LADO + :INCL HAZ "ANGULO :ANGULO +  :INCA]
FIN
```



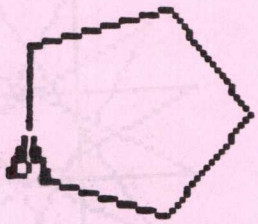
POLIESPIRAL2 10 0 0 26



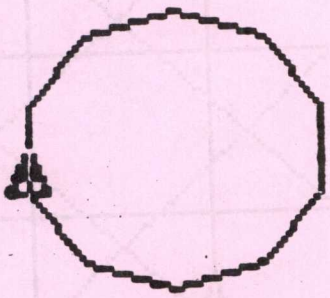
POLIESPIRAL2 10 0 0 31

7) Crear un procedimiento que dibuje un polígono regular cualquiera, a partir de la longit6ud del lado y del número de lados.

```
PARA POLIGONO  :LADO :N
REPITE :N [AV :LADO GD 360/:N]
FIN
```

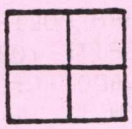


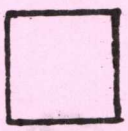
POLIGONO 30 5



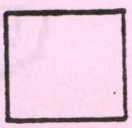
POLIGONO 20 10

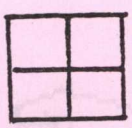
8) Definir un procedimiento que haga:

a) el dibujo 

b) utilizando el "borrador" quede 

c) utilizando el comando de "inversión" quede alternativamente



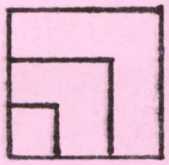


```
PARA FIGURA1
CUADCENT 60
REPITE 2 [AV 30 RE 60 AV 30 GD 90]
GI 180
FIN

PARA FIGURA2
FIGURA1
GOMA REPITE 2 [AV 30 RE 60 AV 30 GD 90]
GI 180
FIN

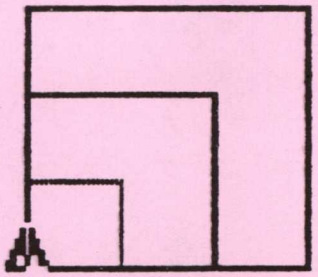
PARA FIGURA3
FIGURA2
IL FIGURA1
FIN
```

9) Dibujar tres cuadrados como en la figura

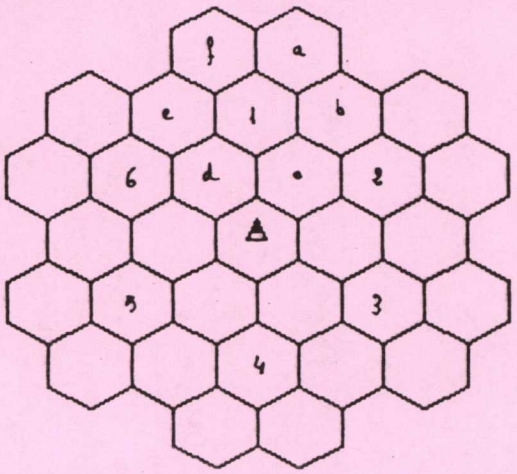


de forma que el más pequeño sea marrón, el siguiente verde y el mayor rojo.

```
PARA CUADCOLOR
PONPALETA 0 PONCL 3 CUADRADO 20
PONCL 1 CUADRADO 40
PONCL 2 CUADRADO 60
FIN
```



10) Definir un procedimiento que dibuje una colmena como en la figura:



```
PARA COLMENA :L
REPITE 6 [COLOCAR2 :L CELDA :L VOLVER2
:L GD 60]
FIN

PARA COLOCAR1 :L
SL AV :L GD 120 BL
FIN

PARA COLOCAR2 :L
SL AV :L * 3 BL
FIN

PARA VOLVER1 :L
SL GI 120 RE :L BL
FIN

PARA VOLVER2 :L
SL RE :L * 3 BL
FIN

PARA CELDA :L
REPITE 6 [COLOCAR1 :L EXAGONOI :L VOLVER1 :L GD 60]
FIN

PARA EXAGONOI :L
REPITE 6 [AV :L GI 60]
FIN
```

GEOMETRIA DE LA TORTUGA

COLOCANDO LA TORTUGA EN UNA POSICION

Se utilizan unas órdenes para posicionar la tortuga en cualquier punto de la pantalla. Pero hay que tener en cuenta que si no se da la orden previa de SUBELAPIZ dibujará su recorrido hasta la posición ordenada. La pantalla se supone como unas coordenadas cartesianas

Las instrucciones son:

PONPOS [ ]
PONX
PONY
CENTRO

Así, CENTRO lleva la tortuga al origen de coordenadas, dejándola en posición vertical, y si el lápiz está bajado realiza un trazo desde la posición en la que está al centro.

- PONPOS coloca la tortuga en las coordenadas (X,Y) dadas.

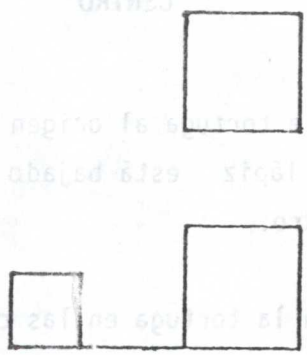
PONX coloca la tortuga en el punto de abscisa dada (X) (y traza una línea horizontal si el lápiz está bajado), y la coordenada no cambia.

SETY coloca la tortuga en el punto de ordenada dada (Y) (y traza una línea vertical si el lápiz está bajado), y la coordenada X no cambia.

Ejemplo:
Realizaremos un dibujo de 3 cuadrados cuyos primeros vértices estén en los puntos (15,40), (15,-30), (-45,-30) respectivamente.

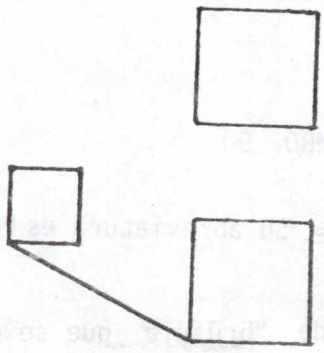
```
SL
PONPOS [15 40]
BL
CUADRADO 40
SL
PONPOS [15 -30]
BL
CUADRADO 40
PONX -45
CUADRADO 25
```

Y realiza el dibujo de la figura.



Nótese que antes de dar coordenadas al primer vértice del segundo cuadrado hemos programado SUBELAPIZ por lo que la tortuga no dibuja y los cuadrados aparecen separados. Al contrario sucede del segundo al tercero: hemos programado PONX sin SUBELAPIZ y la tortuga dibuja la trayectoria haciendo avanzar la X y manteniendo la Y constante.

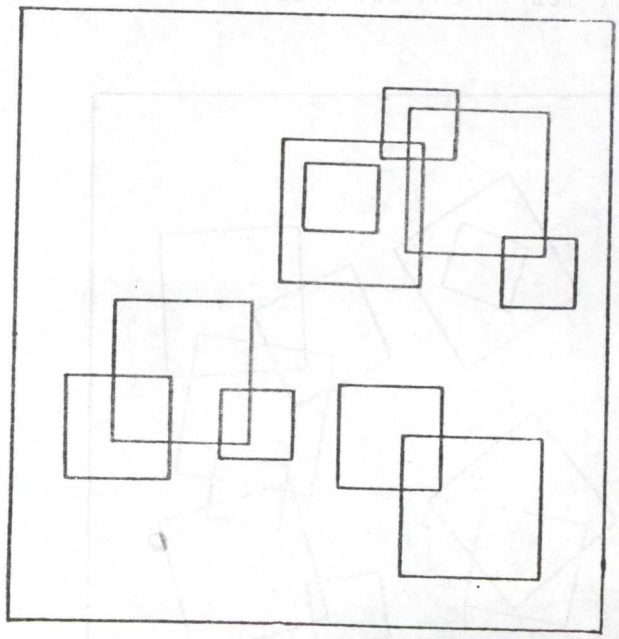
Un programa idéntico, excepto en la instrucción PONX -45 que cambiaremos por PONPOS [-45 0] realizaría el siguiente dibujo:



Aquí las dos coordenadas varían hasta alcanzar el punto programado.

Ejemplo:
Realizaremos un dibujo de 10 cuadrados de lado elegido al azar entre los valores 10 y 50, donde las coordenadas del primer vértice se elegirán igualmente al azar entre los valores $-100 \leq X \leq 100$, y $-100 \leq Y \leq 100$.
El programa sería:

```
REPITE 10 [SL PONPOS [-100 + AZAR 201 -100 + AZAR 201] BL CUADRADO 10 + AZAR 41]
```



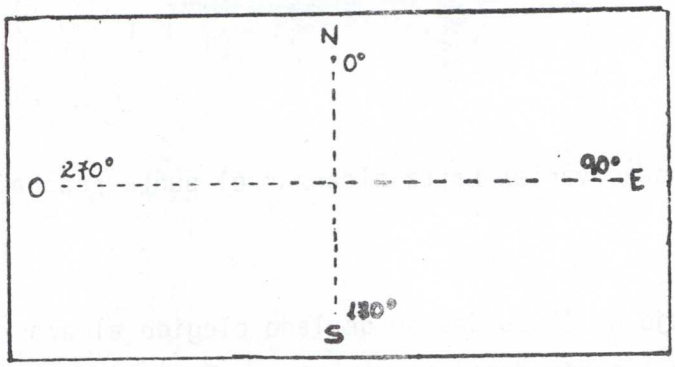
ORIENTANDO LA TORTUGA

Con la instrucción PONRUMBO , se puede orientar la tortuga en la dirección deseada. Así, por ejemplo:

PONRUMBO 90

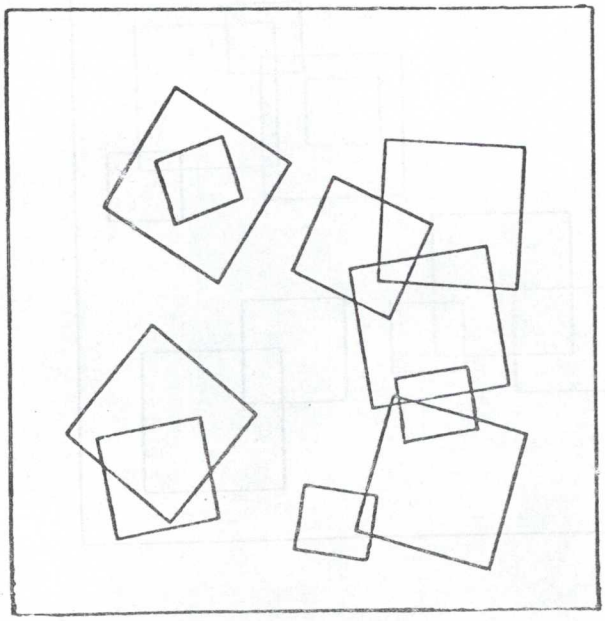
orienta la tortuga en dirección Este. Su abreviatura es PONR

En la figura se muestra el sistema de "brújula" que se utiliza:



Ejemplo:
Si en el ejemplo anterior hacemos también variable la orientación de la tortuga al realizar los cuadrados, resulta:

```
REPITE 10 [SL PONPOS [-100 + AZAR 201 -100 + AZAR 201] PONRUMBO AZAR 360 BL CUAD
RADO 10 + AZAR 40]
```



POSICION Y ORIENTACION DE LA TORTUGA

El "estado" de la tortuga, al ser un punto orientado, viene dado por sus coordenadas y su oriertación.

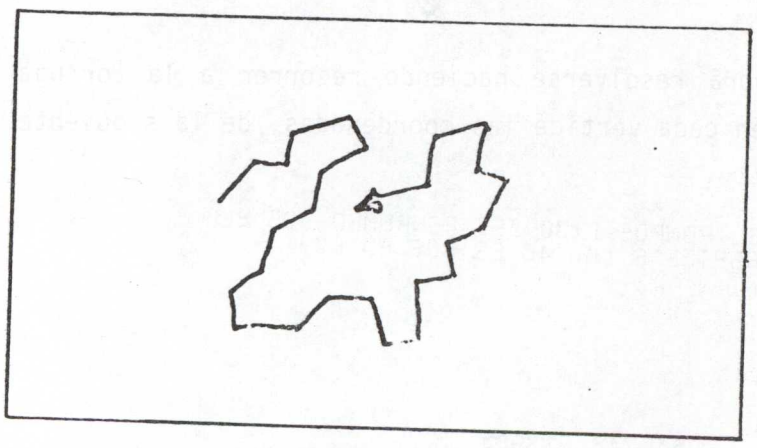
Las funciones siguientes dan como salida información referente al estado de la tortuga:

POS .....Da como salida la lista de coordenadas
COORX .....Da como salida la coordenada X
COORY .....Da como salida la coordenada Y
RUMBO .....Da como salida la orientación

Si queremos visualizarlos valores, debemos ordenar ESCRIBE delante de POS, COORX o COORY.

Ejemplo:
Realizaremos un programa que realiza un movimiento al azar de la tortuga, con pasos de 5 en 5 y rumbo al azar y que imprima las coordenadas y el rumbo en cada momento.

```
REPITE 100 [AV 10 PONRUMBO AZAR 360 ES POS ES RUMBO]
```



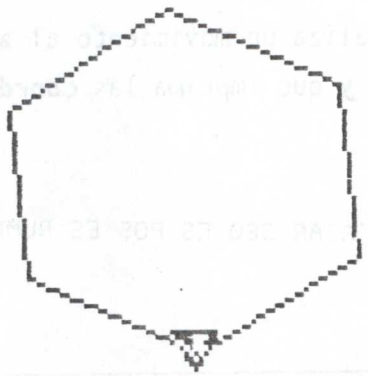
Aquí imprime coordenadas y
rumbo en cada paso

GEOMETRIA DE LA TORTUGA Y GEOMETRIA CARTESIANA

Con las órdenes que nos permiten posicionar la tortuga en los ejes cartesianos (PONPOS, PONX, PONY) junto con las funciones que nos permiten conocer las coordenadas de la tortuga en cada momento (POS,COORX y COORY), se establece una conexión muy interesante entre ambas "geometrías". Se pueden utilizar conjuntamente para resolver problemas como el del siguiente ejemplo:

Hallar las coordenadas de los vértices del hexágono de la figura:

```
?SL PONPOS [130 35] PONRUMBO 295 BL
?REPITE 6 [AV 40 GD 60]
```



El problema podrá resolverse haciendo recorrer a la tortuga el hexágono e imprimiendo en cada vértice las coordenadas, de la siguiente forma:

```
?SL PONPOS [130 35] PONRUMBO 295 BL
?REPITE 6 [AV 40 ES POS GD 60]
```

y dará la lista:

```
63.748 51.904
60.262 91.752
93.027 114.694
129.239 97.79
132.766 57.943
100 35
```

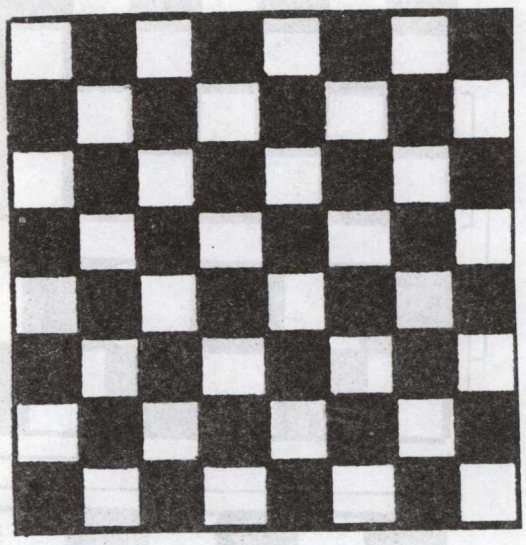
Para obtener estas coordenadas sin la ayuda de la tortuga hubiera sido necesario realizar cálculos trigonométricos.

MODULARIZACION

Ya habíamos hablado en la "Introducción al Logo" de que el trabajo con este lenguaje cultiva ciertas estretegias heurísticas entre ellas, la estrategia de descomponer un problema en partes -módulos- para posteriormente engarzarlos y diseñar el programa principal. A este tipo de estrategias se les suele llamar, en lenguaje informático, análisis "top-down".

Dada la estructuración de LOGO en **procedimientos**, este análisis y realización es no solo acertada sino deseable. Pasamos a realizar un ejemplo siempre teniendo en cuenta que lo resolveremos de una determinada forma pero ésta, como siempre, no es la única.

Ejemplo: Construir un programa de forma que la tortuga dibuje un tablero de ajedrez en la pantalla.



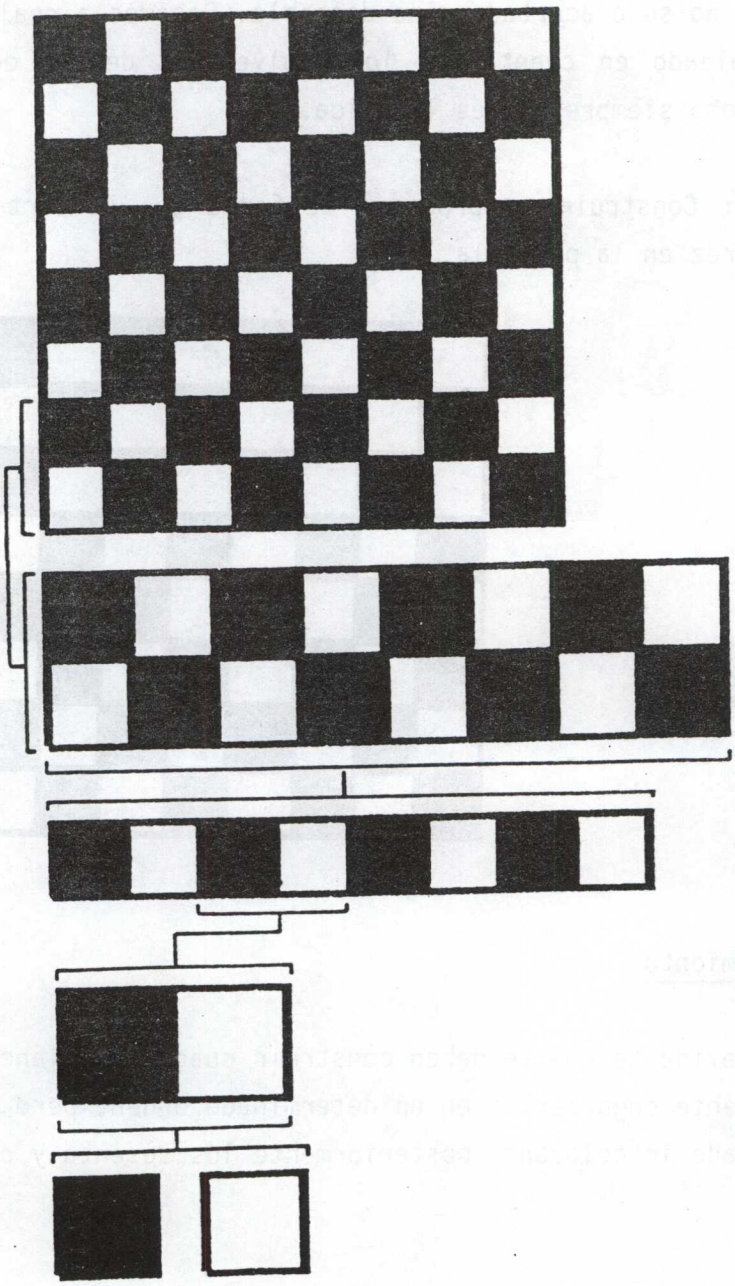
Planteamiento

Parece evidente que se deben construir cuadrados blancos y negros para posteriormente engarzarlos en un determinado orden, pero a simple vista parece complicado ir colocando posteriormente los sesenta y cuatro cuadrados.

Empecemos entonces por contemplar el tablero en su conjunto e intentemos ver las "partes" más grandes del tablero que engarzados den lugar al tablero general. Puede verse que dos filas juntas son la réplica exacta de las dos siguientes y así sucesivamente hasta cuatro veces.

Ya tenemos entonces un procedimiento -aún algo complejo, pero ya más simple- que repetido cuatro veces da el tablero.

¿Cómo lograr otro procedimiento más simple que nos construya las dos filas? Y así sucesivamente hasta lograr la descomposición del problema que se indica en la figura.



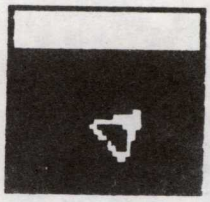
Programemos ahora los módulos que hemos creado:

El cuadrado blanco es un procedimiento ya conocido y no plantea ningún problema.

Podemos suponer que deseamos que los cuadrados pequeños sean de lado 20 o que el tablero sea de lado 160 o cualquier otra dimensión. Supongamos el caso de cuadrados pequeños de lado 20.

```
PARA CUADRADO
REPITE 4 [AV 20 GD 90]
FIN
```

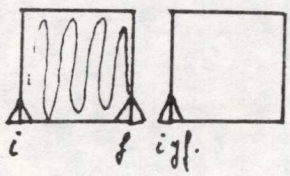
Ahora necesitamos un cuadrado negro del mismo lado que el anterior y esto podría lograrse haciendo que la tortuga lo rellene pasando la pluma por todo el cuadrado. Una forma podría ser:



que dará lugar al procedimiento:

```
PARA CUADRANEGRO
CUADRADO
SL GD 45 AV 10 BL
RELLENA
RE 10 GD 45 AV 20 GI 90
FIN
```

El siguiente paso consistirá en realizar un procedimiento nuevo que una los dos anteriores.



Habrá que tener en cuenta el estado final de la tortuga al realizar el negro para posicionarla y empezar el blanco.

Notemos que si el procedimiento nuevo consta solo de CUADRADO y CUADRANEGRO, la tortuga quedará en su estado final en el medio de los dos, situación que no es deseable si se pretende que una figura igual a ésta se añada por uno de los lados. Escribimos, entonces:

```
PARA DOMINO
CUADRANEGRO CUADRADO
GD 90 AV 20 GI 90
FIN
```

Con razonamientos análogos podríamos diseñar los siguientes procedimientos:

```
PARA FILA
REPITE 4 [DOMINO]
FIN
```

```
PARA BLOQUE
FILA GD 180 FILA
FIN
```

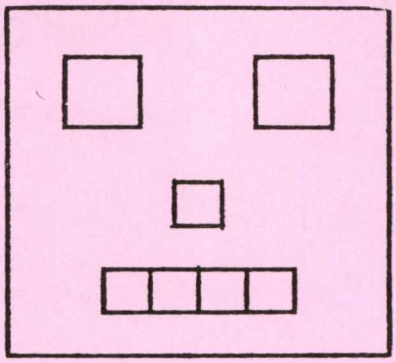
```
PARA TABLERO
REPITE 4 [BLOQUE SL AV 40 GD 180 BL]
FIN
```

Si queremos centrar el tablero en la pantalla, podemos añadir en la definición de tablero:

```
PARA TABLERO
SL PONPOS [-60 60] BL
REPITE 4 [BLOQUE SL AV 40 GD 180 BL]
FIN
```

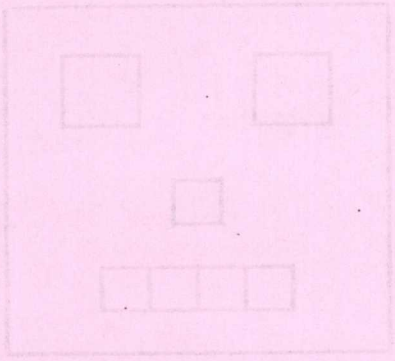
P R A C T I C A S

1) Definir un procedimiento que realice el siguiente dibujo.



2) Definir un procedimiento que realice una circunferencia fija, distinta de la definida para el reloj.

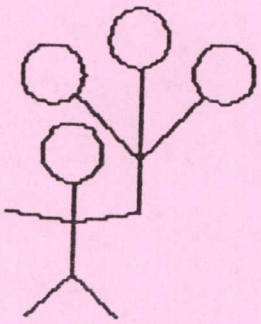
3) Definir un procedimiento que sea capaz de dibujar circunferencias de cualquier tamaño.



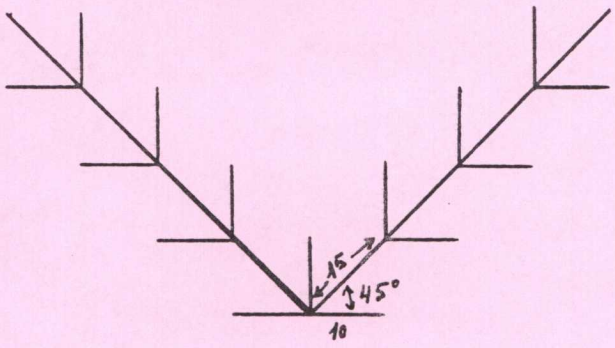
4) En caso de que en el procedimiento anterior no hayan utilizado como variable el radio, haz un programa para construir circunferencias con radio variable.

5) Utiliza el procedimiento anterior para definir otro que construya arcos con grados y radio variables.

6. Realizar el dibujo de la figura modularizando el problema.



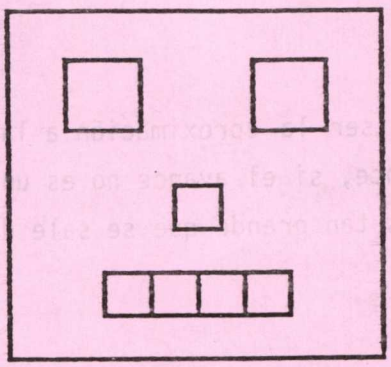
7. Idem con la siguiente figura.



8. Repetir el procedimiento utilizando variables.

S O L U C I O N P R A C T I C A S

1) Definir un procedimiento que realice el siguiente dibujo.



```
PARA CARA
OT
SL PONPOS [-40 -20] BL CUADRADO 80
SL PONPOS [-20 -10] BL REPITE 4 [CUADRADO 10
GD 90 AV 10 GI 90]
SL PONPOS [-5 10] BL CUADRADO 10
SL PONPOS [-28 33] BL CUADRADO 16
SL PONPOS [ 12 33] BL CUADRADO 16
FIN
```

2) Definir un procedimiento que realice una circunferencia fija, distinta de la definida para el reloj.

```
PARA CIRCUNFERENCIA
REPITE 360 [AV 1 GD 1]
FIN
```

3) Definir un procedimiento que sea capaz de dibujar circunferencias de cualquier tamaño.

```
PARA CIR.AV :AVANCE
REPITE 360 [AV :AVANCE GD 1]
FIN
```

Nótese que al ser la aproximación a la longitud circunferencia el número 360* avance, si el avance no es un número pequeño (1,2,3) la circunferencia es tan grande que se sale de la pantalla.

4) En caso de que en el procedimiento anterior no hayas utilizado como variable el radio, haz un programa para construir circunferencias con radio variable.

Podemos utilizar el procedimiento anterior simplemente teniendo en cuenta que la longitud de la circunferencia que nosotros dibujamos es 360 veces el avance y la longitud de una circunferencia en función del radio es 2π Radio. De esta forma

$360 * AVANCE = 2\pi RADIO;$

$AVANCE = \frac{2\pi}{360} * RADIO;$

$AVANCE = 0'017 * RADIO;$

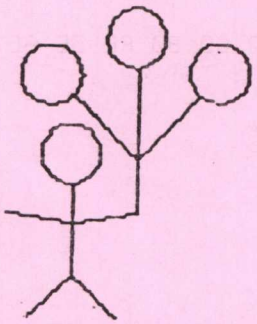
que sustituído en el procedimiento anterior dará lugar a:

```
PARA CIR.RAD :RADIO
REPITE 360 [AV 0.017 * :RADIO GD 1]
FIN
```

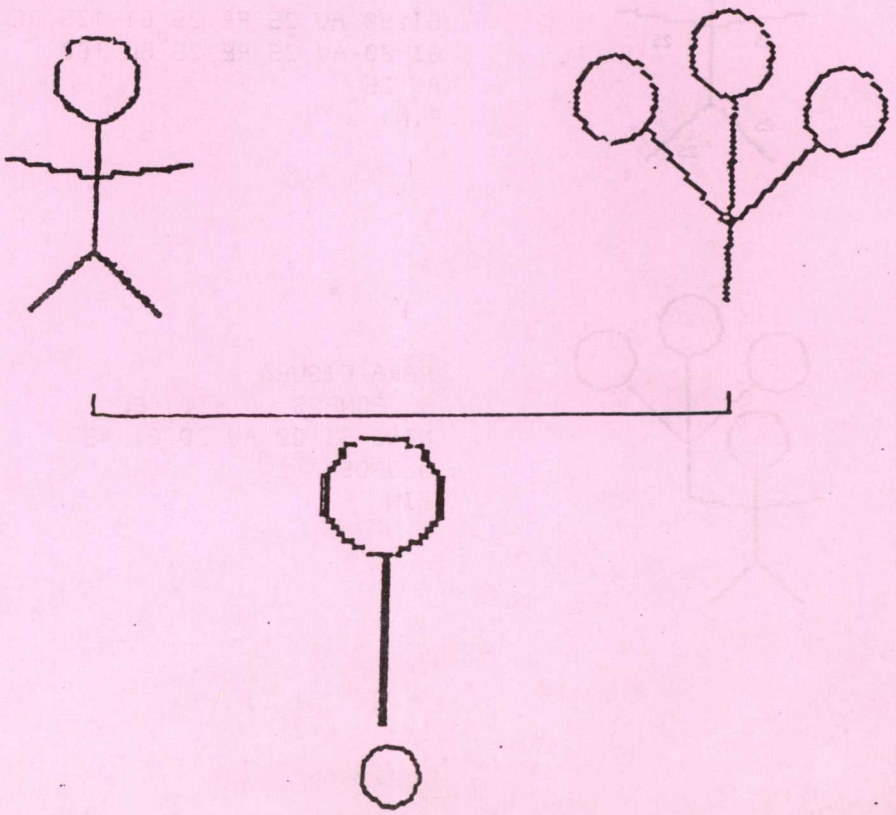
5) Utiliza el procedimiento anterior para definir otro que construya arcos con grados y radio variables.

```
PARA ARCO :GRADOS :RADIO
REPITE :GRADOS [AV 0.017 * :RADIO GD 1]
FIN
```

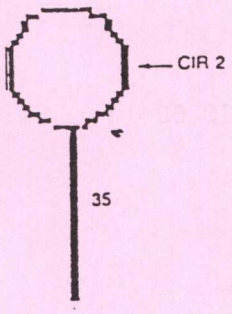
6. Realizar el dibujo de la figura modularizando el problema.



Una posible secuencia de descomposición del problema en partes podría ser:

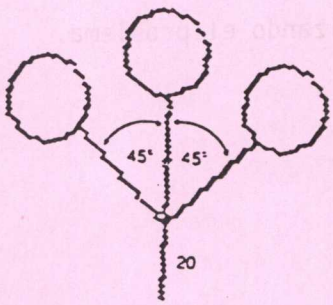


Como ya conocemos el procedimiento circunferencia, el problema podría resolverse así:

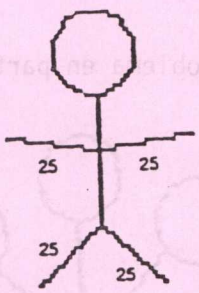



```
PARA CIR :X
REPITE 36 [AV :X GD 10]
FIN
```

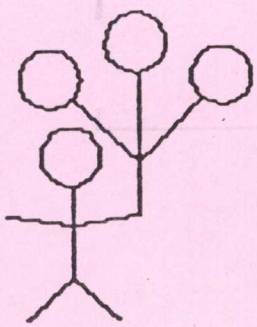
```
PARA GLOBO
OT AV 35 GI 90 CIR 2
FIN
```




```
PARA GLOBOS
OT
REPITE 3 [GLOBO GD 90 RE 35 GD 45]
GI 90 RE 20
FIN
```

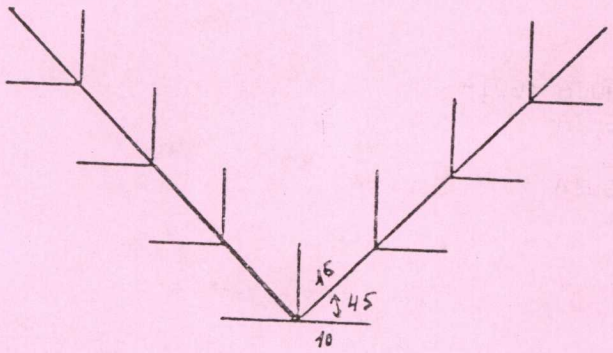



```
PARA NIÑO
GLOBO
GI 90 AV 35 GD 45 AV 25 RE 25
GI 90 AV 25 RE 25 GI 135 RE 20
GI 80 AV 25 RE 25 GD 160
AV 25
FIN
```



```
PARA FIGURA
SL PONPOS [0 -30] BL
NIÑO GI 80 AV 20 GI 45
GLOBOS
FIN
```

7. Idem con la siguiente figura.




```
PARA TROZO
AV 10 RE 10 GI 90
AV 10 RE 10 GI 45
AV 15
FIN

PARA RAMA
REPITE 4 [TROZO GD 45]
FIN
```

```
PARA ARBOL
RAMA
CENTRO
GD 90
RAMA
FIN
```

8. Repetir el procedimiento utilizando variables.

```
PARA TROZO :HOJA :GUIA
AV :HOJA RE :HOJA GI 90
AV :HOJA RE :HOJA GD 45
AV :GUIA
FIN
```

```
PARA RAMA :HOJA :GUIA
REPITE 4 [TROZO :HOJA :GUIA GD 45]
FIN


PARA ARBOL :HOJA :GUIA
RAMA :HOJA :GUIA
CENTRO GD 90
RAMA :HOJA :GUIA
FIN
```

F U N C I O N E S

INTRODUCCION

El lenguaje LOGO posee una serie de funciones y operaciones incorporadas como primitivas, es decir, que comprende directamente, además de la posibilidad de extender las mismas mediante definicionnes de nuevas operacioens o funciones cuando se necesiten para alguna aplicación específica, dado el carácter extensible del lenguaje.

Las funciones que lleva incorporadas el lenguaje varía mucho según la versión LOGO con la que se trabaje, por lo que es conveniente consultar el apéndice y el manual del lenguaje que se utilize para un mayor detalle. En cualquier caso conforme se describen las distintas funciones se hace constar la correspondiente salvedad.

Tanto las operaciones como las funciones son realmente tratadas por LOGO como procedimientos con una serie de entradas, y que generan una salida y es preciso que en definitiva una línea sea un comando, es decir no se puede escribir una línea de la forma

```
? 3 + 2
```

ya que al efectuar la operación daría como resultado una salida 5, y no sabría qué hacer con dicho valor y así lo indicará con un mensaje del tipo

```
NO SE QUE HACER CON 5
```

Serían en cambio válidos los comandos

```
? HAZ "A 3+2
? ES 3+2
5
```

OPERACIONES LOGICAS

Hay variables, denominadas predicados, que pueden tomar dos valores (VERDADERO o FALSO). Así, A B puede ser verdadero o falso, según los valores de A y de B. Se pueden realizar las operaciones .Y , .O y .NO., de acuerdo con las "tablas de verdad siguiente (V significa VERDADERO y F, FALSO)

P1	P2	Y	O
F	F	F	F
F	V	F	V
V	F	F	V
V	V	V	V

Como puede observarse la conjunción Y (operación lógica .Y .) toma el valor verdadero sólo en el caso de que los dos predicados P1 y P2 sean ciertos, mientras que la disyuinción o (operación lógica .O.) toma el valor falso sólo en el caso de que los dos precidados P1 y P2 sean falsos, y por tanto hasta que uno sea cierto para que la función O tenga valor verdadero.

En LOGO las operaciones lógicas Y y O son procedimientos de un número variable de entradas y se realizan de la forma:

```
Y predicado1 predicado2
(Y predicado1 predicado2 precidado3.....)
O predicado1 predicado2
(O predicado1 predicado2 predicado3.....)
NO predicado
```

Observese que cuando en número de entradas es superior a dos, hay que po-

ner paréntesis desde el principio de la operación hasta la última entrada.

Ejemplos:

- La operación

```
Y (COORX < 0) (COORY < 0)
```

da como salida "CIERTO en el caso de que la tortuga se encuentre en el tercer cuadrante y "FALSO en cualquier otro caso.

- La operación

```
O (COORX > 100) (COORX < -100)
```

da como salida "CIERTO en el caso de que el módulo de la coordenada x de la tortuga tenga un valor superior a 100.

```
? PONPOS [-50 -50]
? ES Y (COORX < 0) (COORY < 0)
CIERTO
? ES O (COORX > 100) (COORY < -100)
FALSO
```

Podemos definir nuevos predicados que dén como salida dos valores "CIERTO o "FALSO combinando con operaciones lógicas distintos predicados, como en los siguientes ejemplos:

```
(Y (COORX < 100) (COORX > -100) (COORY < 100) (COORY > -100))
```

toma el valor "CIERTO en el caso de que la tortuga se encuentre en el interior de un cuadrado horizontal de centro el origen y lado 200.

Podemos asignar a una variable el predicado anterior.

Por ejemplo

```
? HAZ "DENTRO (Y (COORX < 100) (COORX > -100) (COORY < 100)
(COORY < -100))
```

```
? ESCRIBE :DENTRO
CIERTO
? HAZ "FUERA NO :DENTRO
? ESCRIBE :FUERA
FALSO
```

FUNCIONES INCORPORADAS

LOGO opera con números enteros y decimales, y permite hacer operaciones de tipo aritmético, funciones trigonométricas, etc.

Algunas funciones como **ENTERO,AZAR,RESTO,REDONDEO,** dan como salida números enteros y otros dependen del resultado de la operación.

A continuación se relacionan las funciones disponibles más frecuentemente.

SUMA: Suma dos o más números. Equivale al signo + (2+3 equivale a SUM 2 3, ya que ambas dan como salida el número 5).

```
? ES SUMA 5 7
12
? ES (SUMA 1 3 3)
7
? ES SUMA 2.5 2.7
5.2
```

Obsérvese cómo si la suma es de más de dos números la función ha de ir entre paréntesis.

PRODUCTO:Multiplica dos o más números, y equivale al signo *.

Ejemplo:

```
? ES PRODUCTO 6 2
12
? ES (PRODUCTO 3 2 5)
30
```

RESTO : Divide los dos números indicados y da como salida el resto obtenido. Da un error si el segundo es 0.

Ejemplo:

```
? ES RESTO 13 5
3
? ES RESTO 13 15
13
? ES RESTO -13 5
-3
```

REDONDEA: Da como salida el valor entero más próximo al número dado.

Ejemplo:

```
? ES REDONDEA 5.3
5
? ES REDONDEA 5.52
6
? ES REDONDEA -5.8
-6
```

ENTERO: Da como salida la parte entera del número dado.

Ejemplo:

```
? ES ENTERO 5.2
5
```

```
? ES ENTERO 5.52
5
? ES ENTERO -5.8
-6
```

AZAR: Da como salida un número al azar entre 0 y el anterior al dado.

```
? ES AZAR 7
3
? REPITE 5 [AZAR 3]
1
0
2
1
0
```

COCIENTE: Da como salida el cociente de dividir dos números. Si el divisor es nulo, genera un mensaje de error. El resultado es idéntico al de la operación "/".

```
? ES COCIENTE 7 5
1.4
? ES COCIENTE 1 3
0.3333333333
```

ARCTAN: Da como salida el arcotangente en grados

```
? ES ARCTAN 1
45
? ES ARCTAN -1
-45
```

SEN: Da como salida el seno de un ángulo, dado en grados.

Ejemplo:

```
? ES SEN 45
0.7071067812
```

COS : Da como sálida el coseno de un ángulo en grados.

Ejemplo:

```
? ES COS 60
0.5
```

RC: Da como salida la raiz cuadrada del número.

Ejemplo:

```
? ES RC 23
4.795831523
```

EXP: Da como salida el número "e" elevado al número de entrada.

```
? ES EXP 2
7.389056099
? ES EXP -1
0.3678794412
```

LN : Da como salida el logaritmo en base "e" (logaritmo natural) del número de entrada. Si el número fuera negativo o nulo genera un mensaje de error:

```
? ES LN 2
0.6931471806
```

Las funciones anteriores se pueden combinar como se muestra seguidamente:

```
? ES ENTERO SUMA 3 5.5
8
? ES RESTO (SUMA 3 2) 3
2
? ES AZAR ENTERO 5.5 + 7.8
8
```

En este último caso, por ejemplo, la función + da como salida el número 13.2 a la siguiente función **ENTERO** que da como salida el número 13 a la siguiente función (AZAR), que genera un número aleatorio entre 0 y 12 y a su vez da salida al procedimiento **ESCRIBE** que ya es un comando que imprime el resultado.

Existen algunas diferencias entre las distintas versionnes de LOGO en este aspecto, de forma que, por ejemplo no existe la función ARCTAN o en algunos casos ni siquiera ninguna función trigonométrica (es el caso de TI-LOGO) o LN y EXP. En cualquier caso es recomendable consultar el manual concreto.

DEFINICION DE FUNCIONES

Cuando se desean definir nuevas funciones o procedimientos que no sean comandos, sino que generan una salida, hay que dar una orden de salida de los parámetros para que se transfieran al procedimiento que lo llama, y al ejecutarse realmente un programa ha de ser un comando, ya que, por ejemplo **AZAR** 5 daría un error, en cambio serían válidos **HAZ** "NUMERO **AZAR** 5 ó **ESCRIBE AZAR** 5. Esta salida de resultados, se realiza con la orden **DEVUELVE**(o abreviado **DEV**)

Ejemplos:
El siguiente procedimiento da como salida la distancia entre dos puntos del plano dados por sus coordenadas en forma de lista.

```
PARA DISTANCIA :POS1 :POS2
HAZ "X (PRIMERO :POS1) - (PRIMERO :POS2)
HAZ "Y (ULTIMO :POS1) - (ULTIMO :POS2)
DEV RC :X * :X + :Y * :Y
FIN


? ESCRIBE DISTANCIA [3 4] [1 2]
2.828427125
```

ESTRUCTURAS DE CONTROL

CONDICIONALES

El lenguaje LOGO posee una estructura condicional realizada mediante la primitiva SI (SI condicional) de forma que permite los dos tipos de ejecución condicional:

1) SI condición ENTONCES acción (ó control unario).
2) SI condición ENTONCES acción 1 EN CASO CONTRARIO acción 2 (ó control binario).

Ambos tipos de control se llevan a cabo con SI:

1) El control unario tiene la forma:

SI condición [lista de instrucciones]

2) El control binario tiene la forma:

SI condición [lista de instrucciones 1] [lista de instrucciones 2]

En el primer caso, la "lista de instrucciones se ejecuta o no dependiendo de que se verifique o no la condición que debe tomar un valor booleano (verdadero o falso).

En el segundo caso se ejecuta siempre una de las dos listas de instrucciones de la línea: si se verifica la condición la lista 1 y en caso contrario la lista 2.
En cualquier caso se ejecuta a continuación la siguiente instrucción.

En la versión del lenguaje TERRAPIN LOGO se utiliza otra notación, que es más frecuente en otros lenguajes. Así, utiliza: IF...THEN e IF...THEN..ELSE (SI...ENTONCES y SI...ENTONCES...EN CASO CONTRARIO respectivamente).

El control unario es, por tanto:

SI condición ENTONCES comando(s)

Y el control binario:

SI condición ENTONCES comando(s) SI NO comando(s)

Ejemplos:

```
SI :EDAD < 18 [ES "MENOR] [ES "MAYOR]
SI :NUMERO < 0 [ES "NEGATIVO]
```

En las dos sentencias anteriores SI se utiliza como comando de ejecución condicional. En la primera siempre se realiza una acción ES "MENOR o ES "MAYOR, en la segunda, si el número es positivo no se realiza ninguna acción.

El procedimiento MONEDA que se define a continuación es una función que da como salida la palabra "CARA o la palabra "CRUZ según que un número al azar sea 0 ó 1.

```
PARA MONEDA
SI 0 = AZAR 2 [DEV "CARA]
DEV "CRUZ
FIN
```

En este procedimiento, si el número al azar es 0 da como salida "CARA, y por tanto finaliza la ejecución de MONEDA dando a quien lo llamó el resultado "CARA. En caso contrario pasa a ejecutar la siguiente instrucción, dando salida "CRUZ.

Otra variante de MONEDA utiliza un SI con tres entradas:

```
PARA MONEDA
SI 0 = AZAR 2 [DEV "CARA] [DEV "CRUZ]
FIN
```

Una versión de MONEDA que utiliza SI como operación y no como comando es la siguiente:

```
PARA MONEDA
DEVUELVE SI 0 = AZAR 2 ["CARA] ["CRUZ]
FIN
```

También se pueden concatenar varios SI en una misma línea, como en el ejemplo siguiente:

```
PARA MAYOR :NUMERO
SI :NUMERO > 0 [SI :NUMERO >10 [ES [NUMERO MAYOR QUE 10]]
   [ES [NUMRO MENOR O IGUAL QUE 10]]] [ES [NUMERO NEGATIVO O
   NULO]]
FIN
```

Una forma adicional para el control condicional que tienen algunas versiones de LOGO es utilizando el trío COMPRUEBA, SICIERTO, SIFALSO

COMPRUEBA almacena el resultado de una operación booleana a lo largo de un procedimiento yen cualquier punto del mismo procedimiento se puede efectuar un control condiconal al valor almacenado en COMPRUEBA con SICIERTO y SIFALSO. Por ejemplo:

```
PARA PRUEBA :NUMERO
COMPRUEBA :NUMERO > 0
SICIERTO [ES [EL NUMERO ES POSITIVO]]
SIFALSO [ES [EL NUMERO ES NEGATIVO O NULO]]
        :
        :
SIFALSO [SI :NUMERO = 0 [ES [EL NUMERO ES NULO]]]
FIN
```

Obsérvese cómo COMPRUEBA almacena el resultado de una prueba en todo el procedimiento.

La estructura que tiene IF en LOGO, permite un control muy flexible y muy claro, a diferencia de otros lenguajes en los que es preciso recurrir a saltos para realizar el control condicional.

DETENCION DE UN PROCEDIMIENTO (ALTO, DEVUELVE)

Se puede hacer que un procedimiento detenga su ejecuci'on y devuelva el control al que lo llamó mediante las instrucciones ALTO, que simplemente detiene el procedimiento en el que se encuentra inserto dicho comando y no del programa, y DEVUELVE (abreviado DEV), que da un valor de salida al procedimiento que lo llamó.

En ambos casos se devuelve el control al procedimiento de nivel superior.

DEVUELVE es una instrucción que permite realizar una programación funcional, en la que cada procedimiento es un módulo-función que da una salida a otro módulo.

Ejemplos:

La siguiente función da como salida el valor absoluto de un número:

```
PARA ABS :NUMERO
SI :NUMERO >0 [DEV :NUMERO] [DEV  (0 - :NUMERO)]
FIN
```

La función que definimos a continuación, RAIZMODULO, da como salida la raiz cuadrada del módulo del número de entrada:

```
PARA RAIZMODULO :NUMERO
DEV RC ABS :NUMERO
FIN
```

Así, al ejecutar:

```
? ES RAIZMODULO -4
2
```

RAIZMODULO llama a ABS con valor -4, y éste le devuelve 4, que a su vez entrega a RC (raiz cuadrada) el valor 4, y ésta a su vez entrega 2 a quien lo llamó (ES) y por tanto, imprime 2.

Si se desea interrumpir el programa deteniendo el proceso completamente, no sirve **ALTO**, ya que ésta sólo interrumpe el procedimeinto en que se encuentra devolviendo el control al que lo llamó.

```
PARA PROCEDIMIENTO1
      :
      :                                  PARA PROCEDIMIENTO2
    PROCEDIMIENTO2                       .....................
      :                 ALTO             SI CONDICION [ALTO]
      :                                  .....................
      :                                  FIN
FIN
```

Para poder devolver el control al nivel más externo o superior, algunas versiones de LOGO poseen la primitiva **ENVIA**. Así, el comando SI condición [ENVIA "NIVELSUPERIOR] interrumpe el programa completo. (Para un mayor detalle de THROW y su asociada **RECIBE** es aconsejable acudir a los correspondientes manuales).

Ejemplo:
El siguiente procedimiento lleva la cuenta de veces que se ejecuta, y caso de ser superior a 10 interrumpe el programa.

```
PARA INC
HAZ "CUENTA :CUENTA + 1
SI :CUENTA > 10 [ES [TE HAS PASADO] ENVIA "NIVELSUPERIOR]
FIN
```

RECURSION

Un procedimiento se dice que es recursivo cuando se define en función de sí mismo. Ya hicimos algunos procedimientos que producían una repetición indefinida, como POLI.

```
PARA POLI :LADO :ANGULO
AV :LADO
GD :ANGULO
POLI :LADO :ANGULO
FIN
```

También sabemos cómo detener un proceso indefinido añadiendo un condicional:

```
PARA POLIESPIRAL :LADO :ANGULO
SI :LADO > 100 [ALTO]
AV :LADO
GD :ANGULO
POLIESPIRAL :LADO + 5 :ANGULO
FIN
```

En el primer caso (POLI), nos limitamos a repetir indefinidamente los comandos AV :LADO y GD :ANGULO, con iguales valores. En el segundo (POLIESPIRAL) la llamada al procedimiento se efectúa con valores diferentes. Así, POLIESPIRAL 20 100, llama a POLIESPIRAL 25 100, y ésta a POLIESPIRAL 30 100, y así sucesivamente hasta la llamada POLIESPIRAL 105 100, que detiene el procedimiento.

Otro ejemplo de procedimiento recursivo es el siguiente:

```
PARA CUENTA.ATRAS :NUMERO
ES :NUMERO
SI :NUMERO = 0 [ES "¡FUEGO! ALTO]
CUENTA.ATRAS :NUMERO - 1
FIN
```

```
? CUENTA.ATRAS 5
5
4
3
2
1
0
¡FUEGO!
```

Este CUENTA.ATRAS 5 llama a CUENTA.ATRAS 4 y así sucesivamente hasta que CUENTA.ATRAS 0 o interrumpe el proceso.

Cuando se utiliza la recursión en procedimientos que dan salida a otros procedimientos, cada uno de los procedimientos llama al siguiente, hasta que se cumple la condición de salida del bucle, en cuyo casose van entregando las salidas en orden inverso al que se fuera llamando.

En general, un procedimiento recursivo debe tener una llamada a sí mismo con un cambiode parámetros que le acerquen a la solución y una condición para finalizar el proceso.

Ejemplos:

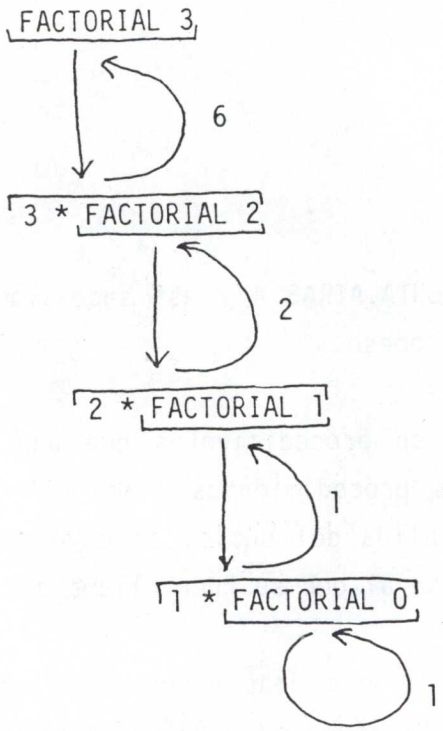
Factorial de un número

La función FACTORIAL es un ejemplo de función definida recursivamente, ya que se puede definir así:
- El factorial de 0 es 1.
- El factorial de N es N por el factorial de N-1.

```
PARA FACTORIAL :N
SI :N = 0 [DEV 1]
DEV :N * FACTORIAL :N - 1
FIN

? ES FACTORIAL 3
6
```

En el esquema se muestra el paso de parámetros y resultados en las sucesivas llamadas de FACTORIAL 3.



Obsérvese cómo en el momento en que se ejecuta FACTORIAL 0 da ya salida 1 a quien lo llamó FACTORIAL 1, y así sucesivamente se completa el cálculo de forma inversa a como se fueron llamando los procedimientos.

Máximo Común Divisor de dos números

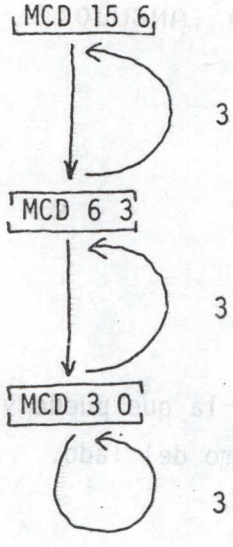
El Algoritmo de Euclides para calcular el máximo común divisor de dos números se puede expresar de forma recursiva.

- El M.C.D. de A y B es A si B = 0
- El M.C.D. de A y B es igual al M.C.D. de B y el resto de dividir A entre B.

```
PARA MCD :MAYOR :MENOR
SI :MENOR = 0 [DEV :MAYOR]
DEV MCD :MENOR RESTO :MAYOR :MENOR
FIN

? ES MCD 15 6
3
```

En el esquema siguiente se muestra el proceso para MCD 15 6.



RECURSION GENERAL

En todos los ejemplos vistos hasta ahora de procesos recursivos, el procedimiento intervenía en su propia definición siempre en la última línea de ésta, de forma que realmente el efecto producido era similar al de un proceso iterativo. A este tipo de recursión se le denomina "recursión por la cola". Cuando el procedimiento aparece en otro lugar de la definición o cuando aparece varias veces en la misma, da lugar a procesos más complejos, y a realizaciones inesperadas. Cuando estudiemos los gráficos recursivos y el uso recursivo de listas, utilizaremos a menudo la recursividad general. Por el momento, y para poder comprender mejor un proceso recursivo general, veremos cómo se transforman algunos de los procedimientos anteriores al cambiar la línea de la definición en la que aparece el propio procedimiento.

Ejemplos:

Primera Sorpresa

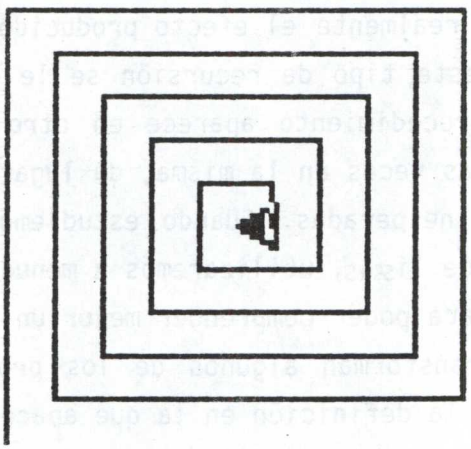
Si en el procediminto POLIESPIRAL anterior modificamos la definición de la forma siguiente:

```
PARA SORPRESA :LADO :ANGULO
SI :LADO > 100 [ALTO]
SORPRESA :LADO + 5 :ANGULO
AV :LADO
GD :ANGULO
FIN
```

Y ejecutamos:

SORPRESA 10 90

aparece el gráfico de la figura, en la que puede verse que es una poliespiral decreciente desde el valor máximo del lado.



Parece algo muy misterioso e inexplicable, pero veamos qué es lo que ha ocurrido.

SORPRESA 10 90 ha llamado a sí misma, tras comprobar que 10 no esmayor que 100, con valores 15 90, es decir SORPRESA 15 90, sin haber hecho nada, y así sucesivamente hasta llamar a SORPRESA 100 90, que al llamar a SORPRESA 105 90 detiene el proceso, devolviendo el control a SORPRESA 100 90, que continua su ejecución donde la dejó, es decir, realiza las órdenes **AV** 100 **GD** 90, finaliza y devuelve el control al que lo llamó SORPRESA 95 90, la que realiza las órdenes pendientes hasta acabar **AV** 95 **GD** 90 ... y así sucesivamente hasta el último pendiente SORPRESA 10 90, que ejecuta las últimas órdenes **AV** 10 **GD** 90, finalizando el procedimiento.

En el esquema siguiente se muestra este proceso. En el proceso hacia abajo no se realiza ninguna acción; éstas tienen lugar en el proceso hacia arriba.

```
SORPRESA 10 90              AV 10
                            GD 90
SORPRESA 15 90
                            AV 15
      :                     GD 90
      :                       :
      :                       :
SORPRESA 100 90             AV 100
                            GD 90
SORPRESA 105 90
                            ALTO
```

Segunda Sorpresa

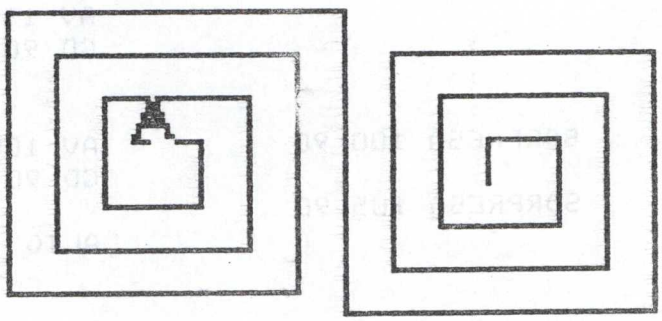
En el ejemplo anterior en el camino descendente no tenía lugar ninguna acción. Para observar el resultado de una acción antes y otra después de la línea en la que aparece el procedimiento, probaremos con el procedimiento SORPRESAS.

```
PARA SORPRESAS :LADO :ANGULO
SI :LADO > 70 [ALTO]
AV :LADO
GD :ANGULO
SORPRESAS :LADO + 5 :ANGULO
RE :LADO
GI :ANGULO
FIN
```

Ejecutamos para los mismos valores de antes:

```
SORPRESAS 10 90
```

dando lugar al gráfico de la figura.



El lector podrá seguir fácilmente el orden de la ejecución de las instrucciones, teniendo en cuenta que en el proceso descendente de llamadas del procedimiento SORPRESAS se ejecutan las órdenes de avanzar y girar a la derecha (AV :LADO y GD 90) y en el proceso ascendente cuando cada procedimiento devuelve el control a quien lo llamó se ejecutan las órdenes de retroceder y girar a la izquierda (RE :LADO y GI 90).

Una vez comprendido el proceso el lector estará en condiciones de no "sorprenderse" de los diseños recursivos y es más, podrá poco a poco aprender a lo largo de este libro cómo utilizarlos como técnica de programación de alto poder.

INTERACCION CON EL EXTERIOR

Se pueden introducir datos desde un dispositivo exterior: teclado, palanca de juegos, tableta gráfica, etc. y modificar desde el teclado interactivamente el flujo de ejecución del programa.Las primitivas que se indican a continuación sirven para leer datos desde el dispositivo de entrada, que supondremos es el teclado.

LEER UN CARACTER

Para leer un caracter se utiliza LEECARACTER (abreviado LC), que es una primitiva qué espera a que haya algún caracter tecleado y da como salida dicho caracter. El ordenador no genera ningún mensaje indicativo de lo que ha leido.
Algunas versiones de LOGO admiten leer varios caracteres (LEECARACTERES n) y leer una palabra(LEEPALABRA o abreviado LP)

LEER UNA LISTA

La función LEELISTA es similar a LEECARACTER pero da como salida una lista con la línea introducida desde el teclado hasta pulsar INTRO Si lo que se

desea es introducir una palabra, como LEELISTA da como salida una lista, hay que extraer de la lista el elemento, aunque éste sea sólo uno, con la función PRIMERO o ULTIMO. Por ejemplo:

```
HAZ "PALABRA.LEIDA PRIMERO LEELISTA
```

Ejemplos:

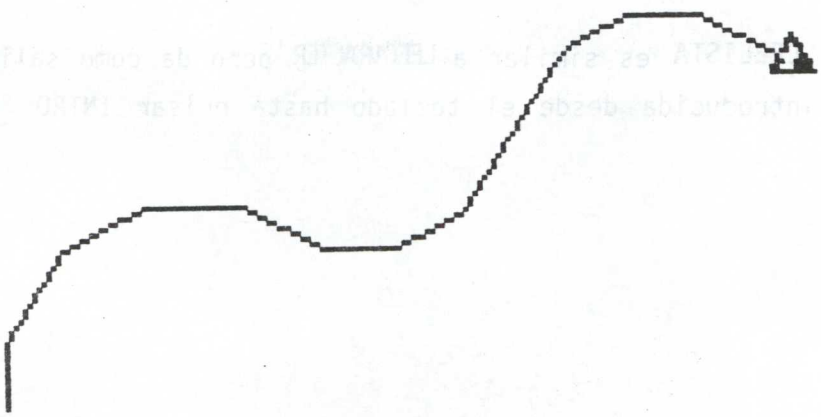
Conducir la tortuga

Desarrollaremos el procedimiento CONDUCIR que dirige la tortuga desde el techado, haciendo que avance 5 pasos pacada vez que se teclee "A, que gire 30° a la derecha o la izquierda cada vez que se teclee "D o "I respectivamente y que pare el proceso de teclear "S.

El procedimiento es un bucle indefinido que lee un caracter y ejecuta la orden pertinente.

```
PARA CONDUCIR
HAZ "ORDEN LC
SI :ORDEN = "A [AV 5]
SI :ORDEN = "D [GD 30]
SI :ORDEN = "I [GI 30]
SI :ORDEN = "P [ALTO]
CONDUCIR
FIN
```

Al ejecutar conducir, la tortuga va evolucionando conforme a las órdenes del teclado.



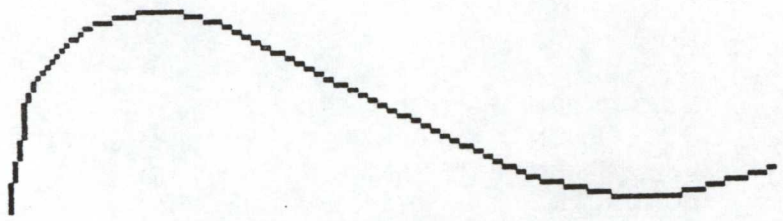
Pilotar.

Con una pequeña modificación, el siguiente ejemplo hace avanzar la tortuga de 5 en 5 unidades y gira 10° a derecha o izquierda cada vez que se teclea D o I.

```
PARA PILOTAR
AV 5
SI TECLA? [HAZ "ORDEN LC] [HAZ "ORDEN " ]
SI :ORDEN = "D [GD 10]
SI :ORDEN = "I [GI 10]
SI :ORDEN = "P [ALTO]
PILOTAR
FIN
```

Y al ejecutarlo:

```
?PILOTAR
```



Se ha utilizado la función TECLA?para saber si hay algún dato del teclado a la espera de leerse.TECLA?es una función que toma el valor"CIERTOo "FALSO según hay algún caracter esperando para ser leido.

Ejemplo:
Si hay alguna tecla accionada, leer el dato e imprimir el mensaje: el caracter leido es el .....

```
?SI  TECLA [ES [EL CARACTER LEIDO ES] ES LC]
EL CARACTER LEIDO ES
```



Aquí imprimirá el carácter que se haya tecleado.

P R A C T I C A S

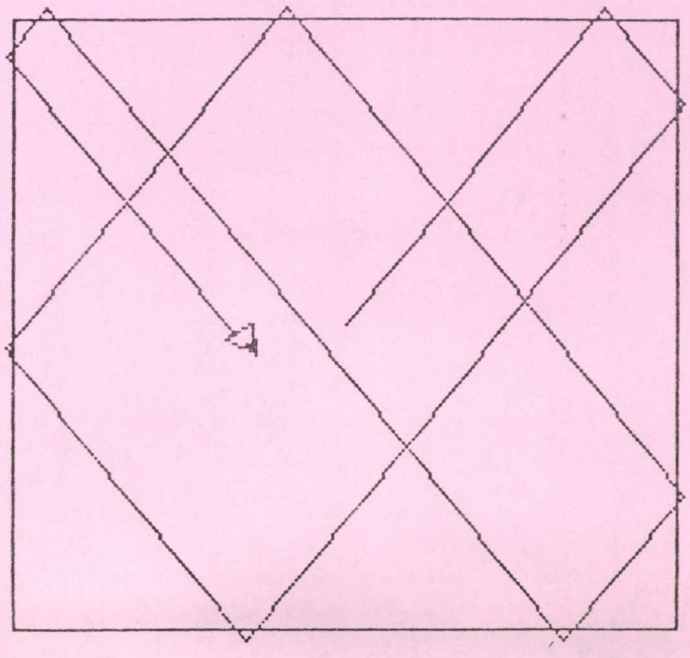
1) a-Diseñar un procedimiento para dibujar conduciendo a la tortuga desde teclado, de forma que al pulsar "A" avance 5 pasos, si se pulsa "R" retrocede 5 pasos, si "D" gira 10º a la derecha, si "I" gire 10º a la izquierda, si "S" suba la plumilla, y si "B" la baje.

b- Modificar el procedimiento anterior para poder rectificar el dibujo, de forma que al darle una orden pueda borrar un trazo.

2) El procedimiento POLI realiza figuras poligonalees que se repiten de forma indefinida. Basándose en él definir un procedimiento que dibuje sólo un ciclo del gráfico, es decir, que cuando haya realizado un ciclo completo, pare el proceso

```
PARA POLI :LADO :ANGULO
AV :LADO
GD :ANGULO
POLI :LADO :ANGULO
FIN
```

3) Realizar un programa que haga moverse indefinidamente a la tortuga en el interior del recinto de la figura, rebotando en sus paredes.



4) SIMULACION DEL MOVIMIENTO DE LOS ANIMALES

a) Simular el movimiento de un animal que se desplaza avanzando una distancia al azar y girando un ángulo al azar (a la derecha y a la izquierda)

Definir un procedimiento general que permita variar parámetros y probar para distinntos pasos y ángulos de giro a la izquierda y a la derecha.

b) Modificarlo de forma que la tortuga no se salga de los límites de un recinto. Para ello se comprobará previamente si se sale fuera de los límites, en cuyo caso se seguirá alguna estrategia que le haga cambiar de trayectoria. Por ejemplo: dar media vuelta, o girar de grado en grado hasta que compruebe que no se salga de los límites en el movimiento de avanzar.

SOLUCION A PRACTICAS

```
PARA DIBUJAR
ESPERA.ORDEN
SI  :ORDEN = "A [AV 5]
SI  :ORDEN = "R [RE 5]
SI  :ORDEN = "D [GD 10]
SI  :ORDEN = "I [GI 10]
SI  :ORDEN = "S [SL]
SI  :ORDEN = "B [BL]
SI  :ORDEN = "C [PONCL 0 AV 5 PONCL 3]
DIBUJAR
FIN

PARA ESPERA.ORDEN
SI TECLA? [HAZ "ORDEN LC ALTO]
ESPERA.ORDEN
FIN
```

2)

Para poder detener el procedimiento POLI cuando haya dibujado una figura completa, podría haberse llevado la cuenta del ángulo girado en total, y cuando hubiera sido múltiplo de 360°, es decir, cuando el resto de dividir el ángulo girado por 360 fuera 0, parar el proceso, es decir, repetir hasta que se cumpla la condición citada:

```
PARA POLI :LADO :ANGULO
AV :LADO GD :ANGULO
SI RUMBO = 0 [ALTO]
POLI :LADO :ANGULO
FIN
```

Si ejecutamos para distintos valores, obtenemos las figuras POLI ya conocidas con la única diferencia de que el proceso se detiene al cerrarse la figura. Por ejemplo, POLI 50 144 de la figura con la tortuga nuevamente en la posición inicial.

3) Tortugas que rebotan en el interior de un recinto.

Haremos un procedimiento que haga moverse indefinidamente a la tortuga rebotando en el interior de un recinto.

El procedimiento principal REBOTES debe dibujar el recinnto, dar un rumbo inicial a la tortuga y repetir indefinidamente el proceso de hacer avanzar a la tortuga y orientarla adecuadamente cuando vaya a superar el recinto. El procedimiento ABS lo definimos nosotros, como hemos visto anteriormente.

```
PARA REBOTES :RUMBOINICIAL
RECINTO
SL CENTRO BL
PONRUMBO :RUMBOINICIAL
REBOTAR
FIN
```

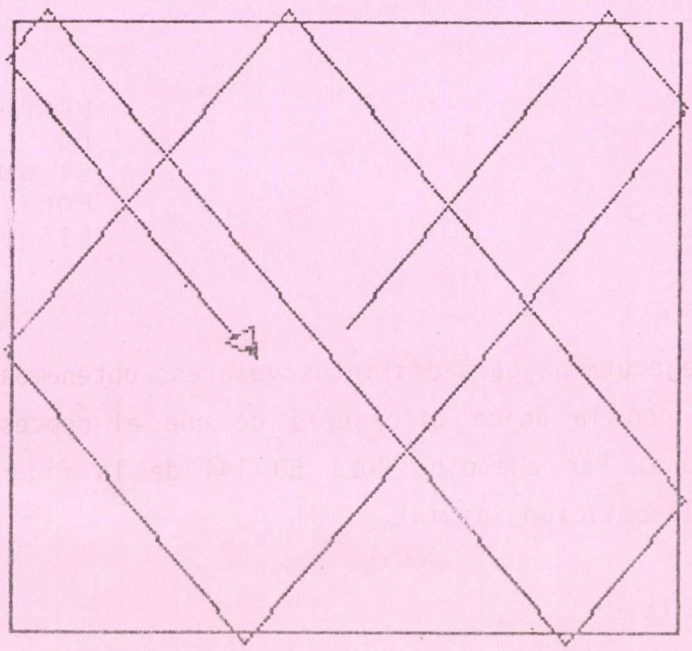
```
PARA RECINTO
BP SL PONPOS [-100 -100] BL
REPITE 4 [AV 200 GD 90]
FIN
```

```
PARA ORIENTAR
SI ABS COORX > 100 [PONR 0 - RUMBO]
SI ABS COORY > 100 [PONR 180 - RUMBO]
FIN
```

```
PARA REBOTAR
AV 5
ORIENTAR
REBOTAR
FIN
```

El procedimiento ORIENTAR cambia de rumbo cuando se superan los límites del recinto.

Ejecutando para un ROMBOINICIAL 37 se obtiene el movimiento de la figura:



P R A C T I C A S (Recursividad)

1.- a) Realizar un trazado de la ejecución del procedimiento MISTERIO para NUMERO = 4. Se deberá predecir el comportamiento del programa antes de ejecutarlo.

```
PARA MISTERIO :NUMERO
SI :NUMERO = 0 [ALTO]
MISTERIO :NUMERO - 1
ES :NUMERO
FIN
```

b) Idem para DIBUJO

```
PARA DIBUJO :CUENTA
SI :CUENTA > 4 [ALTO]
AV 50
GD 90
DIBUJO :CUENTA + 1
AV 40
GI 90
FIN
```

2.- Realizar un procedimiento que utilice la recursividad para dibujar la pila de cajas de la figura primera. Modificarlo para que realice el dibujo en orden inverso, como muestra la figura segunda.

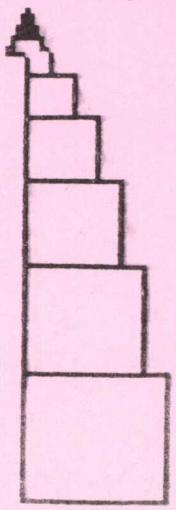


Figura 1

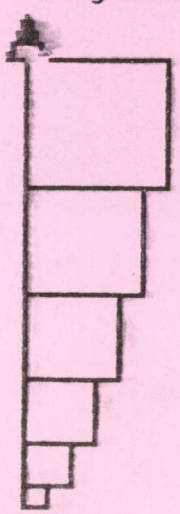
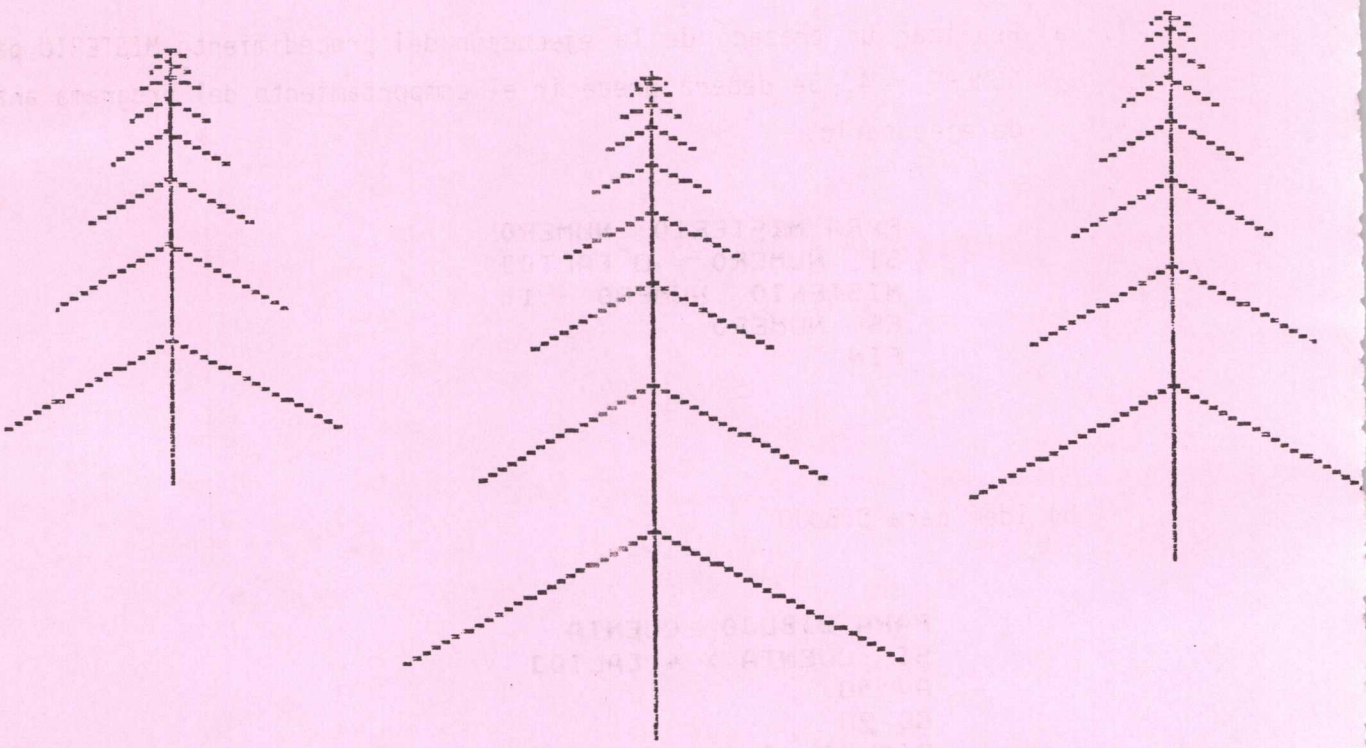


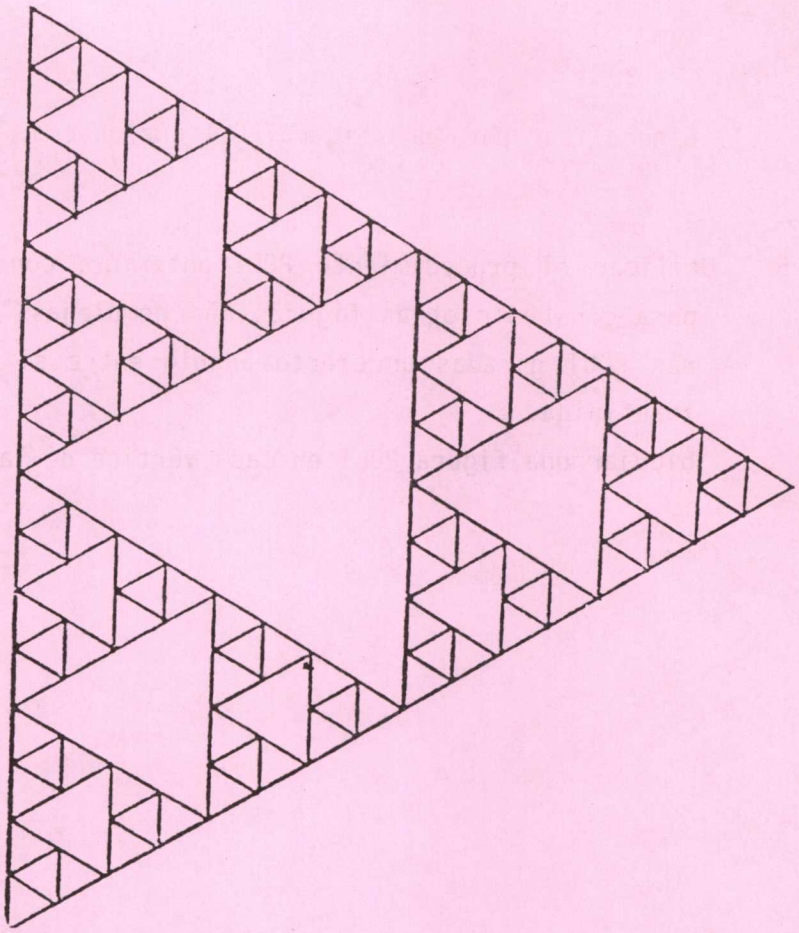
Figura 2

3.- Realizar un procedimiento que utilice la recursividad para dibujar abetos como los de la figura



4.- ANIDAMIENTO DE FIGURAS

Un procedimiento que permite dibujar triángulos como el de la figura es el siguiente:



```
PARA TRIANGULOS :LADO
SI :LADO < 10 [ALTO]
REPITE 3 [TRIANGULOS :LADO / 2 AV :LADO GD 120]
FIN
```

a) Ejecutarlo para diversos valores, observando cómo se forman las figuras.

b) Generalizarlo para poder dibujar otras figuras anidadas de pentágonos, hexágonos, etc.

c) Realizar una modificación del procedimiento para que la salida del proceso recursivo sea, en lugar de por el tamaño del lado, por el nivel que deseemos. Para ello, se introducirá la variable NIVEL como otra entrada a los procedimientos.

d) ¿Qué pasará si cambiamos el orden de los comandos
avanzar : LADO y TRIANGULOS : LADO/2

Probarlo para diversos valores los procesos anteriores
('b' o 'c' y 'd')

e) Generalizar para una figura POLI cualquiera.

5. Utilizar el procedimiento POLI anterior, con la variación introducida, para construir otras figuras más complejas. Sugerencias: Dibujar figuras POLI giradas un cierto ángulo entre sí y que se repiten de forma indefinida.
   Dibujar una figura POLI en cada vértice de la figura POLI original ...

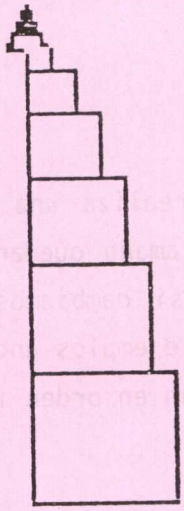
SOLUCION A PRACTICAS (Recursividad)

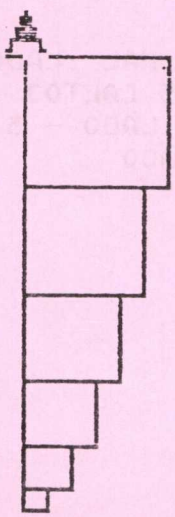
2) Apilar cajas

El siguiente procedimiento realiza una pila de cajas en las que las cajas (cuadrados) de menor tamaño quedan en la parte más alta. El procedimiento es recursivo, y si cambiamos el orden de las líneas, por las mismas causas que en los ejemplos anteriores, se produce un proceso al revés, es decir, se apilan en orden inverso, de menor a mayor tamaño.

```
PARA APILAR.BIEN :LADO
SI :LADO < 5 [ALTO]
CUADRADO :LADO
AV :LADO
APILAR.BIEN :LADO - 5
FIN

PARA APILAR.MAL :LADO
SI :LADO < 5 [ALTO]
APILAR.MAL :LADO - 5
CUADRADO :LADO
AV :LADO
FIN
```



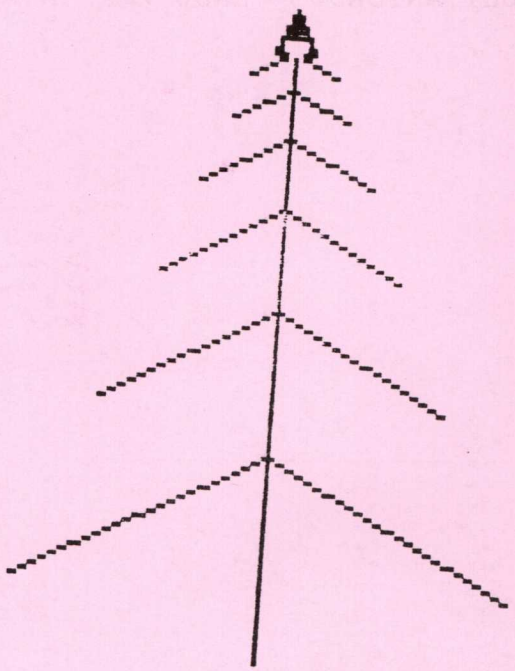


En este ejemplo hay una representación gráfica de cómo opera la recursividad, guardando en una pila las variables de los procedimientos y extrayéndolas en orden inverso.

3.-

```
PARA AGUJAS :LONG
AV :LONG * 0.8 GI 120
AV :LONG RE :LONG GI 120
AV :LONG RE :LONG
GI 120
FIN

PARA ABETO :LONG
SI :LONG < 10 [ALTO]
AGUJAS :LONG
ABETO :LONG * 0.7
FIN
```



4)

```
PARA HEXAGONOS :LADO
SI :LADO < 10 [ALTO]
REPITE 6 [HEXAGONOS :LADO / 2 AV :LADO GD 60]
FIN


PARA TRIANGULOS :LADO :NIVEL
SI :NIVEL = 0 [ALTO]
REPITE 3 [TRIANGULOS :LADO / 2 :NIVEL - 1 AV :LADO GD 120]
FIN


PARA POLI.ANIDADOS :LADO :N
SI :LADO < 10 [ALTO]
REPITE :N [POLI.ANIDADOS :LADO / 2 :N AV :LADO GD 360 / :N]
FIN
```

PALABRAS Y LISTAS

INTRODUCCION

Uno de los mayores atractivos del lenguaje LOGO estriba en su gran potencia para el manejo de listas. Ya se comentó anteriormente que LOGO era un dialecto de LISP, y éste (LISt Processing) es un lenguaje de Inteligencia Artificial diseñado para aplicaciones tales como reconocimiento del lenguaje natural, análisis sintáctico, etc. La estructura básica de datos sobre la que se apoyan los lenguajes de Inteligencia Artificial es la lista, y ésta entendida en un sentido abstracto como conjuntos de elementos que se denominan átomos. A partir de la estructura básica lista, se pueden establecer estructuras jerárquicas, árboles de datos, etc.

Ya contamos cómo el lenguaje LOGO se desarrolla precisamente en el Laboratorio de Inteligencia Artificial del M.I.T. y con él comienza a abrirse camino algo que ya es admitido hoy en día por la mayor parte de los especialistas en el campo de la aplicación de los ordenadores a la Educación, y es que la Inteligencia Artificial, con su desarrollo actual y el que se prevé en un futuro inmediato, está revolucionando los modos de usar los ordenadores en la Enseñanza y que sus posibilidades están tan sólo comenzando a explotarse hoy.

LOGO tiene incorporadas una serie de funciones para manejo de listas, en el sentido abstracto del que ya hemos hablado, y además, y esto es aún más importante, extensibilidad que se puede lograr conforme el usuario puede ir necesitando nuevas funciones, a partir de las primitivas ya existentes. Y este proceso no tiene límite superior.

Al igual que ocurría con los gráficos, en los programas en los que intervienen listas, el instrumento que da potencia a LOGO es la recursividad. Como se verá en este capítulo, el uso recursivo de listas permite con muy pocas sentencias realizar aplicaciones muy complejas.

Mientras que en lenguajes como BASIC los programas para manejar listas son verdaderamente intrincados, en LOGO son conceptualmente claros y modulares, en los que cada módulo realiza una función precisa. Unos procedimientos

se encadenan a otros estableciendo un flujo de datos de salida de unos a otros, como si fueran funciones, con lo cual es factible llevar a cabo una programación orientada a las funciones, o programación funcional.

Todas las familias del lenguaje LOGO derivadas del original desarrollado en el M.I.T. tienen estas capacidades realizadas en el lenguaje, si bien puede haber ligeras diferencias, como pueden ser las existencias de unas primitivas o no, pero que en cualquier caso se pueda construir a partir de las existentes, como puede ser el caso de primitivas para extraer un elemento de una lista, de contar su número de elementos, o el lugar que ocupa un determinado elemento en una lista. Otra diferencia importante es en tener o no primitivas que puedan manejar directamente listas de propiedades (PLIST); éstas se encuentran como parte del lenguaje LOGO desarrollado por LCSI para el PC de IBM, APPLE-LOGO, pero no en ATARI-LOGO, y sí en el TERRAPIN-LOGO. Existe una variante de LOGO, realizada en el Departamento de Inteligencia Artificial de la Universidad de Edimburgo en Escocia, que se adapta más fielmente a LISP, con lo que su potencia en proceso de listas es superior al del resto de la familia LOGO, y se denomina RML (Research Machines LOGO).

PALABRAS Y LISTAS

Una palabra es un conjunto de caracteres. Ejemplos de palabras son:

HOLA
X
XY
3.1415
SUERTE
CUAL?

Cada caracter es un elemento de la palabra. Así, la palabra CUAL?, tiene como elementos: C,U.A,L y ?

Los caracteres blancos sirven de delimitadores, de forma que, por ejemplo, si se ordena:

```
?ES 10/2
10/2
```

ya que interpreta / como un caracter, si bien al ejecutar:

```
?ES 10 / 2
5
```

ya que el caracter blanco sirve de separador, y al analizar la cadena de caracteres que se le manda imprimir, detecta el caracter / como operación y busca a continuación el dividendo (2), efectuando la operación antes de imprimir.

Para asignar a una variable como valor una palabra, se utiliza ésta con comillas, para que lo interprete como cadena de caracteres.

```
?HAZ "NOMBRE "JUAN
```

asigna a la variable NOMBRE la palabra JUAN. Así, si se da la orden de imprimir el contenido de la variable:

```
?ES :NOMBRE
JUAN
```

Las listas se realizan a base de objetos, ya sean palabras u otras listas.

Ejemplos de listas son:

```
[HOLA AMIGO]
[A B C]
[[HOLA AMIGO][COMO ESTAS]]
[A [B C D] E [F G]]
[A]
[1 [2 3] [4 5] [6 7 8] [9 [10 11]]]
```

Así, los elementos de:

```
[[HOLA AMIGO] [COMO ESTAS]]
```

son dos listas: [HOLA AMIGO] y [COMO ESTAS], que a su vez tienen como elementos las palabras HOLA AMIGO y COMO ESTAS respectivamente. Por tanto,

[HOLA AMIGO] es un elemento de la lista, pero no así HOLA.

Los elementos de:

[1 [2 3] [4 5] [6 7 8] [9 [10 11]]]

son:

1

[2 3] , que a su vez tiene como elementos 2 y 3.

[4 5] , que a su vez tiene como elementos 4 y 5.

[6 7 8] , que a su vez tiene como elementos 6, 7 y 8.

[9 [10 11]], que a su vez tiene como elementos 9 y [10 11], que a su vez tiene como elementos 10 y 11.

Una lista vacía o sin elementos se representa como [ ] y una palabra vacía como " seguida de un espacio en blanco.

OPERACIONES CON PALABRAS Y LISTAS

Para manejar objetos del tipo palabra o lista, se utilizan distintas funciones definidas en LOGO que veremos a continuación:

Extracción de sus elementos

PRIMER ELEMENTO

La función **PRIMERO** da como salida el primer elemento de un objeto.

Ejemplo:

```
PRIMERO [A B C] da como salida A
? ES PRIMERO [CARA CRUZ]
CARA
? ES PRIMERO "CARA
C
? ES PRIMERO [C A R A]
C
? ES PRIMERO [[A B] C]
A B
```

Si se utiliza NUESTRA en lugar de ES la lista se muestra como es, con sus corchetes exteriores, así:

```
? MUESTRA PRIMERO [[A B] C]
[A B]
```

Si se aplica a un elemento vacío da un error, al igual que ocurre con las que se describen seguidamente.

ULTIMO ELEMENTO

La función ULTIMO da como salida el último elemento de un objeto.

```
? ES ULTIMO [CARA CRUZ]
CRUZ
? ES ULTIMO [CARA]
CARA
? ES ULTIMO "CARA
A
? ES ULTIMO [A B C [D E]]
D E
? MUESTRA ULTIMO [O [[1 2]]]
[[1 2]]
```

TODOS MENOS EL PRIMER ELEMENTO

Hay una función que da como salida todo el objeto excepto el primer elemento, y es MENOS PRIMERO (abreviado MP)
Ejemplos:

```
? ES MENOSPRIMERO [A B C D]
B C D
? ES MENOSPRIMERO [1 [2 3] 4]
[2 3] 4
? ES MENOSPRIMERO "PALABRA
ALABRA
```

El siguiente procedimiento elimina cada vez un elemento empezando por el primero, imprimiendo en cada paso el resultado.

```
PARA TRIANGULO :OBJETO
SI :OBJETO = " [ALTO]
ESCRIBE :OBJETO
TRIANGULO MENOSPRIMERO :OBJETO
FIN
```

El procedimiento TRIANGULO se llama a sí mismo con una nueva variable OBJETO, que es la anterior menos el primer elemento, y eso hasta que esté vacía. Así, al aplicarlo por ejemplo a ENSEÑANZA, imprime la primera vez ENSEÑANZA, la segunda vez NSEÑANZA, y así hasta A.

Este es el primer ejemplo de manejo de palabras que utiliza la recursividad. Como debe ocurrir en todo proceso recursivo, tiene que haber una salida, una condición de terminación en la definición. En este caso es:

```
SI :OBJETO = "  [STOP]
```

```
?TRIANGULO "ENSEÑANZA
ENSEÑANZA
NSEÑANZA
SEÑANZA
EÑANZA
ÑANZA
ANZA
NZA
ZA
A
```

TODOS MENOS EL ULTIMO ELEMENTO

Existe una función que da como salida todo el objeto excepto el último elemento, y es **MENOSULTIMO (MU)**.
Ejemplos:

```
? ES MENOSULTIMO "PAJAROS
PAJARO
? ES MENOSULTIMO [LOS PAJAROS VUELAN ALTO]
LOS PAJAROS VUELAN
? ES MENOSULTIMO [[LOS PAJAROS] [QUE VUELAN ALTO] [VIAJAN] [A
TIERRAS LEJANAS]
[LOS PAJAROS] [QUE VUELAN ALTO] [VIAJAN]
? MUESTRA MENOSULTIMO [[[LOS PAJAROS] [QUE VUELAN ALTO]]]
[]
```

El siguiente programa elimina cada vez un elemento empezando por el primero e imprimiendo en cada paso el resultado.

```
PARA DISMINUIR :OBJETO
SI :OBJETO = [] [ALTO]
ES :OBJETO
DISMINUIR MENOSULTIMO :OBJETO
FIN
```

```
?DISMINUIR [¡AY! PEPE ASI NO SE PUEDE HACER NADA]
¡AY! PEPE ASI NO SE PUEDE HACER
¡AY! PEPE ASI NO SE PUEDE
¡AY! PEPE ASI NO SE
¡AY! PEPE ASI NO
¡AY! PEPE ASI
¡AY! PEPE
¡AY!
```

El siguiente procedimiento realiza la separación de una "FRASE en las palabras en que se descompone.

```
PARA SEPARAR :FRASE
SI :FRASE = [] [ALTO]
ES PRIMERO :FRASE
SEPARAR MENOSPRIMERO :FRASE
FIN
```

```
?SEPARAR [LOS PAJAROS VUELAN MUY ALTO]
LOS
PAJAROS
VUELAN
MUY
ALTO
```

Como puede observarse, se llama recursivamente al procedimiento SEPARAR, pero cada vez con MENOSPRIMERO :FRASE, es decir, con toda la frase excepto la primera, hasta que no haya más palabras en la frase. Es decir, que llama a SEPARAR con la "FRASE [LOS PAJAROS VUELAN MUY ALTO], después con [PAJAROS VUELAN MUY ALTO],... hasta [ALTO].

Ampliación de objetos con nuevos elementos

Existen primitivas ya construidas en el propio lenguaje que permiten añadir un elemento al principio o al final de una lista o de una palabra. Veremos cómo a partir de estas primitivas se puede añadir un elemento en cualquier posición del objeto.

AÑADIR UN ELEMENTO AL PRINCIPIO DE UNA LISTA

La primitiva PP (abreviatura de PONPRIMERO; o poner primero) da como salida una nueva lista con un elemento más al principio. Las entradas de esta primitiva son el elemento a introducir y la lista.
Ejemplos:

```
? ES PP "ARBOL [FLOR RAMA PLANTA]
ARBOL FLOR RAMA PLANTA
? ES PP "ARBOL MENOSPRIMERO [FLOR RAMA PLANTA]
ARBOL RAMA PLANTA
```

En este último ejemplo hemos sustituido el primer elemento de la lista, ya que MENOSPRIMERO [FLOR RAMA PLANTA] entrega como salida [RAMA PLANTA] a PP y ésta le añade al principio ARBOL, con lo que entrega a ESCRIBE [ARBOL RAMA PLANTA] que ejecuta ya la orden.

AÑADIR UN ELEMENTO AL FINAL DE UNA LISTA

La primitiva PU (abreviatura de PONULTIMO da como salida una nueva lista con un elemento más al final. Las entradas de esta primitiva son el elemento a introducir y la lista.
Ejemplos:

```
? ES PU "ARBOL [FLOR RAMA PLANTA]
FLOR RAMA PLANTA ARBOL
? ES PU "ARBOL MENOSULTIMO [FLOR RAMA PLANTA]
FLOR RAMA ARBOL
```

De forma similar a la anterior, hemos sustituido en el último ejemplo el

último elemento de la lista.

```
?ES PU [D] [A [B C]]
A [B C] [D]
```

AÑADIR UN ELEMENTO EN UNA POSICION CUALQUIERA

Utilizando las primitivas PP y PU se puede definir un procedimiento que permita añadir un elemento dado en una posición cualquiera de la lista.

Para lograrlo definimos un procedimiento que llamaremos AÑADIR, cuyas entradas serán el número de puesto donde deseamos añadir el elemento nuevo y la lista original.

Este procedimiento se definirá en función de sí mismo de forma que cada vez se llamará con la lista disminuida en el primer elemento y el número de puesto decrementado en una unidad hasta que el puesto sea el 1, con lo cual podremos utilizar PP. Además, en cada paso debemos reponer el elemento que vamos quitando.

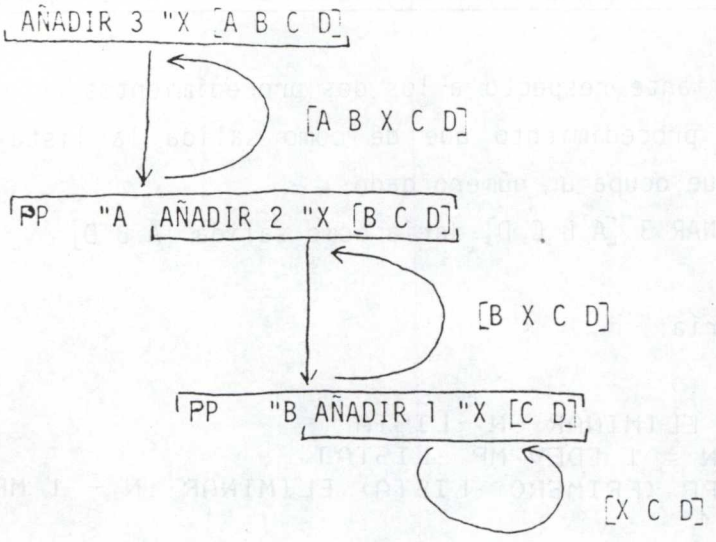
El procedimiento será:

```
PARA AÑADIR :N :ELEMENTO :LISTA
SI :N = 1 [DEV (PP :ELEMENTO :LISTA)]
DEV PP (PRIMERO :LISTA) AÑADIR :N-1 ELEMENTO MP :LISTA
FIN
```

Asi, ejecutando AÑADIR para N=3 ELEMENTO= "X y LISTA = [A B C D], obtenemos:

```
?ES AÑADIR 3 "X [A B C D]
A B X C D
```

Seguiremos el proceso que efectúa el procedimiento al ejecutarse.



Las flechas indican salidas de procedimientos. Así, AÑADIR 3 "X [A B C D] da como salida PP "A AÑADIR 2 "X [B C D], y éste último PP "B AÑADIR 1 "X [C D]. Como :N=1, ya alcanza la condición de salida de la recursión y por tanto da como salida PP "X [C D], es decir [X C D] y que entrega al que lo llamó, y así sucesivamente vuelve atrás hasta entregar la salida definitiva [A B X C D].

REEMPLAZAR UN ELEMENTO DE UNA LISTA

Con una ligera variante, podemos reemplazar el elemento de la lista, si en lugar de colocarlo al principio de :LISTA, lo colocamos al principio de BF :LISTA.

El razonamiento se deja al lector.

```
PARA REEMPLAZAR :N :ELEMENTO :LISTA
SI :N=1 [DEV (PP :ELEMENTO MP :LISTA)]
DEV PP (PRIMERO :LISTA) REEMPLAZAR :N - 1 :ELEMENTO MP :LISTA
FIN

?ES REEMPLAZAR 3 "X [A B C D]
A B X D
```

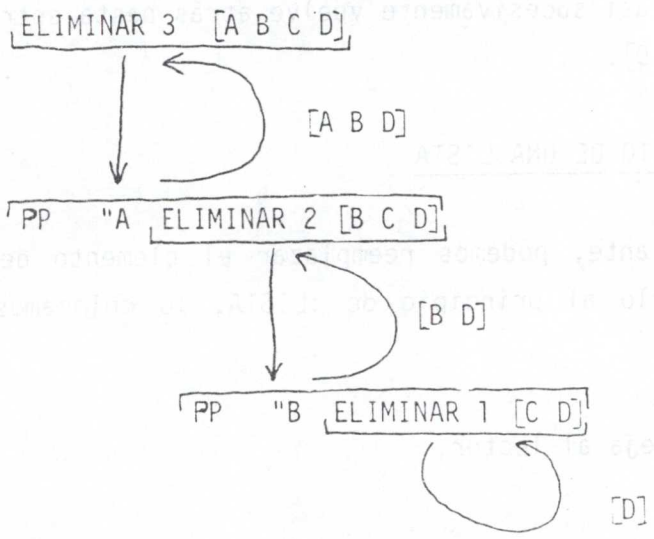
ELIMINACION DE UN ELEMENTO QUE OCUPA UN PUESTO DETERMINADO EN UNA LISTA

Con otra ligera variante respecto a los dos procedimientos anteriores, podemos construir un procedimiento que dé como salida la lista de entrada salvo el elemento que ocupa un número dado.
Por ejemplo: ELIMINAR 3 [A B C D] daría como salida [A B D]

El procedimiento sería:

```
PARA ELIMINAR :N :LISTA
SI :N = 1 [DEV MP :LISTA]
DEV PP (PRIMERO :LISTA) ELIMINAR :N - 1 MP :LISTA
FIN
```

Al ejecutar ELIMINAR 3 [A B C D] se produce el proceso que se esquematiza a continuación.



CODIGO ASCII Y CARACTERES

Los caracteres tienen una codificación en el denominado código ASCII que corresponde a un número. Así, por ejemplo, a las letras mayúsculas y minúsculas les corresponden los siguientes números:

```
A -------- 65          a --------- 97
B -------- 66          b --------- 98
C -------- 67          c --------- 99
:          :           :
Z -------- 90          z ---------122
```

Y a los números del Ø al 9:

```
Ø ------------- 48
1 ------------- 49
2 ------------- 50
3 ------------- 51
.................
9 ------------- 57
```

ASCII

Existe una primitiva (ASCII) que al aplicarse a un caracter da como salida su código ASCII correspondiente.

Ejemplos:

```
? ES ASCII "A
65
?ES ASCII "3
51
```

CA R

Es una primitiva que da como salida el caracter correspondiente al código ASCII.

Ejemplos:

```
? ES CAR 65
A
? ES CAR 97
a
? ES CAR 32 + ASCII "B
b
?ES CAR ASCII "Z
Z
```

Ejemplo: La siguiente función da como salida el caracter de entrada cambiado de mayúscula a minúscula:

```
PARA MINUSCULA :LETRA
HAZ "C 32 + ASCII :LETRA
SI :C = 197 [DEV CAR 164]
SI Y :C > 96 :C < 123 [DEV CAR :C] [DEV :LETRA]
FIN

?ES MINUSCULA "B
b
```

Impresión de listas

Para sacar por el terminal una lista hemos utilizado casi siempre la orden de imprimir (ES), y en algún caso la orden de mostrarMUESTRA. A continuación se muestran todas las posibilidades del lenguaje para dar resultados al exterior.

IMPRIMIR

La orden de imprimir se refiere a un mensaje, en forma de lista, o una palabra entreçomillada o una lista dada como variable.

```
? ES "PALABRA
PALABRA
? ES [HOLA COMO ESTAS]
HOLA COMO ESTAS
? HAZ "LISTA [A B C]
? ES :LISTA
A B C
```

MOSTRAR

La orden MUESTRA imprime un objeto seguido de un retorno, y si el objeto es una lista lo escribe entre corchetes. Es decir, muestra el objeto como es.
Ejemplos:

```
? MUESTRA "AB
AB
? MUESTRA [A B]
[A B]
? MUESTRA PU "DESPACIO [LOS ARBOLES CRECEN]
[LOS ARBOLES CRECEN DESPACIO]
? MUESTRA PU [DESPACIO] [[LOS ARBOLES] [CRECEN]]
[[LOS ARBOLES] [CRECEN] [DESPACIO]]
? MUESTRA PRIMERO [[A B] C]
[A B]
```

ESCRIBIR

La orden TECLEA tiene un efecto similar al de ESCRIBE salvo que escribe a continuación.
Ejemplos:

```
? TECLEA "AA
AA ? TECLEA [A B C] TECLEA "A
A B CA ?
```

Elección de un elemento al azar

Se puede efectuar la elección de un elemento aleatoriamente de una lista, ahora que conocemos la forma de obtener el número de elementos que posee.

```
PARA ELEGIR.AZAR :OBJETO
DEV ELEMENTO 1 + AZAR (CUENTA :OBJETO) :OBJETO
FIN
```

El número de "elemento" a seleccionar lo elige entre 1 y el número de elementos del objeto, ya que 1 + AZAR (CUENTA :OBJETO) da como salida un número al azar entre 1 y CUENTA:OBJETO.

Al ejecutar:

```
? ES ELEGIR.AZAR [COCO PERA FRESA MANZANA]
FRESA
? ES ELEGIR.AZAR [COCO PERA FRESA MANZANA]
PERA
```

Funciones booleanas para manejo de listas y palabras

Existen diversas funciones booleanas (predicados), que pueden tomar dos valores: c i e r t o o f a l s o según se cumpla o no una condición. Estas funciones nos permiten saber si un objeto está vacío, si un elemento es miembro de un objeto, si un objeto es un número, o una lista o una palabra.

OBJETO VACIO

VACIO? es una función booleana que tiene dos posibles valores:CIERTO o FALSO según el objeto esté vacío o no.

Ejemplo:

El siguiente procedimiento imprime el mensaje "LA LISTA ESTA VACIA" o "LA LISTA TIENE ELEMENTOS" dependiendo de la lista de entrada.

```
PARA PRUEBA :LISTA
SI VACIO? :LISTA [ES [LA LISTA ESTA VACIA]] [ES [LA LISTA
TIENE ELEMENTOS]]
```

MIEMBRO

MIEMBRO es una función booleana que tiene dos posibles valores: CIERTO o FALSO, según el elemento dado esté en la lista o no. Su formato es:

MIEMBRO? objeto1 objeto2

Ejemplos:

```
? ES MIEMBRO? "A "PALABRA
CIERTO
? ES MIEMBRO? [A] [A B C]
FALSO
```

El procedimiento BUSCAR imprime un mensaje que indica si un elemento pertenece o no a un conjunto.

```
PARA BUSCAR :ELEMENTO :CONJUNTO
SI MIEMBRO? :ELEMENTO :CONJUNTO [ES [SI ESTA]] [ES [NO ESTA]]
FIN
```

```
? BUSCAR "A [A B]
SI ESTA
? BUSCAR [A] [A B]
NO ESTA
? BUSCAR "A [[A] B]
NO ESTA
```

Número, lista, palabra, iguales

Para averiguar si un determinado objeto es un número, se utiliza NUMERO? que da salida CIERTO o FALSO según se aplique a un número o no. Para saber si es una lista se utiliza LISTA y para ver si es una palabra PALABRA?

Si dos objetos son iguales se utiliza IGUALES?

Obsérvese que las primitivas que son predicados terminan en ?

Ejemplos:

```
? ES NUMERO? [1 2 3 4]
FALSO
? ES LISTA? []
CIERTO
```

```
? ES PALABRA? [PEPE]
FALSO
? ES NUMERO? PRIMERO [1 2 3 4]
CIERTO
? ES PALABRA? "JUAN
CIERTO
? ES LISTA? "LISTA
FALSO
? ES IGUALES? [A] "A
FALSO
```

Construcción de listas y palabras a partir de sus elementos

Existen funciones que permiten formar listas (LISTA FRASE)a partir de elementos y otras que permiten caracterizar palabras (PALABRA)

LISTA

Da como salida una lista cuyos elementos son los dos objetos.
Ejemplo:

```
? MUESTRA LISTA [A B C] [D E F]
[[A B C] [D E F]]
? MUESTRA LISTA [[A B] C] [D [E F]]
[[[A B] C] [D [E F]]]
? MUESTRA LISTA "FLOR "HIERBA
[FLOR HIERBA]
? MUESTRA LISTA "A [B C]
[A [B C]]
```

PALABRA

Da como salida una palabra formada por las dos palabras de entrada.
Ejemplo:

```
? ES PALABRA "JOSE "JUAN
JOSEJUAN
? ES PALABRA PRIMERO [JOSE JUAN] "ANTONIO
JOSEANTONIO
```

FRASE

Abreviado FR, da como salida una lista formada por la serie de sus entradas si son palabras, o la serie de los elementos de sus entradas si son listas.
Ejemplo:

```
? MUESTRA FR "A "B
[A B]
? MUESTRA (FR "ANTONIO "ELISA "JUAN)
[ANTONIO ELISA JUAN]
? MUESTRA FR [A B C] [D E]
[A B C D E]
? MUESTRA (FR [A B C] [D E F] [G H I])
[A B C D E F G H I]
```

Como se puede observar, cuando los elementos de entrada para construir la lista son más de dos, se escribe la función y los elementos entre paréntesis.

La diferencia entre LISTA y FR, se manifiesta en el caso de que los objetos de entrada sean listas, ya que LISTA forma lista de listas y FR produce una lista única eliminando los corchetes intermedios.
Ejemplo:

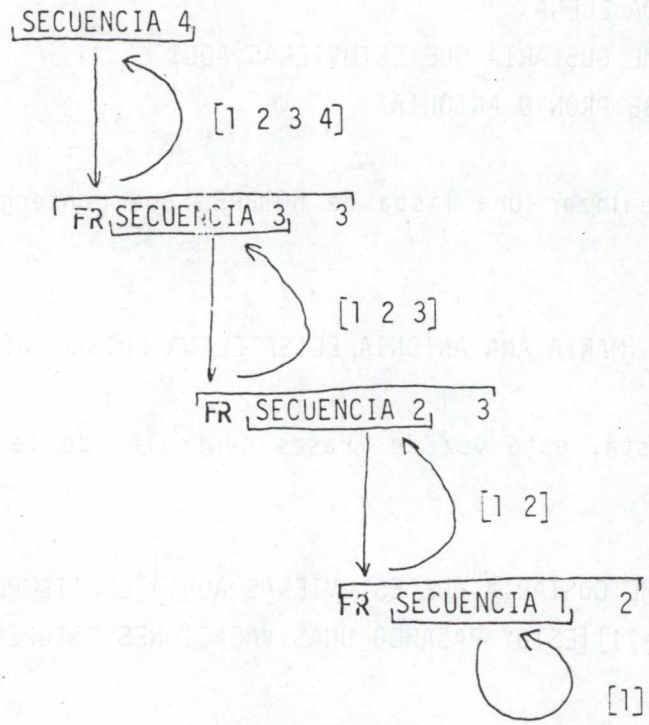
```
? ES LISTA [A B] [C D]
[A B] [C D]
? ES FR [A B] [C D]
A B C D
```

Ejemplo: Construcción de una lista formada por los N primeros números naturales.
Utilizaremos un procedimiento recursivo denominado SECUENCIA cuya entrada es el número de elementos de la lista (N) y se llama a sí mismo con N-1, hasta que N=1, que es la salida del bucle recursivo, e inicia el retorno.

```
PARA SECUENCIA :N
DEV SI :N = 1 [[1]] [FR SECUENCIA :N-1 :N]
FIN
```

```
? MUESTRA SECUENCIA 4
[1 2 3 4]
```

Veamos con ayuda de un esquema el proceso seguido:



Si en lugar de FRASE hubiéramos utilizado LISTA, se hubiera obtenido [[[[1] 2] 3] 4].

El razonamiento se deja al lector.

Aplicación: Postales aleatorias

Antes de continuar con el desarrollo del capítulo, realizaremos una aplicación que puede ser punto de partida para desarrollar aplicaciones de escritura aleatoria.

Realizaremos un programa que escribe postales "aleatorias", como las que se escriben a continuación.

QUERIDA ANA
ME ACUERDO MUCHO DE TI
SALUDOS MARIA

QUERIDA ELENA
COMO ME GUSTARIA QUE ESTUVIERAS AQUI
ESCRIBE PRONTO ANTONIA

Para ello debemos realizar una lista de NOMBRES que contenga los nombres de las personas.

```
HAZ  "NOMBRES [MARIA ANA ANTONIA ELISA ELENA LUISA INMACULADA]
```

Realizaremos otra lista, esta vez de frases centrales de la tarjeta, a la que llamaremos FRASES.

```
HAZ "FRASES [[COMO ME GUSTARIA QUE ESTUVIERAS AQUI] [EL TIEMPO ES ESTUPENDO]
[ME ACUERDO MUCHO DE TI] [ESTOY PASANDO UNAS VACACIONES ESTUPENDAS]]
```

Haremos igualmente una lista de despedidas estandarizadas en otra lista que llamaremos DESPEDIDAS.

```
HAZ "DESPEDIDAS [[SALUDOS] [ESCRIBE PRONTO] [HASTA LA PROXIMA] [MUCHOS BESOS]
[RECUERDOS A TODOS]]
```

El procedimiento TARJETA sería:

```
PARA TARJETA
ES LISTA "QUERIDA NOMBRE
ES CUERPO
TECLEA DESPEDIDA
ES FRASE ". NOMBRE
ES []
TARJETA
FIN
```

El procedimiento se define en función de sí mismo para que se repite indefinidamente. Viene definido en función de otros procedimientos, como NOMBRE CUERPO y DESPEDIDA, que son simplemente procedimientos que eligen al azar

un elemento de la lista correspondiente. Así, sería:

```
PARA NOMBRE
DEV ELEGIR.AZAR :NOMBRES
FIN

PARA CUERPO
DEV ELEGIR.AZAR :FRASES
FIN

PARA DESPEDIDA
DEV ELEGIRAZAR :DESPEDIDAS
FIN
```

En todos ellos, el procedimiento ELEGIRAZAR es el que ya se definió anteriormente.

Al ejecutar se obtienen los siguientes textos:

```
?TARJETA
QUERIDA ANA
ME ACUERDO MUCHO DE TI
ESCRIBE PRONTO LUISA

QUERIDA MARIA
ESTOY PASANDO UNAS VACACIONES ESTUPENDAS
MUCHOS BESOS ELENA

.
.
.
```

P R A C T I C A S (Palabras y Listas)

0) Trabajando de modo inmediato, crear una lista con los días de la semana.

A partir de la lista anterior y con las primitivas oportunas, crear otras dos listas, una de "días laborables" y otra "fin de semana"

Con las dos anteriores, crear una nueva lista "días de la semana"

1) Trabajando en modo inmediato, crear tres listas con varios elementos cada una,

a) Manejar las órdenes básicas **PRIMERO,MP,ULTIMO,MU** para extraer elementos de ellas, crear nuevas listas, etc.

b) Utilizando las órdenes de ampliación de listas PP , PU , FR y LISTArealizar nuevas listas, a partir de las anteriores.

2) Definir un procedimiento que duplique o repita una palabra. Idem con una lista

Ejemplo:

```
?ES DOBLAR "PALABRA
PALABRAPALABRA
```

3) Realizar un procedimiento que "deletree" una palabra.
Idem con una lista separando cada objeto de ella.

4) Realizar un procedimiento para escribir frases aleatorias sintácticamente válidas. Para ello, se introducirá en distintas listas NOMBRE, ADJETIVOS, ARTICULOS, ADVERBIOS, etc.

5) Realizar un procedimiento (VECES) que dé como salida el número de veces que aparece un elemento en una lista.

Ejemplo:

```
?ES VECES "A [A  B  A  C]
2
```

6) Realizar un procedimiento que dé como salida el lugar que ocupa un elemento en una lista.

Ejemplo:

```
?ES LUGAR  "C  [A  B  C  D]
3
```

SOLUCIONES PRACTICAS (Palabras y listas) (I)

2)

```
PARA DOBLAR :PAL
DEVUELVE PALABRA :PAL :PAL
FIN

PARA DOBLAR1 :LIS
DEVUELVE FRASE :LIS :LIS
FIN
```

3)

```
PARA DELETREAR :PAL
SI VACIO? :PAL [ALTO]
ES PRIMERO :PAL
DELETREAR MP :PAL
FIN
```

5)

Número de veces que aparece un elemento en una lista.

Cuando en una lista hay elementos repetidos, interesa disponer de un procedimiento que averigüe el número de veces que aparece un elemento dado en una lista.

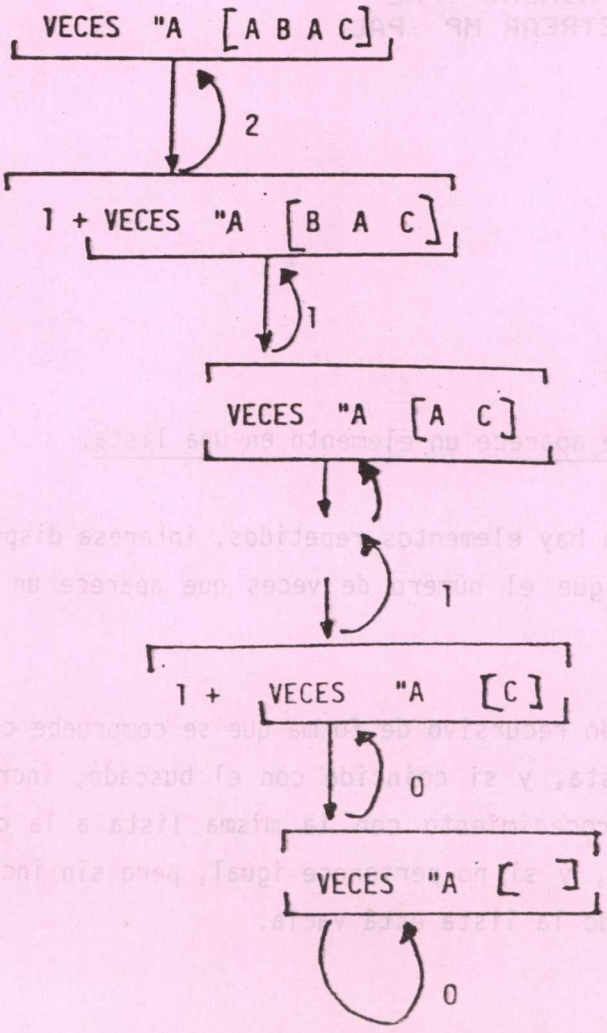
Seguiremos un método recursivo de forma que se compruebe cada vez el primer elemento de la lista, y si coincide con el buscado, incrementar en uno y llamar al mismo procedimiento con la misma lista a la que se ha quitado el primer elemento, y si no pertenece igual, pero sin incrementar. El proceso finaliza cuando la lista está vacía.

El procedimiento será por tanto:

```
PARA VECES :ELEMENTO :LISTA
SI :LISTA = [] [DEV 0]
SI :ELEMENTO = PRIMERO :LISTA [DEV 1 + VECES :ELEMENTO MF
   :LISTA]
DEV :VECES :ELEMENTO MF :LISTA
FIN

? ES VECES "A [A B A C]
2
```

El esquema del proceso de cálculo al ejecutarse el procedimiento se muestra a continuación.



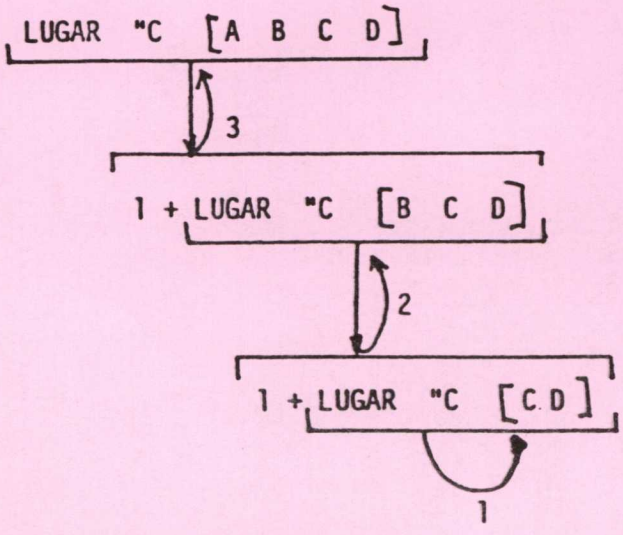
Lugar que ocupa un elemento en una lista

Es la función opuesta a ELEMENTO. Sus entradas son un elemento y una lista, y su salida es el número que ocupa en dicha lista.
La definición se efectúa de forma recursiva: Si ELEMENTO es el primero de LISTA, da salida 1, y si no da como salida 1 + el propio procedimiento pero con la lista disminuída en su primer elemento (MP : LISTA)
La salida del procedimiento puede ser porque la lista esté vacía, en cuyo caso indica que no pertenece a la lista, o bien si es miembro, se detiene el proceso al comprobar que ELEMENTO coincide con el primer elemento.
El procedimiento es el siguiente:

```
PARA LUGAR :ELEMENTO :LISTA
SI VACIO? :LISTA [DEV 1]
SI :ELEMENTO = PRIMERO :LISTA [DEV 1]
DEV 1 + LUGAR :ELEMENTO MP :LISTA
FIN
```

```
?ES LUGAR "C [A B C D]
3
```

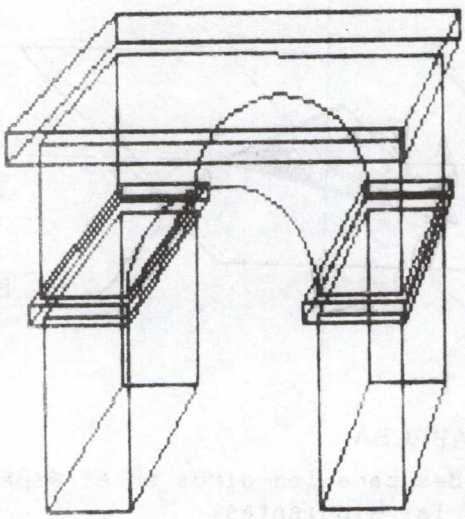
El flujo de datos de salida entre procedimientos se muestra en el esquema siguiente:



Si por el contrario no fuera elemento, seguiría el proceso hasta que la lista estuviera vacía, en cuyo caso siempre da como salida un número superior en una unidad al tamaño de la lista. Por tanto este procedimiento se aplicará cuando el ELEMENTO pertenezca a la lista, lo que se puede comprobar con MIEMBRO? o bien, una vez recibido el dato el procedimiento que lo llama verificando si es mayor que el número de elementos de la lista (CUENTA : LISTA)

LA TORTUGA TRIDIMENSIONAL

Se pueden realizar dibujos en perspectiva sobre la pantalla gracias a la implementación en LOGO de primitivas que permiten manejar a la tortuga en el espacio. Para realizar ésto no se necesitan conocimientos avanzados de LOGO ni es necesario recurrir a las reglas de la perspectiva.



Conocemos a la tortuga que se mueve sobre un plano definido por las coordenadas X e Y. La tortuga tridimensional necesita como referencia los ejes de coordenadas espaciales X, Y, Z.

Para poder trabajar en el espacio tridimensional es necesario utilizar la primitiva ESPACIO que se contrapone a PLANO, la cual permite regresar al espacio bidimensional.

En el espacio tridimensional solamente se puede usar una tortuga y ésta no cambia de forma con los giros.

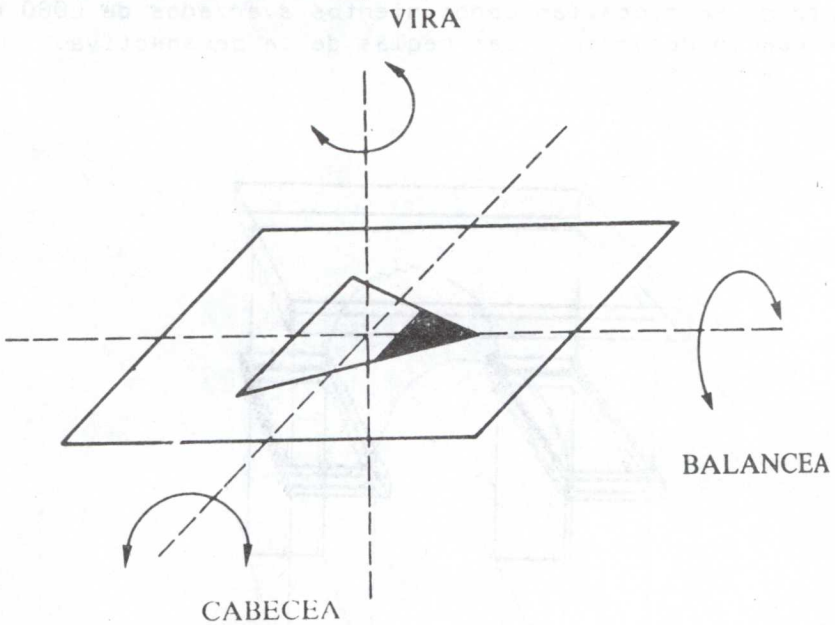
La tortuga tridimensional tiene en cada momento una posición y una orientación, que podemos modificar. La posición se cambia mediante las primitivas ANDA y DESANDA:

ANDA número-de-pasos (abreviado AN)
hace avanzar a la tortuga el "número-de-pasos" dado en un espacio de tres dimensiones.

DESANDA número-de-pasos (abreviado DA)
hace retroceder a la tortuga el "número-de-pasos" dado en un espacio tridimensional.

GIROS EN EL ESPACIO

Al moverse en el espacio existe la posibilidad de que la tortuga realice tres giros, uno alrededor de cada eje.



Las primitivas usadas para los giros en el espacio tridimensional son las siguientes:

VIRADERECHA número-de-grados (abreviado VD)
la tortuga gira hacia la derecha el "número-de-grados" dado, sobre su propio eje.

VIRAIZQUIERDA número-de-grados (abreviado VI)
la tortuga gira hacia la izquierda el "número-de-grados" dado, sobre su propio eje.

CABECEARRIBA número-de-grados (abreviado CA)
la tortuga gira, inclinando su eje longitudinal hacia arriba, el "número-de-grados" dado.

CABECEABAJO número-de-grados (abreviado CB)
la tortuga gira, inclinando su eje longitudinal hacia abajo, el "número-de-grados" dado.

BALANCEADERECHA número-de-grados (abreviado BD)
la tortuga gira, inclinando su eje transversal hacia la derecha, el "número-de-grados" dado.

BALANCEAIZQUIERDA número-de-grados (abreviado BI)
la tortuga gira, inclinando su eje transversal hacia la izquierda, el "número-de-grados" dado.

PUN DE MIRA Y POSICION DE LA TORTUGA

El punto de mira de un observador puede variar y entonces la representación tridimensional de un objeto cambia en función de aquel. Para definir la posición del observador tenemos la primitiva:

PUNTODEMIRA lista-de-coordenadas (abreviado pm)
los dos primeros valores definen las coordenadas X e Y y el tercer valor define la coordenada Z es decir la distancia del observador a la pantalla.

En el espacio tridimensional existe la posibilidad de desplazar la tortuga a un punto cualquiera determinado por las coordenadas X, Y, Z:

POSICIONATE lista-de-coordenadas (abreviado PO)
situa a la tortuga, sin variar su orientación, en la posición dada por "lista-de-coordenadas".

PRIMITIVAS DE USO COMUN EN EL PLANO Y EN EL ESPACIO

Al trabajar en el espacio no disponemos de todas las primitivas de gràficos que existen en el plano. Si usamos alguna de las no admitidas nos darà un mensaje de error.

Las primitivas que podemos usar en el plano y en el espacio indistintamente son:

BAJALAPIZ
BORRAPANTALLA
COLORLAPIZ
FONDO
GOMA
INVIERTELAPIZ
LAPIZ
LIMPIA
MUESTRATORTUGA
OCULTATORTUGA
PALETA
PONCOLORLAPIZ
PONFONDO
PONLAPIZ
PONPALETA
SUBELAPIZ
VISIBLE?

EJEMPLOS DE PROCEDIMIENTOS EN EL ESPACIO

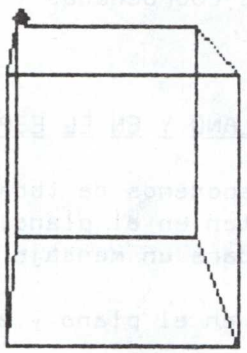
1) Vamos a realizar en primer lugar un procedimiento que dibuje un CUBO en el espacio a partir del procedimiento CUADRADO.

```
PARA CUADRADO :LADO
REPITE 4 [ANDA :LADO VIRADERECHA 90]
FIN

PARA CUBO :LADO
REPITE 4 [CUADRADO :LADO ANDA :LADO CABECEARRIBA 90]
FIN
```

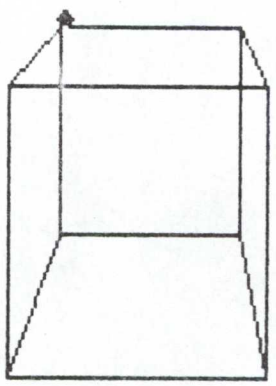
Ejecutamos

```
? CUBO 60
```

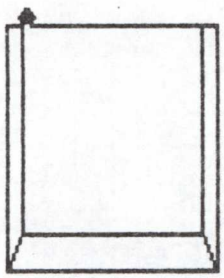


Cambiamos la posición del observador:

```
? PUNTODEMIRA 40 40 200
? CUBO 60
```



```
? PM 30 0 400
? CUBO 60
```



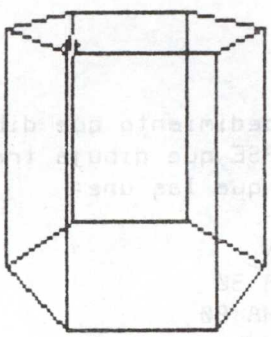
2) Ahora vamos a realizar el procedimiento PRISMA que nos permitirá dibujar prismas de :N caras a partir de un procedimiento RECTANGULO que se repite despues de la orden CABECEARRIBA o CABECEABAJO.

```
PARA RECTANGULO :L1 :L2
REPITE 2 [ANDA :L1 VIRADERECHA 90 ANDA :L2
     VIRADERECHA 90]
FIN

PARA PRISMA :N :L1 :L2
REPITE :N [RECTANGULO :L1 :L2 ANDA :L1
     CABECEABAJO 360/:N]
FIN
```

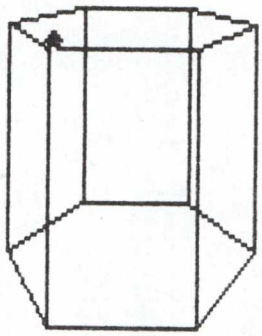
Ejecutamos el procedimiento asi:

```
? PRISMA 6 50 80
```

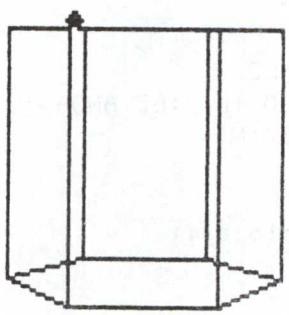


Podemos cambiar la posición del observador:

```
? PM 40 40 200
? PRISMA 6 50 80
```



```
? PM 30 0 400
? PRISMA 6 50 80
```



3) Vamos a realizar un procedimiento que dibuje una SILLA a partir del procedimiento BASE que dibuja tres lineas, dos patas y el eje transversal que las une:

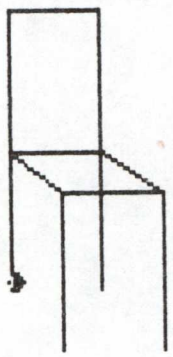
```
PARA BASE :PATA :UNION
ANDA :PATA VIRADERECHA 90
ANDA :UNION VIRADERECHA 90
ANDA :PATA VIRADERECHA 90
FIN
```

```
PARA PATAS :PATA :UNION
VIRAIZQUIERDA 90 REPITE 4 [BASE :PATA :UNION
     BALANCEADERECHA 90]
FIN

PARA SILLA :P :U
PATAS :P :U
ANDA :P
BASE :P :U
VIRAIZQUIERDA 90
ANDA :U
VIRAIZQUIERDA 90
ANDA :P
VIRAIZQUIERDA 90
FIN
```

EJEMPLO

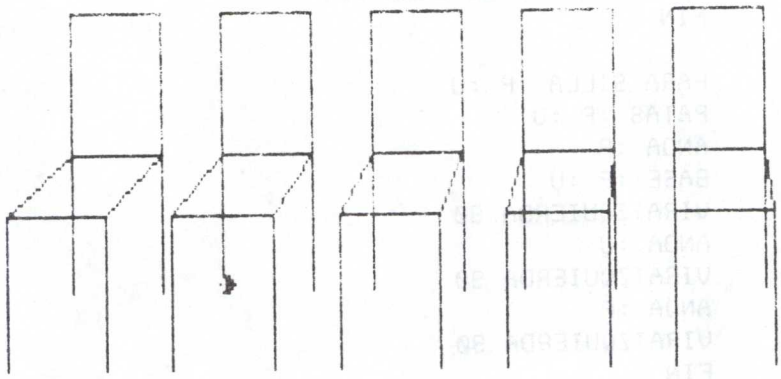
```
? pm -125 125 250
? SILLA 40 30
```



Si queremos dibujar una fila de sillas podemos teclear las siguientes ordenes

```
? BP PM 100 200 300
? SILLA 40 30
? SL PO 50 0 0 BL
? SILLA 40 30
? SL PO 100 0 0 BL
? SILLA 40 30
? SL PO -50 0 0 BL
? SILLA 40 30
? SL PO -100 0 0 BL
? SILLA 40 30
```

El dibujo que obtenemos es:



Observa que cambiando el "PUNTODEMIRA" obtenemos perspectivas totalmente diferentes del dibujo.

ANEXO

PROCEDIMIENTOS QUE PERMITEN EL USO DE LA TORTUGA 3-D

Existen algunas versiones del lenguaje LOGO que no tienen implementadas las primitivas que posibilitan el manejo de la tortuga tridimensional. Podemos, de todos modos, crear los procedimientos que nos permiten realizar los distintos movimientos y giros:

ESPACIO

El procedimiento ESPACIO sera el que nos permite comenzar a actuar en tres dimensiones y al mismo tiempo nos proporcionará unos valores iniciales que determinan las posiciones del observador y de la tortuga y su orientación en el espacio.

```
PARA ESPACIO
HAZ "OX 0 HAZ "OY 1 HAZ "OZ 0
HAZ "DX 1 HAZ "DY 0 HAZ "DZ 0
HAZ "PX 0 HAZ "PY 0 HAZ "PZ 1
HAZ "X 0 HAZ "Y 0 HAZ "Z 0
PUNTODEMIRA 0 0 600
FIN
```

PUNTODEMIRA

El procedimiento PUNTODEMIRA situa al observador en el punto dado por sus tres coordenadas.

```
PARA PUNTODEMIRA
HAZ "OBX :A
HAZ "OBY :B
HAZ "DIST :C
SL POSICIONATE :X :Y :Z BL
FIN
```

ANDA

El procedimiento ANDA hace avanzar a la tortuga en el espacio. Si la entrada es un número negativo el efecto producido es el mismo que el que se consigue en algunas versiones de LOGO con la primitiva DESANDA.

```
PARA ANDA :DISTANCIA
HAZ "X :X + :DISTANCIA * :OX
HAZ "Y :Y + :DISTANCIA * :OY
HAZ "Z :Z + :DISTANCIA * :OZ
POSICIONATE :X :Y :Z
FIN
```

POSICIONATE

El procedimiento POSICIONATE situa a la tortuga en la posición dada por los valores de las tres coordenadas que requiere como entrada.

```
PARA POSICIONATE
LOCAL "ESC
HAZ "ESC :DIST/(:DIST - :Z)
PONPOS LISTA (:OBX+(:X - :OBX) * :ESC)
     (:OBY+(:Y - :OBY) * :ESC)
FIN
```

PROCEDIMIENTOS PARA DEFINIR GIROS ESPACIALES

Para que la tortuga gire en el espacio necesitamos definir tres procedimientos básicos, que denominaremos VIRA, CABECEA y BALANCEA. Estos procedimientos necesitan una entrada que será el número de grados a girar.

Algunas versiones de LOGO tienen primitivas diferentes según la dirección que tomen los giros. En nuestro caso la dirección vendrá determinada por el signo, positivo o negativo, de la entrada.

```
PARA VIRA :ANGULO
(LOCAL "C "S "TX "TY "TZ)
HAZ "C COS :ANGULO HAZ "S SEN :ANGULO
HAZ "TX :C * :OX + :S * :DX
HAZ "TY :C * :OY + :S * :DY
HAZ "TZ :C * :OZ + :S * :DZ
HAZ "DX :C * :DX - :S * :OX
HAZ "DY :C * :DY - :S * :OY
HAZ "DZ :C * :DZ - :S * :OZ
HAZ "OX :TX HAZ "OY :TY HAZ "OZ :TZ
FIN
```

```
PARA CABECEA :ANGULO
(LOCAL "C "S "TX "TY "TZ)
HAZ "C COS :ANGULO HAZ "S SEN :ANGULO
HAZ "TX :C * :OX - :S * :PX
HAZ "TY :C * :OY - :S * :PY
HAZ "TZ :C * :OZ - :S * :PZ
HAZ "PX :C * :PX + :S * :OX
HAZ "PY :C * :PY + :S * :OY
HAZ "PZ :C * :PZ + :S * :OZ
HAZ "OX :TX HAZ "OY :TY HAZ "OZ :TZ
FIN
```

```
PARA BALANCEA :ANGULO
(LOCAL "C "S "TX "TY "TZ)
HAZ "C COS :ANGULO HAZ "S SEN :ANGULO
HAZ "TX :C * :DX + :S * :PX
HAZ "TY :C * :DY + :S * :PY
HAZ "TZ :C * :DZ + :S * :PZ
HAZ "PX :C * :PX - :S * :DX
HAZ "PY :C * :PY - :S * :DY
HAZ "PZ :C * :PZ - :S * :DZ
HAZ "DX :TX HAZ "DY :TY HAZ "DZ :TZ
FIN
```

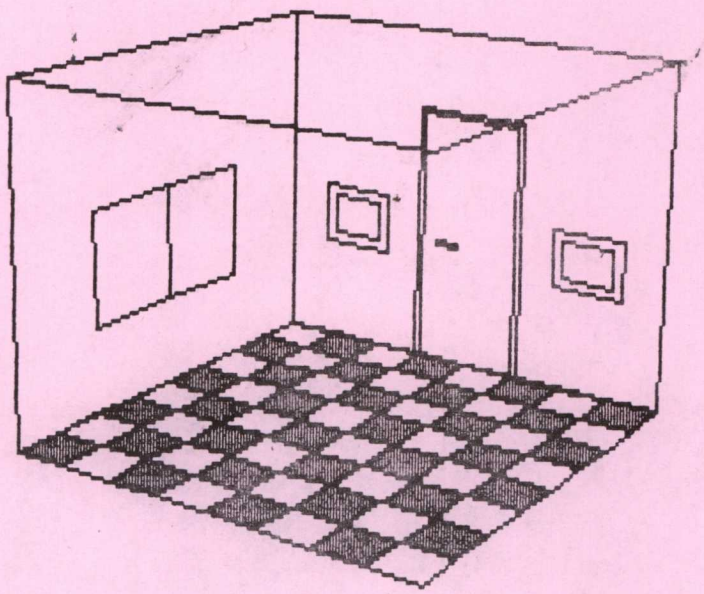
PRACTICAS DE TORTUGA TRIDIMENSIONAL

1) Realiza un procedimiento que dibuje un campo de fútbol y ejecútalo de la manera en que sería visto desde la banda, desde las porterías, desde distintos puntos de las gradas y desde el centro del campo.

2) A partir del procedimiento CUBO anteriormente definido debes realizar otro que dibuje edificios de un numero variable de pisos en el que cada piso este representado por un cubo.

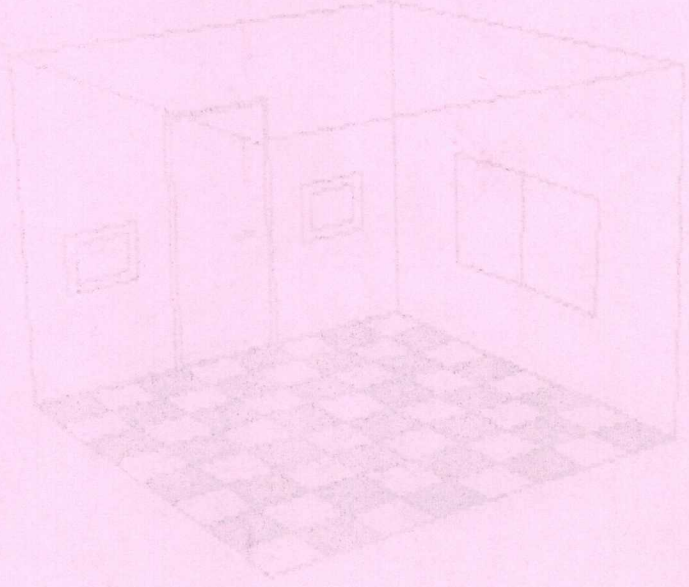
3) Realiza un procedimiento que dibuje cilindros de manera similar al procedimiento PRISMA.

4) Realiza un procedimiento que dibuje una habitación en forma de cubo y que en una de sus paredes tenga un cuadro, en otra una ventana y que el suelo sea de baldosas cuadradas.



5) A partir del procedimiento SILLA realiza otro que permita dibujar filas de diferente numero de sillas y vistas desde posiciones diversas.

6) Realiza un procedimiento que dibuje un vagón de un tren para ver luego la unión de diversos vagones desde distintos puntos de vista.



EDITOR DE FORMAS EN LOGO

Para cambiar la forma de la tortuga en LOGO nos podemos encontrar en dos situaciones distintas según si el LOGO con el que estamos trabajando tenga o no editor de formas. De todas maneras, hay que tener en cuenta que casi ninguno puede hacer girar las figuras editadas, cuestión a tener en cuenta cuando se trabaje con formas no estándar.

LOGO CON EDITOR DE FORMAS

Si el logo con el que estamos trabajando posee un editor de formas, la primitiva que nos introduce en el editor de formas es EDFORMA "<nombre de la forma>.

La primitiva FORMA nos indica cuál es la forma actual de la tortuga; antes de cambiar de forma es conveniente que sepamos cuál es el nombre de la tortuga estándar, por medio de:

```
? ES FORMA
```

La forma de la tortuga estándar, normalmente, es un número determinado o la cadena vacía; así pues, si queremos volver a la forma normal, teclearemos:

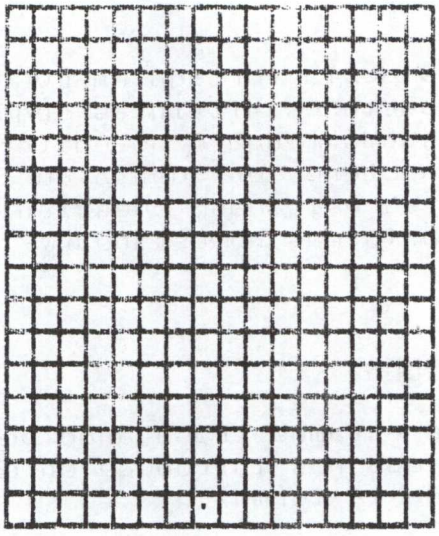
```
? PONFORMA "
```

Ya estamos en disposición de cambiar la forma de la tortuga, supongamos que queremos que nuestra tortuga se parezca a un coche. En primer lugar, buscamos un nombre para nuestra forma, por ejemplo COCHE, tecleando:

```
? EDFORMA "COCHE
```

En pantalla aparecerá una cuadrícula de 16 x 16 cuadraditos que representan un pixel de pantalla cada uno. En uno de ellos aparecerá un cuadradito más pequeño que indica la

posicion del cursor:



El sistema para crear la forma nueva consiste en colocarse en el cuadrado que queremos rellenar y presionar la tecla R (rellena), el cuadro tomará el color activo.

Al mismo tiempo podemos ver en la parte derecha de la pantalla la forma que va tomando la tortuga al tamaño con que aparecerá posteriormente.

El cuadradito, que nos indica el cuadrado activo, se mueve con las teclas de cursor; seleccionándose el color pulsando los números 0, 1, 2 ó 3, correspondiendo el 0 al color del fondo (se puede usar para borrar si nos equivocamos) y el resto de los números a los colores de la paleta activa. Cuando pulsamos un número, todos los cuadrados sobre los que pulsemos la tecla R se rellenarán del color asociado a ese número mientras no sea pulsado otro.

La salida del editor de formas se realiza del mismo modo que del editor de procedimientos: F1, si queremos que quede fijada la forma, o [ALT]+F2, si nos queremos salir sin actualizar la forma.

Una vez que la forma está creada, si queremos que aparezca en pantalla tecleamos PONFORMA "<nombre de la forma> y nuestra

tortuga tomará la forma asociada a ese nombre.

A continuación se va a ver un pequeño juego que nos permitirá explicar sobre la marcha algunos detalles de funcionamiento. El juego consiste en dibujar una casa que simula ser la "CASA DE LAS FRACCIONES" donde se meterá un número que por extrañas razones, saldrá transformado en otro distinto que se colocará en un extremo de la pantalla; esta operación se repetirá cinco veces y luego se preguntará al alumno cuál es la salida del sexto número.

Los procedimientos necesarios son:

```
PARA FRACCIONES
DIBUJA.CASA
ELIGE.FRACCION
DA.EJEMPLOS 5
PREGUNTA.AL.ALUMNO
FIN
```

```
PARA DIBUJA.CASA
BP OT
CASA
FIN
```

```
PARA CASA
HAZ "LADO1 50
HAZ "LADO2 100
SL
PONPOS [-50 -25]
BL
RECTANGULO :LADO1 :LADO2
AV :LADO1
PONPOS [0 50]
PONPOS [50 25]
SL
PONPOS [-30 -25]
BL
RECTANGULO 20 10
FIN
```

```
PARA RECTANGULO :LADO1 :LA
    DO2
REPITE 2 [AV :LADO1 GD 90
    AV :LADO2 GD 90]
FIN
```

```
PARA ELIGE.FRACCION
HAZ "NUMERADOR 1 + AZAR 6
HAZ "DENOMINADOR 1 + AZAR 6
FIN
```

```
PARA DA.EJEMPLOS :CUANTOS
HAZ "NUM.ELEGIDOS []
EJEMPLOS :CUANTOS
FIN

PARA EJEMPLOS :N
SI :N = 0 [ALTO]
LOCAL "ELEGIDO
HAZ "ELEGIDO NUM.VALIDO
METE :ELEGIDO
SACA TRANSFORMADO :ELEGIDO
    :ELEGIDO :N
EJEMPLOS :N - 1
FIN
```

```
PARA PON :N
HAZ "NUM.ELEGIDOS PP :N
    :NUM.ELEGIDOS
FIN


PARA METE :NUM.ELEGIDO
OT SL PONPOS [-125 -16]
PONFORMA "COCHE
PONRUMBO 90
MT
AVANZA.NUMERO :NUM.ELEGIDO
    [0 11] 80
PONCURSOR [11 11]
REPITE 2 [TECLEA CAR 32]
FIN


PARA AVANZA.NUMERO :NUM.
    ELEGIDO :INICIAL :LO
    NGITUD
SI :LONGITUD < 1 [ALTO]
PONCURSOR :INICIAL
TECLEA CAR 32
TECLEA :NUM.ELEGIDO
AV 4
HAZ "INICIAL PP ((PR :INI
    CIAL)+0.5) MP :INICI
    AL
AVANZA.NUMERO :NUM.ELEGID
    O :INICIAL :LONGITUD
    - 4
FIN


PARA SACA :NUMERO1 :NUMERO2
    :ORDEN
SACAR.NUMERO :NUMERO1
COLOCAR.NUMERO :NUMERO2 :NU
    MERO1 :ORDEN
FIN
```

```
PARA NUM.VALIDO
LOCAL "N
HAZ "N AZAR 26
SI Y (NO MIEMBRO? :N "NUM.
    ELEGIDOS) (ENTERO?
    TRANSFORMADO :N) [PON
    :N DEV :N]
DEV NUM.VALIDO
FIN
PARA ENTERO? :N
DEV IGUALES? ENTERO :N :N
FIN


PARA TRANSFORMADO :N
DEV :N*:NUMERADOR/DENOMINA
    DOR
FIN


PARA COLOCAR.NUMERO :NUME
    RO1 :NUMERO2 :ORDEN
BAJAR.GRUA
ATRAPAR.NUMERO
SUBIR.GRUA :ORDEN
GI 90 AV 20 OT
ACTIVA [0]
PONER.NUMERO :NUMERO1 :NU
    MERO2 :ORDEN
FIN


PARA SACAR.NUMERO :SALIDA
OT SL
PONPOS [60 -16]
PONRUMBO 90
MT
AVANZA.NUMERO :SALIDA [23
    11] 70
FIN
```

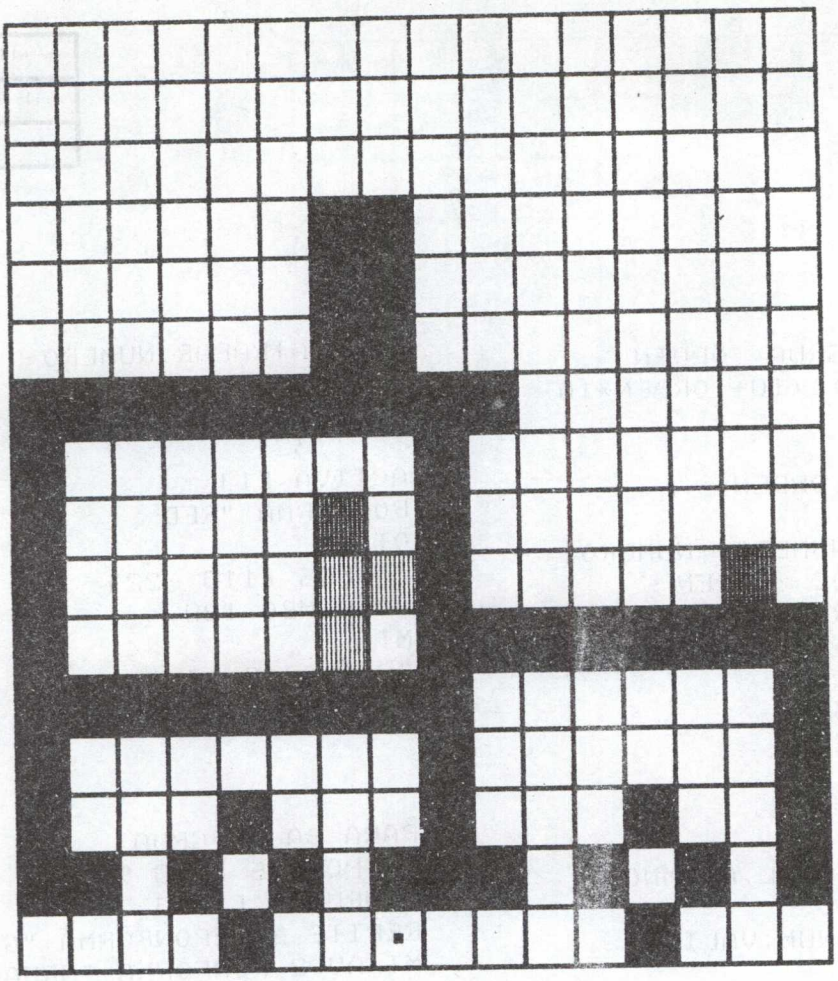
```
PARA SUBIR.GRUA :ORDEN
SI (COORY) > (10+:ORDEN*10)
     [ALTO]
RE 4
SUBIR.GRUA :ORDEN
FIN
PARA PONER.NUMERO :NUMERO1
     :NUMERO2 :ORDEN
PONCURSOR FRASE "34 (5-:OR
     DEN)
TECLEA :NUMERO1
TECLEA CAR 32
TECLEA :NUMERO2
FIN

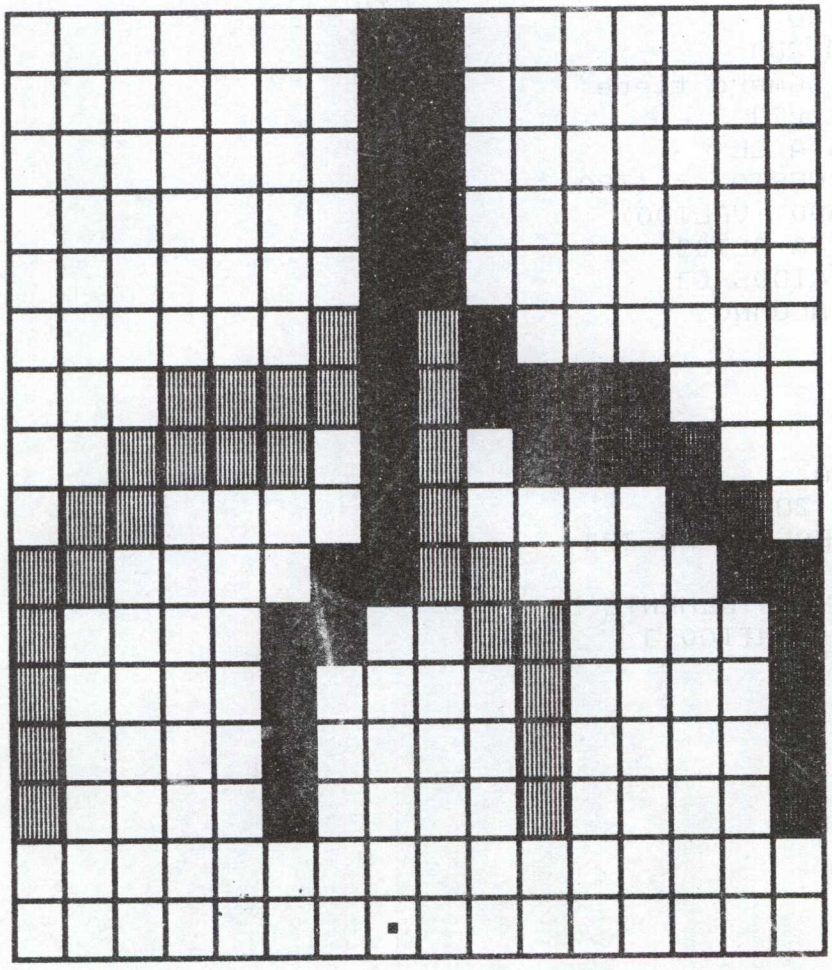

PARA PREGUNTA.AL.ALUMNO
LOCAL "VALIDO
HAZ "VALIDO NUM.VALIDO
METE "VALIDO
PONCURSOR [23 11]
TECLEA :VALIDO
PONCURSOR [0 20]
TECLEA [Qué número tiene
     que salir?]
HAZ "RESPUESTA LL
SI (PR :RESPUESTA) = (TRA
     NSFORMADO :VALIDO)
     [FELICITA ALTO]
HAZ "NUM.ELEGIDOS []
PREGUNTA.AL.ALUMNO
FIN

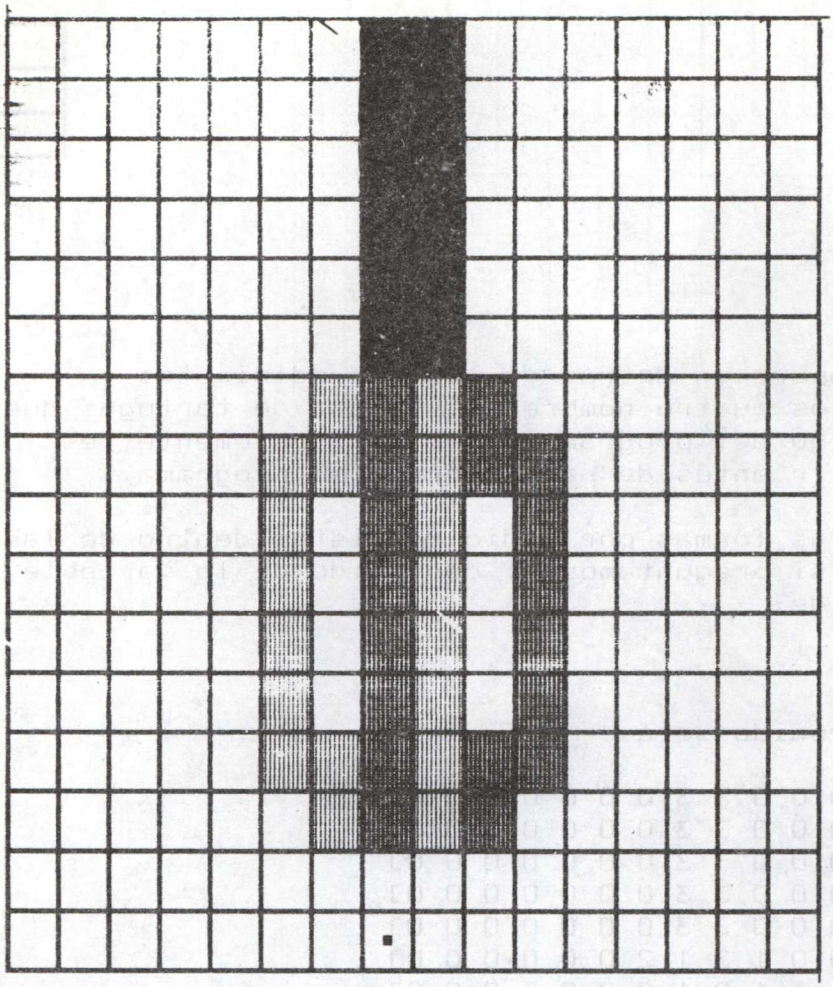

PARA FELICITA
PONCURSOR [0 20]
REPITE 39 [TECLEA CAR 32]
PONCURSOR [0 20]
TECLEA [ESTOY TOTALMENTE DE
     ACUERDO CONTIGO!]
ES []
FIN
```

```
PARA ATRAPAR.NUMERO
PONCURSOR [33 11]
REPITE 4 [TECLEA CAR
     32]
ACTIVA [1]
PONFORMA "RED
OT SL
PONPOS [110 -22]
PONRUMBO 180
MT
ACTIVA [0 1]
FIN

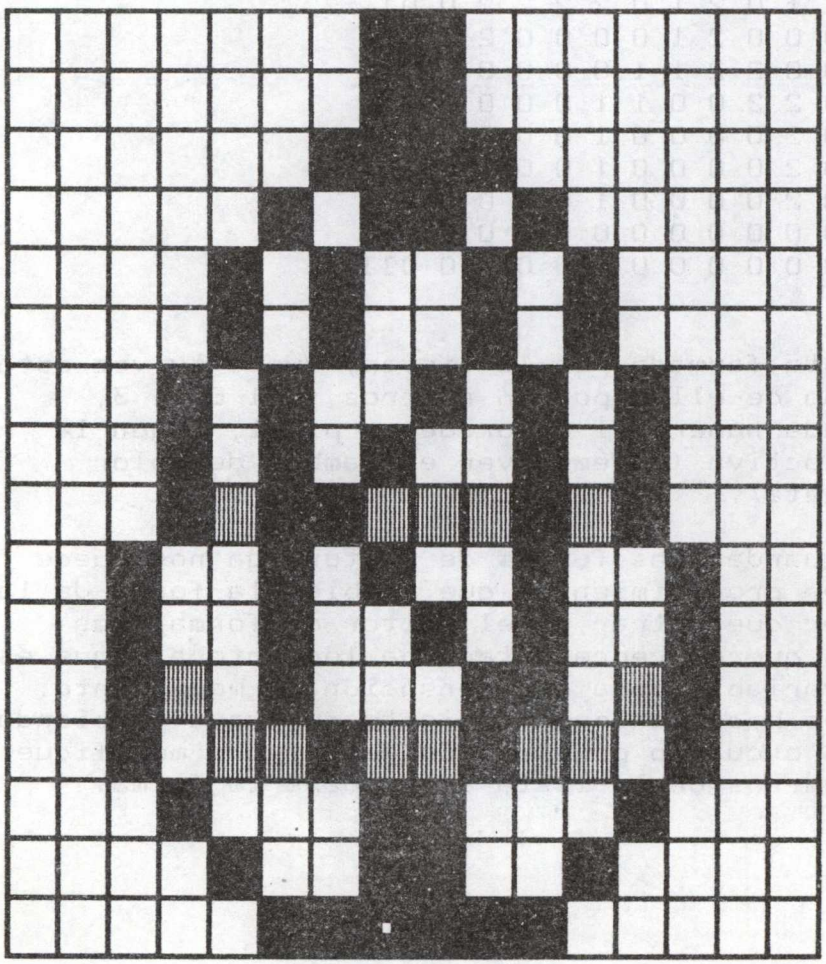

PARA BAJAR.GRUA
OT PONPOS [110 90]
PONRUMBO [180]
REPITE 24 [PONFORMA "GRUA
MT AV 2 PONFORMA "GRUAC
     MT AV 2]
FIN
```













MINISTERIO DE EDUCACION Y CIENCIA - BIBLIOTECA

Las formas que aparecen detrás de los procedimientos corresponden a los cuatro nombres distintos de tortugas que se usan a lo largo del programa y que, evidentemente, es preciso introducir antes de hacer correr el programa.

El LOGO guarda las formas por medio de listas dentro de las variables; así, si preguntamos el contenido de la variable "GRUA por medio de:

```
? ES :GRUA
```

El resultado obtenido será:

```
[[0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 2 1 2 0 0 0 0 0 0]
 [0 0 0 1 1 1 1 2 1 2 2 2 2 0 0 0]
 [0 0 1 1 1 1 0 2 1 0 2 2 2 2 0 0]
 [0 1 1 0 0 0 0 2 1 0 0 0 0 2 2 0]
 [1 1 0 0 0 0 2 2 1 1 0 0 0 0 2 2]
 [1 0 0 0 0 2 2 0 0 1 1 0 0 0 0 2]
 [1 0 0 0 0 2 0 0 0 0 1 0 0 0 0 2]
 [1 0 0 0 0 2 0 0 0 0 1 0 0 0 0 2]
 [1 0 0 0 0 2 0 0 0 0 1 0 0 0 0 2]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

Es decir una lista formada por 16 listas, que a su vez están formadas cada una de ellas por 16 números, del 0 al 3, representando cada número el color de un pixel, según la paleta qué esté activa (podemos ver el cambio de color cambiando la paleta).

Esta manera de guardar las formas de la tortuga nos puede servir para hacer procedimientos que cambien la forma de la tortuga sin tener que entrar en el editor de forma. Por ejemplo, podemos querer representar una locomotora y que ésta suelte un humo variable para dar sensación de movimiento. Para ésto podemos hacer un procedimiento que vaya cambiando el color de tres o cuatro pixeles. Es decir, que modifique tres o cuatro números de la lista asociada a la forma.

Los primeros procedimientos son claramente entendibles por lo que no nos detendremos en ellos, se trata de dibujar una casa, elegir una fracción y buscar un número que, al aplicarle la fracción, produzca otro entero (para no complicar las series) y que no esté previamente elegido.

Una vez que hemos encontrado un número, lo metemos en la casa arrastrado por una tortuga con forma de coche (PONFORMA "COCHE del procedimiento METE); puede verse, con respecto a la animación que, mientrás las tortugas pueden avanzar de pixel en pixel, los caracteres tienen posiciones fijas y sólo se pueden moverse módulos de 8 pixel; también es destacable que las coordenadas de las posiciones de los caracteres no coinciden con las de las tortugas.

Suponemos que el número entra en la casa y que dentro de ella sufre una maquiavélica modificación que lo convierte en otro distinto que también sale arrastrado por el coche.

Cuando el número ha llegado a un punto determinado, una agrasiva grua, que abre y cierra sus fauces, bajará a recogerlo para colocarlo en su posición. Nótese el efecto de animación que produce la combinación de varias figuras parecidas, en este caso GRUA y GRUAC, en el procedimiento BAJAR.GRUA. Por medio de la sucesión de las dos formas por la repetición de PONFORMA "GRUA y PONFORMA "GRUAC conseguimos que parezca que es la misma grua que abre y cierra las garras.

Si en este momento detenemos el programa y preguntamos la forma de la tortuga:

```
? ES FORMA
```

La respuesta puede ser GRUA o GRUAC según el punto en el que lo hayamos detenido.

Una vez que la grua ha bajado, podemos observar otra aplicación de la edición de formas, se trata de construir un movimiento con varias tortugas de diversas formas. Para conseguir la actuación simultánea de varis tortugas empleamos la primitiva ACTIVA.

En el procedimiento ATRAPAR.NUMERO, después de borrar el número que había trasladado el coche, activamos una segunda tortuga (ACTIVA [1], recuérdese que los ordenadores empiezan a contar desde cero), le damos la forma deseada (PONFORMA

"RED) y la colocamos en su sitio preciso, ya que como no habia sido usada anteriormente, su posición era [0 0]; una vez que tenemos la tortuga en su sitio y con su orientación la mostramos y activamos la nueva junto con la antigua (ACTIVA [0 1])

En este momento tenemos dos tortugas activas, lo cual podemos comprobar parando el proceso y tecleando:

```
? ES QUIEN
```

Y la respuesta será

```
[0 1]
```

Todas las órdenes que demos ahora a la tortuga las ejecutarán simultáneamente ambas tortugas, por lo que hace el efecto de ser una sola más grande.

LOGO SIN EDITOR DE FORMAS

Anque el LOGO no posea un editor de formas, si suelen tener posibilidad de cambiar la tortuga a un simbolo ASCII determinado; así si tecleamos:

```
? PONFORMA 67
```

La tortuga se convertirá en una C, pero si ahora tecleamos:

```
? GD 90
? AV 100
```

Veremos, en la mayoría de los casos, que la C sigue en la misma posición aunque se haya desplazado hacia la derecha.

El sistema para crear una nueva forma de tortuga será crear una figura y darle un número del código ASCCI y luego cambiar la tortuga a ese código.

La primitiva que asocia una forma a un código ASCII es MODELO n, que asocia al número ASCII n la parte de pantalla que cubre la tortuga. este número ha de estar comprendido entre el 128 y 255.

En este punto hay que tener en cuenta dos cuestiones: la parte de pantalla sombreada por la tortuga suele ser mayor que lo que se ve en pantalla y no olvidarse de ocultar la tortuga antes de asociar una forma a un número.

La zona que cubre la tortuga suele ser un rectángulo de 10 x 8 pixel cualquiera que sea la forma de ésta; cuyas esquinas, si la tortuga está en el centro, serán [-3 5], [-3 -4], [4 -4] y [4 5]; así pues, si queremos saber realmente la zona cubierta por la tortuga podemos teclear:

```
? PONFORMA 219
```

Y en pantalla aparecerá un cuadradito (▒) que nos cubre toda la zona que sombrea la tortuga y en negativo el dibujo que hay debajo.

Probemos a cambiar el cáracter 175 (») por una cruz, por medio del procedimiento siguiente:

```
PARA CAMBIAR.CARACTER
BP
AV 5
RE 2
GD 90
RE 2
AV 4
SL CENTRO BL
MODELO 175
FIN
```

Ejecutamos nuestro procedimiento y tecleamos:

```
? ES CAR 175
```

Comprobando nuestra cruz. Pero... ¡Algo falla!, el nuevo carácter 175 no se parece ni remotamente a lo que intentábamos; nos hemos olvidado de ocultar la tortuga para que ésta no forme parte del espacio sombreado; modifiquemos el procedimiento ocultandola antes de grabar el modelo:

```
PARA CAMBIAR.CARACTER
BP
AV 5
RE 2
GD 90
RE 2
AV 4
OT
SL CENTRO BL
MODELO 175
FIN
```

Ahora ya funciona a la perfección, el carácter 175 es una cruz.

Si quisiéramos que la tortuga tuviera esta forma nos bastaría con teclear:

```
? BP PONFORMA 175 MT
```

Y podremos comprobar que nuestra tortuga tiene forma de cruz con las órdenes AV o RE, también podemos comprobar que no gira la figura con GD o GI (si lo hace ¡felicidades!).

Ya sólo nos queda hacer un procedimiento que reuna todo lo anterior. La secuencia de movimientos se la daremos en forma de una lista, pero a menudo sucede que las listas se hacen más largas que la línea que acepta el LOGO por lo que en lugar de listas de órdenes suele ser más sencillo usar listas de procedimientos.

Necesitaremos primero, por tanto, fabricar un procedimiento que nos vaya haciendo el dibujo ejecutando los elementos de una lista:

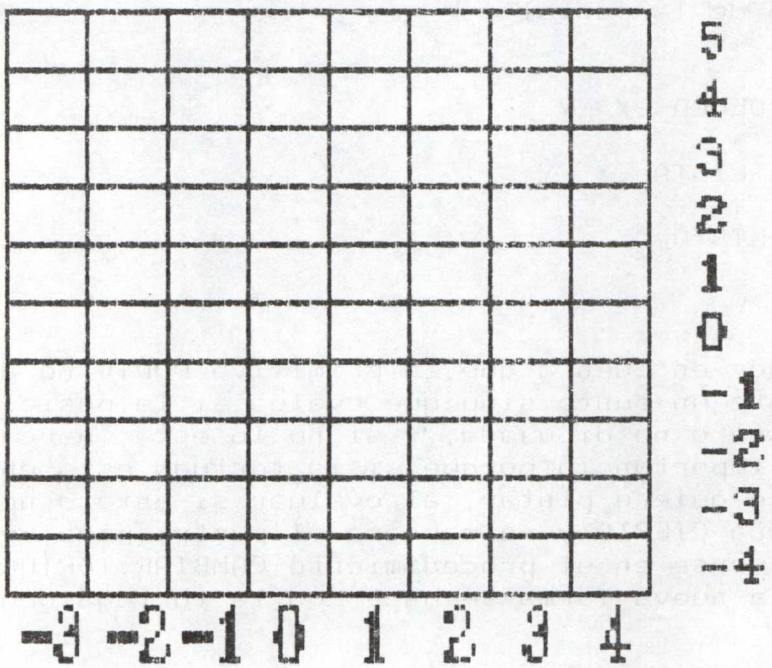
```
PARA DIBUJO :LISTA
SI LISTA = [] [ALTO]
EJECUTA PRIMERO :LISTA
DIBUJO MF :LISTA
FIN
```

Ahora definimos el procedimiento para cambiar la forma de la tortuga:

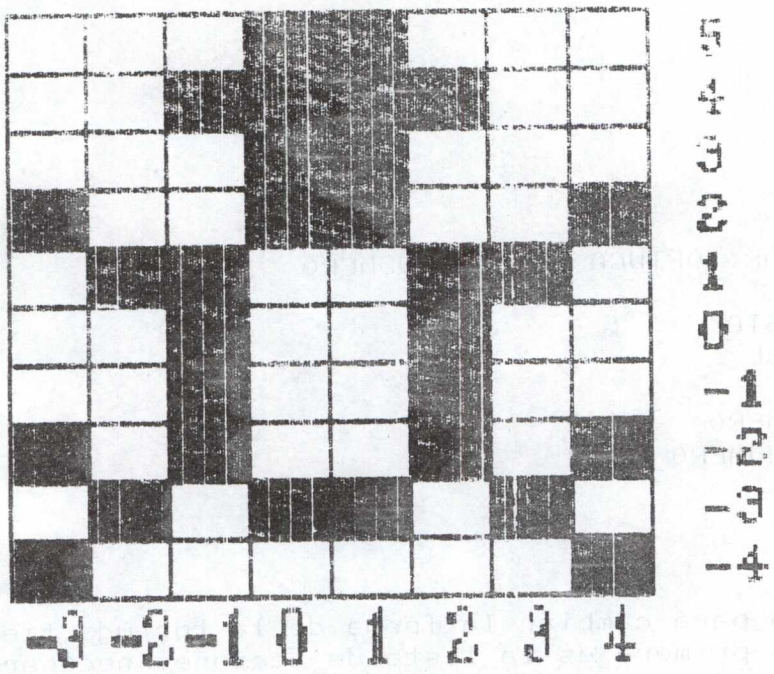
```
PARA CAMBIAR.TORTUGA :LISTA :NUMERO
BP OT
DIBUJO :LISTA
SL CENTRO BL
OT
MODELO :NUMERO
PONFORMA :NUMERO
BP MT
FIN
```

El procedimiento para cambiar la forma de la tortuga tiene dos entradas, la primera es la lista de acciones necesarias para hacer el dibujo, la segunda es en número ASCII que queremos dar a la nueva forma.

Para dibujar la nueva forma es conveniente que nos hagamos en una cuadrícula un rectángulo de la forma de la sombra de la tortuga:



Supongamos que queremos que nuestra tortuga se parezca más a una tortuga para lo cual marcamos en la cuadrícula las casillas que nos interesa que queden sombreadas,



Ahora debemos hacer que la tortuga dibuje las zonas sombreadas, ésto lo podemos hacer de diversas maneras, como es frecuente que muchas de las zonas sombradas sean puntos, creamos un procedimiento que nos dibuje un punto en un lugar determinado de la pantalla:

```
PARA COLOCA :X :Y
SL
PONPOS LISTA :X :Y
BL
PUNTO LISTA :X :Y
FIN .
```

Hay que tener en cuenta que la primitiva PUNTO no dibuja estrictamente un punto sino que evalúa si la posición indicada está o no dibujada, y si no lo está, marca el punto. Esto tiene importancia porque si la tortuga está encima del punto que se quiere pintar, al evaluar si está o no encendido el pixel dará CIERTO y no marcará el punto; para evitar ésto, como puede verse en el procedimiento CAMBIAR.TORTUGA, todo el dibujo de la nueva forma se hace con la tortuga oculta.

Para las pequeñas líneas modificamos el procedimiento anterior:

```
PARA COLOCA.AVANZA :X :Y :LONGITUD
COLOCA :X :Y
AV :LONGITUD
FIN
```

Estos dos procedimientos nos simplifican bastante la terea de dibujar nuestra nueva tortuga:

```
PARA TORTUGA
COLOCA -3 -4
COLOCA -3 2
COLOCA -2 -3
COLOCA -2 1
COLOCA.AVANZA -1 -2 3
COLOCA -1 4
COLOCA 0 -3
COLOCA.AVANZA 0 2 3
COLOCA 1 -3
COLOCA.AVANZA 1 2 3
COLOCA.AVANZA 2 -2 3
COLOCA 2 4
COLOCA 3 -3
COLOCA 3 1
COLOCA 4 -4
COLOCA 4 2
FIN
```

Y una vez que tenemos el procedimiento que nos dibuja la tortuga lo podemos introducir en CAMBIAR.TORTUGA pero antes sepamos cuál es la notación de la tortuga estándar por medio de:

```
? ES FORMA
```

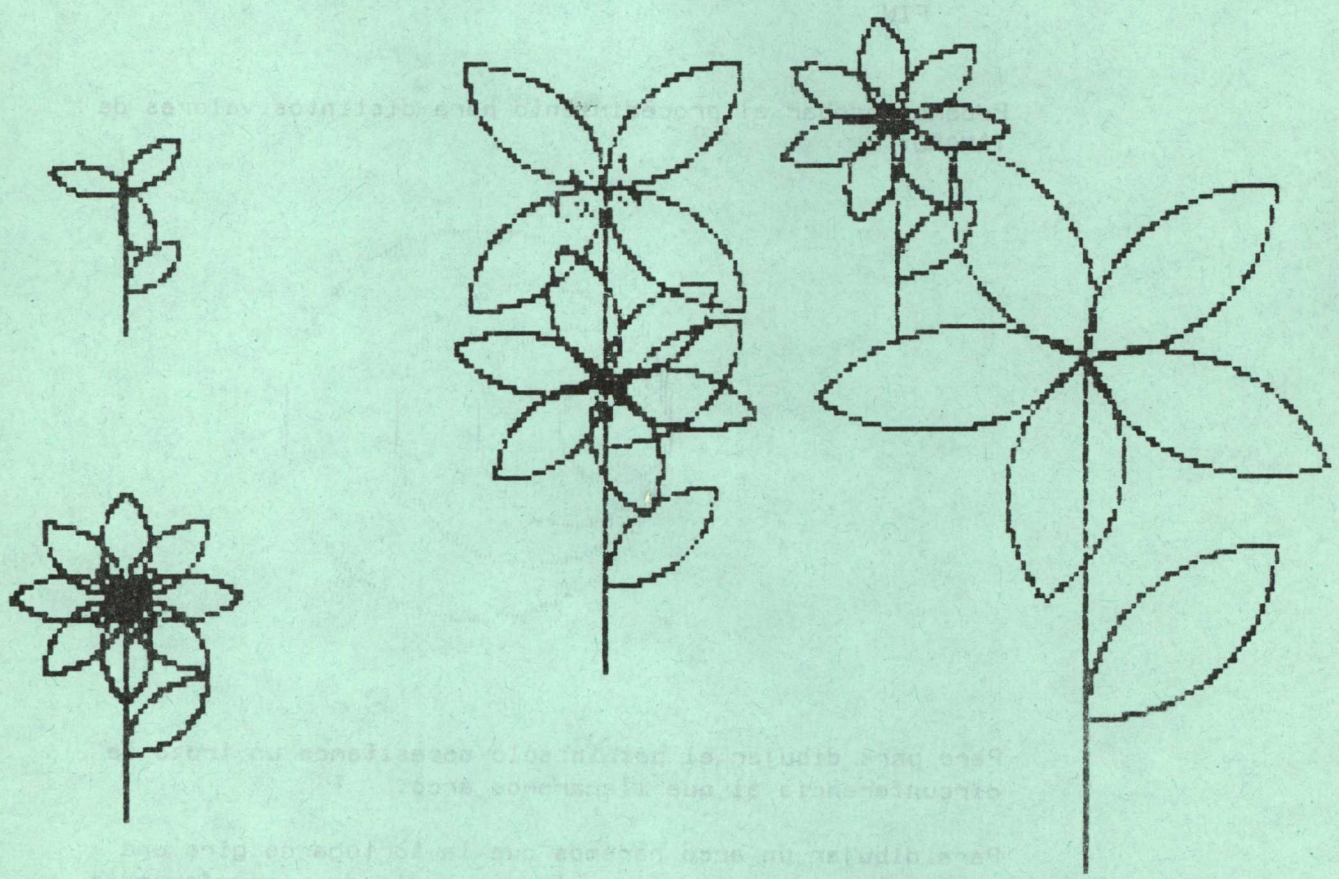
El resultado será la forma que tendremos que teclear cuando queramos tener la tortuga normal. Ya estamos en disposición de que nuestra tortuga sea una verdadera tortuga (aunque no gire):

```
? CAMBIAR.TORTUGA [[TORTUGA]] 250
```

Y, si todo ha ido bien, y no nos hemos equivocado al teclear, tendremos una tortuguita moviéndose por la pantalla.

APLICACION: DIBUJO DE UNA FLOR

Como ha empezado la primavera, hoy nos proponemos dibujar un jardín con flores del siguiente estilo:



Nuestro jardín está formado por la repetición de un elemento, que llamaremos "flor", que a su vez se compone de un número indeterminado de "pétalos".

Luego nuestro primer trabajo será descubrir un procedimiento que nos dibuje pétalos. Un pétalo está formado por dos trozos opuestos de circunferencia.

Recuerda que para hacer un círculo se podía emplear el procedimiento:

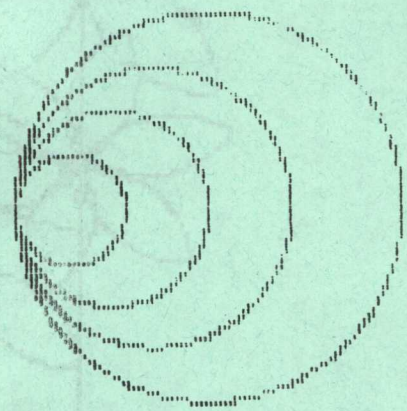
```
PARA CIRCULO
REPITE 360 [AV 1 GD 1]
FIN
```

Pero, claro, este procedimiento tiene el defecto que siempre nos hace los círculos del mismo tamaño; para solucionarlo, haremos que la longitud del camino recorrido por la tortuga

antes de cada giro pueda ser distinto:

```
PARA CIRCULO :TAMAÑO
REPITE 360 [AV :TAMAÑO GD 1]
FIN
```

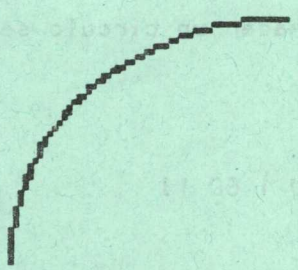
Podemos probar el procedimiento para distintos valores de TAMAÑO:



Pero para dibujar el pétalo sólo necesitamos un trozo de circunferencia al que llamaremos arco.

Para dibujar un arco haremos que la tortuga no gire una vuelta completa sino tan solo una parte de circunferencia, más concretamente un cuarto de vuelta; para ésto le ordenaremos a la tortuga que repita el movimiento la cuarta parte de 360 es decir 90 veces:

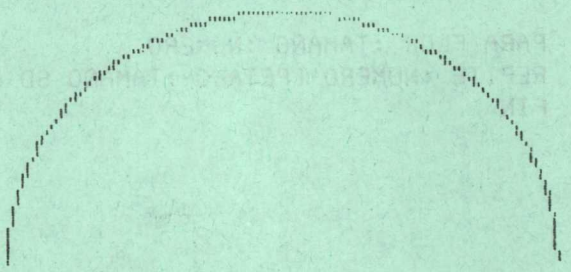
```
PARA ARCO :TAMAÑO
REPITE 90 [AV :TAMAÑO GD 1]
FIN
```



Una vez conseguido el arco probemos a formar el pétalo uniendo dos arcos:

```
PARA PETALO :TAMAÑO
REPITE 2 [ARCO :TAMAÑO]
FIN
```

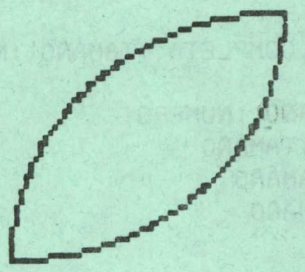
Probamos con PETALO 1



¡Oh! Nuestro pétalo no se parece en nada a lo que queríamos conseguir, entre los dos arcos la tortuga tiene que hacer un giro. Prueba hasta que el resultado sea el deseado.

```
? REPITE 2 [ARCO 1 GD .... ]
```

Habrás visto que necesitas girar 90 grados la tortuga antes de hacer el siguiente arco para que el pétalo quede completo:

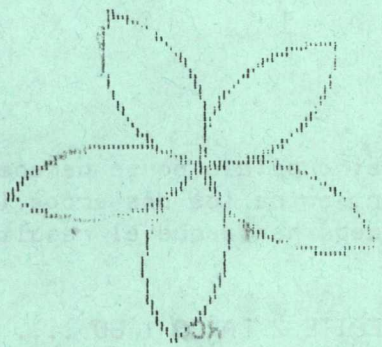


Una vez que tenemos el pétalo, tratemos de hacer una flor uniendo diversos pétalos; para que la flor quede cerrada, entre pétalo y pétalo la tortuga tendrá que girar un número diferente de grados según el número de pétalos que tenga la flor:

```
PARA FLOR :TAMAÑO :NUMERO
REPITE :NUMERO [PETALO :TAMAÑO GD .... ]
FIN
```

Debemos encontrar una estrategia que nos relacione el número de pétalos con el giro entre cada pétalo y el siguiente; probemos a dibujar una flor de 4 pétalos, de 3, de 5, etc; hasta que encontremos la solución:

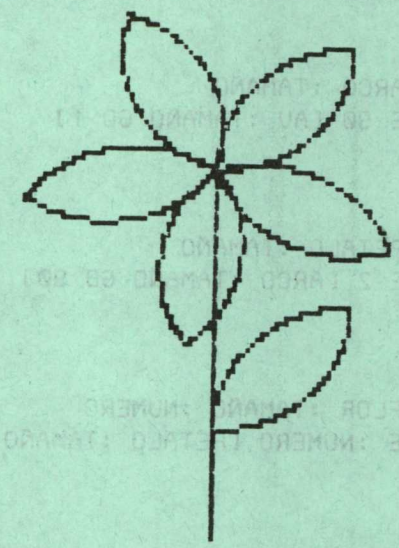
```
PARA FLOR :TAMAÑO :NUMERO
REPITE :NUMERO [PETALO :TAMAÑO GD 360 / :NUMERO ]
FIN
```



Nuestra flor está prácticamente terminada, sólo nos falta hacerle el tallo mediante un RETROCEDE y otro pétalo en el tallo. El procedimiento puede ser del estilo:

```
PARA FLOR.COMPLETA :TAMAÑO :NUMERO
OT
FLOR :TAMAÑO :NUMERO
RE 120 * :TAMAÑO
PETALO :TAMAÑO
RE 60 :TAMAÑO
FIN
```

Nuestra flor está terminada:



Nuestro jardín se compone de una serie de flores dibujadas en distintos puntos de la pantalla; para que la tortuga se traslade de un punto a otro definimos el procedimiento TRASLADAR:

```
PARA TRASLADAR
SL
PONPOS LISTA (150 - AZAR 300) (120 - AZAR 240)
BL
FIN
```

Ahora definimos el jardín como una repetición de flores de diferentes colores, posiciones y tamaños:

```
PARA JARDIN
PG BP VENTANA
REPITE (5 + AZAR 15) [PONCL (1 + AZAR 3) TRASLADAR
   FLOR.COMPLETA ((1 + AZAR 7) / 10) 2 + AZAR 6
FIN
```

En este último procedimiento, usamos cuatro veces la primitiva AZAR; en primer lugar (5 + AZAR 15) para que se dibujen un número de flores entre 5 y 19; después usamos (1 + AZAR 3) para definir el color de cada una ; posteriormente el tamaño se define con ((1 + AZAR 7) / 10 entre 0.1 y 0.7 y, por último, 2 + AZAR 6 delimita el número de pétalos entre 2 y 7.

LISTADO DE TODOS LOS PROCEDIMIENTOS USADOS EN JARDIN

```
PARA ARCO :TAMAÑO
REPITE 90 [AV :TAMAÑO GD 1]
FIN

PARA PETALO :TAMAÑO
REPITE 2 [ARCO :TAMAÑO GD 90]
FIN

PARA FLOR :TAMAÑO :NUMERO
REPITE :NUMERO [PETALO :TAMAÑO GD 360 / :NUMERO ]
FIN

PARA FLOR.COMPLETA :TAMAÑO :NUMERO
OT
FLOR :TAMAÑO :NUMERO
RE 120 * :TAMAÑO
PETALO :TAMAÑO
RE 50 :TAMAÑO
FIN

PARA TRASLADAR
SL
PONPOS LISTA (150 - AZAR 300) (120 - AZAR 240)
BL
FIN

PARA JARDIN
PG BP VENTANA
REPITE (5 + AZAR 15) [PONCL (1 + AZAR 3) TRASLADAR
   FLOR.COMPLETA ((1 + AZAR 7) / 10) 2 + AZAR 6
FIN
```

A P L I C A C I O N

PROGRAMACION DE UN RELOJ

Plantearemos ahora el problema de realizar un reloj analógico, con dos agujas, minutero y horaria, a las que haremos avanzar conforme pase el tiempo de minuto en minuto.

Realizaremos un análisis modular del problema, las fases a seguir corresponden a los distinntos procedimientos que desarrollaremos más tarde.

- El procedimiento principal (RELOJ) tendrá como entradas las HORAS y MINUTOS del tiempo actual, y se compondrá de:

  . ESFERA, que dibujará la esfera del reloj.
  . PONER EN HORA, que dibujará las agujas en la posición correspondiente a la hora correcta.
  . AVANZAR TIEMPO, que deberá ir variando las posiciones de las agujas conforme pasan los minutos.

Así pues, el procedimiento RELOJ se define:

```
PARA RELOJ :HORAS :MINUTOS
ESFERA
PONER.EN.HORA :HORAS :MINUTOS
AVANZAR.TIEMPO
FIN
```

Dibujo de la esfera

El procedimiento ESFERA, dibuja un círculo (CIRCULO) y dibuja las 12 señales, correspondientes a cada hora en la esfera horaria.

```
PARA ESFERA
SL PONX -72 BL
CIRCULO
SL CENTRO BL
REPITE 12 [MARCA GD 30]
FIN


PARA CIRCULO
REPITE 72 [AV 6.28 GD 5]
FIN


PARA MARCA
SL AV 50 BL AV 10 SL RE 60 BL
FIN
```

Puesta en hora del reloj

El procedimiento PONER EN HORA, tiene como entradas el tiempo real (HORAS y MINUTOS), calcula el ángulo de cada una de las agujas del reloj (MINUTERO y HORARIA) y llama a un procedimiento que dibuja ambas agujas (DIBUJAR AGUJAS)

```
PARA PONER.EN.HORA :HORAS :MINUTOS
HAZ "MINUTERO :MINUTOS * 6
HAZ "HORARIA :HORAS * 30 + :MINUTOS * 0.5
DIBUJAR.AGUJAS :MINUTERO :HORARIA
FIN
```

El cálculo del ángulo de las agujas tiene en cuenta que cada minuto se corresponde con 6º de ángulo del MINUTERO, cada hora son 30º de ángulo de la aguja HORARIA y 0.5º cada minuto más.

El procedimiento DIBUJAR AGUJAS, de acuerdo con el valor de sus entradas,

que se corresponden con los ángulos de ambas agujas, dibuja los trazos correspondientes representando a ambas, orientando previamente la tortuga.

```
PARA DIBUJAR.AGUJAS :MINUTERO :HORARIA
PONR :MINUTERO AV 40 RE 40
PONR :HORARIA AV 20 RE 20
FIN
```

Movimiento de las agujas

El procedimiento AVANZAR.TIEMPO debe simular el movimiento de las agujas del reloj, y para ello lo que haremos será borrar cada minuto las agujas que estaban dibujadas y dibujar las nuevas, y ésto indefinidamente. Naturalmente hay que esperar un minuto para realizar esta acción y hay que actualizar los valores de las variables MINUTERO y HORARIA antes de dibujarlas en la nueva posición, lo cual se consigue incrementando cada minuto MINUTERO en 6 grados y HORARIA en 0.5 grados, con lo que al llamar al procedimiento DIBUJAR.AGUJAS con los nuevos valores, las dibuja en su nueva posición.

```
PARA AVANZAR.TIEMPO
ESPERA 1000                          (El valor de la espera depende
BORRAR.AGUJAS                         del ordenador que se use)
HAZ "MINUTERO :MINUTERO + 6
HAZ "HORARIA :HORARIA + 0.5
DIBUJAR.AGUJAS :MINUTERO :HORARIA
AVANZAR.TIEMPO
FIN
```

El procedimiento BORRAR.AGUJAS es idéntico al de DIBUJAR.AGUJAS, salvo que las dibuja con el mismo color del fondo (**GOMA**), restaurando fi-

```
PARA BORRAR.AGUJAS
GOMA
PONR :MINUTERO AV 40 RE 40
PONR :HORARIA AV 20 RE 20
PONCL 1
BL
FIN
```

```
SETH  :  HORARIA  FD  20 BK 20
SETPC 1
PD
END
```

En la figura se muestra el funcionamiento del programa

A continuación se dan todos los listados de los procedimientos definidos.

```
PARA RELOJ :HORAS :MINUTOS
ESFERA
PONER.EN.HORA :HORAS :MINUTOS
AVANZAR.TIEMPO
FIN
```

```
PARA ESFERA
SL PONX -72 BL
CIRCULO
SL CENTRO BL
REPITE 12 [MARCA GD 30]
FIN
```

```
PARA CIRCULO
REPITE 72 [AV 6.28 GD 5]
FIN
```

```
PARA MARCA
SL AV 50 BL AV 10 SL RE 60 BL
FIN
```

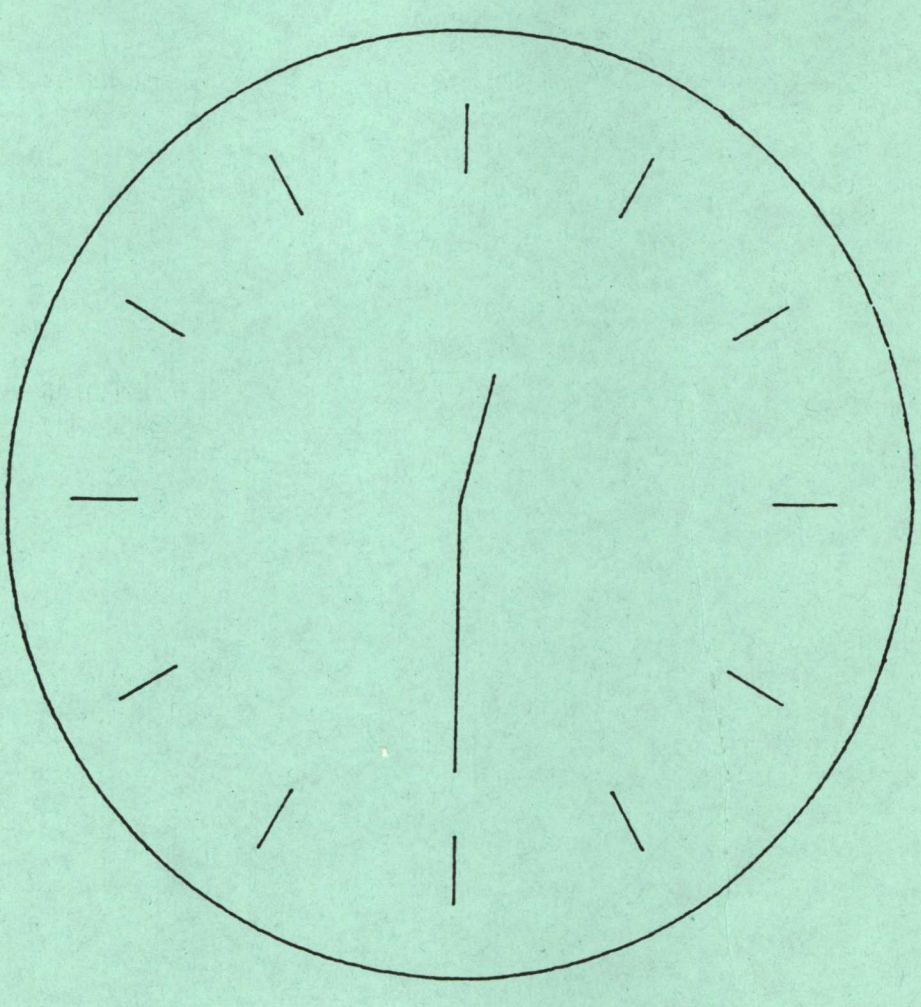
```
PARA PONER.EN.HORA :HORAS :MINUTOS
HAZ "MINUTERO :MINUTOS * 6
HAZ "HORARIA :HORAS * 30 + :MINUTOS * 0.
DIBUJAR.AGUJAS :MINUTERO :HORARIA
FIN
```

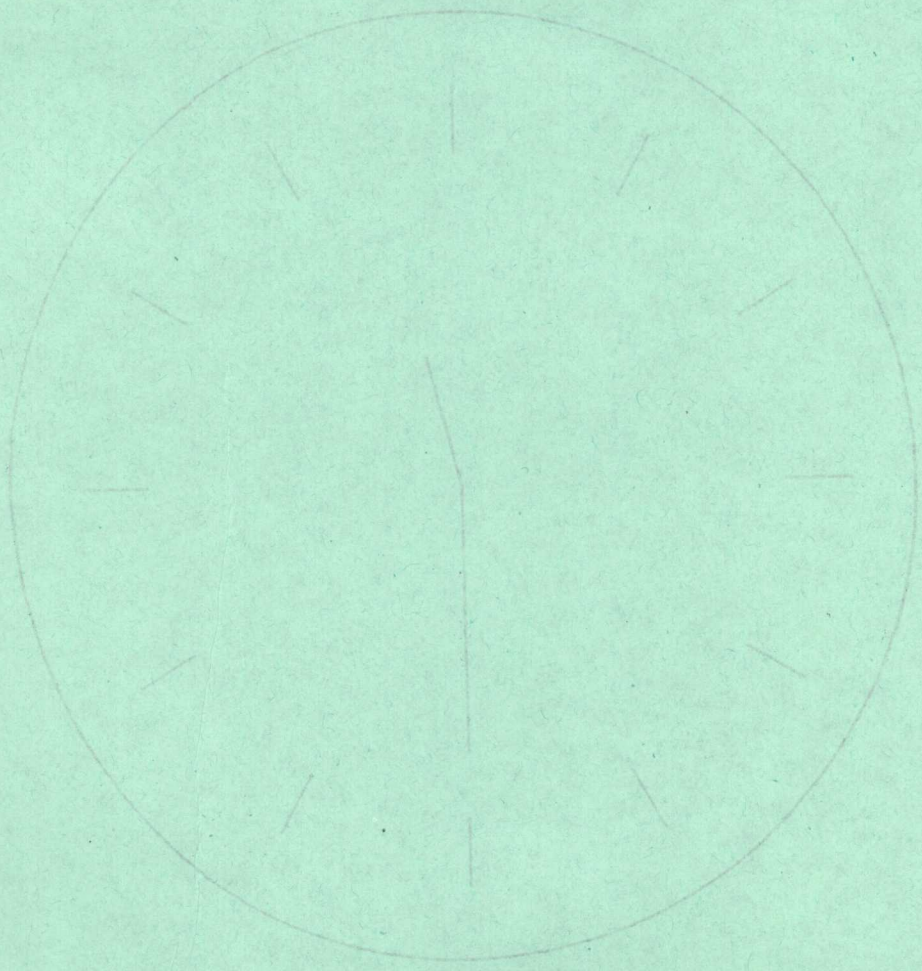
```
PARA AVANZAR.TIEMPO
ESPERA 1000                    (El valor de la espera depende
BORRAR AGUJAS                   del ordenador que se use)
HAZ "MINUTERO :MINUTERO + 6
HAZ "HORARIA :HORARIA + 0.5
DIBUJAR.AGUJAS :MINUTERO :HORARIA
AVANZAR.TIEMPO
FIN
```

```
PARA DIBUJAR.AGUJAS :MINUTERO :HORARIA
PONR :MINUTERO AV 40 RE 40
PONR :HORARIA AV 20 RE 20
FIN
```

```
PARA BORRAR.AGUJAS
GOMA
PONR :MINUTERO AV 40 RE 40
PONR :HORARIA AV 20 RE 20
PONCL 1
BL
FIN
```

Se deja como ejercicio al lector la ampliación del problema para que dibuje el segundero y lo haga avanzar cada 5 segundos, y que imprima simultáneamente las horas, minutos y segundos.





EL CUENTO INTERMINABLE

Erase una vez un niño y una niña, Arturo y Cristina que estaban aburridos en casa. Era sábado y llovía tanto que era impensable salir a jugar al parque.

Arturo, que era el mayor, propuso:

- Podríamos leer un cuento.

- ¡Bah!. Ya nos hemos leido todos los que tenemos - respondió Cristina con cara aburrida.

- Inventémonos uno - insistió Arturo.

- Si nos inventamos un cuento, cuando lo terminemos ya lo habremos leido y seguiremos en la misma situación. - replicó su hermana.

- Tienes razón, Cristina, tendríamos que hacer un cuento que no hubiésemos leido, un cuento largo, que no terminase nunca, que siempre fuera distinto. ¿Tu crees que nuestra tortuga nos podría ayudar?

- Podemos probar.

Y dicho y hecho, acudieron a su ordenador y teclearon con toda ilusión:

```
? CUENTO
```

Pero la tortuga respondió lacónicamente:

```
NO SE COMO HACER PARA CUENTO
```

El desánimo empezó a cundir, la tortuga no era tan lista como parecía, ni siquiera sabía lo que era un cuento.

- Podríamos enseñarle lo que es un cuento. - Sugirió Cristina

- ¡Claro!. Así podría contar cuentos que nosotros no supiéramos. Total, un cuento no es más que una serie de personajes que realizan una serie de acciones.

- Sí, pero los personajes tienen que salir de un modo inesperado y realizar acciones audaces, arriesgadas, impensables, ... Empecemos por darle intriga al cuento haciendo que la tortuga elija al azar un elemento de una lista con nuestro viejo procedimiento:

```
PARA ELIGE :LISTA
DEV ELEMENTO (1 + AZAR (CUENTA :LISTA)) :LISTA
FIN
```

A continuación construyeron un procedimiento para que la tortuga eligiera un personaje:

```
PARA PERSONAJES
HAZ "PERSONAJES [[EL REY] [EL MAGO] [EL HADA] [LA BRUJA]
 [EL DRAGON] [LA LUNA] [LA TORTUGA]]
DEV ELIGE :PERSONAJES
FIN
```

Y probaron su nuevo procedimiento:

```
? ES PERSONAJES
LA LUNA
? ES PERSSONAJES
LA BRUJA
   :
   :
```

El cuento empezaba a funcionar, ya tenían una serie de personajes; ahora estos personajes tenían que hacer algo. De un modo parecido al anterior definieron un procedimiento para las acciones:

```
PARA ACCIONES
HAZ "ACCIONES [[MIRABA] [COMIA] [ILUMINABA] [AMABA]
 [ODIABA] [DEVORABA] [CONSOLABA]]
DEV ELIGE :ACCIONES
FIN
```

Y otra vez probaron su procedimiento:

```
? ES ACCIONES
AMABA
? ES ACCIONES
ILUMINABA
    :
    :
```

Ya tenían personajes y acciones para el cuento, sólo faltaba obligar a los personajes a realizar las acciones, para esto construyeron el procedimiento ORACION:

```
PARA ORACION
DEV (FRASE PERSONAJES ACCIONES PERSONAJES)
FIN
```

El procedimiento funcionaba del siguiente modo:

```
? ES ORACION
EL DRAGON MIRABA EL REY
? ES ORACION
LA LUNA DEVORABA LA TORTUGA
     :
     :
```

Empezaba a aproximarse a un cuento pero las oraciones no eran correctas, además todas eran afirmativas: era posible que el dragón pudiera no mirar a alguien o la luna no iluminar algo.

Era preciso crear dos procedimientos, uno para que las oraciones apareciesen de forma correcta y otro para que pudieran ser negativas:

El procedimiento para las frases negativas se hizo del siguiente modo:

```
PARA SINO
SI 0 = AZAR 2 [DEV "NO]
DEV "
FIN
```

Para arreglar las oraciones hubo que crear el procediminto COMPLEMENTO y, para hacer las cosas más sencillas, incluyeron éste en otro llamado PREDICADO

```
PARA COMPLEMENTO
LOCAL "COM
HAZ "COM PERSONAJES
SI "EL = PRIMERO :COM [DEV FRASE "AL MP :COM]
DEV FRASE "A :COM
FIN


PARA PREDICADO
DEV (FRASE SINO ACCIONES COMPLEMENTO)
FIN
```

Con estos dos nuevos procedimientos arreglaron el de ORACION convirtiéndolo en ORACION1:

```
PARA ORACION1
DEV FRASE PERSONAJES PREDICADO
FIN
```

El nuevo procedimiento ya construía oraciones de un modo correcto:

```
? ES ORACION1
EL REY NO ILUMINABA A LA TORTUGA
? ES ORACION1
LA TORTUGA MIRABA A LA LUNA
        :
```

El cuento ya funcionaba. Al cabo de un rato de jugar con él, pensaron que había más tipos de oraciones y que si le enseñaban a la tortuga a construirlas el cuento sería aún más divertido.

Pensaron que un personaje podía hacer dos acciones al mismo tiempo y crearon un procedimiento para que las pudiera hacer. Dicho procedimiento requería la elección de un enlace entre las dos acciones que se podía realizar de una manera similar a los procedimientos a como se hizo para los personajes y las acciones:

```
PARA ENLACE
HAZ "ENLACE [[PERO] [SIN EMBARBO] [NO OBSTANTE] [Y]
  [ADEMAS] [,] [AUNQUE] [EN OTRO LUGAR]]
DEV ELIGE :ENLACE
FIN
```

Una vez solucionado el tema del enlace, la construcción de la frase no presentaba dificultad ninguna:

```
PARA ORACION2
DEV (FRASE ORACION1 ENLACE PREDICADO)
FIN
```

Ya tenían dos tipos de oraciones, el cuento se enriquecía:

```
? ES ORACION2
EL MAGO AMABA A LA BRUJA Y CONSOLABA AL MAGO
? ES ORACION2
LA TORTUGA CONSOLABA AL REY AUNQUE NO AMABA AL MAGO
        :
        :
```

Pero aún se podían contruir oraciones diferentes, a Cristina se le ocurrió otro tipo de oraciones:

```
PARA ORACIONE3
DEV (FRASE PERSONAJES "QUE PREDICADO PREDICADO)
FIN
```

Procedimiento que construía frases del estilo:

```
? ES ORACION3
EL DRAGON QUE NO MIRABA A LA BRUJA ILUMINABA AL HADA
? ES ORACION3
LA BRUJA QUE MIRABA AL REY MIRABA A LA TORTUGA
        :
        :
```

Por fin, integraron todos los procedimientos en otro que se llamó CUENTO:

```
PARA CUENTO
ES EJECUTA ELIGE [[ORACION1] [ORACION2] [ORACION3]]
CUENTO
FIN
```

Que cada vez se parecía más a un cuento de verdad:

```
? CUENTO
EL MAGO AMABA AL REY
LA LUNA ODIABA AL MAGO EN OTRO LUGAR MIRABA AL HADA
EL DRAGON QUE NO AMABA A LA TORTUGA NO AMABA AL MAGO
                :
                :
```

Pasaron el resto de la tarde, y aún lo hacen los días de lluvia, perfeccionando su cuento; desarrollaron muchos tipos de estructuras teniendo en cuenta los lugares, el tiempo, las cosas, etc...

Y colorín colorado.....

APRENDIZAJE DEL CONCEPTO DE VARIABLE

OBJETIVO GENERAL

Se pretende con esta aplicación didáctica que los alumnos del Ciclo Superior de E.G.B. realicen un acercamiento al concepto de variable.

OBJETIVOS ESPECIFICOS

- Definir procedimientos LOGO con entradas.

- Desarrollar la posibilidad de creación de variables en LOGO

- Observar la variación de los resultados de un procedimiento al cambiar el contenido de las variables.

- Reconocer la potencia de LOGO al admitir el uso y la manipulación de variables en sus procedimientos.

- Realizar procedimientos que supongan el manejo y la creación de variables.

- Investigar las operaciones con variables.

METODOLOGIA

El alumno enseña al ordenador a realizar figuras concretas a través de los gráficos de la tortuga, que pueden ser repetidos una y mil veces al llamarlos por el mismo nombre.

A partir de esto, se trata de que el alumno descubra la posibilidad de que los procedimientos que realice tengan una función más general. Para ello debe llegar a constatar la posibilidad de que los valores numéricos, las ideas,etc pueden ser representados mediante un simbolo.

No se persigue una meta única, sinó que desde un punto de partida común para todos se potencie la investigación del alumno y a través de ella, éste llegue a asimilar conceptos que a menudo no han sido fáciles, pero que con el uso de LOGO pueden aprehenderse de manera sencilla y clara.

CONCEPTO DE VARIABLE

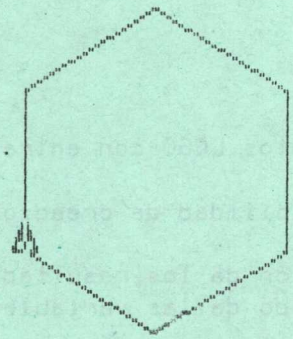
Podemos decir de un modo genérico que variable es una magnitud susceptible de tomar distintos valores en cada caso.

ACTIVIDADES

PROCEDIMIENTOS SIN ENTRADAS

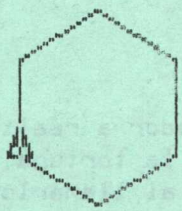
Para comenzar vas a realizar un procedimiento que dibuje un exagono cuyo lado sea 50.

```
? EXAGONO
```

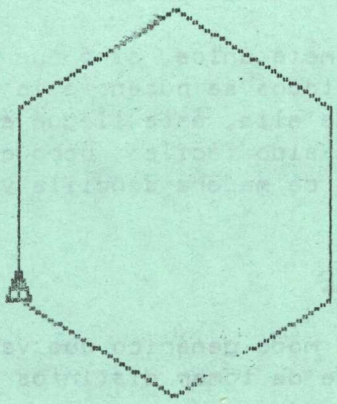


Ahora debes entrar en el EDITOR para modificar la longitud del lado, es decir la cantidad de pasos que avanza la tortuga, para conseguir exagonos de diferentes tamaños.

```
? EXAGONO
```



```
? EXAGONO
```



PROCEDIMIENTOS CON UNA ENTRADA

Como puedes comprobar, resulta bastante dificultoso tener que entrar en el EDITOR cada vez que se pretenda modificar el tamaño de un polígono. Para solventar este problema existe en LOGO la posibilidad de realizar procedimientos que dibujen polígonos cuyos lados sean diferentes cada vez. Para ello a la hora de definir el nuevo polígono, los pasos que avanzará la tortuga no se los daremos como algo concreto sino bajo un símbolo que llamaremos :LADO.

Los : son la clave que hace que el ordenador sepa que lo que va a continuación - LADO - tiene que recibir un contenido, en este caso una cantidad.

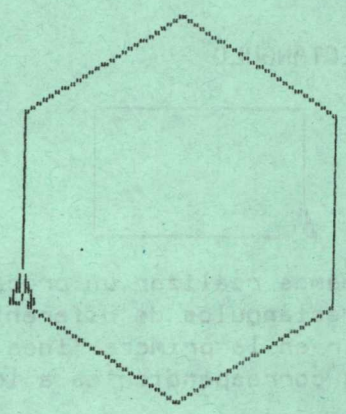
En LOGO las entradas al procedimiento han de estar en la primera linea de definición.

EJEMPLO :

```
PARA EXAGONO :LADO
REPITE 6 [AVANZA :LADO GIRADERECHA 60]
FIN
```

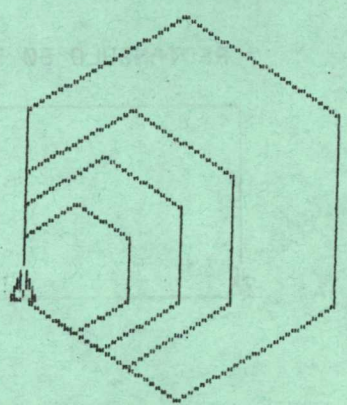
Para ejecutarlo tecleamos el nombre del procedimiento seguido del valor que damos a :lado.:

```
? EXAGONO 60
```



Vamos dando distintos valores:

```
? EXAGONO 20
? EXAGONO 30
? EXAGONO 40
.............
.............
? EXAGONO 80
.............
```



EJERCICIOS

1) Realiza procedimientos que dibujen un triangulo equilatero, un cuadrado, un pentagono y un decágono, cuyos lados sean de longitud variable.

Observa que :LADO contiene una información diferente en cada caso, por ello le denominamos variable.

Los nombres de las variables son palabras que sirven para designar objetos, sus valores o funciones.

En toda variable hay que distinguir dos partes: su nombre - :lado - y su contenido - 20, 30, 40, 60,.....-.
La caracteristica fundamental de los valores de una variable es que pueden ser modificados.

PROCEDIMIENTOS CON DOS ENTRADAS

En los procedimientos realizados anteriormente teníamos un dato variable en cada caso. Vamos a ver un nuevo caso de polígono en el que son necesarias dos entradas en el procedimiento que lo define:

```
PARA RECTANGULO
REPITE 2 [AV 40 GD 90 AV 70 GD 90]
FIN

? RECTANGULO
```



Como deseamos realizar un procedimiento que nos permita dibujar rectángulos de diferentes tamaños tendremos que introducir en la primera linea de definición dos entradas variables correspondientes a los dos lados diferentes.

```
PARA RECTANGULO :ALTO :ANCHO
REPITE 2 [AV :ALTO GD 90 AV :ANCHO GD 90]
FIN

? RECTANGULO 50 80
```



EJERCICIOS

1) A partir del procedimiento RECTANGULO definido con antérioridad puedes obtener distintas figuras modificando los valores de las entradas. Averigua los valores que deben tener las entradas para que el resultado sea:

   a) un cuadrado

   b) una linea horizontal

   c) una linea vertical

2) Utilizando el procedimiento RECTANGULO realiza otro que dibuje banderas de diferente tamaño.

PROCEDIMIENTOS CON VARIAS ENTRADAS

Los procedimientos en LOGO pueden tener más de un valor variable. Asi si nos fijamos en el ejemplo anterior:

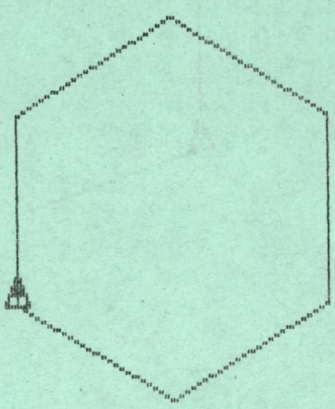
```
PARA EXAGONO :LADO
REPITE 6 [AV :LADO GD 60]
FIN
```

Podemos sustituir los valores constantes 6 y 60 por otros que sean variables. Lo mismo podríamos hacer con los procedimientos que dibujan cuadrados, pentagonos, etc. Ello nos permitirá definir un procedimiento que dibuje polígonos y que nos sirva para todos y cada uno de los ejemplos mencionados:

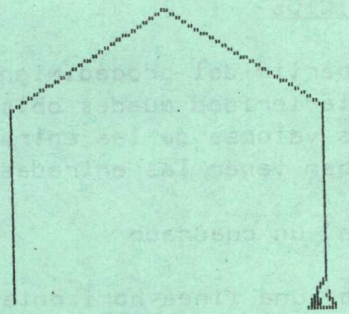
```
PARA POLIGONO :LADO :ANGULO :NUMERO.LADOS
REPITE :NUMERO.LADOS [AV :LADO GD :ANGULO]
FIN
```

Que podemos ejecutar asi:

```
? POLIGONO 60 60 6
```

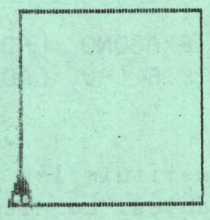


? POLIGONO 60 60 4

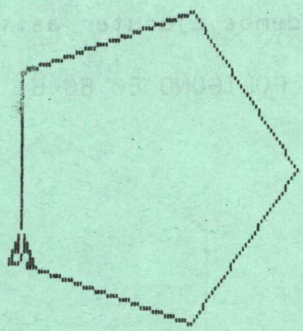


Hemos intentado hacer un polígono de 4 lados pero vemos que no es suficiente con darle a la variable :LADO el valor 4, tenemos que modificar tambien el valor del ángulo.

? POLIGONO 60 90 4



? POLIGONO 60 72 5



OPERACIONES CON VARIABLES

En estos ejemplos hemos mantenido constante la longitud del lado pero hemos variado el número de lados y al mismo tiempo el ángulo, pues como hemos comprobado en el caso del cuadrado fallido la tortuga debe realizar el TEOREMA DEL VIAJE COMPLETO para realizar un polígono, es decir sumar 360 grados en la realización de todos sus giros.

Podemos observar que en los anteriores procedimientos existe una relación entre el número de lados de cada polígono y el ángulo que gira la tortuga, que podemos formular asi:

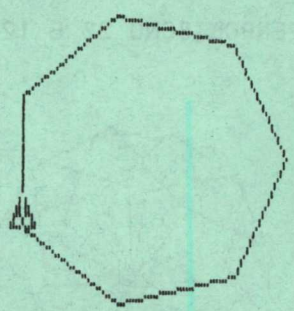
```
:ANGULO * :NUMERO.LADOS = 360.
```

Los valores de las variables pueden relacionarse entre si o con otros mediante operadores matemáticos para obtener otro u otros valores reutilizables. Asi podemos realizar un procedimiento que dibuje polígonos de distinto número de lados, sin necesidad de preocuparnos del ángulo, ya que este nos vendrá dado por la relación 360 / :NUMERO.LADOS.:

```
PARA POLIGONO :LADO :NUMERO.LADOS
REPITE :NUMERO.LADOS [AV :LADO GD 360 / :NUMERO.LADOS]
FIN

EJEMPLO :

? POLIGONO 40 7
```



Podemos tambien transmitir valores de un procedimiento a otro.:

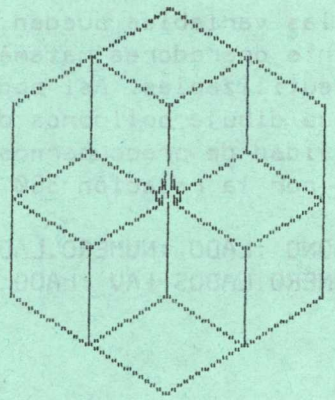
```
PARA SUPERPOLIGONO :LADO :NUMERO.LADOS :VECES
REPITE :VECES [POLIGONO :LADO :NUMERO.LADOS
       GD 360 / :VECES]
```

El procedimiento SUPERPOLIGONO llama al procedimiento POLIGONO y puesto que éste necesita dos valores, para ejecutarse correctamente, usa el contenido de las variables :LADO y :NUMERO.LADOS que figuran en la primera linea de definición.

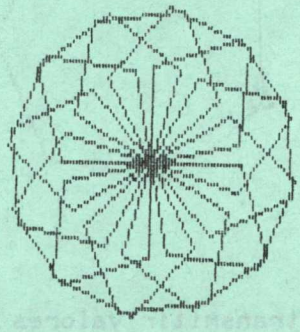
La variable :VECES requiere un valor que indicará el número de veces que se repetirá el polígono girando.

EJEMPLO :

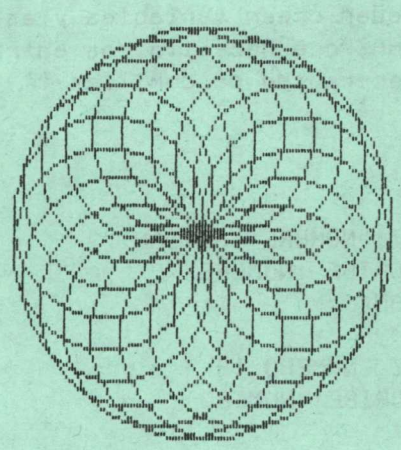
```
? SUPERPOLIGONO 30 6 6
```



```
? SUPERPOLIGONO 20 5 12
```



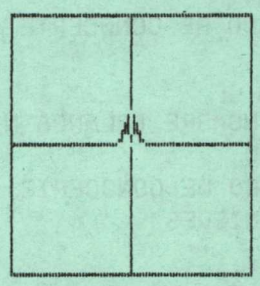
```
? SUPERPOLIGONO 10 20 20
```

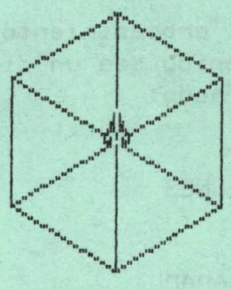


EJERCICIOS

1) Utilizando el procedimiento POLIGONO realiza otro que apile un número variable de poligonos de tamaño tambien variable.

2) Realizar el procedimiento que permita dibujar las siguientes figuras mediante composiciones de cuadrados y de triángulos cuyos lados sean variables.





CREACION DE VARIABLES

Hemos estudiado las variables en lo que se denomina procedimientos con parámetros de entrada. Ahora vamos a ver que se pueden crear variables y asignarles valores mediante la orden HAZ que necesita dos entradas: el nombre de la variable precedido de comillas (") y el valor de la variable.

EJEMPLOS :

```
? HAZ "NOMBRE "ADRIAN
? ESCRIBE :NOMBRE
ADRIAN

? HAZ "NUMERO 18
? ESCRIBE :NUMERO
18
```

Vamos a realizar un procedimiento que requiera la creación de una variable:

```
PARA SALUDO
ESCRIBE [ENCANTADO DE CONOCERTE]
ESCRIBE :NOMBRE
FIN
```

Creamos la variable y ejecutamos el procedimiento:

```
? HAZ "NOMBRE "LUIS
? SALUDO
ENCANTADO DE CONOCERTE
LUIS

? HAZ "NOMBRE [BLANCA NIEVES]
? SALUDO
ENCANTADO DE CONOCERTE
BLANCA NIEVES
```

EJERCICIOS

1) Definir el procedimiento FECHA que tenga una variable :DIA cuyo contenido sea un dia de la semana y que actue del siguiente modo:

```
? FECHA
HOY ES LUNES

? FECHA
HOY ES SABADO
```

2) Definir el procedimiento ENCABEZAMIENTO que tenga dos variables :CIUDAD y :FECHA y que funcione así:

```
? ENCABEZAMIENTO
MADRID 17 de NOVIEMBRE de 1.982

? ENCABEZAMIENTO
LEON 30 de ABRIL de 1.986
```

3) Definir el procedimiento SALUDAR :NOMBRE en el que la variable :NOMBRE no necesita ser creada por aparecer en la primera linea de definición como entrada. Su funcionamiento será:

```
? SALUDAR "ANDRES
BIENVENIDO ANDRES

? SALUDAR [JUAN PEDRO]
BIENVENIDO JUAN PEDRO
```

APROXIMACION AL CONCEPTO DE FUNCION

OBJETIVO GENERAL

La presente unidad está destinada a los alumnos de Ciclo Superior de E.G.B.; por tanto, no se pretende que el alumno capte totalmente el concepto de función que requiere un alto poder de abstracción, simplemente se intenta:

- Aproximar intuituvamente al alumno al concepto de función.

OBJETIVOS ESPECIFICOS

- Manejar las primitivas de LOGO que actúan como funciones en el tratamiento de palabras y listas.

- Realizar las distintas operaciones aritméticas básicas que permite el LOGO.

- Distinguir las funciones según sus imágenes (palabras, listas, verdadero/falso)

- Observar la variación de la imagen de una función al cambiar los antecedentes.

- Inventar funciones sencillas que produzcan una imagen determinada a partir de un antecedente dado.

- Investigar la composición de funciones.

METODOLOGIA

Como ya es sabido, los ambientes de aprendizaje LOGO se prestan a la manipulación directa por parte del alumno sin la realización de los largos y tediosos cálculos que han dado a las matemáticas una tradicional fama de ciencia aburrida.

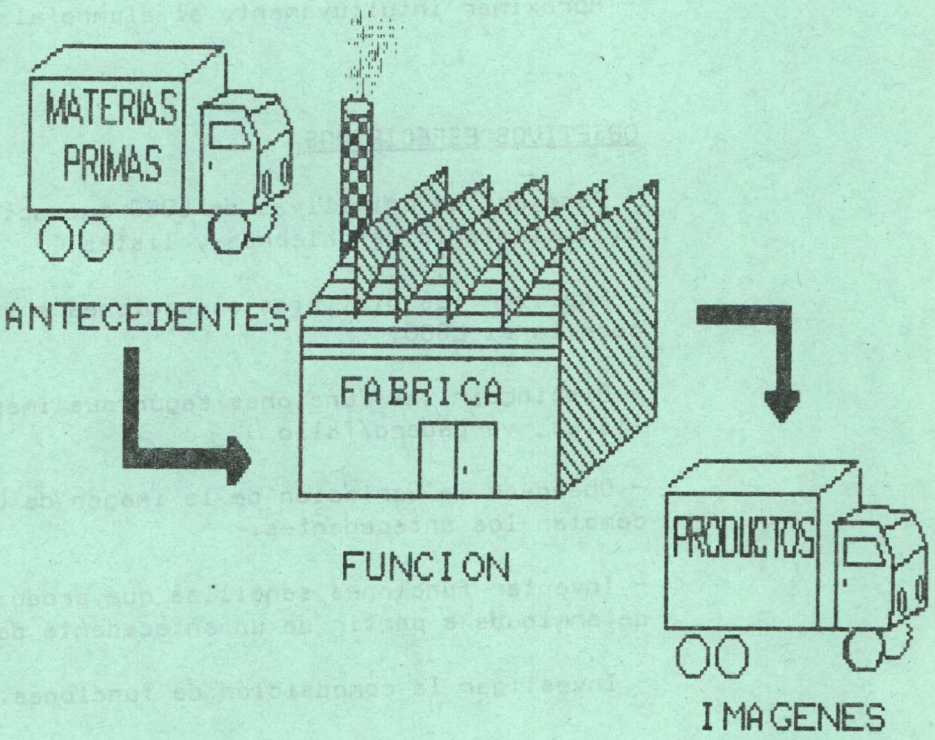
En esta unidad pretendemos que el alumno manipule las "casas-función" y observe el resultado de forma inmediata, sin necesidad de aburrirse en cálculos, y con la posibilidad de que el resultado sea inesperado o erróneo sin temer la más o menos velada reprobación del profesor.

Démosle al alumno las herramientas y las guias de trabajo sin adelantarle los resultado y dejemos que llegue por sus propios medios a una solución válida. Si no lo consigue, ya vendrá a pedirnos ayuda.

CONCEPTO DE FUNCION

Podemos representar de una manera gráfica una función como una "fabrica" que recibe una serie de "materias primas" y las transforma en "productos".

A estas "materias primas", los matemáticos llaman antededentes de la función y a los "productos" los llaman imágenes de la función.



ACTIVIDADES

PALABRAS Y LISTAS

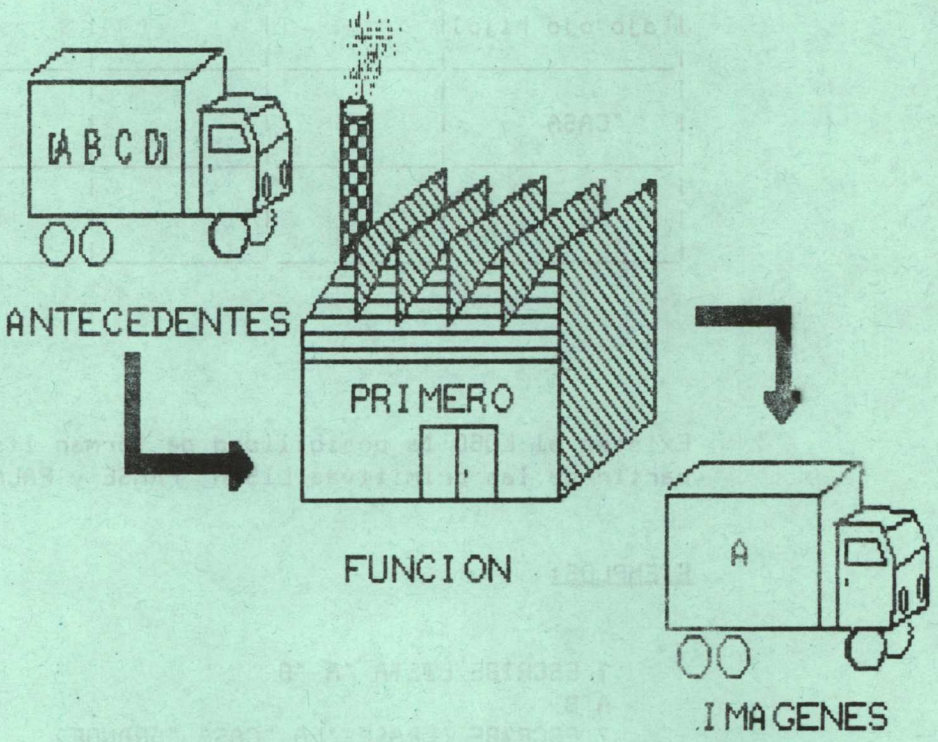
Con las primitivas PRIMERO, ULTIMO, MENOSPRIMERO y MENOSULTIMO se pueden manejar los objetos de LOGO llamados palabras y listas.

EJEMPLOS:

```
? ESCRIBE PRIMERO [A B C D]
A
? ESCRIBE PRIMERO [4 3 2 1]
4
? ESCRIBE PRIMERO "BARCELONA
B
```

```
? ESCRIBE ULTIMO [MI PAPA FUMA EN PIPA]
PIPA
? ESCRIBE MENOSPRIMERO [MI MAMA ME MIMA]
MAMA ME MIMA
? ESCRIBE MENOSULTIMO [CASI LLEGO TARDE]
CASI LLEGO
```

Se puede observar lo que ocurre cuando le pedimos al ordenador que realice lo anterior:



Vemos que en cada momento las imágenes dependen de los elementos dados como antecedentes.

EJERCICIOS

Completa la tabla siguiente:

ANTECEDENTE	PRIMERO	ULTIMO	MP	MU
[A E I O U]				
[1 3 5 7]				
[ajo ojo hijo]				
"CASA				
5				

Existen el LOGO la posibilidad de formar listas y palabras a partir de las primitivas LISTA, FRASE y PALABRA.

EJEMPLOS:

```
? ESCRIBE LISTA "A "B
A B
? ESCRIBE (FRASE "LA "CASA "GRANDE)
LA CASA GRANDE
? ESCRIBE PALABRA "HIERBA "BUENA
HIERBABUENA
```

EJERCICIOS

Construye con la primitiva FRASE distintas listas en las que el sujeto y el verbo se mantengan constantes y el predicado varíe.

OPERACIONES ARITMETICAS

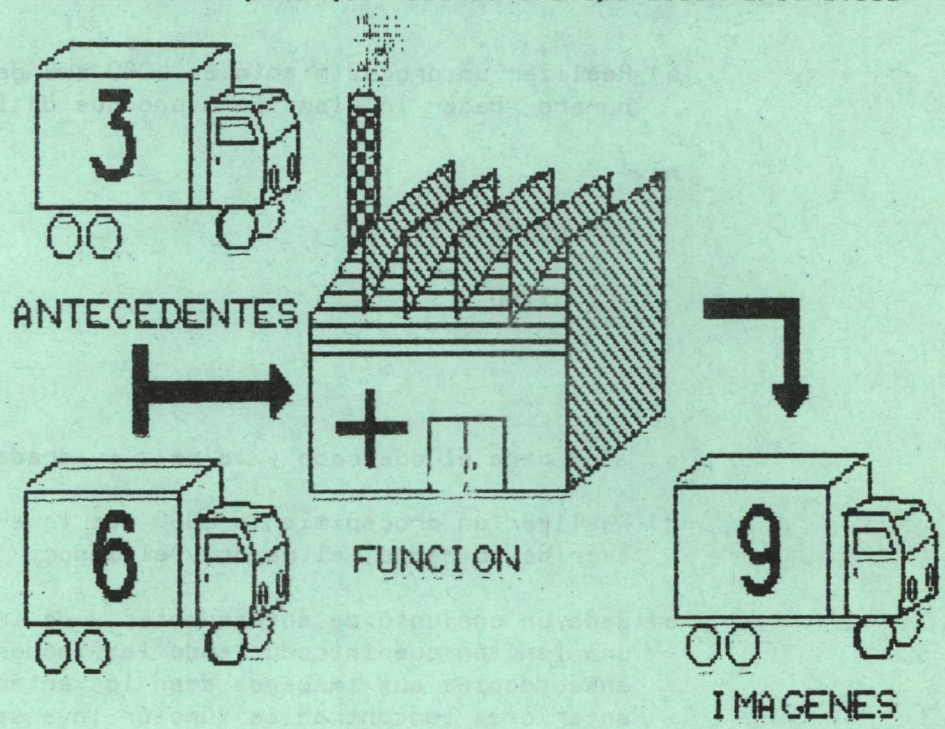
El lenguaje LOGO dispone de primitivas para realizar las operaciones aritméticas básicas; veamos algunos ejemplos:

```
? ESCRIBE SUMA 3 6
9
? ESCRIBE DIFERENCIA 8 2
6
? ESCRIBE PRODUCTO 3 7
21
? ESCRIBE COCIENTE 35 7
5
```

También podriamos haberlo realizado con los signos aritméticos usuales:

```
? ESCRIBE 3 + 6
9
? ESCRIBE 8 - 2
6
? ESCRIBE 3 * 7
21
? ESCRIBE 35 / 7
5
```

Vamos a reflexionar un poco sobre lo que hace el ordenador cuando se le pide que realice una operación aritmética:



Podemos decir que la suma de dos números está en función de ellos mismos, podemos ver como varían las imágenes al variar uno o varios de los antecedentes:

EJERCICIOS

a) Comprobar la variación de la imagen al variar uno de los antecedentes para las primitivas PRODUCTO y COCIENTE:

PRODUCTO

*	ANTEC.	IMAGEN

COCIENTE

/	ANTEC.	IMAGEN

b) Realizar un procedimiento en LOGO que de el doble de un número, hacer lo mismo para uno que de la mitad:

ej.:

```
? DOBLE 8
16
? MITAD 8
4
```

c) Idem para el cuadrado y la raiz cuadrada

d) Realizar un procedimiento LOGO que tome cuatro números y escriba su media, el mayor y el menor.

e) Dada un conjunto de antecedentes y de imágenes, encontrar una función que introduciendo las imágenes anteriores como antecedentes sus imágenes sean los antecedentes anteriores (encontrar la función inversa)

ej.:

COMPLETAR LA TABLA

ANTEC.	FUNCION	IMAGEN	ANTEC.	FUNCION	IMAGEN
4		6	6		4
5	--- ___ ---	7	7	--- ____ ---	5
3		5	5		3

CASA		ASAC	ASAC		CASA
PERRO	-- ___ ---	ORREP	ORREP	--- ___ ---	PERRO
SOLOS		SOLOS	SOLOS		SOLOS

OPERACIONES LOGICAS

Existen multitud de situaciones que sólo pueden tener dos valores: SI/NO, VERDADERO/FALSO, 1/0, ect; a estas situaciones las llamamos "binarias".

Por ejemplo, un interruptor puede estar aabierto o cerrado, un número puede ser, o no ser, mayor mayor que otro, un alumno puede estar dentro o fuera de la clase, etc.

En el siguiete ejemplo se puede comprobar que, dependiendo de la situación de la tortuga, se obtienen distintas respuestas a las preguntas referidas a su localización en DENTRO.PEQUEÑO? (cuadrado pequeño), DENTRO.GRANDE? (cuadrado grande), DENTRO.RECINTO? (espacio comprendido entre los dos cuadrados) y FUERA? (exterior del cuadrado grande). El juego se trata de colocar la tortuga en diferentes posiciones y predecir el resultado de las preguntas referentes a su localización.

Comencemos por definir el procedimiento para dibujar un cuadrado que ya es un viejo amigo nuestro:

```
PARA CUADRADO :LADO
REPITE 4 [AV :LADO GD 90]
FIN
```

Posteriormente definimos un procedimiento que nos dibuje los dos cuadrados con las coordenadas:

```
PARA COORDENADAS
BP
SL PONPOS [-100 -100] BL
CUADRADO 200
SL PONPOS [-50 -50] BL
CUADRADO 100
SL RE 10 BL
PONCL 1
RELLENA
PONCL 3
SL PONPOS [0 0] BL
REPITE 2 [RE 120 AV 240 RE 120 GD 90]
PONRUMBO 0
FIN
```

Ahora definiremos los procedimientos que nos indican si la tortuga está dentro o fuera de los espacios indicados anteriormente:

```
PARA PEQUEÑO?
DEV (Y COORX < 50 COORX > -50 COORY < 50 COORY > -50)
FIN


PARA GRANDE?
DEV (Y COORX < 100 COORX > -100 COORY < 100 COORY >
     -100)
FIN

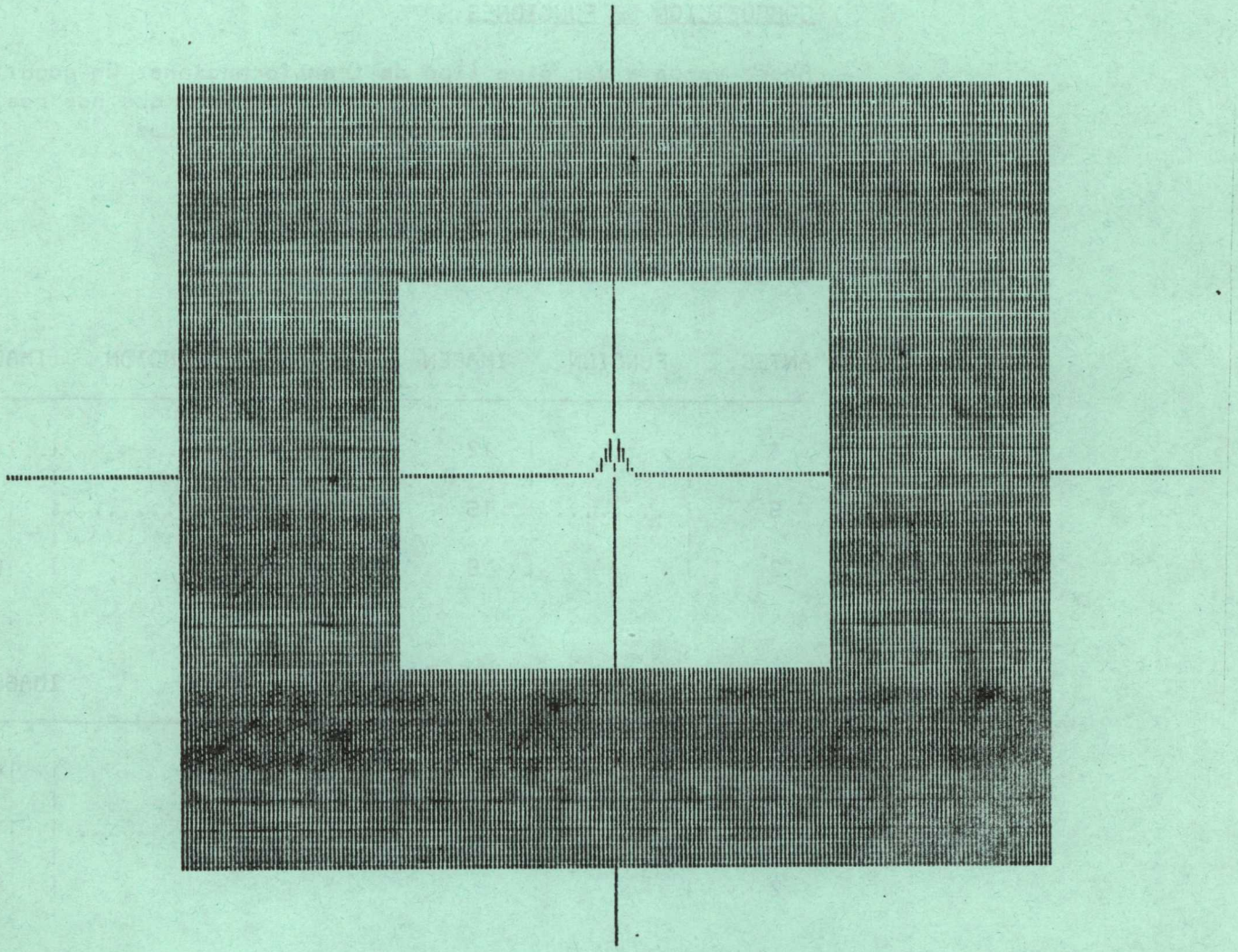

PARA RECINTO?
DEV Y GRANDE? NO PEQUEÑO?
FIN


PARA DENTRO.PEQUEÑO?
ES PEQUEÑO?
FIN


PARA DENTRO.GRANDE?
ES GRANDE?
FIN


PARA DENTRO.RECINTO?
ES RECINTO?
FIN


PARA FUERA?
ES NO GRANDE?
FIN
```



El profesor podrá hacer reflexionar a los alumnos sobre las posibilidades de los distintos valores de las respuestas:

ej.:

Situar la tortuga en una posición que cumpla los resultados de la tabla:

PEQUEÑO	GRANDE	RECINTO	FUERA	
C	C	F	F	
F	C	C	F	
C	C	C	F	!!
F	F	F	C	
F	F	F	F	!!

COMPOSICION DE FUNCIONES

Ahora vamos a ver otro tipo de transformaciones un poquito más complejas. Intentamos descubrir procesos que nos realicen de una sola vez lo mismo que otros dos seguidos.

EJEMPLOS:

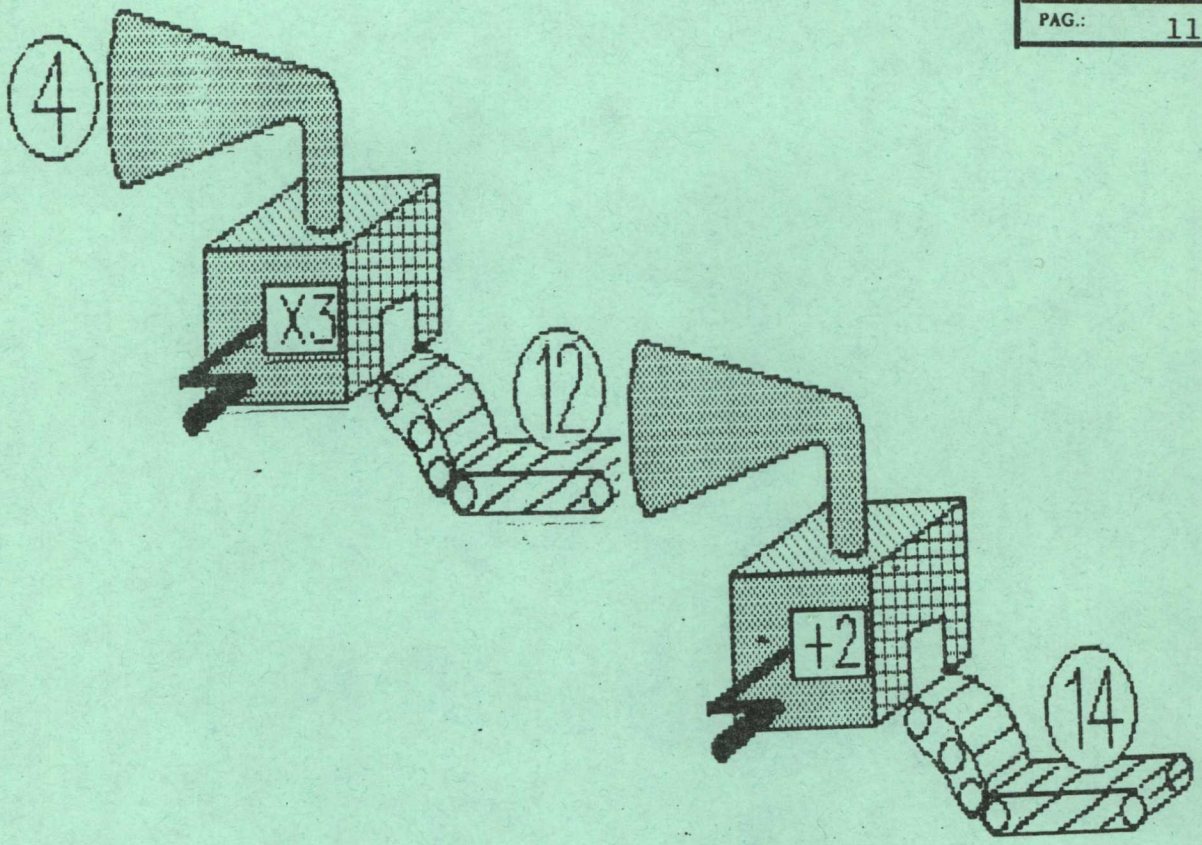
Completa las siguientes tabla:

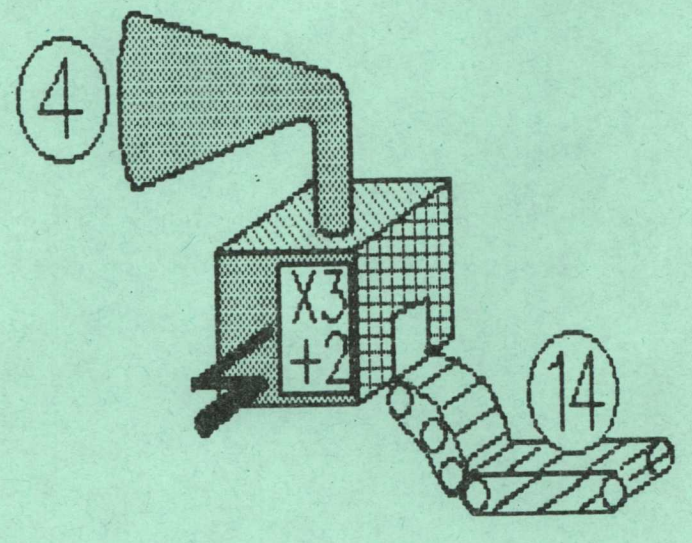
ANTEC.	FUNCION	IMAGEN	ANTEC.	FUNCION	IMAGEN
4		12	12		14
5		15	15		17
2		6	6		8

ANTEC.	FUNCION	IMAGEN
4		14
5		17
2		8

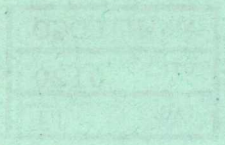
ANTEC.	FUNCION	IMAGEN	ANTEC.	FUNCION	IMAGEN
MESA		ESA	ESA		ES
VASO		ASO	ASO		AS
LUNA		UNA	UNA		UN

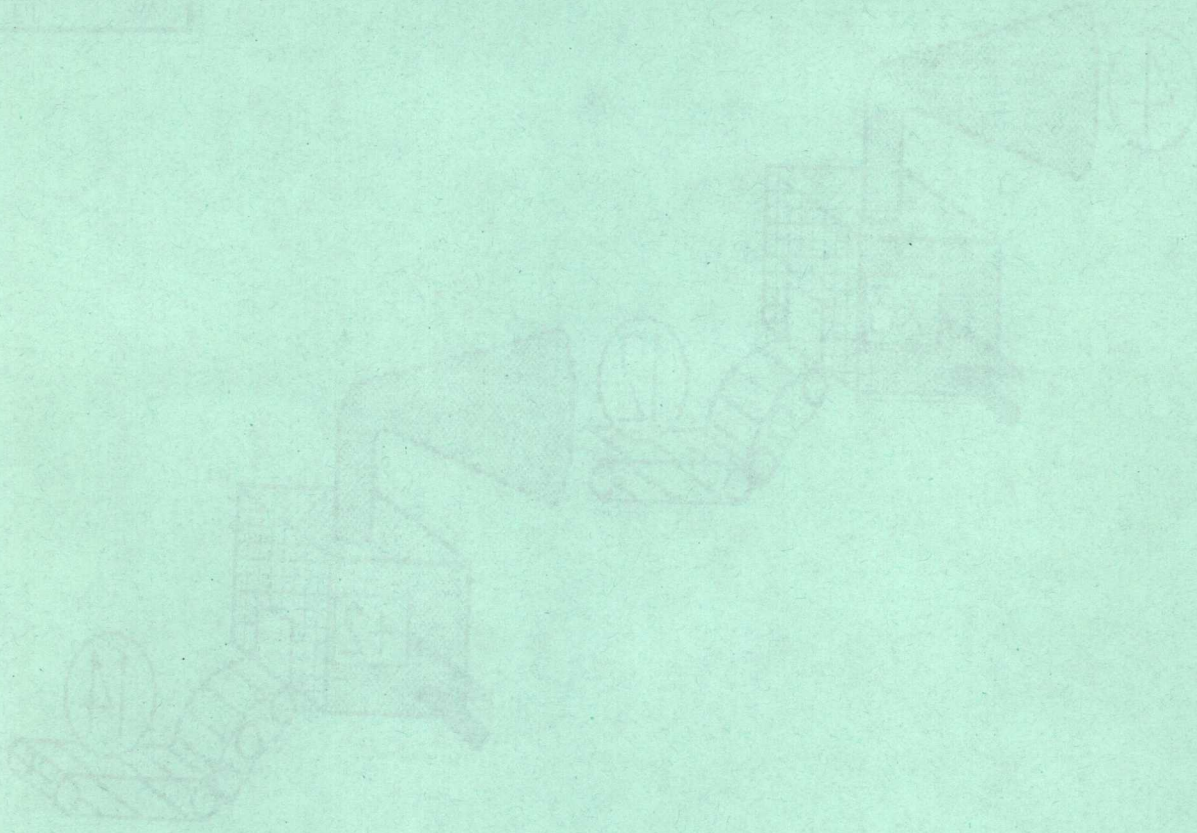
ANTEC.	FUNCION	IMAGEN
MESA		ES
VASO		AS
LUNA		UN

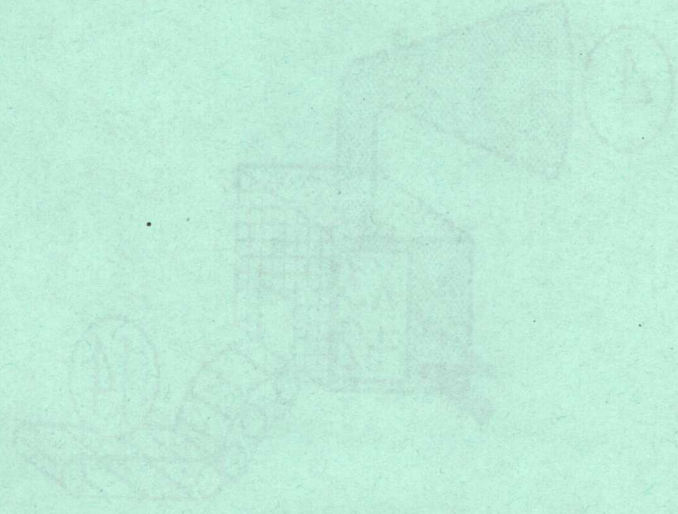




Como verás, una imagen puede ser a su vez el antecedente de otra función; cuando ocurre esto, podemos buscar una función que produzca el mismo efecto que las otras dos juntas, a esto se llama composición de funciones y a la función que resume el efecto de las otras dos la llamamos función compuesta.







A P L I C A C I O N

LA TORTUGA SALE DE UN LABERINTO

Con ligeras modificaciones los procedimientos anteriores nos servirán para sacar a la tortuga de un laberinto. Lo realizaremos desde el principio como modelo y que pretende lograr los siguientes objetivos al plantearse en un aula LOGO.

Objetivos Didácticos

- Mostrar un ejemplo de programación modular en LOGO ameno y atractivo en el que se manejan gráficos de tortuga y listas conjuntamente.

- Introducir el concepto de "memoria"

Planteamiento del problema

Realizaremos ahora un ejemplo de algo más complicado, en el que pretendemos introducir una tortuga en un laberinto comandándola desde el teclado con órdenes sencillas (A para adelante, D para girar 90º a la derecha e I para girar 90º a la izquierda), para que seguidamente consiga salir por sí misma del laberinto.

En este caso, como se deduce del propio enunciado, debemos dotar de una cierta "inteligencia" a la tortuga, si bien en este caso basta con que "recuerde" el camino que se le ha hecho seguir.

Resolución del problema

Dividiremos el problema en partes y después desarrollaremos c a una de las partes.

El primer procedimiento necesario será la creación del LABERINTO, y es el más simple, ya que basta con hacer recorrer a la tortuga subiendo y bajando plumilla en los lugares adecuados, las paredes del laberinto deseado. Seguidamente situamos la tortuga en un punto cualquiera fuera del laberinto, y en una lista que llamaremos "TRAZADO, iremos almacenando las coordenadas por las que va pasando.

El procedimiento principal TORTUGACONMEMORIA se compone pues, de LABERINTO unas órdenes de iniciación para poner la tortuga en una posición inicial, e iniciar la lista "TRAZADO, y de unos procedimientos para entrar en el laberinto y salir de él que llamaremos "ENTRELABERINTO y "SALIRLABERINTO respectivamente.

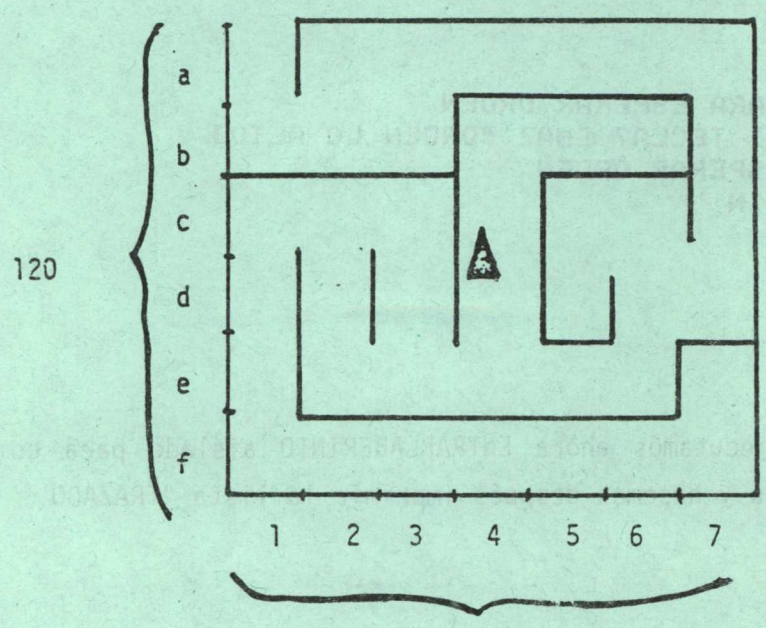
```
PARA TORTUGA.CON.MEMORIA
LABERINTO
SL PONPOS [-100 80]
HAZ "TRAZADO []
ENTRAR.LABERINTO
SALIR.LABERINTO
FIN
```

El procedimiento LABERINTO hace que la tortuga dibuje un laberinto en la pantalla.

```
PARA LABERINTO
AV 60
GD 90 AV 60
GD 90 AV 40
RE 20
GD 90 AV 40
GI 90 AV 40
GI 90 AV 20
GI 90 AV 20
SL PONPOS [-60 80] PONR 180
BL AV 120
GI 90 AV 140
GI 90 AV 120
GI 90 AV 120
GI 90 AV 20
SL AV 40
```

```
BL AV 40
GI 90 AV 100
GI 90 AV 20
GD 90 AV 20
SL PONPOS [-60 40] PONR 90
BL AV 60
SL PONPOS [-20 20]
PONR 180
BL AV 20
FIN
```

Al ejecutar "LABERINTO aparecerá el laberinto de la figura:



ENTRARLABERINTO

El procedimiento "ENTRARLABERINTO introduce la tortuga en el laberinto comando desde el teclado, y además genera la lista con las sucesivas posiciones de la tortuga, mediante la orden:

```
HAZ "TRAZADO PU POS :TRAZADO
```

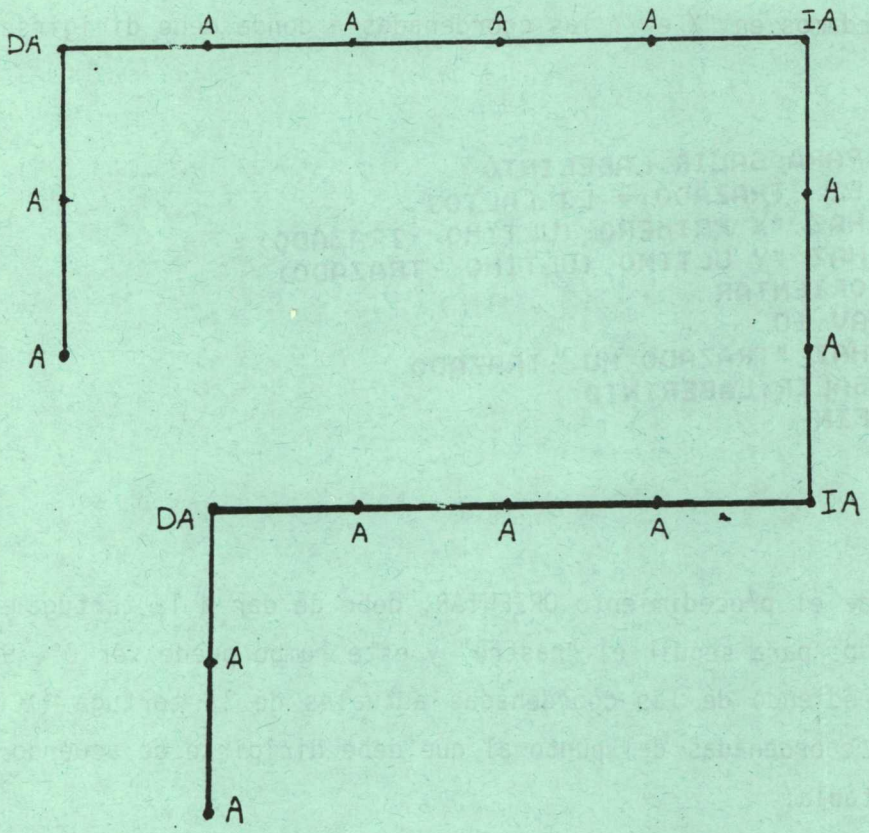
que añade cada vez las nuevas coordenadas de la tortuga a la lista "HILO. Naturalmente esto lo realiza solamente cuando cambian las coordenadas, es decir cuando se le da la orden de avanzar.

El programa completo es:

```
PARA ENTRAR.LABERINTO
ESPERAR.ORDEN
SI :ORDEN = "A [AV 10 HAZ "TRAZADO PU POS :TRAZADO]
SI :ORDEN = "D [GD 90]
SI :ORDEN = "I [GI 90]
SI :ORDEN = "P [ALTO]
ENTRAR.LABERINTO
FIN
```

```
PARA ESPERAR.ORDEN
SI TECLA? [HAZ "ORDEN LC ALTO]
ESPERAR.ORDEN
FIN
```

Si ejecutamos ahora ENTRARLABERINTO aislado para comprobar su funcionamiento y hacemos después imprimir la lista "TRAZADO



?MUESTRA : TRAZADO

[[0 10] [0 20] [10 20] [20 30] [30 20] [40 20] [40 30] [40 40] [40 50]
[30 50] [20 50] [10 50] [0 50] [-10 50] [-10 40] [-10 30]]

SALIRLABERINTO

El procedimiento "SALIRLABERINTO, ha de seguir el "rastro" dejado en la lista "TRAZADO por el procedimiento anterior y hacer avanzar a la tortuga en la dirección adecuada, y para ello se va recordando la lista "TRAZADO empezando por las últimas coordenadas, y así vuelve recursivamente a llamar al propio procedimiento "SALIRLABERINTO.

Guardamos en "X e "Y las coordenadas a donde debe dirigirse la turtuga.

```
PARA SALIR.LABERINTO
SI :TRAZADO = [] [ALTO]
HAZ "X PRIMERO (ULTIMO :TRAZADO)
HAZ "Y ULTIMO (ULTIMO :TRAZADO)
ORIENTAR
AV 10
HAZ "TRAZADO MU :TRAZADO
SALIR.LABERINTO
FIN
```

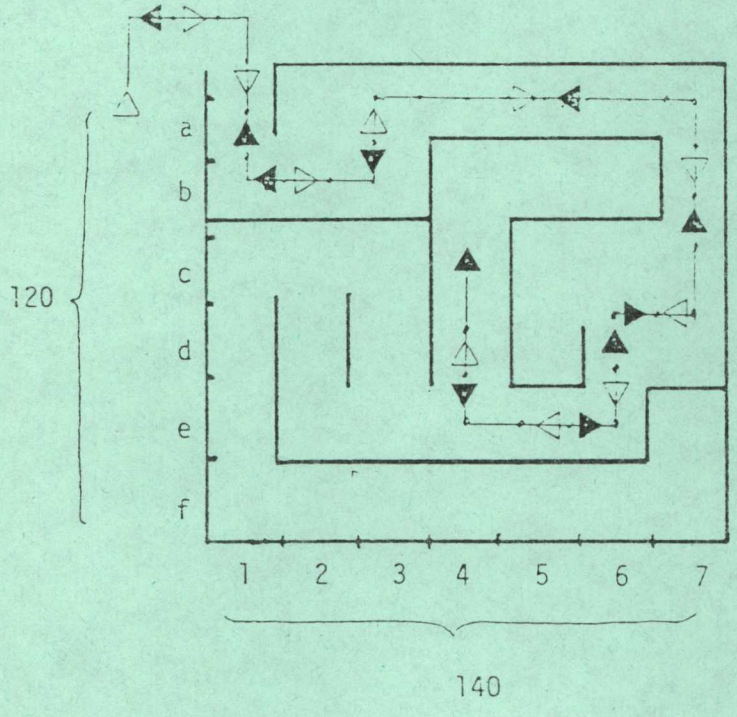
Donde el procedimiento ORIENTAR, debe de dar a la tortuga el rumbo adecuado para seguir el "rastro" y este rumbo puede ser 0°, 90°, 180° ó 270° dependiendo de las coordenadas actuales de la tortuga (XCOR e YCOR) y de las coordenadas del punto al que debe dirigirse de acuerdo con la siguiente tabla:

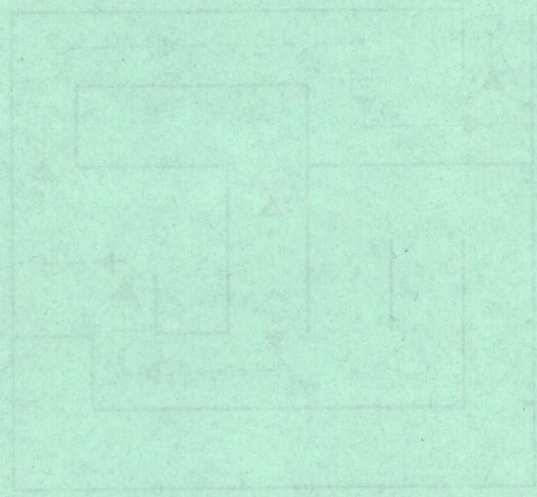
	COORY = Y	COORY > Y	COORY < Y
COORX = X	----	180°	0°
COORX > X	270°	---	----
COORY < X	90°	---	----

Y el procedimiento "ORIENTAR será:

```
PARA ORIENTAR
SI :X = COORX [ (SI :Y > COORY [PONR 0]) (SI :Y < COORY [PONR
   180])]
SI :Y = COORY [ (SI :X > COORX [PONR 90]) (SI :X < COORX
   [PONR 270])]
FIN
```

Y ejecutando el procedimiento TORTUGACONMEMORIA se obtendrá la salida de la gráfica.


a
b
c
d
e
f
120
1
2
3
4
5
6
7
140



POLIGONOS ANIDADOS

Desarrollaremos en este apartado un "micromundo" de gran poder estético y geométrico, como es el de dibujar polígonos anidados, es decir, trazar figuras poligonales que se inbrican unas dentro de otras de forma que si partimos de los vértices del polígono inicial, se dibujará el mismo, y a partir de los vértices dados, calcula otros nuevos, que por ejemplo estén en los centros de los lados del polígono original y el ñuevo polígono así formado sirve de base para realizar la misma operación y así sucesivamente. En la figura se muestra la gráfica obtenida para un cuadrado con un nivel de anidamientos de 10 y tomando el punto medio.

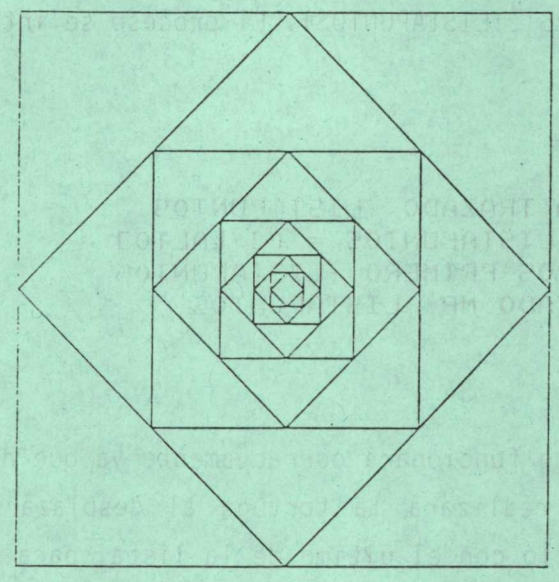


Fig. 1: Anidamiento de cuadrados

Veremos cómo variando los puntos, el número de lados del polígono y el criterio de anidamiento, realizaremos diseños que nos introducirán en conceptos geométricos de alto nivel, de un modo exploratorio y por descubrimientos sucesivos.

Dibujo de un polígono

Desarrollaremos en primer lugar un procedimiento que realice el dibujo de un polígono cualquiera a partir de sus vértices, dados en una lista de coordenadas. Por ejemplo, la lista de puntos de un cuadrado podría ser:

[[100 100] [100 -100] [-100 -100] [-100 100]]

Para realizar la operación de unir los vértices dibujando el trazado, desarrollamos el procedimiento TRAZADO, cuya entrada será la lista de vértices dada por LISTAPUNTOS.

El procedimiento lo haremos recursivo de forma que va colocando la tortuga en el primer punto de la lista de vértices con la orden:

PONPOS PRIMERO :LISTAPUNTOS

para, a continuación, llamarse a sí mismo con la lista excepto su primer elemento (BUT-FIRST :LISTAPUNTOS). El proceso se interrumpe cuando la lista está vacía.

```
PARA TRAZADO :LISTAPUNTOS
SI :LISTAPUNTOS = [] [ALTO]
PONPOS PRIMERO :LISTAPUNTOS
TRAZADO MP :LISTAPUNTOS
FIN
```

El programa aún no funcionará correctamente ya que hay que eliminar el primer trazado que realizará la tortuga al desplazarse al primer vértice y además falta unirlo con el último de la lista, para ello definimos POLIGONO un procedimiento que basándose en TRAZADO, dibuja ya correctamente el polígono completo.

```
PARA POLIGONO :LISTAPUNTOS
SL PONPOS ULTIMO :LISTAPUNTOS BL
TRAZADO :LISTAPUNTOS
FIN
```

Comprobemos su funcionamiento para, por ejemplo, una lista de 8 puntos, y nos dibuja el octógono de la figura:

```
?POLIGONO [[100 100] [120 0] [80 -100] [20 -112] [-90 -90] [-120 -20] [-90 85] [10 110]]
```

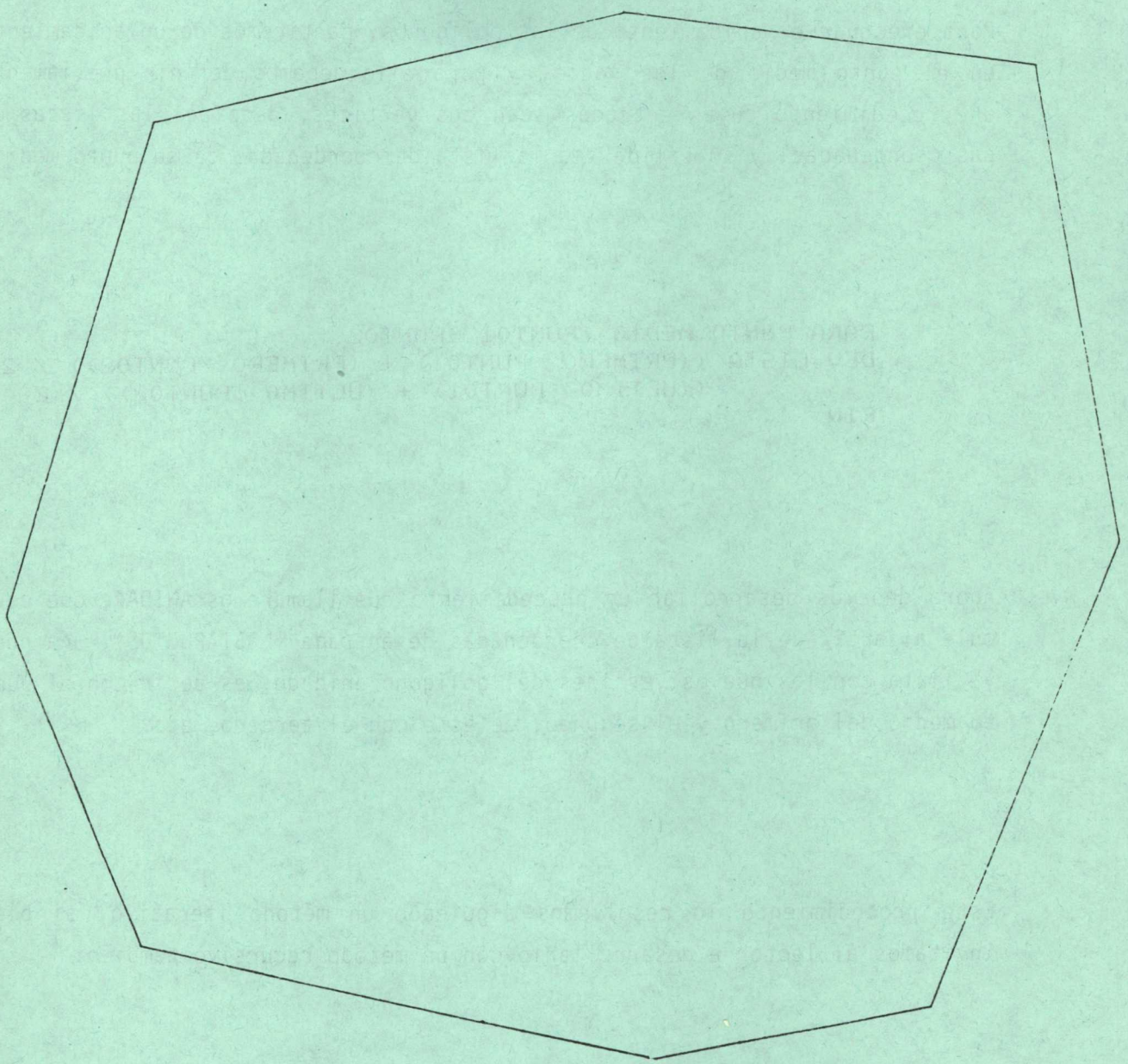


Fig. 2: Octógono

Procedimiento FIGURA

Para efectuar el anidamiento de los polígonos, partiremos de un anidamiento en el punto medio de los lados, y para ello debemos definir previamente un procedimiento cuyas entradas sean dos vértices, dado por las listas de sus coordenadas, y su salida sea la lista de coordenadas de su punto medio.

```
PARA PUNTO.MEDIO :PUNTO1 :PUNTO2
DEV LISTA ((PRIMERO :PUNTO1) + (PRIMERO :PUNTO2)) / 2
          ((ULTIMO :PUNTO1) + (ULTIMO :PUNTO2)) / 2
FIN
```

Ahora debemos desarrollar un procedimiento que llamaremos ANIDAR, que calcule a partir de la lista de coordenadas de entrada (LISTAPUNTOS), una nueva lista con los nuevos vértices del polígono anidado, es decir con el punto medio del primero y el segundo, de éste con el tercero, etc.

Este procedimiento lo resolvemos siguiendo un método iterativo, si bien invitamos al lector a desarrollarlo con un método recursivo también.

A partir de una lista vacía (LISTA), vamos añadiendo al final de la misma las sucesivas coordenadas, que son el punto medio de los items 1 y 2, 2 y 3 y así sucesivamente hasta N-1 y N, siendo N el número de vértices, así pues la orden:

```
HAZ "LISTA PU (PUNTO.MEDIO ELEMENTO :I :LISTAPUNTOS ELEMENTO
     ( :I + 1) :LISTAPUNTOS) :LISTA
```

y esto repetido tantas veces como vértices haya menos uno, variando el valor de I cada vez.

```
REPITE (CUENTA :LISTAPUNTOS) - 1 [.....]
```

Finalmente, añadimos al principio de la lista el punto medio del primero y último vértice y damos la lista resultante como salida del procedimiento.

El listado se da a continuación:

```
PARA ANIDAR :LISTAPUNTOS
(LOCAL "I "LISTA)
HAZ "I 1
HAZ "LISTA []
REPITE (CUENTA :LISTAPUNTOS) - 1 [HAZ "LISTA PU (PUNTO.MEDIO
       ELEMENTO :I :LISTAPUNTOS ELEMENTO ( :I + 1)
       :LISTAPUNTOS ) :LISTA HAZ "I :I + 1]
HAZ "LISTA PP (PUNTO.MEDIO PR :LISTAPUNTOS UL :LISTAPUNTOS)
     :LISTA
DEV :LISTA
FIN
```

Comprobemos su correcto funcionamiento, para la lista de puntos anterior, del procedimiento ANIDAR.

```
?MUESTRA ANIDAR [[100 100][120 0][80 -100][20 -112][-90 -90][-120 -20][-90
  85][10 110]]
[[55 105][110 50][100 -50][50 -106][-35 -101][-105 -55][-105 32.5][-40 97.5]]
```

Para culminar el proceso sólo nos queda remetir el número de veces que deseemos, según el nivel de anidamiento a realizar el procedimiento POLIGONO, cambiando cada vez la lista de vértices por la nueva obtenida del procedimiento ANIDAR.

```
PARA FIGURA :LISTAPUNTOS :NIVEL
REPITE :NIVEL [POLIGONO :LISTAPUNTOS HAZ "LISTAPUNTOS ANIDAR
       :LISTAPUNTOS]
FIN
```

Ejecutando para el caso del cuadrado, obtenemos el dibujo de la figura 1.

```
?FIGURA [[100 100][100 -100][-100 -100][-100 100]] 10
```

y para la lista de puntos del octógono con un nivel 40, obtenemos la gráfica de la figura 3.

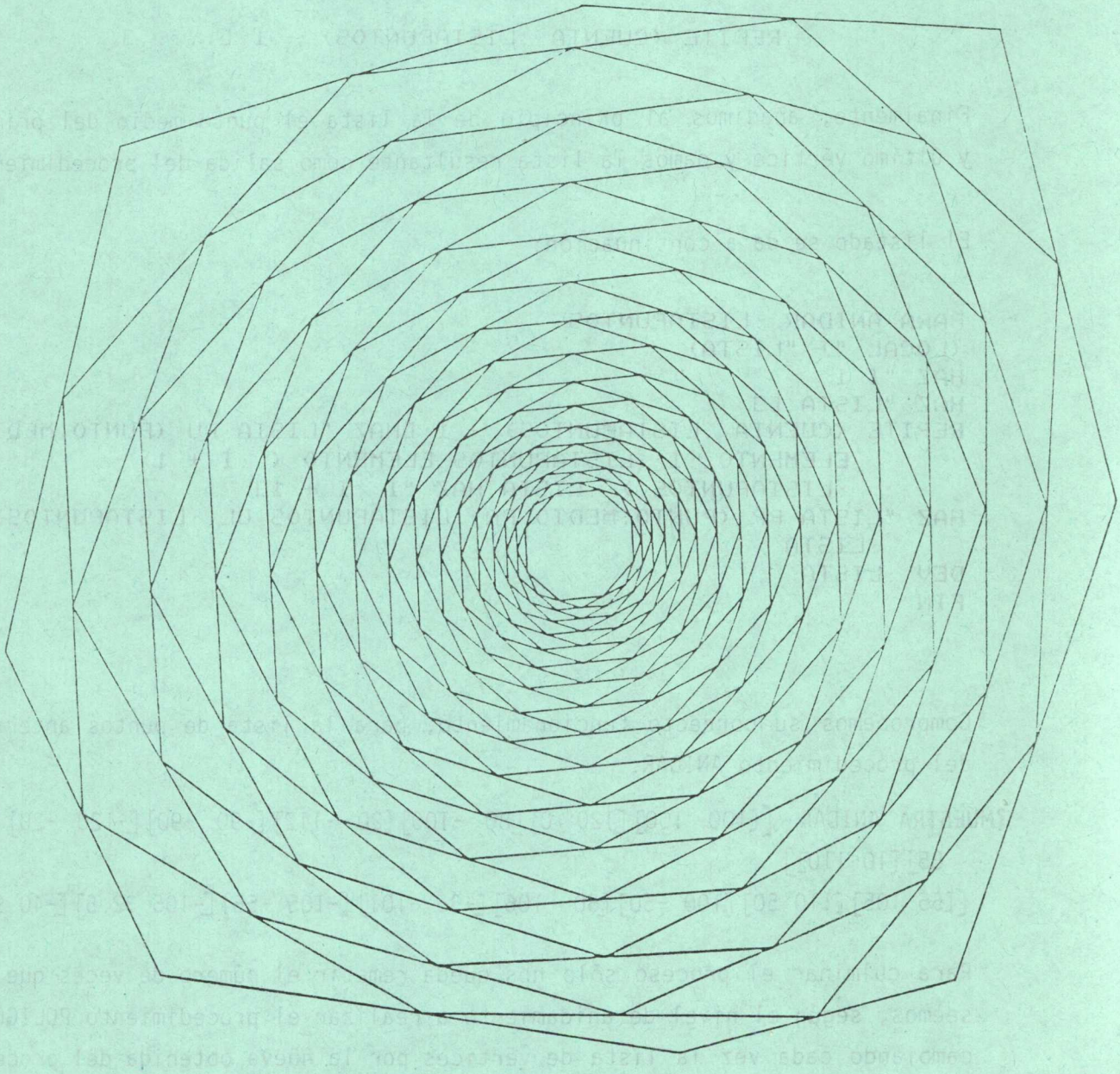


Fig. 3: Octógonos Anidados

Se listan a continuación juntos, todos los procedimientos desarrollados.

```
PARA PUNTO.MEDIO :PUNTO1 :PUNTO2
DEV LISTA ((PRIMERO :PUNTO1) + (PRIMERO :PUNTO2)) / 2
          ((ULTIMO :PUNTO1) + (ULTIMO :PUNTO2)) / 2
FIN


PARA TRAZADO :LISTAPUNTOS
SI :LISTAPUNTOS = [] [ALTO]
PONPOS PRIMERO :LISTAPUNTOS
TRAZADO MP :LISTAPUNTOS
FIN


PARA POLIGONO :LISTAPUNTOS
SL PONPOS ULTIMO :LISTAPUNTOS BL
TRAZADO :LISTAPUNTOS
FIN


PARA FIGURA :LISTAPUNTOS :NIVEL
REPITE :NIVEL [POLIGONO :LISTAPUNTOS HAZ "LISTAPUNTOS ANIDAR
       :LISTAPUNTOS]
FIN


PARA ANIDAR :LISTAPUNTOS
(LOCAL "I "LISTA)
HAZ "I 1
HAZ "LISTA []
REPITE (CUENTA :LISTAPUNTOS) - 1 [HAZ "LISTA PU (PUNTO.MEDIO
       ELEMENTO :I :LISTAPUNTOS ELEMENTO ( :I + 1)
       :LISTAPUNTOS ) :LISTA HAZ "I :I + 1]
HAZ "LISTA PP (PUNTO.MEDIO PR :LISTAPUNTOS UL :LISTAPUNTOS)
     :LISTA
DEV :LISTA
FIN
```

Otros anidamientos

Podemos ahora introducir variantes al procedimiento anterior de forma que en lugar de unir los puntos medios de los vértices, una, por ejemplo, los puntos que están distando la tercera parte de un lado del vértice y dos terceras partes del otro, lo que nos daría, por ejemplo, para el cuadrado de la figura 1, un gráfico como el de la figura 4.

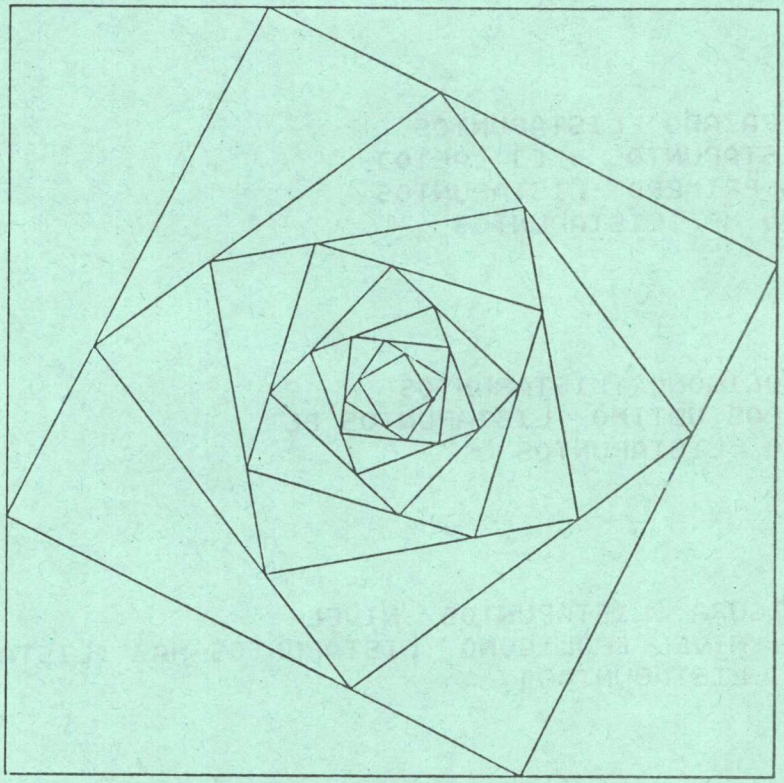


Fig. 4: Cuadrados anidados

Si redefinimos el procedimiento PUNTOMEDIO para que nos dé el punto deseado

```
PARA PUNTO.MEDIO :PUNTO1 :PUNTO2
DEV LISTA ((PRIMERO :PUNTO1) + (PRIMERO :PUNTO2)) / 3
          ((ULTIMO :PUNTO1) + (ULTIMO :PUNTO2)) / 3
FIN
```

y ejecutamos nuevamente para la lista de puntos del octógono, obtenemos la gráfica de la figura 5.

?FIGURA [[100 100] [120 0] [80 -100] [20 -112] [-90 -90] [-120 -20] [-90 85] [10 110]]

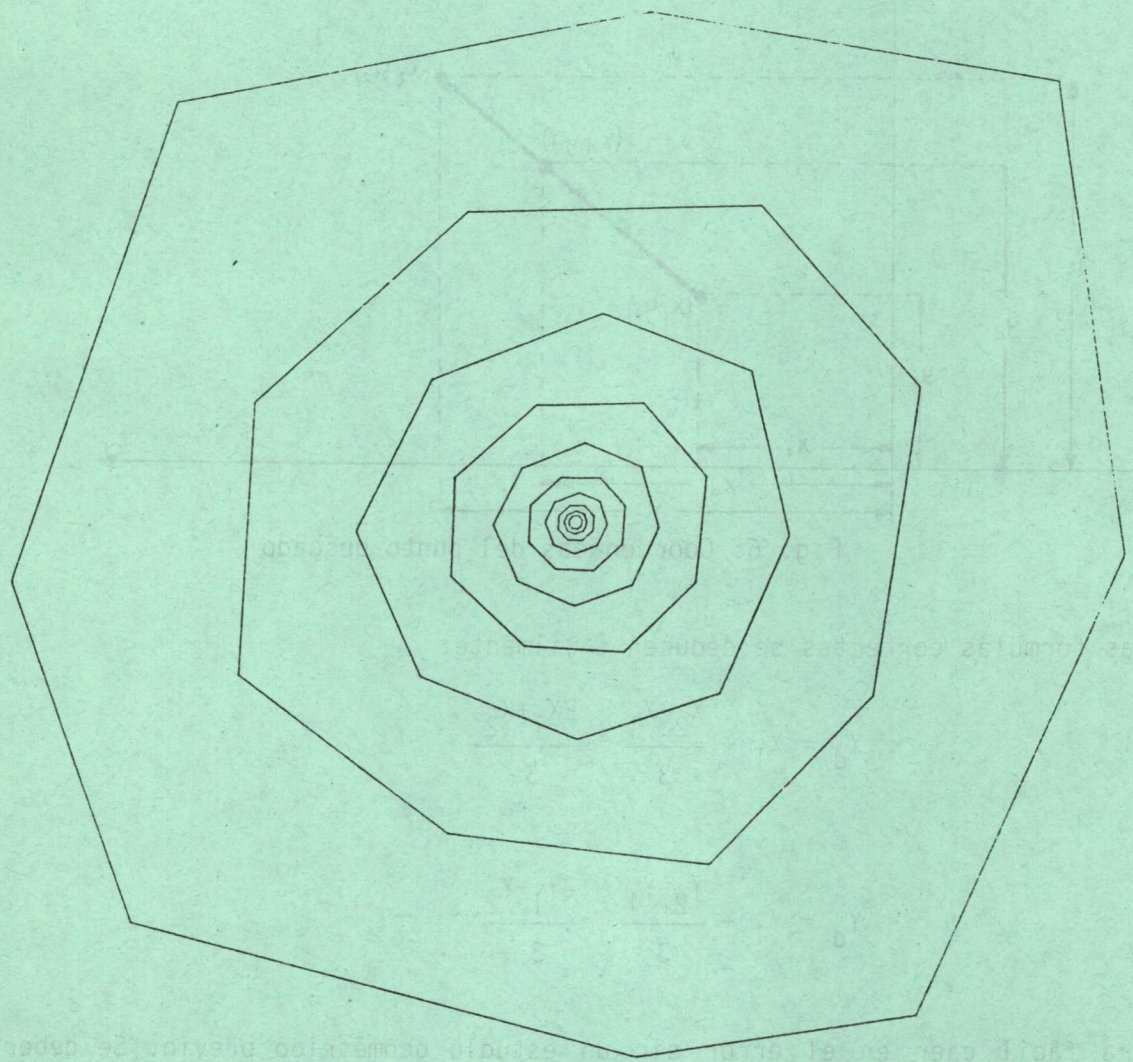


Fig. 5: Anidamiento de octógonos

Algo no funciona correctamente, hay un error,la figura obtenida no es la deseada. Bien, el posible error tal vez esté en el cálculo del punto para hallar los nuevos vértices. Comprobémoslo:

?MUESTRA PUNTOMEDIO [10 10] [40 40]

[16.66666667 16.66666667]

En efecto, éste no es el punto deseado, haremos un análisis geométrico más depurado. En la figura 6 se muestran las coordenadas de los extremos de un segmento y el punto deseado (X_d Y_d).

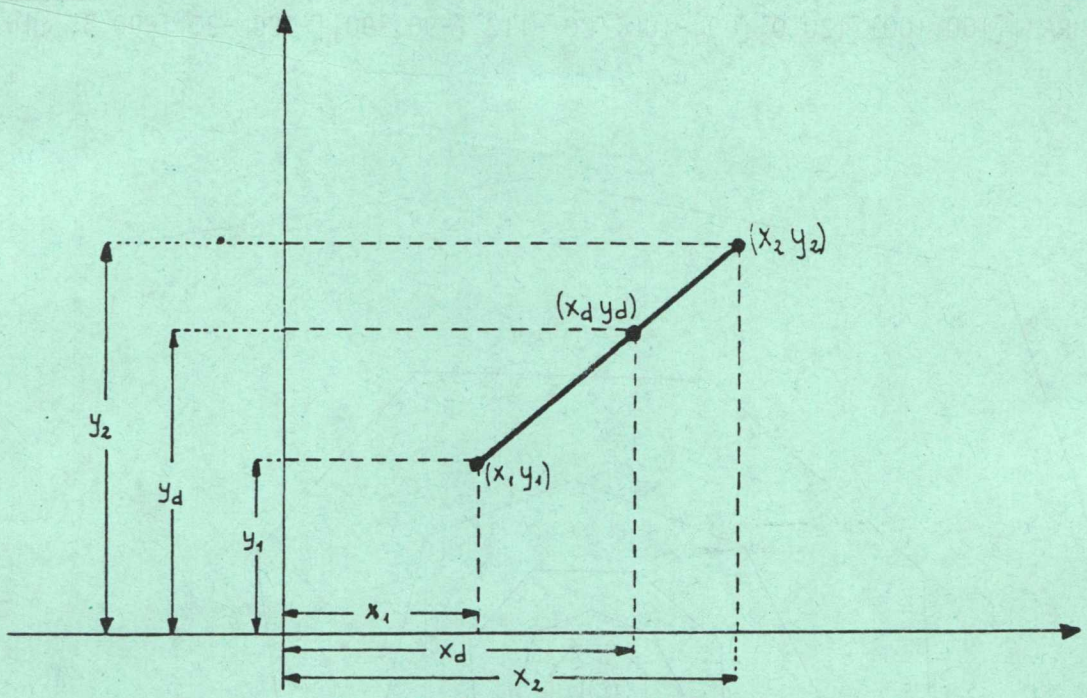


Fig. 6: Coordenadas del punto buscado

Las fórmulas correctas se deducen fácilmente:

$$X_d = X_1 + \frac{X_2 - X_1}{3} = \frac{2X_1 + X_2}{3}$$

$$Y_d = Y_1 + \frac{Y_2 - Y_1}{3} = \frac{2Y_1 + Y_2}{3}$$

Era fácil caer en el error sin un estudio geométrico previo. Se debería haber probado el procedimiento PUNTOMEDIO antes de realizar la figura, pero no importa, el error ya sabemos que forma una parte importante en el proceso de aprendizaje con LOGO.

Aprovecharemos ya para en lugar de tomar el punto a un tercio del lado, lo tomaremos a un factor variable 1/K, y así, el que buscamos ahora será un caso particular. Las fórmulas generales serán:

$$X_d = ((K-1)X_1 + X_2)/K$$

$$Y_d = ((K-1)Y_1 + Y_2)/K$$

Y el procedimiento, al que llamaremos ahora PUNTOLADO tiene tres entradas: PUNTO1, PUNTO2 y FACTOR.

```
TO PUNTOLADO :PUNTO1 :PUNTO2 :FACTOR
OP LIST ((FIRST :PUNTO2) + (FIRST :PUNTO1)*( :FACTOR-1))/
  :FACTOR ((LAST :PUNTO2)+(LAST :PUNTO1)*( :FACTOR-1))/ :FACTOR
END
```

Comprobemos que funciona bien antes de seguir:

```
?MUESTRA PUNTOLADO [10 10] [40 40] 3
  [20 20]
```

Modificaremos ligeramente los procedimientos FIGURA, al que llamaremos ahora FIGURA2, y ANIDAR, al que llamaremos ahora ANIDAR2. Ambos han de tener una nueva entrada para el FACTOR.

```
PARA PUNTO.LADO :PUNTO1 :PUNTO2 :FACTOR
DEV LISTA ((PRIMERO :PUNTO2) + (PRIMERO :PUNTO1) * ( :FACTOR
        - 1)) / :FACTOR  ((ULTIMO :PUNTO2) + (ULTIMO
        :PUNTO1) * (:FACTOR - 1)) / :FACTOR
FIN
```

```
PARA FIGURA2 :LISTAPUNTOS :NIVEL :FACTOR
REPITE :NIVEL [POLIGONO :LISTAPUNTOS HAZ "LISTAPUNTOS ANIDAR2
      :LISTAPUNTOS :FACTOR]
FIN
```

```
PARA ANIDAR2 :LISTAPUNTOS :FACTOR
(LOCAL "I "LISTA)
HAZ "I 1
HAZ "LISTA []
REPITE (CUENTA :LISTAPUNTOS) - 1 [HAZ "I ( :I + 1) HAZ "LISTA
      PU (PUNTO.LADO ELEMENTO :I :LISTAPUNTOS ELEMENTO ( :I
      + 1) :LISTAPUNTOS :FACTOR ) :LISTA HAZ "I :I + 1]
HAZ "LISTA PP (PUNTO.LADO PR :LISTAPUNTOS UL :LISTAPUNTOS
    :FACTOR ) :LISTA
DEV :LISTA
FIN
```

Ejecutemos ahora para el octógono de antes.

```
?FIGURA2 [[100 100] [120 0] [80 -100] [20 -112] [-90 -90] [-120 -20] [-90 85]
  [10 110]] 50 3
```

Hemos dado un nuevo perímetro (FACTOR=3). Obtenemos el gráfico de la figura 7.

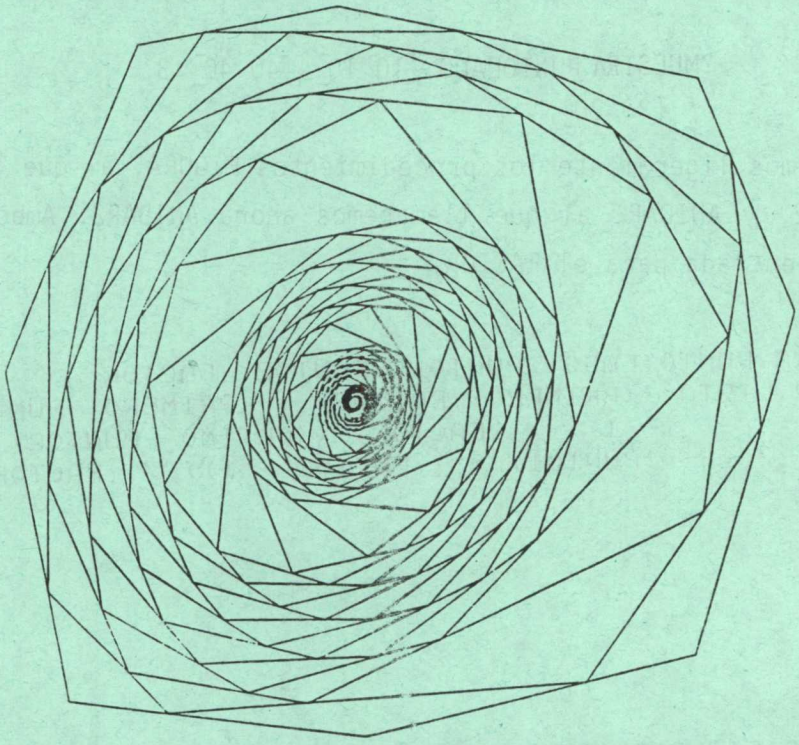


Fig. 7: Anidamiento de octógonos

La figura obtenida parece algo rara, ya que los octógonos van formando una especie de espiral. Esto es algo inesperado, y bonito, pero veamos si hay algún fallo, los octógonos al irse anidando van haciéndose más irregulares e inclinándose. Si observamos la figura con detalle, podemos ver cómo en uno de los lados en cada octógono, los nuevos vértices no están a 1/3, sino a 2/3, y esto va inclinando irregularmente la figura.

Nuevamente hay un error, pero que nos ha conducido a un diseño curioso.

Comprobemos el procedimiento ANIDAR2, ya que los puntos los calculaba correctamente PUNTOLADO. Nuevamente nos damos cuenta de la conveniencia de probar cada procedimiento por separado.

```
?MUESTRA ANIDAR2 [[100 100] [100 -100] [-100 -100] [-100 100]] 3
[[33.33333333 100] [100 33.33333333]
[33.33333333 -100] [-100 -33.33333333]]
```

En efecto, esto es incorrecto, pero antes de resolver el problema, probaremos el resultado que produce sobre el cuadrado. Así, ejecutamos:

```
?FIGURA2 [[100 100] [100 -100] [-100 -100] [-100 100]] 10  3
```

El resultado se muestra en la figura 8.

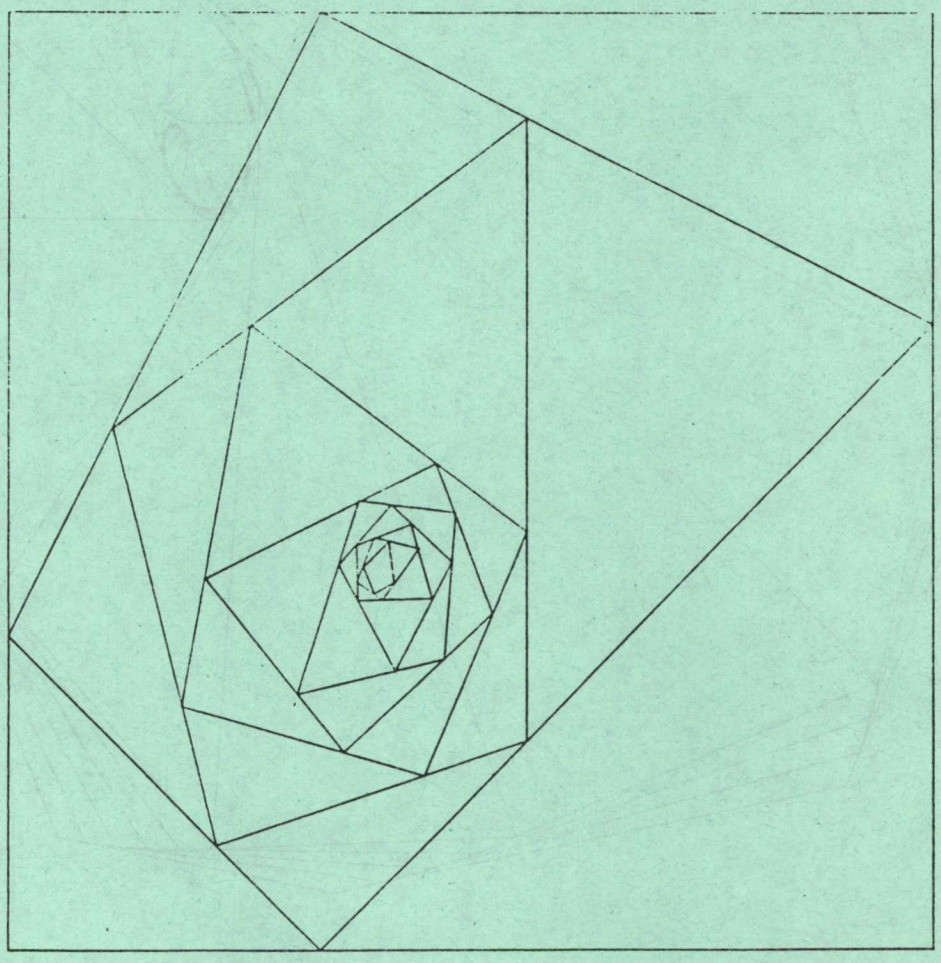


Fig. 8: Cuadrados anidados

Probando ahora en el octógono y con un factor 10:

```
?FIGURA2 [[100 100] [120 0] [80 -100] [20 -112] [-90 -90] [-120 -20] [-90 85]
  [10 110]] 40  10
```

El resultado se muestra en la figura 9.

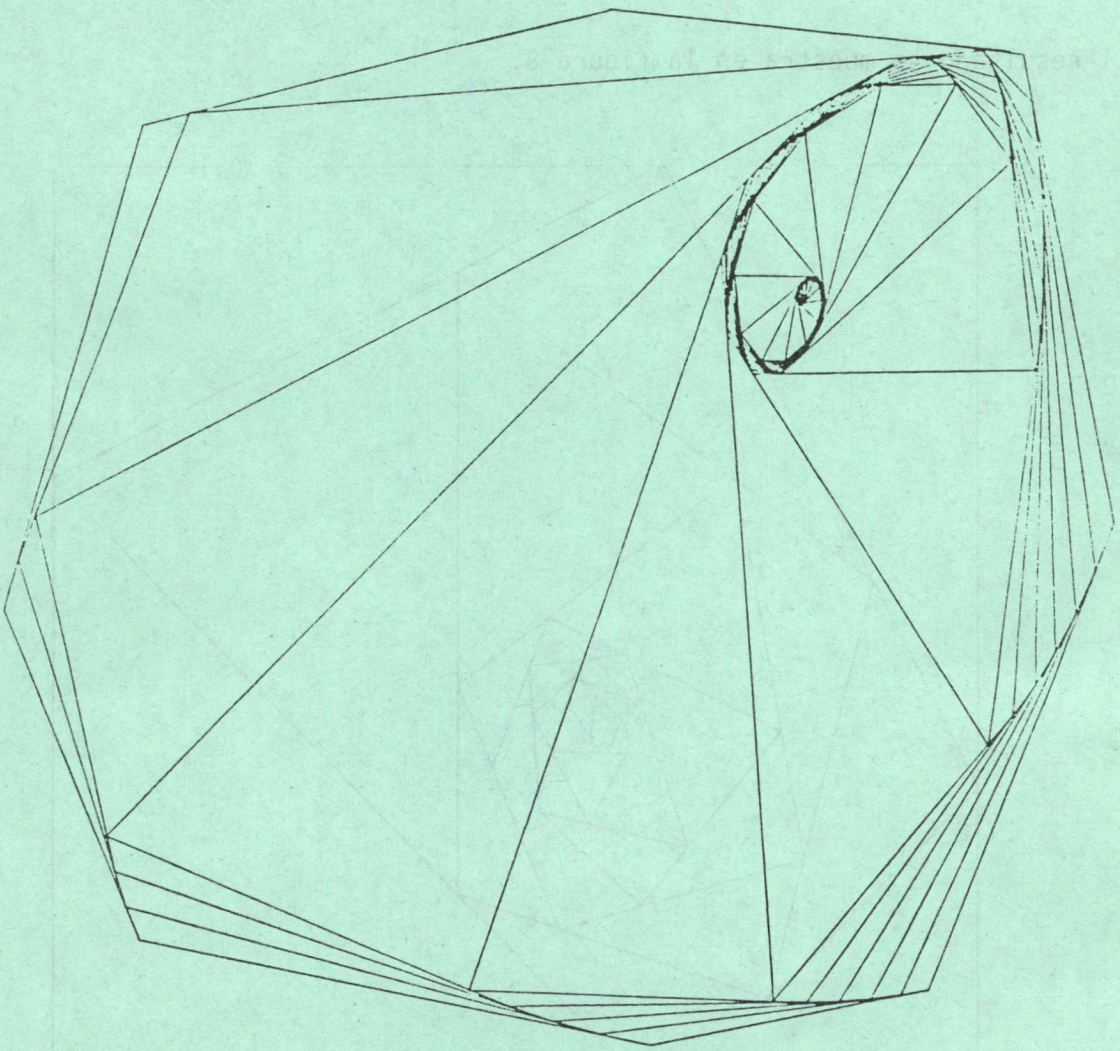


Fig. 9: Anidamiento de octógonos

El error es fácil de localizar, ya que si observamos el procedimiento ANIDAR2, cuando introducimos el elemento último, que es el que aparecerá como primero de LISTA, lo calculamos con el primero y el último de LISTAPUNTOS,

y eso antes no daba problemas por ser el punto medio, pero ahora, al haber una asimetría debida al factor de proporcionalidad (FACTOR), calcula ese punto para FACTOR=3 por ejemplo, a un tercio de un punto y a dos tercios del otro punto, y por tanto, ahora el orden afecta.

Lo corregimenos, véanse los listados al final, y volvemos a ejecutar para distintos casos, obteniendo las figuras que se indican a continuación:

```
FIGURA3 [[100 100] [100 -100] [-100 -100] [-100 100]] 10  3
```

(Figura 4)

```
FIGURA3 [[100 100] [120 0] [80 -100] [20 -112] [-90 -90] [-120 -20] [-90 85]
  [10 110]] 40  3
```

Estas dos ejecuciones (figura 4 y figura 9) producen 10 cuadrados y 40 octógonos, con un factor de 3.

Si probamos ahora con el cuadrado, con un factor de 13 y dibujando 50, obtenemos el gráfico de la figura 10.

```
FIGURA3 [[100 100] [100 -100] [-100 -100] [-100 100]] 50  13
```

Obsérvese que el lugar geométrico de los vértices describe una especie de curva espiral, que conicide con la solución al problema de persecución de las 4 pulgas desarrollado en otro capítulo, y que describe el movimeinto realizado por 4 pulgas, una en cada vértice de un cuadrado cuando cada una persigue a la pulga del siguiente vértice, y la del último a su vez a la del primero, cerrando el ciclo.

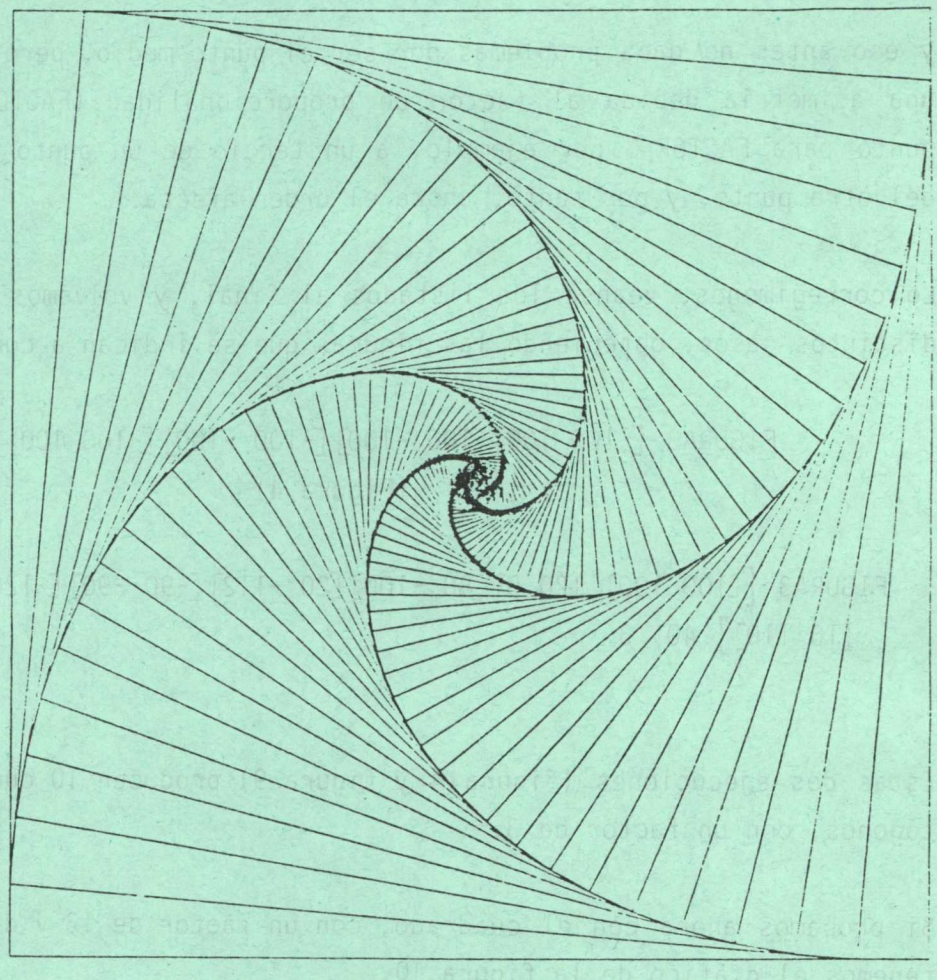


Fig. 10: Anidamiento de cuadrados

Esto se puede generalizar a un número de puntos N y así para el octógono, obtendríamos las curvas descritas por 8 pulgas cada una persiguiendo a la siguiente a su derecha.

Esto se consigue ejecutando:

```
FIGURA3 [[100 100][120 0][80 -100][20 -112][-90 -90][-120 -20][-90 85][10 110]]60
```

Gráfico en la figura 12.

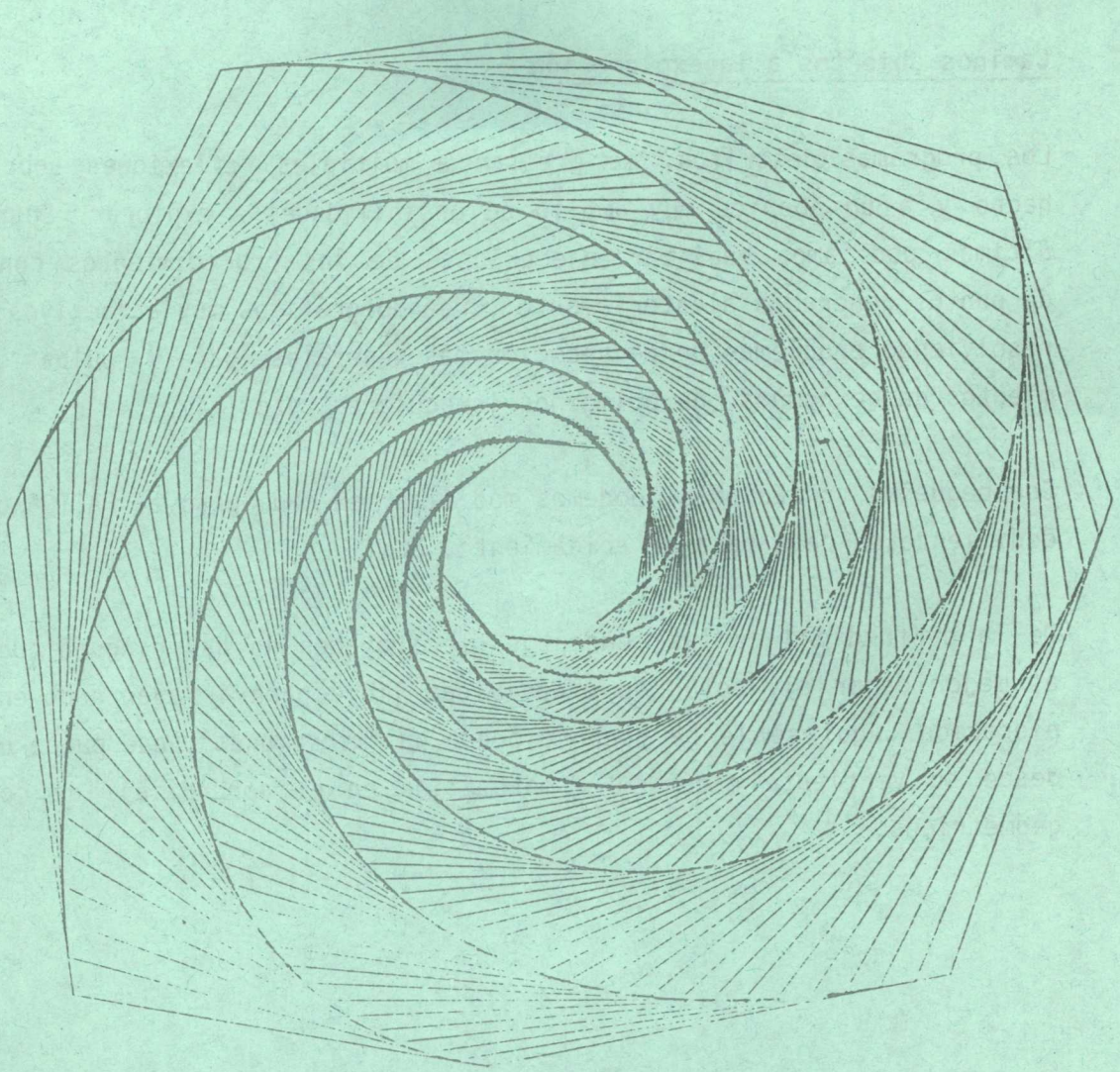


Fig. 11: Anidamiento de octógonos

A continuación se transcriben los listados de los procedimientos utilizados en esta nueva versión.

```
PARA FIGURA3 :LISTAPUNTOS :NIVEL :FACTOR
REPITE :NIVEL [POLIGONO :LISTAPUNTOS HAZ "LISTAPUNTOS ANIDAR3
       :LISTAPUNTOS :FACTOR]
FIN


PARA ANIDAR3 :LISTAPUNTOS :FACTOR
(LOCAL "I "LISTA)
HAZ "I 1
HAZ "LISTA []
REPITE (CUENTA :LISTAPUNTOS) - 1 [HAZ "LISTA PU (PUNTO.LADO
        ELEMENTO :I :LISTAPUNTOS ELEMENTO ( :I + 1)
        :LISTAPUNTOS :FACTOR ) :LISTA HAZ "I :I + 1]
HAZ "LISTA PP (PUNTO.LADO UL :LISTAPUNTOS PR :LISTAPUNTOS
    :FACTOR ) :LISTA
DEV :LISTA
FIN
```

Caminos abiertos a la exploración

Los programas descritos, se prestan a múltiples reflexiones sobre lo ya hecho y a un abanico muy amplio de posibilidades a explorar. Entre estas últimas podríamos realizar ejecuciones de los procedimientos con listas de puntos generadas aleatoriamente, creando gráficos muy atractivos, o bien resolviendo problemas de persecución más generales, para N "pulgas", anidamientos de hexágonos, de polígonos irregulares, etc.

Con pequeñas variaciones, podemos modificar en cada paso el FACTOR del procedimiento en cada paso de anidamiento, etc.

De la profundidad de los conceptos que subyacen no hay ninguna duda, pero es mejor dejar al lector su creatividad libre para ir avanzando en su exploración. Aquí se le ha proporcionado un "micromundo", que puede utilizar desde su vertiente estética o matemática en el aprendizaje de conceptos geométricos.

Lista de procedimientos primitivos LOGO en castellano.

Gráficos de la tortuga

AVANZA distancia (abreviado **AV**)
hace avanzar la tortuga la distacia indicada.

BAJALAPIZ (abreviado **BL**)
baja el lápiz, haciendo que, a partir de ese momento, la tortuga efectúe el dibujo dejando un rastro a su paso.

BORRAPANTALLA (abreviado **BP**)
borra la pantalla de gráficos, colocando la tortuga en posición [0 0] y con orientación 0.

CARGADIB fichero
carga el fichero de tipo pantalla (con extensión .PIC) y lo visualiza en la pantalla gráfica

CENTRO
mueve la tortuga al origen y la sitúa con orientación 0 grados

COLORLAPIZ (abreviado **CL**)
da como salida un número que representa el color del lápiz en uso.

COORX
da como salida la coordenada x correspondiente a la posición de la tortuga.

COORY
da como salida la coordenada y correspondiente a la posición de la tortuga.

ENLAZA
hace que la tortuga reaparezca por el lado opuesto de la pantalla cuando su posición excede de los límites de la pantalla, es decir, que aplique geometría toroidal.

ESCALA
da como salida la relación de aspecto de la pantalla, es decir, el cociente entre el paso de tortuga del eje vertical y el del horizontal.

ESTAMPA
deja una imagen fija de la tortuga de acuerdo con su forma actual en la posición en la que se encuentre.

FONDO
da como salida el número correspondiente al color del fondo.

FORMA
da como salida el número correspondiente a la forma que tiene la tortuga en uso.

GIRADERECHA ángulo (abreviado **GD**)
gira la tortuga a la derecha el ángulo dado en grados.

GIRAIZQUIEDA ángulo (abreviado **GI**)
gira la tortuga a la izquierda el ángulo dado en grados.

GOMA
activa el borrador, de forma que la tortuga borra las líneas dibujadas. Realmente lo que hace es dibujar con el color del fondo.

GUARDADIB fichero
guarda la pantalla (gráficos y texto) en el fichero especificado.

HACIA lista-de-coordenadas
da como salida la orientación que debería tener la tortuga para estar orientada hacia la posición dada por la lista--de-coordenadas.

INVIERTELAPIZ (abreviado **IL**)
pone el lápiz en modo inverso, haciendo que dibuje trazos cuando no haya líneas y borre las líneas ya dibujadas.

LAPIZ (abreviado **L**)
da como salida una lista con información sobre el estado actual del lápiz de la tortuga.

LIMITA
limita el campo de la tortuga acotándolo a unos valores. Si se da un comando a la tortuga que, caso de ejecutarse, la llevaría fuera de los límites, daría un error, y no ejecutaría la orden.

LIMPIA
borra la pantalla de gráficos sin afectar al estado de la tortuga.

MODELO n
reemplaza la forma de la tortuga por el gráfico sobre el que se encuentra.

MUESTRATORTUGA (abreviado **MT**)
hace visible la tortuga si ésta se encuentra oculta.

OCULTATORTUGA (abreviado **OT**)
hace la tortuga invisible.

PONCOLORLAPIZ número-de-color (abreviado **PONCL**)
da el color correspondiente a número-de-color al lápiz de la tortuga.

PONFONDO número-de-color
da al fondo el número-de-color especificado.

PONFORMA número
da a la tortuga la forma correspondiente al codigo ASCII especificado por número.

PONLAPIZ lista
asigna al lápiz el estado correspondiente a la lista de entrada.

PONPALETA número-de-paleta
hace que la paleta en uso sea la especificada por número-de-paleta.

PONPOS lista-de-coordenadas
sitúa a la tortuga, sin variar su orientación, en la posición dada por lista-de-coordenadas.

PONRUMBO ángulo (abreviatura **PONR**)
orienta la tortuga según el ángulo especificado.

PONX número
desplaza horizontalmente la tortuga, sin variar su orientación, hasta el punto de coordenada-x especificada.

PONY número
desplaza la tortuga verticalmente, sin variar su orientació, hasta el punto de coordenada-y.

POS
da como salida una lista con las coordenadas de la tortuga.

PUNTO lista-de-coordenadas
coloca un punto en el lugar definido por la lista-de--coordenadas dada.

RELLENA
rellena con el color actual del lápiz una zona cerrada que rodee a la tortuga.

RETROCEDE distancia (abreviado **RE**)
hace retroceder a la tortuga la distancia indicada.

RUMBO
da como salida la orientación de la tortuga en uso.

SUBELAPIZ (abreviado **SL**)
levanta el lápiz, haciendo que la tortuga no deje huella al efectuar sus desplazamientos.

VENTANA
hace que la pantalla sea como una ventana de observación del campo de movientos de la tortuga, permitiendo a ésta moverse fuera de los límites de la pantalla.

VISIBLE?
da como salida "CIERTO o "FALSO, según la tortuga esté visible o no.

Palabras y listas

ASCII carácter
da como salida el valor ASCII del carácter especificado.

CAR número
da como salida el carácter cuyo valor ASCII es el número de entrada.

CUENTA objeto
da como salida el número de elementos del objeto especificado.

ELEMENTO número objeto
da como salida el elemento que ocupa el lugar especificado por el número dado en el objeto de entrada.

FRASE objeto1 objeto2 (abreviado **FR**)
da como salida una lista formada por los objetos de entrada. Es similar a LISTA, salvo que elimina los corchetes más externos de cada uno de los objetos de entrada.

IGUALES? objeto1 objeto2
da como salida un valor booleano "CIERTO o "FALSO según sean iguales o no los objetos de entrada.

LISTA objeto1 objeto2
da como salida una lista cuyos elementos son los objetos dados como entrada.

LISTA? objeto
da como salida un valor booleano "CIERTO o "FALSO según sea o no una lista el objeto de entrada.

MENOSPRIMERO objeto (abreviado **MP**)
da como salida el objeto especificado sin su primer elemento.

MENOSULTIMO objeto (abreviado **MU**)
da como salida el objeto especificado sin su último elemento.

MIEMBRO? objeto1 objeto2
da como salida un valor booleano "CIERTO o "FALSO según sea o no el objeto1 un elemento del objeto2.

NUMERO? objeto
da como salida un valor booleano "CIERTO o "FALSO según sea o no el objeto especificado un número.

PALABRA palabra1 palabra2
da como salida una palabra formada por todas las palabras dadas como entrada, unas detrás de otras.

PALABRA? objeto
da como salida un valor booleano "CIERTO o "FALSO según sea o no una palabra el objeto especificado.

PONPRIMERO objeto lista (abreviado **PP**)
da como salida una lista, que es la obtenida al añadir el objeto especificado como nuevo elemento al principio de la lista original.

PONULTIMO objeto lista (abreviado **PU**)
da como salida una lista, que es la obtenida al añadir el objeto especificado como nuevo elemento al final de la lista original.

PRIMERO objeto (abreviado **PR**)
da como salida el primer elemento del objeto especificado.

ULTIMO objeto (abreviado **UL**)
da como salida el último elemento del objeto especificado.

VACIO? objeto
da como salida un valor booleano "CIERTO o "FALSO según el objeto especificado sea o no vacío.

Variables

ASIGNA objeto nombre
da al objeto el valor de nombre. Es equivalente a HAZ salvo que el orden de las entradas está invertido.

EDVARS
carga los valores de todas las variables del área de trabajo en el registro del editor de pantalla, pudiendo modificar su valor trabajando desde éste.

HAZ nombre objeto
crea la variable nombre y le da el valor objeto.

LOCAL nombre
hace que la varaiable cuyo nombre se especifica como entrada sea accesible sólo desde el procedimiento en el que está incluida y en todos aquellos subprocedimientos que sean llamados desde éste.

VALOR nombre
da como salida el valor de la variable nombre.

VAR? nombre
da como salida un valor booleano, "CIERTO o "FALSO, según sea o no una variable el nombre especificado.

Operaciones aritméticas

ARCTAN número
da como salida el arco en grados cuya tangente es el número de entrada.

AZAR número
da como salida un número al azar comprendido entre 0 y número menos 1.

COCIENTE número1 número2
da como salida el cociente de las entradas.

COS ángulo
da como salida el coseno del ángulo, dado en grados.

DIFERENCIA número1 número2
da como salida la diferencia entre las entradas.

ENTERO número (abreviado **ENT**)
da como salida la parte entera del número de entrada.

EXP número
da como salida el número 'e' elevado al número especificado como entrada.

FORMATO n a (b)
da como salida el número n con a digitos antes del punto decimal y b digitos después. Si no se especifica b, no hay dígitos después del punto decimal.

LN número
da como salida el logaritmo natural, o neperiano, del número especificado.

MISMOAZAR
hace que la siguiente expresión aleatoria reproduzca la misma anterior.

NOTACION n a
da como salida un número n en notación científica, con a dígitos. a debe ser positivo y real entre 1 y 1000. Si a no es entero, se redondea al entero más próximo.

PI
da como salida el valor de la constante 3.1415925.. .

PONPRECISION n
asigna un número n de dígitos significativos para las operaciones que se realicen a partir de ese momento.

POTENCIA n a
da como salida el número n elevado a a.

PRECISION
da como salida el número de dígitos significativos al que se redondea cuando se efectúan operaciones.

PRODUCTO a b
da como salida el producto de a por b.

RAIZCUADRADA número (abreviado **RC**)
da como salida la raíz cuadrada del número de entrada.

REDONDEA número
da como salida el resultado de redondear al entero más próximo el número especificado como entrada.

RESTO a b
da como salida el resto obtenido al hacer la división entera de a por b.

SEN ángulo
da como salida el seno del ángulo dado en grados.

SUMA a b
da como salida la suma de a más b.

Definición de procedimientos

COPIADEFINICION nuevo-nombre nombre
define el procedimiento nuevo-nombre copiando su definición de la de nombre.Ninguno de los argumentos de entrada puede ser una primitiva, a no ser que la variable REDEF? tenga el valor "CIERTO.

DEFINE nombre lista
define un procedimiento con el nombre especificado, utilizando para ello la lista de instrucciones indicada.

DEFINIDO? nombre
da como salida un valor booleano, "CIERTO o "FALSO, según esté o no definido el procedimiento especificado por nombre.

EDITA nombre (lista de nombres) (abreviado **ED**)
carga el(los) procedimiento(s) cuyo(s) nombre(s) se indican, en el registro del editor de pantalla, y entra en el entorno del editor de pantalla. Si el procedimiento no existe en el área de trabajo, se inicia el proceso con la primera línea de la definición. Si no se da entrada, vuelve a editar el último procedimiento que se editó.

PARA procedimiento (:entradas)
indica el comienzo de la definición de un procedimiento con las entradas que se indiquen.

PRIMITIVA? nombre
da como salida un valor booleano, " CIERTO o "FALSO, según que el nombre especificado sea o no una primitiva del sistema.

TEXTOPROC procedimiento
da como salida un lista que corresponde a la definición del procedimiento dado como entrada.

Control

ALTO
detiene la ejecución del procedimiento y devuelve el control al procedimiento que lo llamó, o al NIVEL SUPERIOR, o nivel más externo.

COMPRUEBA condición
realiza la comprobación de si la condición de entrada es "CIERTO o "FALSO, y almacena este resultado para ser comprobado por SICIERTO o SIFALSO.

CONT
finaliza una pausa generada mediante una orden PAUSA en el programa, o provocada mediante el teclado. Se utiliza para la depuración de los programas. No se puede utilizar en el interior de un procedimiento.

DEVUELVE objeto (abreviado **DEV**)
hace que el objeto de entrada sea la salida del procedimiento, interrumpiendo su ejecución, y devolviendo objeto al que lo llamó.

EJECUTA lista-de-instrucciones
ordena la ejecución de la lista-de-instrucciones dada como entrada.

ENVIA nombre
ejecuta la línea identificada por el nombre de entrada en una expresión RECIBE anterior. Si se desea devolver el control al nivel más externo, nombre será la palabra reservada "NIVEL SUPERIOR.

ERROR
da como salida una lista cuyos elementos describen el error detectado más recientemente. Esta lista contiene seis elementos: 1) número que identifica el error, 2) mensaje explicativo del error, 3) nombre del procedimiento que contiene la expresión errónea, 4) la línea completa que contiene la expresión errónea, 5) el nombre de la primitiva causante del error, 6) el nombre del objeto que causa el error.

ETIQUETA nombre
identifica la línea que se ha de ejecutar por medio de una instrucción PARA nombre.

PAUSA
suspende la ejecución del procedimiento que se está ejecutando, permitiendo así la interacción con el intérprete o el editor. Para continuar el proceso, se ordena CO.

RECIBE nombre lista-de-instrucciones
ejecuta la lista-de-instrucciones. Si se ejecuta un ENVIA con el mismo nombre mientras se está ejecutando la lista-de--instrucciones, se devuelve el control a RECIBE. El nombre se utiliza para hacer coincidir RECIBE y ENVIA.

REPITE número lista-de-instrucciones
ordena la ejecución repetida de la lista-de-instrucciones el número de veces especificado.

SI condición lista-de-instrucciones1 (lista-de-instrucciones2)
ejecuta una de las dos listas-de-instrucciones dadas como entrada, según se verifique o no la condición dada como entrada. Si no existe la segunda lista de instrucciones, y no realiza ninguna acción caso de que la condición no se verifique.

SICIERTO lista-de-instrucciones (abreviado **SIC**)
ejecuta la lista-de-instrucciones de entrada si el último COMPRUEBA realizado dio como resultado "CIERTO.

SIFALSO lista-de-instrucciones (abreviado **SIF**)
ejecuta la lista-de-instrucciones de entrada si el último COMPRUEBA realizado dio como resultado "FALSO.

VE etiqueta
ordena la ejecución de la instrucción identificada por la entrada etiqueta, y que se ha de encontrar definida como ETIQUETA en alguna otra parte del procedimiento.

Operaciones lógicas

NO predicado
da como salida el valor complementario, en sentido lógico, del predicado de entrada. Es decir, si la entrada es "FALSO la salida es "CIERTO, y recíprocamente.

O predicado1 predicado2
da como salida la función booleana O de las entradas, es decir, da salida "CIERTO en todos los casos excepto cuando los dos predicados de entrada son "FALSO.

Y predicado1 predicado2
da como salida la función booleana Y de las entradas, es decir, da salida "CIERTO en el caso de que los dos predicados de entrada sean "CIERTO y "FALSO en cualquier otro caso.

Comunicación con el exterior

ABIERTOS
da como salida una lista con los nombres de todos los ficheros abiertos.

ABRE fichero/dispositivo
abre el fichero o el dispositivo especificado. Es imprescindible dar la orden de abrir el fichero o dispositivo antes de empezar a enviarle caracteres o a recibirlos del mismo.

ADIOS
devuelve el control al sistema operativo.

ARCHIVO? archivo
da como salida un valor booleano, "CIERTO o "FALSO, según el exista o no en el disco un fichero con el nombre especificado.

BORARCHIVO archivo
borra el archivo del disco cuyo nombre se especifica.

BOTON? número
da como salida un valor booleano "CIERTO o "FALSO según esté activado o no el botón correspondiente del "paddle" o "joystick" especificado por número.

CIERRA fichero/dispositivo
cierra el fichero o dispositivo especificado. Se produce un error si se intenta cerrar un dispositivo o fichero que no esté abierto. Sucede lo mismo si se intenta cerrar un fichero ECO.

CIERRATODO
cierra todos los ficheros abiertos.

DIR (dispositivo : fichero)
imprime todos los ficheros del disco de la unidad de disco especificada (o la que estuviera activa por defecto). Si se especifica la entrada fichero imprime los que corresponden. Se pueden utilizar "*" y "?" para obtener el directorio de un determinado modelo de fichero, donde "*" representa cualquier palabra y "?" cualquier carácter.

DISCO
da como salida el nombre de la unidad de disco por defecto.

ECO dispositivo/fichero
inica el proceso de enviar una copia de los caracteres de texto al fichero o dispositivo especificado. ("LPT1 para la impresora.

EDARCHIVO fichero-entrada (fichero-salida)
edita el fichero-entrada, guardándolo con el mismo nombre si no se especifica fichero-salida como segunda entrada. No debe existir en el disco un fichero con ese segundo nombre si se elige esta segunda opción. Se puede utilizar para editar ficheros ASCII pero no binarios.

ESCRIBE objeto (abreviado **ES**)
visualiza el o los objetos de entrada en la pantalla de texto.

ESCRITURA
devuelve el nombre del archivo o el código del dispositivo abierto para escritura.

ESPERA número
detiene la ejecución durante el tiempo especificado por número.

FINESC?
da como salida un valor booleano "CIERTO, si la posición actual del fichero de escritura es la del final de fichero, y "FALSO en caso contrario.

FINLEC?
da como salida un valor booleano, "CIERTO si la posición actual del fichero es la de fin de fichero, y "FALSO, en caso contrario.

IMARCHIVO fichero
abre el fichero especificado como entrada y lo imprima en la pantalla.

LECTURA
da como salida el dispositivo o fichero que está activado para lectura.

LEECAR (abreviado **LC**)
da como salida el primer carácter tecleado. Al ejecutarse esta primitiva, queda a la espera de recibir un carácter, y, una vez tecleado, continúa la ejecución, sin que aparezca éste en la pantalla.

LEECARS n (abreviado **LCS**)
da como salida los n primeros caracteres tecleados. Al ejecutar esta primitiva queda a la espera de recibir n caracteres, y una vez tecleados, continuará la ejecución, sin que aparezcam en la pantalla.

LEELISTA (abreviado **LL**)
da como salida una lista formada por la línea escrita por el teclado. Al ejecutarse esta primitiva, se queda a la espera de recibir un retorno, en cuyo caso ya asigna a las palabras tecleadas a elementos de la lista de salida de la primitiva.

LEEPALABRA (abreviado **LP**)
da como salida una palabra introducida por el teclado, y seguida de un retorno.

LONGARCHIVO archivo
da como salida la longitud en bytes del archivo especificado como entrada. Este archivo ha de estar abierto, en caso contrario, se produciría un error.

MANDO número
da como salida un número que corresponde a la posición del mando especificado por el número de entrada.

MUESTRA objeto
visualiza el objeto de entrada en la pantalla de texto. Si el objeto es una lista, aparece tal como es, es decir, con los corchetes exteriores.

NOECO
desactiva la escritura de los caracteres al archivo previamente activado por el comando ECO.

PONCOMUNICACION n baudios paridad bits-de-datos bits-de-parada
inicia, de acuerdo con los parámetros argumento, la línea serie de comunicaciones asíncronas.

PONDISCO disco
hace que el disco por defecto seaa el disco especificado por la entrada.

PONESCRITURA dispositivo / archivo
asigna como dispositivo o archivo de escritura al especificado. A partir de esta orden, las escrituras de datos a realizar por las diferentes primitivas de escritura (ESCRIBE, TECLEA, etc.) se efectúan al mismo.

PONLECTURA dispositivo / archivo
asigna como dispositivo o archivo de lectura al especificado. A partir de esta orden, las lecturas de datos pedidos por las distintas primitivas de lectura (LC, LL, etc.) se efectúan del mismo.

PONPOSESCRITURA posición-del-fichero
sitúa el puntero del fichero de escritura en el número de byte especificado por posición-de-fichero.

PONPOSLECTURA posición-del fichero
sitúa el puntero del fichero de lectura en el número de byte especificado por posición-del-fichero, permitiendo lectura de acceso directo.

POSESCRITURA
da como salida la posición del puntero de escritura en el fichero.

POSLECTURA
da como salida lal posición actual del byte que se está leyendo del fichero.

TECLA?
da como salida un valor booleano, "CIERTO o "FALSO, según haya o no un carácter tecleado en espera de ser leído.

TECLEA objeto...
imprime los objetos de entrada en la pantalla de texto, pero sin que se produzca un cambio de línea después del último objeto visualizado.

TONO frecuencia duración
da como salida una nota de la frecuencia y duración especificada por las entradas correspondientes.

Texto y pantalla

ANCHURA
da como salida la anchura actual de la pantalla (de 2 a 80 caracteres por línea).

BORRATEXTO (abreviado **BT**)
borra la parte de texto de la pantalla.

COLORTEXTO (abreviado **CT**)
da como salida una lista con dos números, el primero de ellos corresponde al color de los caracteres, y el segundo, al del fondo del texto.

CURSOR
da como salida una lista que contiene los números de columna y de línea que corresponden a la posición del cursor en la ventana de texto.

PANTALLA
devuelve un número que representa el número de pantallas utilizadas: 1 si hay una sola pantalla y 2 si hay una pantalla de color y una pantalla monocroma.

PANTALLAGRAFICA (abreviado **PG**)
pone a la pantalla en modo gráfico, haciendo desaparecer el texto sin borrarlo.

PANTALLAMIXTA (abreviado **PM**)
pone la pantalla en modo mixto, con la parte superior para gráficos y la inferior para texto.

PANTALLATEXTO (abreviado **PT**)
selecciona el modo de pantalla de texto.

PONANCHURA número
asigna a la anchura de la pantalla el número especificado de caracteres (entre 0 y 80).

PONCOLORTEXTO lista-de-colores (abreviado **PCT**)
da al texto los colores correspondientes dados en list-de--colores, de forma que el primer elemento de la lista es el color de los caracteres, y el segundo, el color del fondo de los caracteres.

PONCURSOR lista de coordenadas
coloca el cursor en la posición indicada por la lista-de--coordenadas que da el número de columna y el de fila.

PONMAYUSCULAS predicado
si la entrada es "CIERTO, los caracteres aparecen en mayúsculas tanto si está bloqueada la tecla CAPS LOCK como si no. Si la entrada es "FALSO, los caracteres aparecen en mayúsculas o minúsculas según esté bloqueada o no la tecla CAPS LOCK

PONPANTALLA n
fija el texto y el grágico sobre una o dos pantallas, según que n sea 1 o 2. Si n es 2, una pantalla será para los gráficos y otra para el texto, si n es 1 se utilizará una sola pantalla.

PONTEXTO n
asigna el número dado de líneas a la ventana de texto en el modo mixto de pantalla.

Area de Trabajo

AGRUPA nombre-del-paquete nombre-del-objeto(s)
realiza una agrupación colocando los procedimientos y variables especificadas por nombre-del-objeto en el paquete cuyo nombre se indica. Si el paquete ya existe, añade los nuevos objetos al paquete ya existente. Cada procedimiento o variable, puede estar sólo en un paquete.

AGTODO nombre-del-paquete
coloca todos los procedimientos y variables que no se hallen en paquetes, en el paquete cuyo nombre-de-paquete se especifica como entrada a la primitiva.

BOPROCS (paquete/lista de paquetes)
borra todos los procedimientos del área de trabajo o de los paquetes especificados en las entradas, caso de especificarse.

BORRA nombre-de-procedimiento/lista de procedimientos
(abreviado **BO**)
borra los procedimientos, cuyos nombres se especifican a la entrada, del área de trabajo.

BOTODO (paquete/lista de paquetes)
borra todos los procedimientos y variables del área de trabajo, o, si se especifica paquete o lista-de-paquetes de entrada, sólo las correspondientes a éstos.

BOVAR nombre/lista-de-nombres
borra las variables cuyos nombres se especifica en la entrada del área de trabajo.

BOVARS (paquete/lista-de-paquetes)
borra todas las variables del área de trabajo, y si se especifica un paquete(s), de entrada, sólo las correspondientes a éstos.

NUCLEO: LOGO
UTI: 0123
PAG.: 17

CARGA fichero
carga el fichero cuyo nombre se especifica en el área de trabajo. El fichero debe de haber sido cargado con la orden GUARDA previamente.

CONTENIDOS
da como salida una lista con todos los objetos del área de trabajo.

DESTAPA nombre-del-paquete/lista-de-paquetes
pone nuevamente bajo el control de los comandos del área de trabajo el o los paquetes especificado, es decir, los "desentierra", pudiendo listarlos o editarlos con los comandos habituales.

GUARDA archivo (paquete/list-de-paquetes)
escribe el contenido del área de trabajo o del paquete o lista-de-paquetes en el archivo cuyo nombre se especifica.

IM procedimiento/lista-de-procedimientos
imprime las definiciones de los procedimientos especificados.

IMPS (paquete/lista-de-paquetes)
cuando no tiene entrada, escribe la definición de cada procedimiento del área de trabajo, excepto los que están tapados. Con una entrada escribe los procedimientos del paquete o paquetes, incluso si están tapados.

IMTODO (nombre-del-paquete/lista-de-paquetes)
imprime las definiciones de todos los procedimientos y variables del área de trabajo o de los paquetes que se especifiquen, caso de que se den como entrada.

IMTS (nombre-del-paquete/lista-de-paquetes)
imprime los títulos (nombres y entradas) de todos los procedimientos del área de trabajo o de los paquetes especificados, si se dan como entrada.

IMVS (nombre-del-paquete/lista-de-paquetes)
imprime los nombres y valores de todas las variables del área de trabajo o de los paquetes especificados como entradas.

NODOS
da como salida el número de nodos libres en el área de trabajo. Así se puede, conociendo la disponibilidad de memoria, efctuar una limpieza del área de trabajo.

RECICLA
efectúa una limpieza dl área de trabajo ("recogida de basura" o "garbage collection"), reorganizándola, y liberando tantos nodos como es posible.

RECOPILA
cuando se define o se bora un procedimiento, LOGO necesita cambiar la interpretación de otros procedimientos del área de trabajo.

TAPA paquete/lista-de-paquetes
oculta del área de trabajo el paquete o la lista-de--paquetes especificados en la entrada, con lo cual no aparecerán al pedir realizar cualquier orden que afecte al área de trabajo, mientras tengan la propiedad de estar "enterrados". Para "desenterrarlos" se utiliza DESTAPA.

Lista de propiedades

BOPROP nombre propiedad
elimina la propiedad dada en su entrada, de la lista de propiedades asociadas a nombre.

IMPROPS
imprime las listas de propiedades del área de trabajo.

LPROP nombre
da como salida la lista de propiedades asociada al nombre dado como entrada.

PONPROP nombre propiedad objeto
añade una nueva propiedad a la lista de propiedades de nombre, y cuyo valor asociado es objeto.

PROP nombre propiedad
da como salida el valor asociado al objeto dado por nombre, por la propiedad.

Lenguaje ensamblador

CARGABIN especarchivo segmento desplazamiento
carga en memoria un archivo binario (escrito en lenguaje ensamblador).

DEPOSITA n byte
escribe el byte en la dirección de máquina n.

EXAMINA n
da como salida el contenido de la dirección de máquina n.

GUARDABIN especarchivo segmento desplazamieno longitud
copia una parte de la memoria en el archivo especificado.

LLAMA base
transfiere el control a la subrutina de lenguaje máquina que comienza en la dirección base determinada por la entrada.

Palabras Especiales

ARRANQUE
Variable del sistema. Si es una lista, LOGO la ejecuta automáticamente al cargar en memoria un fichero que la contenga, y por tanto sirve para crear un fichero que se autoinicializa.

CIERTO
entrada especial para operaciones booleanas (Y, SI,NO...).

EDITOR
entrada para el comando de edición, contiene el registro del editor.

ERRACT
variable del sistema. Si es "CIERTO, LOGO efectúa una pausa al ocurrir un error.

FALSO
entrada especial para operaciones booleanas (Y, SI, NO...).

FIN
indica el fin de una definición de procedimiento que se inició con PARA.

GRUPOPROC
propiedad asociada al nombre de un procedimiento, y cuyo valor es el nombre del paquete en que se encuentra.

GRUPOVAR
propiedad asociada a las variables que se encuentran dentro de un paquete, y su valor es el nombre de dicho paquete.

NIVELSUPERIOR
asociado a ENVIA, cuando se desea devolver el control al nivel más alto, en el que las órdenes se ejecutan directamenate, y no dentro de un procedimiento.

REDEF?
variable del sistema. Si tiene el valor "CIERTO, permite redefinir las primitivas.

TORTUGA
entrada para PONFORMA o salida para FORMA, corresponde a la forma normal de la tortuga.

Otras primitivas

ACTIVA número-de-tortuga/lista-de-tortugas
hace que los comandos que se den a las tortugas, a partir de ese momento, se dirijan a la(s) tortuga(s) indicadas por número-de-tortuga o elementos de lista-de-tortugas.

BALANCEA ángulo
balancea la tortuga un ángulo dado.

BOPROPS (paquete/lista-de-paquetes)
borra todos los procedimientos del área de trabajo o de los paquetes especificados en las entradas, caso de especificarse.

BOTONPAL n
activa el botón especificado por el número.

CABECEA ángulo
hace cabecear a la tortuga el ángulo dado.

CADA lista-de-instrucciones
hace ejecutar la lista-de-instrucciones a cada una de las tortugas por separado.

CONTACTO número-de-tortuga1 número-de-tortuga2
da como salida un número simbolizando una colisión entre las dos tortugas número-de-tortuga1 y número-de-tortuga2.

CUANDO condición lista-de-instrucciones
activa un sensor o "demon", de forma que cuando ocurra la condición especificada (choque entre tortugas...), ejecuta la lista-de-instrucciones, y si ésta es la lista vacía, desactiva el sensor.

EDFORMA número-de-forma
inicializa el editor de formas, apareciendo en la pantalla la forma correspondiente al número-de-forma solicitado.

IMSENSORES
imprime todas las condiciones y acciones a realizar de todos los sensores (demons) activados previamente por una orden CUANDO.

PALANCA número
da como salida un número que corresponde a la posición del joystick especificado por el número de entrada.

PIDE número-de-tortuga lista-de-instrucciones
pide a la(s) tortuga(s), indicadas por un número-de-tortuga (o por una lista de números), que ejecuten las instrucciones de la lista-de-instrucciones.

PONVEL velocidad
da a la tortuga en uso la velocidad indicada.

PONVELX coordenada-x
desplaza horizontalmente a la tortuga, sin variar su orientación, a la coordenada-x especificada en la entrada.

PONVELY coordenada--y
desplaza verticalmente a la tortuga, sin variar su orientación, a la coordenada-y especificada en la entrada.

QUIEN
da como salida el número de la tortuga en uso.

VEL
da como salida la velocidad de la tortuga en uso.

VIRA ángulo
gira la tortuga un ángulo dado.

B I B L I O G R A F I A

LENGUAJE LOGO Y ENSEÑANZA

ABELSON, Harold (1984). **Apple LOGO.** New York, McGraw-hill.

ABELSON, Harold and A. diSESSA (1986). **Geometria de la tortuga. El ordenador como medio de exploración de las Matemáticas.** Madrid. Anaya Multimedia.

O'SHEA, Tim and J. SELF (1985). **Enseñanza y aprendizaje con ordenadores: Inteligencia Artificial en educación.** Madrid, Anaya.

PAPERT, Seymour (1982). **Desafío a la mente. Computadoras y educación.** Buenos Aires, Galápago.

REGGINI, Horacio C. (1982). **Alas para la mente. LOGO: Un leenguaje de computadoras y un estilo de pensar.** Buenos Aires, Galápago.

REGGINI, Horacio C. (1985). **Ideas y formas: Explorando el espacio con LOGO.** Buenos Aires, Galápago.

RODRIGUEZ-ROSELLO, Luis (1986). **LOGO: De la tortuga a la Inteligencia Artificial.** Madrid, Vector.

SEGARRA, M. Dolors y J. GAYAN (1985). **LOGO para maestros. El ordenador en la escuela: Propuesta de uso.** Barcelona, Gustavo Gili.

TURCKLE, Sherry (1984). **El segundo yo.** Buenos Aires, Galápago.

WATT, Daniel (1984). **Aprendiendo con LOGO.** Madrid, McGraw-Hill.

MARTI, Eduardo (1984). "El ordenador como metáfora. Las posibilidades educativas de LOGO. **Infancia y aprendizaje**, no 26, pp 47-64.

Revistas

The International LOGO Exchange (ILX).

The National LOGO exchange (NLX).

LOGO and Educational Computing Journal.

The Computing Teacher

Boletín de la Asociación LOGO

Boletín del Centro LOGO.

HOJA ELECTRONICA DE OPEN ACCESS

INTRØDUCCION

Las Hojas de Calculo son programas que proporcionan al usuario un ambiente de trabajo con gran poder de calculo, pero fundamentalmente le dan gran flexibilidad para realizar modificaciones en dichos calculos.Permiten simular diferentes situaciones hipoteticas y recalculan facilmente todos los aspectos afectados ante cualquier modificacion sobre los planteamientos iniciales.

Se pueden considerar la herramienta ideal para realizar previsiones de cualquier tipo:de producción,de ventas,de capitalizacion,de beneficios,etc., ya que permiten ajustar,con gran velocidad,todos los calculos ante cualquier nuevo supuesto.En definitiva podemos decir que estan pensadas para responder a la pregunta:¿Que pasaria si...?

Acceso a la Hoja de Calculo

Para acceder debemos seleccionar Hoja de Calculo en el **Menú de Opciones** del Open Access (figura 1).

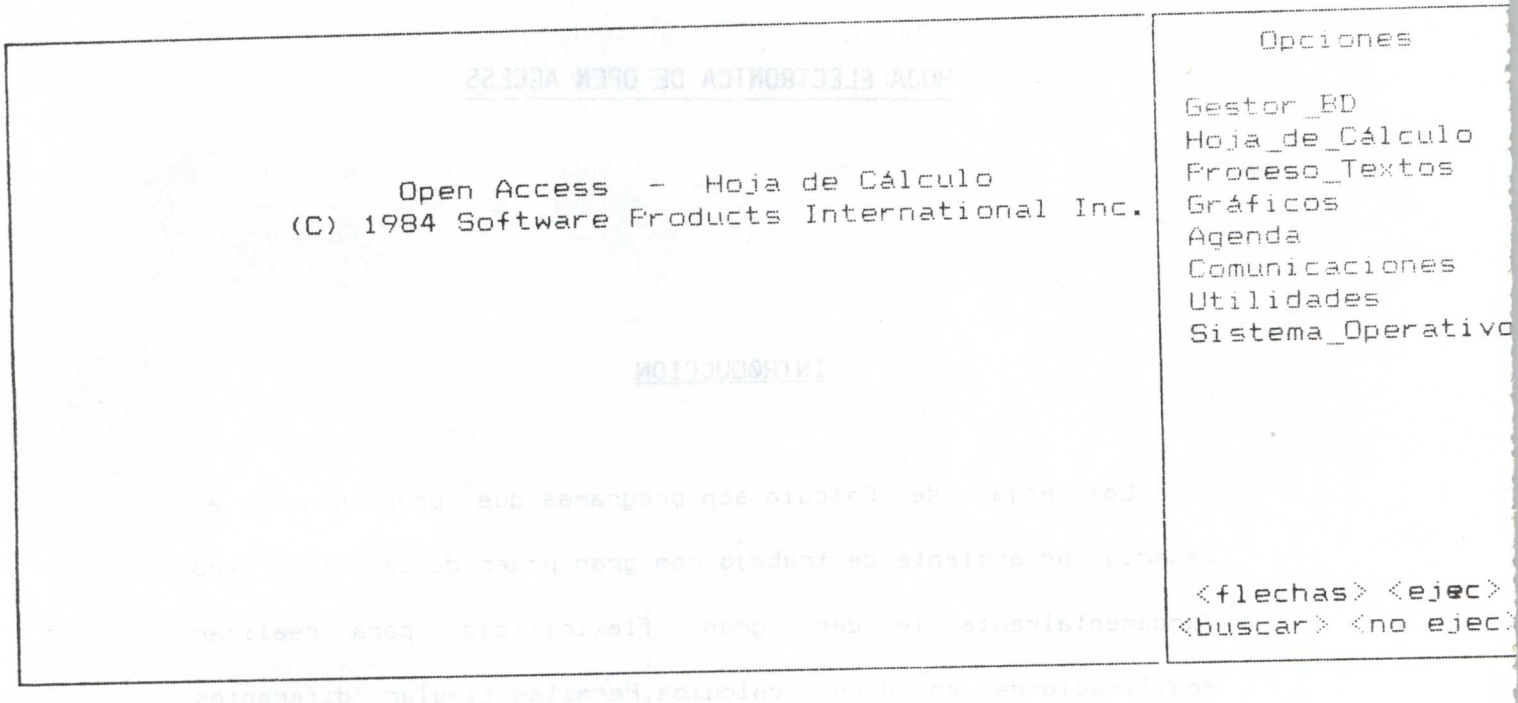


figura 1

La seleción nos conduce al **Menú de Selección de Modelo** (figura 2), que nos permite realizar las siguientes tareas:

SELECCIONAR: Carga un modelo de Hoja ya existente.

NUEVO: Permite crear un nuevo modelo.Hay que darle un nombre (de 1 a 8 caracteres) y opcionalmente una palabra clave.

COPIAR: Sirve para crear un nuevo modelo con el contenido de otro ya existente.

BORRAR: Elimina del disco el modelo indicado.

EST_CONSOLIDACION: Permite establecer la lista de modelos que se han de incluir en **Consolidación.**

LISTA ORDINARIA

LISTA RECURSIVA

CONSOLIDACION: Totaliza datos numericos constantes varios modelos en uno.

OPCIONES: Presenta las posibilidades de conexión, con otros módulos del Open Access, desde la Hoja de Calculo.

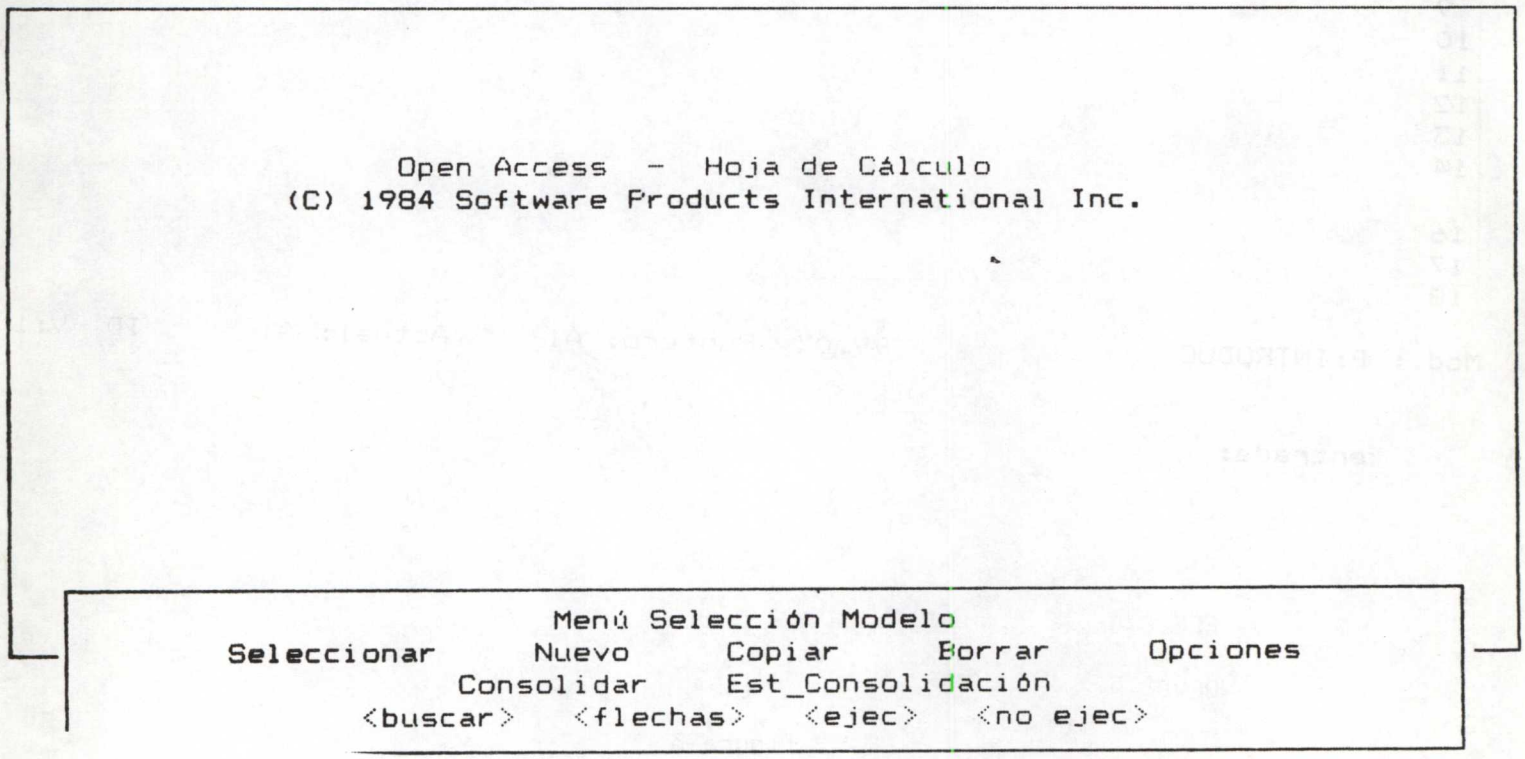


figura 2

Con la ejecución de **Seleccionar** ó **Nuevo** se accede a la Hoja de Calculo.

Descripción de la Hoja de Calculo

Se presentan como unas matrices compuestas de celdas, que se pueden referenciar individualmente o por grupos y que constituyen las areas de trabajo.(figura 3).

```
V1 A          |B          |C          |D          |E          |F          |
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14

16
17
18

Mod.: B:INTRODUC                  99.0%  Puntero: A1      Actual: A1        ID  V:1

      entrada:
```

figura 3

La Hoja de Calculo del Open Access dispone de una capacidad máxima de 216 columnas (de la A a la HH) por 3000 lineas (de la 1 a la 3000), por lo que visualizamos en la pantalla solamente una parte de la misma,que denominaremos ventana abierta sobre la Hoja de Calculo.Con las teclas de movimiento del cursor podemos presentar en la pantalla el total de las celdas, en sucesivas ventanas.(figura 4).

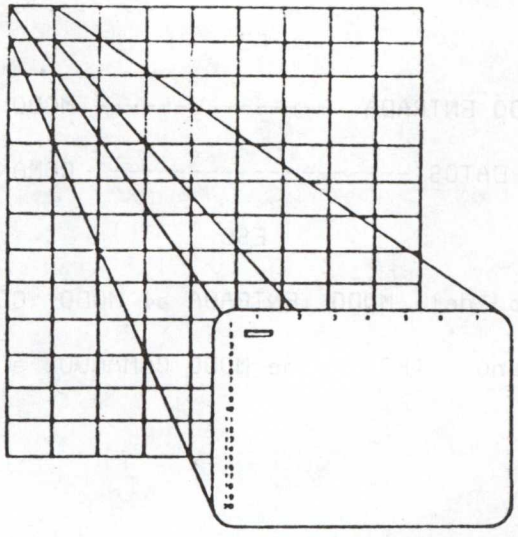


figura 4

Tambien es posible visualizar parte de una ventana, junto con parte de otra,pudiendo llegar a simultanear hasta un total de 8 ventanas.

Ademas del contenido de las diferentes celdas,la ventana presenta, bajo la ultima fila, las caracteristicas siguientes:

Mod.: Indica la unidad de disco y nombre del modelo.

núm. % Espacio disponible en el modelo.

Puntero: Coordenada actual del puntero.

Actual: Posición del puntero.

+ Indica que el avance esta activo.

ID/AA Establece el movimiento del puntero con avance ó F9.

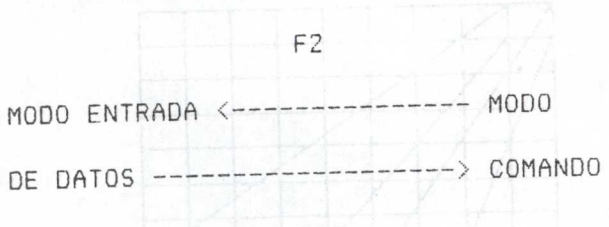
V: núm. Identificador de ventana.

 # núm. Identificador del número de canal.

Asi mismo, debajo aparece **entrada:** que es la línea en la que se introduce la información.

Modalidades de trabajo

Existen dos modalidades de trabajo en la Hoja de Calculo:



El paso del MODO ENTRADA al MODO COMANDOS se realiza pulsando <menú> (F2) y de MODO COMANDOS a MODO ENTRADA <no ejec> (ESC).

Entrada de datos

En el MODO ENTRADA de datos, la información introducida se carga en la celda seleccionada, que es la unidad básica de trabajo de la Hoja de Calculo.La información puede ser de tipo **numerico** ó de **texto**.El primer caracter introducido define si se trata de textos (una letra) ó números (una cifra ó los signos + ó -), excepto que lo modifiquemos con la tecla <cambiar> (F6).

La información de las celdas puede, cuando no ocupa el espacio total de las mismas, ajustarse a izquierda, a derecha, ó al centro, mediante la tecla <ajustar> (F7), que al pulsarla consecutivamente, recorre los diferentes tipos de ajuste.Tambien pueden definirse de forma generica atributos a un rango de celdas (**Formato ----> Atributos**).

En las celdas en que se trabaje con información numerica pueden introducirse constantes, variables, funciones,e incluso expresiones, que permiten obtener valores para ciertas celdas en función de otras,lo cual conduce a poder recalcular el contenido de una celda, cuando hay cualquier cambio en aquellas de las que es función.

Entrada de comandos

En este modo se puede seleccionar la ejecución de un comando de la Hoja de Calculo.

Esta selección se puede realizar:

-Visualizando el **Menú de Comandos**.En este caso basta con posicionar el cursor en el comando correspondiente del **Menú** y pulsar <ejec> (F10), ó teclear la inicial del comando ó las dos primeras letras, cuando hay dos que comienzan con la misma letra.

-Sin visualizar el **Menú de Comandos**.Los comandos se referencian por una sola letra, introducida en la **Linea de Comandos**.

La visualización ó no del **Menú de Comandos** se define con el comando **Establecer ----> Menú**.

Teclas de Función

F1 - AYUDA, F2 - MENU, F4 - BUSCAR, F5 - CAMBIAR VENTANA

F6 - CAMBIAR TEXTO/NUMERICO, F7 - AJUSTAR, F8 - CALC/TODO

F9 - FILA ó COLUMNA SIGUIENTE, F10 - EJECUTAR

Selección de celdas

La selección de celdas puede hacerse de dos formas:

1.Posicionando el cursor sobre la celda correspondiente.

-Teclas de movimiento del cursor.

-Tecla de tabulación.Se situa en la posición indicada a continuación.

-Tecla <RET> mueve a la siguiente celda según el orden establecido (**Establecer ----> Orden ----> ID/AA**) si se ha previsto el avance automatico (**Establecer**

----> Avance ----> +).

-F9 mueve a la siguiente celda segun el orden establecido.

2.Indicando en la linea de entrada, en respuesta a los comandos que lo necesiten, la referencia a la celda ó rango de celdas.

-Una celda por las coordenadas de columna/fila.Por ejemplo B7,C700,AL25,etc.

-Una linea ó columna por su propia referencia.Por ejemplo B,AL,700,25,etc.

-Un área ó rango por las coordenadas de las dos celdas extremas separadas por dos puntos.Por ejemplo B7:C700,AL25:AL50,etc.

COMANDOS

ACTUALIZAR: Se utiliza para realizar simulaciones, ya que permite hacer cambios temporales en los datos numericos constantes, (para calculos y formulas **copiar ---> valores**) y recalcular el modelo,pudiendo despues restaurar los valores originales.El calculo se expresa comenzando por el signo "!".La restitución de los valores iniciales con "=".

AUTO: Salta automaticamente a las celdas previamente seleccionadas (atributo **forma <--- formato**).

BLANCO: Borra una celda ó área sin reajustar el modelo.Simplemente deja en **blanco** los contenidos.No se pueden dejar en blanco entradas protegidas.

BORRAR: Suprime una o más lineas ó columnas enteras del modelo, reajustandolo tras el borrado.

FILA

COLUMNA

COPIAR: Permite copiar entradas de un lugar a otro del modelo.Cuando en el contenido hay formulas de calculo,existen 2 opciones:

VALORES: Copia solamente el valor de la celda.

TODO: Copia el valor de la celda y la formula que permite su obtención.Admite 3 posibilidades:

RELATIVA: La copia se hace modificando las referencias a coordenadas de acuerdo con la nueva posición de la expresión.

ABSOLUTA: Las referencias a coordenadas son las de la formula original.

PREGUNTADA: Permite escoger entre copiar **relativa** ó **absolutamente** para cada una de las variables de la formula.

RELATIVA

ABSOLUTA

EDITAR: Permite realizar correcciones sobre el contenido de una celda.

<--- --->: Movimiento del cursor.

INS: Inserta 1 espacio.

DEL: Borra 1 caracter.

F6: Pasa de numerico a texto y viceversa.

ESTABLECER: Permite preparar las caracteristicas generales para la **Entrada** de información.

ORDEN: Controla la dirección en que se mueve el puntero (ID/AA).

AVANCE: Establece si el avance definido en **orden** es ó no automatico(+) al introducir una entrada.

ATRIBUTOS: Permite prefijar los atributos para las entradas, de acuerdo con los de una celda de referencia ó los asumidos por defecto.

TEXTO

NUMERICO

TOMAR ATRIBUTOS

ATRIBUTOS POR DEFECTO

MENU: Selecciona el modo de introducir los comandos de la Hoja de Calculo.

VISUALIZA

NO VISUALIZA

MODO: Predispone el estado al que regresa tras la ejecución de un comando.

ENTRADA

COMANDO

EXTERNO: Establece asociaciones de canal entre un modelo principal y hasta un máximo de otros 3 modelos.

ABRIR/CERRAR CANAL #

FORMATO: Define las caracteristicas de la información.

ANCHURA: Permite modificar la anchura de la columna/s que especifiquemos (de 0 a 72 espacios).Con 0, la información numerica previa se hace invisible, pero se puede referenciar.Pueden definirse hasta un máximo de 32 columnas por pantalla.

ATRIBUTOS: Define las caracteristicas de un dato .Existen 8 atributos, que podemos modificar, situandonos en el interrogante posicional correspondiente y recorriendo ciclicamente las diferentes alternativas con **<cambiar> (F6).**

????????

12345678

1.Ajustar: A derecha(D),izquierda(I),centro(C),

_izquierda(i).

2.Proteger(P)/Invisible(I).

3.Forma(F)/Consolidar(C).

4.Realzar(R)/Subrayar(R)/Italica(I)/Negrilla(N).

5.Precisión: Decimales de 0 A 9, Grafico(*).

6.Signo especial ($,£,Pt) ó bien transformar en %.

7.Coma de miles(,),fecha(F),booleano(B).

8.Negativo/Positivo.

RELLENAR: Permite introducir un caracter dado, de forma repetida, en los espacios no significativos de la celda ó área que indiquemos.

TIPO_DATOS: Cambia el tipo de la información de una celda ó área especificada.

NUMERICO

TEXTO

IMPRIMIR: Permite imprimir los modelos e informes.Hay que indicar el área a imprimir y el dispositivo o fichero de salida.

CABECERA: Permite poner titulo a nuestro informe.

IMPRIMIR_AREA: Imprime el contenido del área.

VOLCAR_ECUACIONES: Se obtiene una copia impresa de todas las informaciones relativas a una celda ó área.Es decir, textos, valores fijos, formulas y especificaciones de cada una de las entradas.

ECUACIONES_REFERENCIA: Adjunta una lista de las coordenadas que hacen referencia a cada celda volcada.

SIN_REFERENCIA

INSERTAR: Crea nuevas lineas ó columnas vacias en el modelo.Las entradas existentes reajustan su posición.

FILA

COLUMNA

LOCALIZAR: Situa el cursor en la primera celda en que encuentra el **texto** indicado.Pulsando <RET> localiza la siguiente aparicion.Distingue mayusculas de minusculas.

NOMBRE: Sirve para dar un nombre a una celda ó área, que permita referencias posteriores.Puede tener hasta 9 caracteres, empezando por letra.Cuando se utilice un nombre es necesario que vaya precedido de apostrofe (').Al ejecutar **<buscar>** (F4) dentro del comando **nombre**,nos presenta una ventana con **nombre-área-indice**.

Si se selecciona un nombre existente,aparecen tres opciones:

CAMBIAR: Modifica el área asignada a un nombre.

NO_PONER: Cancela una asociación nombre-área.

INDICE: Permite asignar un indice a las entradas, en el área con nombre.Si **nombre** representa un área, al especificar **nombre [n]**, el sistema devuelve el elemento n-simo.Se puede indexar una lista identificada por un nombre **(nombre1)**, con otra lista identificada por otro nombre **(nombre2)**, mediante **nombre1(nombre2)**, que tomara el valor posicional correspondiente.

ORDENAR: Permite clasificar datos de un área o de la totalidad de la Hoja de Calculo.

PERSECUCION DE OBJETIVOS: Permite especificar los valores a conseguir ("objetivos") para una o más variables, y entonces obtener los valores de las entradas, de que dependan, que los verificarian.

VARIABLE DEPENDIENTE: Es la celda que se quiere definir como objetivo.

VALOR ACTUAL: Es el resultado para cada variable dependiente.

VALOR OBJETIVO: Es el valor que se quiere alcanzar para la variable dependiente.

VARIABLE INDEPENDIENTE: Es la celda cuyo valor se debe modificar para alcanzar el valor objetivo.

VALOR REQUERIDO: Es el valor que debe tener la variable independiente correspondiente.

ITERACION: Es el número de iteraciones que se han efectuado para llegar a los resultados.

¿ALMACENO VARIABLES INDEPENDIENTES?: Permite que los resultados se introduzcan en las celdas correspondientes, entrando el programa en recalculo.

RECALCULAR: Permite volver a calcular ecuaciones, cuando se han hecho cambios en los datos de que dependen.

RECALCULAR LAZO: Permite realizar aproximaciones sucesivas, cuando existen, en la Hoja, cálculos que no se pueden resolver directamente.En este caso hay que indicar el número de iteraciones.

TERM/GUARD: Salva en el disco la Hoja en curso, y segun se indique, continua en un ambiente de trabajo diferente.

SELECCION_MODELO: Nos situa en el **Menu de Selección de Modelo**, permitiendo continuar con otra Hoja.

OPCIONES: Sale al **Menu General del O.A.**, para pasar a otro trabajo.

CONTEXTO: Transfiere los datos de la Hoja a otro módulo del O.A.

SEGURIDAD: Permite continuar en la misma Hoja.

ESCRIBIR_EN: Copia la Hoja en curso en otra nueva, de la que hay que especificar su **nombre**, continuando en la Hoja en curso.

TRANSFERIR: Permite trasladar información desde un modelo a un fichero y viceversa.

EXPORTAR_DATOS: Transfiere de **modelo** a **fichero**.

FILAS

CULUMNAS

IMPORTAR_COLUMNAS: Transfiere de **fichero** a **modelo**.

FILAS

COLUMNAS

VENTANA: Controla la apariencia de la pantalla.

DIVIDIR: Parte la ventana actual en dos.El número máximo de ventanas abiertas es de 6.

HORIZONTALMENTE

VERTICALMENTE

UNIR: Permite unir las dos ventanas obtenidas en la ultima división.

CONECTAR: Permite recorrer dos ventanas juntas al mover el cursor en una de ellas.Para desconectar, se utiliza el mismo comando y la conexión se efectua con la **ventana 0**.

TITULOS: Permite visualizar la **fila 1**, la **columna A** ó ambas (son las que habitualmente contienen los títulos), desde cualquier posición de la Hoja.

SIN_TITULOS: Hace desaparecer los visualizados con anterioridad.

FILA 1

COLUMNA A

AMBAS

BORDE: Rediseña la pantalla, situando en el angulo superior izquierdo la celda seleccionada.Para volver a la situación anterior, se pulsa TAB.

VISTA: Permite ver todas las ecuaciones asociadas con un modelo, en una ventana dada.

ECUACIONES: Permite ver, alternativamente (<F10>), ecuaciones ó contenidos de las celdas.

CANAL_EXTERNO: Permite ver varios canales simultaneamente.Previamente han de abrirse los canales con el comando **Externo.**

GUIA DE OPERACION

El objeto de esta guía es iniciar el uso de la Hoja de Calculo, siguiendo el desarrollo de un sencillo ejemplo, que permita ir introduciendo los comandos basicos y familiarizarnos con el modo de proceder de este módulo del Open Access.

El ejemplo va a consistir en la situación presentada por una empresa (ATENE S.A.), sobre la que iremos estudiando diferentes supuestos, que nos conduzcan a la realización de las operaciones fundamentales de un proceso de calculo y de previsiones.

Vamos a desarrollar los siguientes apartados:

1.Acceso a la Hoja de Calculo.

2.Creación de un modelo de Hoja de Calculo.

3.Impresión de un modelo.

4.Comunicación entre modelos.

5.Utilización de funciones.

6.Obtención de previsiones.

7.Comunicación con otros módulos del Open Access.

1.ACCESO A LA HOJA DE CALCULO

a)**Con dos unidades de diskette.**

Desde ambiente del sistema operativo (A>), introducir el disco 1 del Open Access en la unidad A y teclear OA y <RET>.Aparecera una pantalla como la de la figura 1.

```
Open Access    E1.01    COMPUTEC
(C) 1984 Software Products International Inc.

Fecha de hoy:  16/12/85
```

figura 1

Actualizar ó confirmar la fecha y pulsar **<RET>**, la pantalla presentara el Menú de **OPCIONES** del Open Access (figura 2).

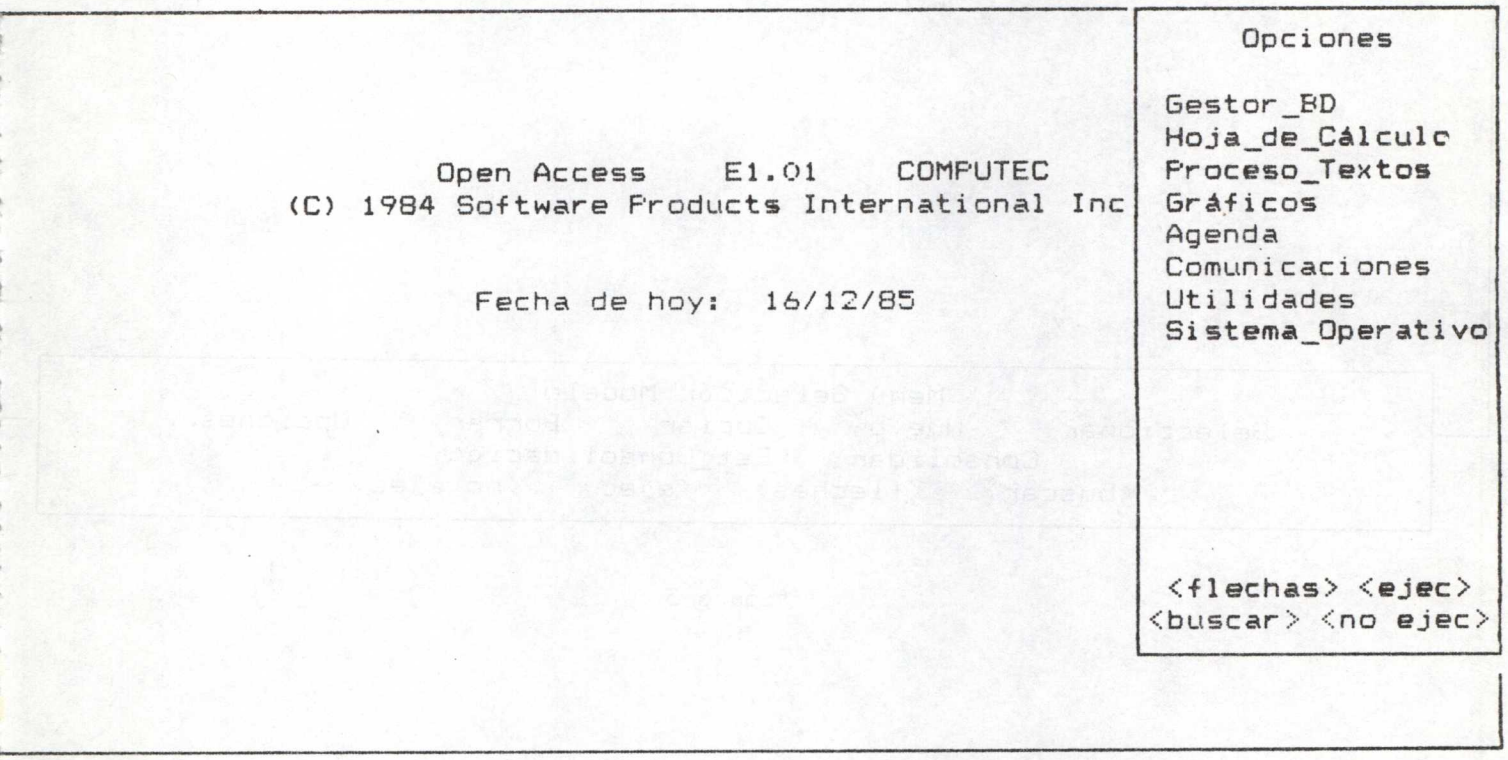


figura 2

Situar el cursor en Hoja de Calculo y pulsar **<RET>** (ó pulsar hc).El Open Access pedirá el disco de Hoja de Calculo (Codigo_3), que habrá que introducir en la unidad A, sustituyendo al de arranque, pulsando a continuación **<ejec>** (F10).La pantalla presentará el **Menú de selección de modelo** (figura 3).

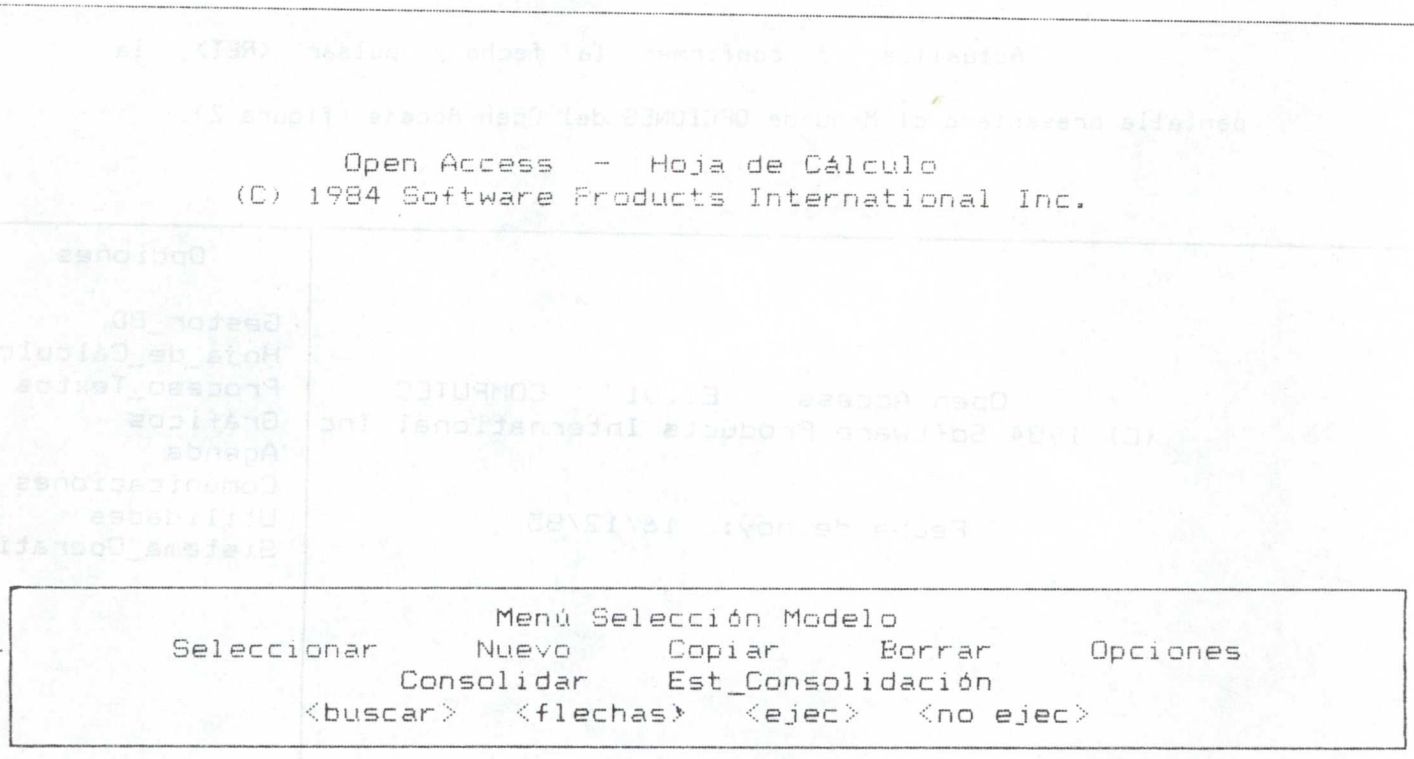


figura 3

Introducir el disco de usuario en la unidad B, y ya nos encontramos en condiciones de trabajar con la Hoja de Calculo.

b)**Con unidad de disco duro.**

El proceso es análogo, con las siguientes diferencias:

-El punto de partida es la situación en el directorio del Open Access.

-El unico diskette necesario es el de arranque, ya que los módulos del Open Access y el área de trabajo del usuario se encuentran en el disco duro.

2.CREACION DE UN MODELO.

Vamos a crear un modelo de Hoja de Calculo como el siguiente:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4	TOTAL
UNID.VEND.	200.0	250.0	160.0	400.0	1010.0
P.V.P.	50.0	50.0	50.0	50.0	50.0
VENTAS B.	10000.0	12500.0	8000.0	20000.0	50500.0
DESC.CLI.	750.0	937.5	600.0	1500.0	3787.5
VENTAS N.	9250.0	11562.5	7400.0	18500.0	46712.5
COMIS.VEND.	1156.3	1445.3	925.0	2312.5	5839.1
NETO	8093.8	10117.2	6475.0	16187.5	40873.4

figura 4

Para ello, desde el **Menú de Selección de Modelo**, situar el cursor en el comando **Nuevo** y pulsar **F10**.La Hoja de Calculo pedirá el nombre del modelo a crear (figura 5), y al contestar ATENESA incluirá automaticamente la unidad asumida por defecto, según la configuración establecida, y la extensión correspondiente a este módulo (.FMD).Así mismo, nos permitirá, si lo deseamos, incluir una palabra clave para protejerlo.

```
Nombre del modelo a crear: B:ATENESA.FMD

Palabra clave para nuevo modelo, <ret> = ninguna:
```

figura 5

Terminadas las especificaciones para la denominación del modelo, la pantalla nos presentará la Hoja de Calculo vacia (figura 6).

```
   A       |B          |C          |D          |E          |F          |
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18

Mod.: B:ATENESA                    99.3%  Puntero: A1      Actual: A1        ID  V

      entrada:
```

figura 6

Con F2 pasar del modo **Entrada** al modo **Comandos** (figura 7) y seleccionando **Establecer** definir las caracteristicas básicas para iniciar la entrada de datos.(Por ejemplo: +AA, visualizar menú, modo entrada).

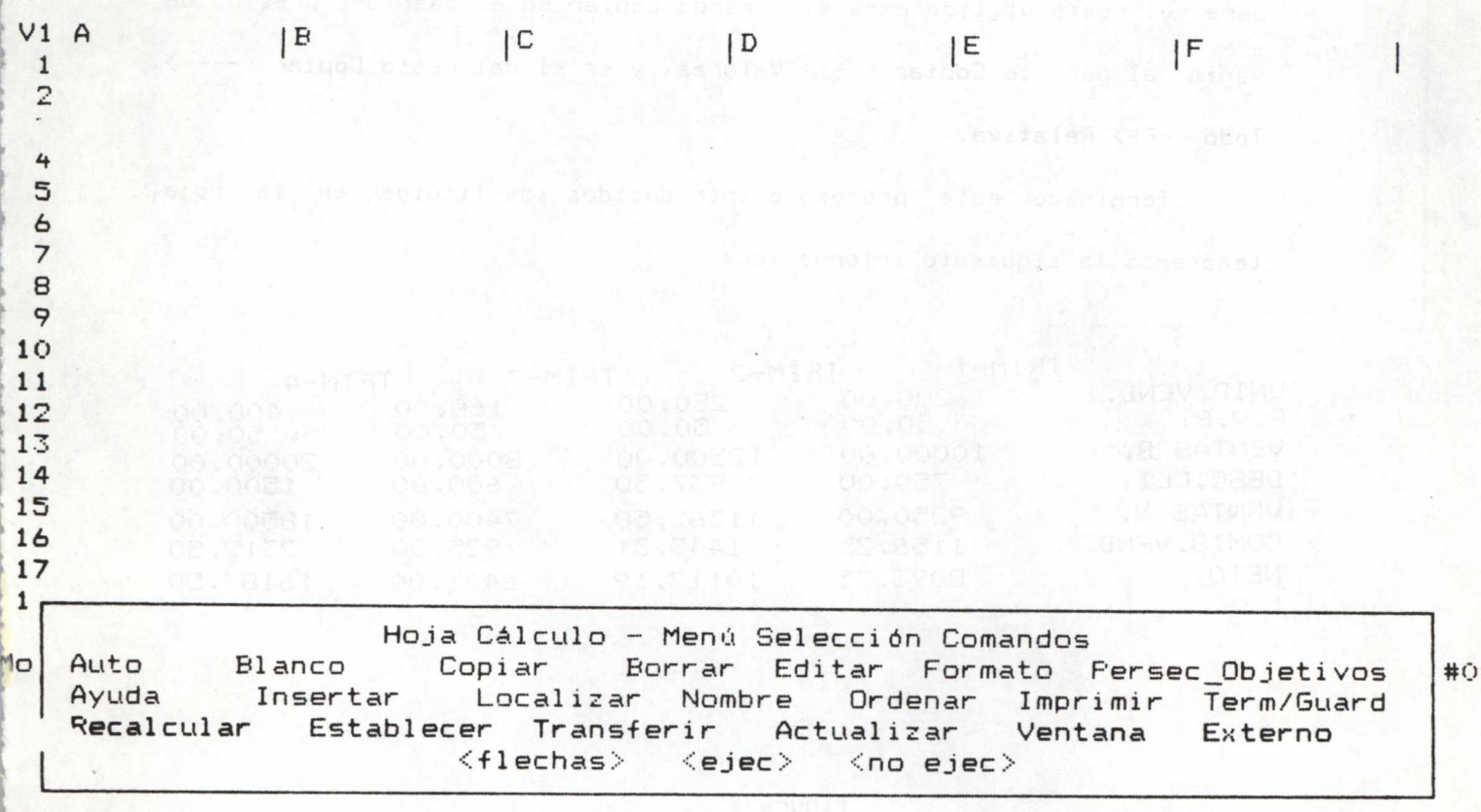


figura 7

Introducir datos numericos correspondientes a las unidades vendidas y precio de venta al publico del primer trimestre.Para los siguientes datos de este trimestre, introducir la expresión de calculo, es decir, para ventas brutas +B2*B3, para descuento a clientes 7.5%B4, para ventas netas +B4-B5, para comisiones vendedor 12.5%B6 y para neto +B6-B7.Observese la necesidad de incluir el signo

+ delante de las expresiones que comiencen con letra para evitar que sean interpretadas como texto.(Tambien puede cambiarse el caracter de la entrada con F6).

En las columnas correspondientes a los otros trimestres solamente introduciremos los datos de la fila unidades vendidas, ya que para el resto utilizaremos el comando **Copiar**.En el caso del precio de venta al publico **Copiar** ----> **Valores**, y en el del resto **Copiar** ----> **Todo** ----> **Relativa**.

Terminado este proceso e introducidos los titulos, en la hoja tendremos la siguiente información:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4
UNID.VEND.	200.00	250.00	160.00	400.00
P.V.P.	50.00	50.00	50.00	50.00
VENTAS B.	10000.00	12500.00	8000.00	20000.00
DESC.CLI.	750.00	937.50	600.00	1500.00
VENTAS N.	9250.00	11562.50	7400.00	18500.00
COMIS.VEND.	1156.25	1445.31	925.00	2312.50
NETO	8093.75	10117.19	6475.00	16187.50

figura 8

Observense los valores asumidos por defecto por la Hoja de Calculo al introducir la información en las celdas.Las que contienen texto quedan ajustadas a la izquierda y las de datos numericos se ajustan a la derecha e incluyen 2 decimales.

A continuación vamos a crear una nueva columna que totalice los 4 trimestres.Para ello solamente introducimos la expresión de la fila unidades vendidas (+B2+C2+D2+E2), y volvemos a aprovechar las facilidades del comando **Copiar**.Obtendremos en la hoja:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4	TOTAL
UNID.VEND.	200.00	250.00	160.00	400.00	1010.00
P.V.P.	50.00	50.00	50.00	50.00	50.00
VENTAS B.	10000.00	12500.00	8000.00	20000.00	50500.00
DESC.CLI.	750.00	937.50	600.00	1500.00	3787.50
VENTAS N.	9250.00	11562.50	7400.00	18500.00	46712.50
COMIS.VEND.	1156.25	1445.31	925.00	2312.50	5839.06
NETO	8093.75	10117.19	6475.00	16187.50	40873.44

figura 9

Finalizada la entrada de información de nuestro modelo, vamos a preocuparnos de la presentación, para conseguir que quede como el de la figura 4.

En los lugares donde tenemos que incluir - ó *, tenemos que intercalar nuevas filas ó columnas, lo cual se realiza con el comando **Insertar**.Como la anchura por defecto de las columnas es de 12 espacios, y nosotros solo necesitamos 1 para rellenar de asteriscos, tenemos que estrechar las 4 columnas insertadas con **Formato ----> Anchura**.

Una vez creados los espacios adecuados, los simbolos elegidos como separadores se introducen con gran facilidad gracias a **Formato ----> Rellenar**.

Tambien necesitamos, encolumnar de forma adecuada, los titulos con los datos numericos, para lo cual disponemos de la tecla de función F7, ó bien de **Formato ----> Atributos ----> Ajustar**.El uso de F7 obliga a situarnos en cada celda y realizar el ajuste una a una, el comando **Formato** permite realizar el ajuste de todo el área de una vez.

Tambien modificamos el número de decimales de de nuestros datos,reduciendolos a 1, para lo cual utilizamos **Formato ---->**

Atributos ----> Decimales (5˜ ?).

Finalizado totalmente nuestro modelo, tanto en la entrada de datos como en los detalles de presentación, lo guardamos en el disco para usos posteriores.Esto lo hacemos con el comando **Term/Guard.**

3.IMPRESION DE UN MODELO.

Suponiendo que iniciamos nueva sesión de trabajo, desde el **Menu de Selección de Modelo**, con el comando **Seleccionar**, accedemos al modelo creado.(En nuestro caso, de momento, solo tenemos ATENESA, pero cuando haya varios y no recordemos todos los nombres, usaremos **F4**).

Una vez situados dentro del modelo, podriamos realizar una "fotocopia" del contenido de la pantalla pulsando **Mayusculas-PrtSc**, como en cualquier otro ambiente de trabajo, pero la Hoja de Calculo tiene el comando **Imprimir**, especialmente pensado para esta función.

Con este comando, podemos poner un titulo a nuestro modelo, **Imprimir ----> Cabecera**, que encabece el contenido de un modelo (supongamos que queremos indicar que las ventas corresponden a Madrid), y por supuesto imprimir el contenido del modelo, **Imprimir ----> Imprimir_Area**, (figura 10).

VENTAS DE ATENE S.A. EN EL EJERCICIO 1984 EN LA AUTONOMIA DE MADRID

	TRIM-1	TRIM-2	TRIM-3	TRIM-4	TOTAL
UNID.VEND.	200.0	250.0	160.0	400.0	1010.0
P.V.P.	50.0	50.0	50.0	50.0	50.0
VENTAS B.	10000.0	12500.0	8000.0	20000.0	50500.0
DESC.CLI.	750.0	937.5	600.0	1500.0	3787.5
VENTAS N.	9250.0	11562.5	7400.0	18500.0	46712.5
COMIS.VEND.	1156.3	1445.3	925.0	2312.5	5839.1
NETO	8093.8	10117.2	6475.0	16187.5	40873.4

figura 10

Tambien permite obtener información, en un listado, sobre las caracteristicas de la infomación almacenada, **Imprimir ----> Volcar_Ecuaciones**.En donde se indica el tipo (numerica ó de texto), el ajuste, y en informaciones numericas el número de decimales, la expresión que ha permitido obtener esos datos, y si así se le indica, la lista de las coordenadas que hacen referencia a cada celda volcada.

```
A2 TXTiI          ¿ UNID.VEND.
A3 TXTiI          ¿ P.V.P.
A4 TXTiI          ¿ VENTAS B.
A5 TXTiI          ¿ DESC.CLI.
A6 TXTiI          ¿ VENTAS N.
A7 TXTiI          ¿ COMIS.VEND.
A8 TXTiI          ¿ NETO
B1 TXTiI          ¿ TRIM-1
B2 NUMiD  2  -¿ 200=                200.00
B3 NUMiD  2  -¿ 50=                  50.00
B4 NUMiD  2  -¿ +B2*B3=           10000.00
B5 NUMiD  2  -¿ 7.5%B4=             750.00
B6 NUMiD  2  -¿ +B4-B5=            9250.00
B7 NUMiD  2  -¿ 12.5%B6=           1156.25
B8 NUMiD  2  -¿ +B6-B7=            8093.75
C1 TXTiI          ¿ TRIM-2
C2 NUMiD  2  -¿ 250=                250.00
C3 NUMiD  2  -¿ 50=                  50.00
C4 NUMiD  2  -¿ +C2*C3=           12500.00
C5 NUMiD  2  -¿ 7.5%C4=             937.50
C6 NUMiD  2  -¿ +C4-C5=           11562.50
C7 NUMiD  2  -¿ 12.5%C6=           1445.31
C8 NUMiD  2  -¿ +C6-C7=           10117.19
D1 TXTiI          ¿ TRIM-3
D2 NUMiD  2  -¿ 160=                160.00
D3 NUMiD  2  -¿ 50=                  50.00
D4 NUMiD  2  -¿ +D2*D3=             8000.00
D5 NUMiD  2  -¿ 7.5%D4=              600.00
D6 NUMiD  2  -¿ +D4-D5=             7400.00
D7 NUMiD  2  -¿ 12.5%D6=             925.00
D8 NUMiD  2  -¿ +D6-D7=             6475.00
E1 TXTiI          ¿ TRIM-4
E2 NUMiD  2  -¿ 400=                400.00
E3 NUMiD  2  -¿ 50=                  50.00
E4 NUMiD  2  -¿ +E2*E3=            20000.00
E5 NUMiD  2  -¿ 7.5%E4=             1500.00
E6 NUMiD  2  -¿ +E4-E5=            18500.00
E7 NUMiD  2  -¿ 12.5%E6=           2312.50
E8 NUMiD  2  -¿ +E6-E7=            16187.50
F1 TXTiI          ¿ TOTAL
F2 NUMiD  2  -¿ +B2+C2+D2+E2=                1010.00
F3 NUMiD  2  -¿ +B3=                  50.00
F4 NUMiD  2  -¿ +B4+C4+D4+E4=               50500.00
F5 NUMiD  2  -¿ +B5+C5+D5+E5=                3787.50
F6 NUMiD  2  -¿ +B6+C6+D6+E6=               46712.50
F7 NUMiD  2  -¿ +B7+C7+D7+E7=                5839.06
F8 NUMiD  2  -¿ +B8+C8+D8+E8=               40873.44
```

figura 11

1. CREACION DE UN MODELO

Vamos a crear un modelo de Hoja de Calculo como el siguiente:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4	TOTAL
UNID.VEND.	200.0	250.0	160.0	400.0	1010.0
P.V.P.	50.0	50.0	50.0	50.0	50.0
VENTAS B.	10000.0	12500.0	8000.0	20000.0	50500.0
DESC.CLI.	750.0	937.5	600.0	1500.0	3787.5
VENTAS N.	9250.0	11562.5	7400.0	18500.0	46712.5
COMIS.VEND.	1156.3	1445.3	925.0	2312.5	5839.1
NETO	8093.8	10117.2	6475.0	16187.5	40873.4

Para ello, desde el **Menú de Selección de Modelo**, situar el cursor en el comando **Nuevo** y pulsar **F10**.La Hoja de Calculo pedirá el nombre del modelo a crear (figura 5), y al contestar ATENESA incluirá automaticamente la unidad asumida por defecto, según la configuración establecida, y la extensión correspondiente a este modulo (.FMD).Así mismo, nos permitirá, si lo deseamos, incluir una palabra clave para protejerlo.

```
Nombre del modelo a crear: B:ATENESA.FMD
Palabra clave para nuevo modelo, <ret> = ninguna:
```

Terminadas las especificaciones para la denominación del modelo, la pantalla nos presentara la Hoja de Calculo vacia (figura 6).

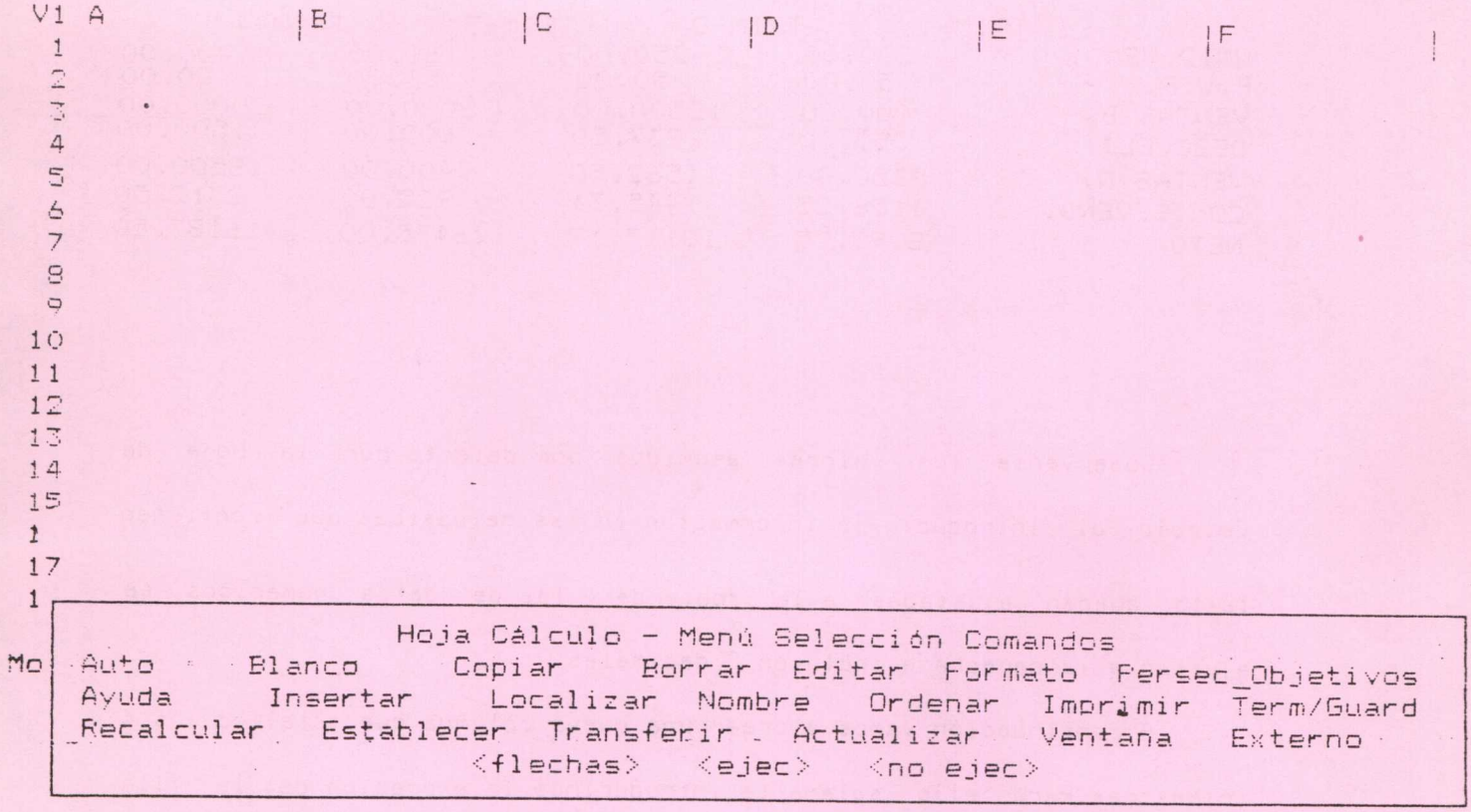
Terminadas las especificaciones para la denominación del modelo, la pantalla nos presentará la Hoja de Calculo vacia (figura 6).

```
V1 A          |B          |C          |D          |E          |F          |
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18

Mod.: B:ATENESA                99.3%  Puntero: A1      Actual: A1      ID  V:1

      entrada:
```

Con F2 pasar del modo **Entrada** al modo **Comandos** (figura 7) y seleccionando **Establecer** definir las caracteristicas básicas para iniciar la entrada de datos.(Por ejemplo: +AA, visualizar menú, modo entrada).



Introducir datos numericos correspondientes a las unidades vendidas y precio de venta al publico del primer trimestre.Para los siguientes datos de este trimestre, introducir la expresión de calculo, es decir, para ventas brutas +B2*B3, para descuento a clientes 7.5%B4, para ventas netas +B4-B5, para comisiones vendedor 12.5%B6 y para neto +B6-B7.Observese la necesidad de incluir el signo + delante de las expresiones que comiencen con letra para evitar que sean interpretadas como texto.(Tambien puede cambiarse el caracter de la entrada con F6).

En las columnas correspondientes a los otros trimestres solamente introduciremos los datos de la fila unidades vendidas, ya que para el resto utilizaremos el comando **Copiar**.En el caso del precio de venta al publico **Copiar ----> Valores**, y en el del resto **Copiar ----> Todo ----> Relativa.**

Terminado este proceso e introducidos los titulos, en la hoja tendremos la siguiente información:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4
UNID.VEND.	200.00	250.00	160.00	400.00
P.V.P.	50.00	50.00	50.00	50.00
VENTAS B.	10000.00	12500.00	8000.00	20000.00
DESC.CLI.	750.00	937.50	600.00	1500.00
VENTAS N.	9250.00	11562.50	7400.00	18500.00
COMIS.VEND.	1156.25	1445.31	925.00	2312.50
NETO	8093.75	10117.19	6475.00	16187.50

Observense los valores asumidos por defecto por la Hoja de Calculo al introducir la información en las celdas.Las que contienen texto quedan ajustadas a la izquierda y las de datos numericos se ajustan a la derecha e incluyen 2 decimales.

A continuación vamos a crear una nueva columna que totalice los 4 trimestres.Para ello solamente introducimos la expresión de la fila unidades vendidas (+B2+C2+D2+E2), y volvemos a aprovechar las facilidades del comando **Copiar**.Obtendremos en la hoja:

	TRIM-1	TRIM-2	TRIM-3	TRIM-4	TOTAL
UNID.VEND.	200.00	250.00	160.00	400.00	1010.0
P.V.P.	50.00	50.00	50.00	50.00	50.0
VENTAS B.	10000.00	12500.00	8000.00	20000.00	50500.0
DESC.CLI.	750.00	937.50	600.00	1500.00	3787.5
VENTAS N.	9250.00	11562.50	7400.00	18500.00	46712.5
COMIS.VEND.	1156.25	1445.31	925.00	2312.50	5839.0
NETO	8093.75	10117.19	6475.00	16187.50	40873.4

Finalizada la entrada de información de nuestro modelo, vamos a preocuparnos de la presentación, para conseguir que quede como el de la figura 4.

En los lugares donde tenemos que incluir - ó *, tenemos que intercalar nuevas filas ó columnas, lo cual se realiza con el comando **Insertar**.Como la anchura por defecto de las columnas es de 12 espacios, y nosotros solo necesitamos 1 para rellenar de asteriscos, tenemos que estrechar las 4 columnas insertadas con **Formato ----> Anchura.**

Una vez creados los espacios adecuados, los simbolos elegidos como separadores se introducen con gran facilidad gracias a **Formato ----> Rellenar.**

Tambien necesitamos, encolumnar de forma adecuada, los titulos con los datos numericos, para lo cual disponemos de la tecla de función F7, ó bien de **Formato ----> Atributos ----> Ajustar**.El uso de F7 obliga a situarnos en cada celda y realizar el ajuste una a una, el comando **Formato** permite realizar el ajuste de todo el área de una vez.

Tambien modificamos el número de decimales de de nuestros datos,reduciéndolos a 1, para lo cual utilizamos **Formato ----> Atributos ----> Decimales** (5° ?).

Finalizado totalmente nuestro modelo, tanto en la entrada de datos como en los detalles de presentación, lo guardamos en el disco para usos posteriores.Esto lo hacemos con el comando **Term/Guard.**

A P E N D I C E S

A) TECLADO

Teclas de Función Especiales de Open Access

FUNCIONES	TECLA COMPUTEC XP	DESCRIPCION
<ayuda>	Pulse F1	explica las posibilidades ex
<menú>	Pulse F2	accede a los comandos de mód
<impr>	Pulse F3	imprime pantalla o informe.
<buscar>	Pulse F4	encuentra cadenas de texto/f
<cambiar>	Pulse F6	cambia entrada/tipo de menú.
<calc>	Pulse F8	accede a la calculadora.
<pf1>	Pulse MAYUS y F1	Teclas de Funcion
<pf?>	Pulse MAYUS y F2	definibles por el
<pf2>	Pulse MAYUS y F3	usuario.
<macro>	Pulse Home	accede a Ejecutar/Definir.
<ejec>	Pulse F10	acepta elección en pantalla o contesta "si".
<no ejec>	Pulse ESC	rechaza elección en pantalla o contesta "no".

Teclas para Movimientos del Cursor e Introducción

FUNCION	TECLA COMPUTEC XP	DESCRIPCION
<arr>	Pulse ↑	mueve el cursor arriba.
<abj>	Pulse ↓	mueve el cursor abajo.
<izda>	Pulse ←	mueve el cursor a la izquier
<dcha>	Pulse →	mueve el cursor a la derecha.
<salto arr>	Pulse END y ↑	movimiento largo arriba.
<salto abj>	Pulse END y ↓	movimiento largo abajo.
<salto izda>	Pulse END y ←	movimiento largo izquierda.
<salto dcha>	Pulse END y →	movimiento largo derecha.
<Pg Ar>	Pulse PgUp	mueve una pantalla arriba.
<Pg Ab>	Pulse PgDn	mueve una pantalla abajo.
<tab>	Pulse →|	movimiento preseleccionado.
<tab atrás>	Pulse MAYUS y |←	movimiento preseleccionado.
<atrás>	Pulse ←	borra un caracter.
<ins>	Pulse INS	inserta entradas.
<borr>	Pulse DEL	borra entradas.
<ins lin>	Pulse END y INS	inserta parámetros/ficheros.
<borr lin>	Pulse END y DEL	borra parámetros/ficheros.
<ret>	Pulse ↵	acepta datos o ficheros.

Teclas de Función Dependientes del Módulo.

MODULO	NOMBRE FUNCION	TECLA COMPUTEC XP	DESCRIPCION
Gestor de Base de Datos	<borr ent>	Pulse F5	Borra entrada.
	<mover ent>	Pulse F7	Mueve entrada.
	<calc>	Pulse F8	Evaluar.
	<entrada>	Pulse F9	Hacer entrada (situar).
Hoja de Cálculo	<ventana>	Pulse F5	Ventana.
	<ajustar>	Pulse F7	Ajustar.
	<calc/todo>	Pulse F8	Calcular/todo o Imprimir/ todo.
	<sig col/fila>	Pulse F9	Comienzo de la siguiente columna o fila.
Proceso de Textos	<par adel>	Pulse F5	Párrafo adelante.
	<par atrás>	Pulse ALT y F5	Párrafo atrás.
	<frase adel>	Pulse F7	Frase adelante.
	<frase atrás>	Pulse ALT y F7	Frase atrás.
	<formato>	Pulse F8	Formato.
	<pal adel>	Pulse F9	Palabra adelante.
	<pal atrás>	Pulse ALT y F9	Palabra atrás.
Agenda	<notas>	Pulse F5	Notas (cuaderno).
	<cambiar>	Pulse F6	Cambia (Borra Calendario).
	<fecha>	Pulse F7	Fecha (calendario).
	<sgte>	Pulse F9	Siguiente (cita).
Comunicación	<comenzar>	Pulse F5	Empieza comunicación.
	<parar>	Pulse F7	Para comunicación.
	<cortar>	Pulse F9	Corta comunicación.
Gráficos	<pintar>	Pulse F7	Pinta.

NOTA: Esta tabla está almacenada permanentemente en Open Access. Siempre que este usando Open Access puede visualizar esta tabla simplemente pulsando dos veces <ayuda>.

B) FUNCIONES

Funciones matemáticas

Función	Resultado
ABS(x)	calcula el valor absoluto de x.
CONT(lista)	calcula el número de valores de la lista.
COS(x)	calcula el coseno de x, con x en radianes.
DESV(lista)	calcula la desviación normal de los valores de la lista.
ENT(x)	calcula la parte entera de x.
EXP(x)	calcula el resultado de elevar el número e a la potencia x.
LN(x)	calcula el logaritmo natural o neperiano de x.
MAX(lista)	calcula el máximo valor de la lista.
MEDIA(lista)	calcula la media aritmética de los valores de la lista.
MIN(lista)	calcula el mínimo valor de la lista.
NO(x)	devuelve valor 0 si x es cierto y 1 si es falso.
POT(x,p)	calcula el resultado de elevar el valor x a la potencia p.
RCUAD(x)	calcula la raíz cuadrada de x.
RESTO(x,y)	calcula el resto de la división entera de x entre y.
SEN(x)	calcula el seno de x, con x en radianes
SUM(lista)	calcula el sumatorio de los valores de la lista.
VAR(lista)	calcula la varianza de los valores de la lista.

Funciones generales

FECHA.... Permite introducir la fecha en una celda

HORA..... Permite introducir la hora en una celda.

LISTA ... Devuelve un valor de una lista dando su localización.

TABLA ... Busca en una tabla un valor, y devuelve su correspondiente.

EXTRN ... Permite hacer referencias a coordenadas de otros modelos.

Funciones empresariales

DPRA = Depreciación por el Método de Suma de los Años.
DPRD = Depreciación por el Método de Compensación por Doble Desvalorización.
ESTLIN = Estimación Lineal.
PAGO = Pago.
TRI = Tasa de Rendimiento Interno.
TRIM = Tasa de Rendimiento Interno Modificada.
VA = Valor Actual.
VAA = Valor Actual de una Anualidad.
VF = Valor futuro.

PRACTICAS DE HOJA DE CALCULO

1. Crear un modelo nuevo, siguiendo el apartado 2 de la Guía de Operación.

2. Imprimir un modelo, siguiendo el apartado 3 de la Guía de Operación.

3. Practicar la utilización de la función tabla, copiando el modelo ATENESA en otro TABLATEN, y siguiendo el apartado 5 de la Guía de Operación.

4. Crear el modelo correspondiente a ura lista de alumnos, que incluya las notas de las 5 evaluaciones. Deberá tener las siguientes columnas:

   APELLIDOS NOMBRE EV1 EV2 EV3 EV4 EV5 NOTA_FINAL

   Una vez introducidos los datos correspondientes a uno seis o siete alumnos, realizar lo siguiente:

   a) Crear las celdas que nos den la nota final de cada alumno y las medias de la clase por cada evaluación y final.

   b) Utilizando la función Tabla, asignar la calificación literal a cada alumno.

   c) Imprimir la lista de alumnos ordenada, primero por notas finales, y después por orden alfabético.

5. Practicar la comunicación entre modelos, siguiendo el apartado 4 de la Guía de Operación.

HOJA DE CALCULO : APLICACION DIDACTICA

CUADRO DE AMORTIZACION DE UN PRESTAMO

Introducción

Esta aplicación tiene como objetivo ver las posibilidades educativas de la hoja de cálculo en la asignatura de Matemáticas de BUP y Matemáticas Comerciales de Formación Profesional.

Se trata de la obtención del cuadro de amortización de un préstamo como apoyo a la comprensión del concepto de interés compuesto y anualidades constantes de amortización.

La estructura de la hoja es válida cualquiera que sea el capital, el tipo de interés y un período de amortización no superior a cincuenta años.

Objetivos

- Cálculo de la anualidad de amortización.
- Cálculo del capital amortizado en cada período.
- Desglose de la anualidad en capital e intereses en cada momento.
- Obtención del capital a solicitar en función de un período de amortización, un tipo de interés y una anualidad determinados.
- Obtención del período de amortización correspondiente a un capital, un tipo de interés y una anualidad definidos previamente.

Descripción de la hoja de cálculo

En la siguiente figura se muestra parte de la hoja tal como aparece en la pantalla.

```
V1 A     |B                |C                |D                |E                |F
 1                                  CUADRO DE AMORTIZACION DE UN PRESTAMO
 2                                  ====================================
 3
          CAPITAL.....                    0 Pt
 5        INTERES(%)..                 0.00                    Anualidad...              0 Pt
 6        AÑOS........                    0                    ****************************
 7
 8        Total de intereses .......    0 Pt
 9 ------------------------------------------------------------------------------------------
10  Año        Capital ini          Principal          Intereses     Capital fin
11
12   1                0 Pt               0 Pt               0 Pt               0 Pt
13   2                0 Pt               0 Pt               0 Pt               0 Pt
14   3                0 Pt               0 Pt               0 Pt               0 Pt
15   4                0 Pt               0 Pt               0 Pt               0 Pt
16   5                0 Pt               0 Pt               0 Pt               0 Pt
17   6                0 Pt               0 Pt               0 Pt               0 Pt
18   7                0 Pt               0 Pt               0 Pt               0 Pt

od.: B:EJEMPLO                  69.9% Puntero: C7      Actual: C7      +AA  V:1 #0
```

entrada:

La aplicación está definida en el área A1:F61. En la parte superior aparecen el título y los valores del préstamo solicitado, tipo de interés aplicado y período de amortización. En la celda F5 se describe la cuantía de la anualidad correspondiente a estos datos de entrada como resultado de la fórmula:

$$a = \frac{Cr\ (1+r)^n}{(1+r)^n - 1}$$

esto es, C4*C5/100*POT(1+C5/100,C6)/(POT(1+C5/100,C6)-1)

Finalmente, en la celda D8 se expresa el total de intereses abonados a lo largo de todo el proceso, como suma de los contenidos de la columna D.

En la parte inferior de la hoja se representa una tabla que expresa en cada período (fila) el capital inicial, el capital amortizado, la cuantía del interés y el capital pendiente después de haber abonado la correspondiente anualidad. Las fórmulas de las celdas se muestran en la siguiente figura, que se pueden visualizar utilizando el comando VENTANA - VISTA - ECUACIONES.

```
V1 A     |B                |C                |D                |E                |F                |
 1                                  CUADRO DE AMORTIZACION DE UN PRESTAMO
 2                                  ====================================
 3
 4        CAPITAL.....                          0
 5        INTERES(%)..                          0                 Anualidad...    (C4*C5/100*
 6        AÑOS........                          0                 ***************************
 7
 8        Total de intereses .......     SUM(D12:D61)
 9 ------------------------------------------------------------------------------------------
10  Año       Capital ini        Principal       Intereses     Capital fin
1.
12    1                 C4          F5-D12          C5%B12          B12-C12
13 A12+1    A13<=C6|E12|0 D13>0|F5-D13|0          C5%B13          B13-C13
14 A13+1    A14<=C6|E13|0 D14>0|F5-D14|0          C5%B14          B14-C14
15 A14+1    A15<=C6|E14|0 D15>0|F5-D15|0          C5%B15          B15-C15
16 A15+1    A16<=C6|E15|0 D16>0|F5-D16|0          C5%B16          B16-C16
17 A16+1    A17<=C6|E16|0 D17>0|F5-D17|0          C5%B17          B17-C17
18 A17+1    A18<=C6|E17|0 D18>0|F5-D18|0          C5%B18          B18-C18

lod.: B:EJEMPLO                  69.9%  Puntero: F5      Actual: F5      +AA  V:1 #0

)UM [DP  OR,-] (C4*C5/100*POT(1+C5/100,C6))/(POT(1+C5/100,C6)-1)
Comando   <Y> para lista :
```

La celda B12 contiene el valor de C4, ya que el capital inicial en el primer año coincide con la cuantía del préstamo solicitado. Las restantes celdas de la columna B de capital inicial coinciden con el valor del capital final del período anterior, si el año correspondiente a cada fila es menor o igual al período de amortización, C6.

Las celdas de la columna D de intereses abonados se obtiene aplicando el tipo de interés del préstamo al capital pendiente de amortización al comienzo de cada período.

Las celdas de la columna C representan la parte de la anualidad que corresponde a capital amortizado y se obtienen restando de la anualidad (F5) los intereses de cada período.

Por fin, las celdas de la columna E se calculan restando el capital amortizado del capital inicial.

El formato definido en cada celda es el siguiente:

Todas las celdas numéricas está definidas como números enteros, salvo la celda C5 que define el tipo de interés cuya precisión es de 2 decimales. Las celdas que representan cantidades monetarias tienen formato . Las celdas de los valores de entrada: capital, tipo de interés y plazo de amortización tienen el atributo F para poder acceder a ellas automáticamente a través del comando AUTO, si se desea. Las celdas que contienen fórmulas están protegidas contra la escritura desde el teclado, al efecto de evitar su pérdida, y es el gestor del Open Access el que las actualiza a través del comando 'Recalcular'.

Utilización de la hoja de cálculo

1.- Obtención de la cuantía de la anualidad y cuadro de amortización para los siguientes datos:

Capital.....5.000.000

Intereses.....16

Número de años.....10

Escribimos el capital en C4, el interés en C5 y el número de años en C6. A continuación accedemos al comando 'Recalcular' sobre toda el área de trabajo (pulsando F8), obteniendo el siguiente cuadro:

CUADRO DE AMORTIZACION DE UN PRESTAMO

CAPITAL.....	5,000,000 ₧		
INTERES(%)..	16.00	Anualidad...	1,034,505 ₧
AÑOS.....	10	**********	**********

Total de intereses ..... 5,345,054 ₧

Año	Capital ini	Principal	Intereses	Capital fin
1	5,000,000 ₧	234,505 ₧	800,000 ₧	4,765,495 ₧
2	4,765,495 ₧	272,026 ₧	762,479 ₧	4,493,468 ₧
3	4,493,468 ₧	315,550 ₧	718,955 ₧	4,177,918 ₧
4	4,177,918 ₧	366,039 ₧	668,467 ₧	3,811,879 ₧
5	3,811,879 ₧	424,605 ₧	609,901 ₧	3,387,275 ₧
6	3,387,275 ₧	492,541 ₧	541,964 ₧	2,894,733 ₧
7	2,894,733 ₧	571,348 ₧	463,157 ₧	2,323,385 ₧
8	2,323,385 ₧	662,764 ₧	371,742 ₧	1,660,621 ₧
9	1,660,621 ₧	768,806 ₧	265,699 ₧	891,815 ₧
10	891,815 ₧	891,815 ₧	142,690 ₧	0 ₧

2.- Obtención del préstamo que debemos solicitar sabiendo que estamos dispuestos a pagar una anualidad de 1.500.000 durante 10 años al 16%.

Accedemos a 'Perseguir Objetivos'. Como variable dependiente escribimos F5 (anualidad).

Valor objetivo....1.500.000

Variable independiente...C4

Después de una iteración se obtiene un valor requerido de 7.249.841 pts. A continuación pulsamos Return para modificar en la hoja el valor de C4.

Si queremos obtener la tabla actualizada para este nuevo capital, salimos de la opción 'Perseguir objetivos' pulsando la tecla Esc, con lo cual accedemos automáticamente al comando 'Recalcular'. Pulsamos F8 para indicar que deseamos recalcular toda la tabla observando al cabo de unos segundos los resultados actualizados:

CUADRO DE AMORTIZACION DE UN PRESTAMO

CAPITAL.....	7,249,841 Pt	
INTERES(%)..	16.00	Anualidad... 1,500,000
AÑOS.....	10	**********

Total de intereses ..... 7,750,159 Pt

Año	Capital ini	Principal	Intereses	Capital fin
1	7,249,841 Pt	340,025 Pt	1,159,975 Pt	6,909,816 Pt
2	6,909,816 Pt	394,429 Pt	1,105,571 Pt	6,515,386 Pt
3	6,515,386 Pt	457,538 Pt	1,042,462 Pt	6,057,848 Pt
4	6,057,848 Pt	530,744 Pt	969,256 Pt	5,527,104 Pt
5	5,527,104 Pt	615,663 Pt	884,337 Pt	4,911,440 Pt
6	4,911,440 Pt	714,170 Pt	785,830 Pt	4,197,271 Pt
7	4,197,271 Pt	828,437 Pt	671,563 Pt	3,368,834 Pt
8	3,368,834 Pt	960,987 Pt	539,013 Pt	2,407,848 Pt
9	2,407,848 Pt	1,114,744 Pt	385,256 Pt	1,293,103 Pt
10	1,293,103 Pt	1,293,103 Pt	206,897 Pt	0 Pt

3.- Imaginemos que dicho préstamo resulta insuficiente y que estamos dispuestos a aumentar el plazo de amortización hasta un máximo de 15 años, manteniendo el importe de la anualidad en 1.500.000 pts. ¿Que capital podemos solicitar en este caso?

Escribimos en la celda C6 el valor 15 y accediendo nuevamente a la opción 'Perseguir Objetivos', escribimos:

Variable dependiente.....F5

Valor objetivo....1.500.000

Variable independiente...C4

Después de una iteración se muestra un valor para el capital de 8.363.184 pts.

Pulsando de nuevo Return, Esc y recalculando obtenemos la siguiente tabla:

CUADRO DE AMORTIZACION DE UN PRESTAMO

CAPITAL.....	8,363,184 ₧		
INTERES(%)..	16.00	Anualidad...	1,500,000 ₧
AÑOS.....	15	**********	

Total de intereses ..... 14,136,816 ₧

Año	Capital ini	Principal	Intereses	Capital fin
1	8,363,184 ₧	161,891 ₧	1,338,109 ₧	8,201,294 ₧
2	8,201,294 ₧	187,793 ₧	1,312,207 ₧	8,013,501 ₧
3	8,013,501 ₧	217,840 ₧	1,282,160 ₧	7,795,661 ₧
4	7,795,661 ₧	252,694 ₧	1,247,306 ₧	7,542,967 ₧
5	7,542,967 ₧	293,125 ₧	1,206,875 ₧	7,249,841 ₧
6	7,249,841 ₧	340,025 ₧	1,159,975 ₧	6,909,816 ₧
7	6,909,816 ₧	394,429 ₧	1,105,571 ₧	6,515,386 ₧
8	6,515,386 ₧	457,538 ₧	1,042,462 ₧	6,057,848 ₧
9	6,057,848 ₧	530,744 ₧	969,256 ₧	5,527,104 ₧
10	5,527,104 ₧	615,663 ₧	884,337 ₧	4,911,440 ₧

Nótese que para un incremento del préstamo de poco más de un millón de pts en el capital, se incrementa el total de intereses en casi el doble.

4.- ¿Que plazo de amortización obtendríamos para un préstamo de 8.000.000 pts., abonando una anualidad de 1.500.000 pts.?

Escribimos en la celda C4 el valor 80000000 y volvemos a acceder al comando 'Perseguir objetivos' escribiendo:

Variable dependiente.....F5

Valor objetivo....1.500.000

Variable independiente...C6

Al cabo de tres iteraciones se obtiene un plazo de 12,933477455 años. Si queremos recalcular el cuadro para este nuevo plazo, hay que tener en cuenta que si guardamos la variable independiente pulsando Return, la celda C6, por tener un formato entero, muestra el valor 13, pero el recalculo lo realiza con su valor real de 12,933477455, obteniendo unos datos que no se corresponden con la realidad. En particular no aparecen los valores reales de la fila número 24 correspondiente al último año. Estos resultados serían los siguientes:

CUADRO DE AMORTIZACION DE UN PRESTAMO

CAPITAL..... 8,000,000 Pt
INTERES(%).. 16.00
AÑOS..... 13

Anualidad... 1,500,002 Pt
****

Total de intereses ..... 11,212,933 Pt

Año	Capital ini	Principal	Intereses	Capital fin
1	8,000,000 Pt	220,002 Pt	1,280,000 Pt	7,779,998 Pt
2	7,779,998 Pt	255,202 Pt	1,244,800 Pt	7,524,797 Pt
3	7,524,797 Pt	296,034 Pt	1,203,967 Pt	7,228,763 Pt
4	7,228,763 Pt	343,400 Pt	1,156,602 Pt	6,885,363 Pt
5	6,885,363 Pt	398,343 Pt	1,101,658 Pt	6,487,019 Pt
6	6,487,019 Pt	462,078 Pt	1,037,923 Pt	6,024,941 Pt
7	6,024,941 Pt	536,011 Pt	963,991 Pt	5,488,930 Pt
8	5,488,930 Pt	621,773 Pt	878,229 Pt	4,867,157 Pt
9	4,867,157 Pt	721,256 Pt	778,745 Pt	4,145,901 Pt
10	4,145,901 Pt	836,657 Pt	663,344 Pt	3,309,244 Pt
11	3,309,244 Pt	970,523 Pt	529,479 Pt	2,338,721 Pt
12	2,338,721 Pt	1,125,806 Pt	374,195 Pt	1,212,915 Pt
13	0 Pt	0 Pt	0 Pt	0 Pt

Para obtener los valores reales hay que introducir en la celda C6 el valor 13 y recalcular de nuevo obteniendo entonces una anualidad de 1.497.473 pts como muestra la siguiente figura:

CUADRO DE AMORTIZACION DE UN PRESTAMO

CAPITAL..... 8,000,000 ₧
INTERES(%).. 16.00
AÑOS..... 13

Anualidad... 1,497,473 ₧
****

Total de intereses ..... 11,467,147 ₧

Año	Capital ini	Principal	Intereses	Capital fin
1	8,000,000 ₧	217,473 ₧	1,280,000 ₧	7,782,527 ₧
2	7,782,527 ₧	252,269 ₧	1,245,204 ₧	7,530,259 ₧
3	7,530,259 ₧	292,632 ₧	1,204,841 ₧	7,237,627 ₧
4	7,237,627 ₧	339,453 ₧	1,158,020 ₧	6,898,175 ₧
5	6,898,175 ₧	393,765 ₧	1,103,708 ₧	6,504,410 ₧
6	6,504,410 ₧	456,767 ₧	1,040,706 ₧	6,047,642 ₧
7	6,047,642 ₧	529,850 ₧	967,623 ₧	5,517,792 ₧
8	5,517,792 ₧	614,626 ₧	882,847 ₧	4,903,166 ₧
9	4,903,166 ₧	712,966 ₧	784,507 ₧	4,190,200 ₧
10	4,190,200 ₧	827,041 ₧	670,432 ₧	3,363,159 ₧
11	3,363,159 ₧	959,367 ₧	538,105 ₧	2,403,791 ₧
12	2,403,791 ₧	1,112,866 ₧	384,607 ₧	1,290,925 ₧
13	1,290,925 ₧	1,290,925 ₧	206,548 ₧	0 ₧

P R A C T I C A S

Ejercicios:

1. Calcular la anualidad y el cuadro de amortización de un préstamo de 10.000.000 a pagar en 13 años al 16% de interés.

2. Calcular el capital que debe solicitarse a una entidad financiera que aplica un tipo de interés del 15.5% anual si estamos dispuestos a pagar una anualidad de 1.000.000 durante 8 años.

3. Calcular el número de años necesarios para amortizar un préstamo de 4.000.000 si el tipo de interés a aplicar es del 12% para una anualidad de 700.000

4. Calcular el tipo de interés aplicado a un préstamo de 2.000.000 sabiendo que se amortiza en 5 años pagando una anualidad de 600.000

5. Extender el modelo PRESTAMO ampliando el plazo de amortización posible hasta 60 años. (Indicación: Utilizar el comando COPIAR-TODO-PREGUNTADA)

6. Calcular la anualidad y el cuadro de amortización del préstamo descrito en el ejercicio 1, suponiendo que se los pagos se abonan trimestralmente. (Indicación: El número de plazos de amortización será de 52 y el tipo de interés a aplicar del 4%)

EJEMPLO DE LA HOJA DE CALCULO: INDICE DE PRECIOS AL CONSUMO

Introducción.

La hoja de cálculo es una herramienta muy útil y ágil siempre que se desee trabajar con una tabla que incluya datos, sobre todo numéricos, que estén relacionados entre sí. El ejemplo que se presenta muestra una tabla con las variaciones de los precios de los productos de consumo que intervienen en la obtención del índice general de precios al consumo. A cada producto o tipo de productos se le asigna un peso que repercute en el índice resultante. Éste se calcula como media ponderada de las variaciones de los precios de los 58 productos que intervienen en la tabla. Dichos pesos están determinados en función del consumo de cada producto en el conjunto de la población. La variación de estos pesos tiene una influencia decisiva en el índice final.

Objetivos.

- Utilizar una hoja de cálculo.
- Familiarizarse con algunos comandos de la hoja de cálculo.
- Analizar los datos de una tabla y las relaciones entre los mismos.
- Organizar un trabajo de campo para la obtención de datos.
- Seleccionar los datos que resultan más relevantes.
- Resolver problemas a partir de los datos obtenidos.
- Acostumbrar al alumno al manejo de tablas estadísticas.
- Modificar los datos y observar la repercusión que produce sobre la hoja dichas modificaciones.

Este ejemplo se puede utilizar para el aprendizaje de la hoja y además se pueden realizar ciertos trabajos de campo en el área de Ciencias Sociales, algunos de los cuales se proponen al final.

Presentación y utilización de la hoja.

Inicialmente, la hoja muestra los 58 productos de consumo que intervienen en el I.P.C. en la columna B, el % de variación de los precios de los mismos a lo largo de 1984 (columna C) y el peso asignado a cada uno en % (columna D).

En primer lugar, debemos obtener la repercusión de la variación del precio de cada producto sobre el índice final. Nos situamos sobre la celda E7 e introducimos la fórmula C7*D7/100 con formato numérico (pulsando F6 o introduciendo +C7*D7/100). A continuación copiamos el contenido de esta celda sobre el área E8:E78 con el comando COPIAR - TODO - RELATIVA. Si esta operación la realizamos de una vez, nos aparece en las filas 32, 33, 44, 45, 49, 50, 57, 58, 61, 62, 67, 68, 75 y 76 la expresión ¿n? debido a que el contenido de las celdas correspondientes de las columnas C y D no es numérico. Debemos borrar estas celdas con el comando BLANCO.

En la zona inferior de la hoja aparece un resumen de la tabla por grupos de productos. La suma de los pesos de los productos de un grupo proporciona el peso del grupo y la suma de las repercusiones, la repercusión del grupo. Así, introducimos SUM(D7:D31) en la celda D85, SUM(D34:D43) en D86, SUM(D46:D48) en D87, SUM(D51:D56) en D88, SUM(D59:D60) en D89, SUM(D63:D66) en D90, SUM(D69:D74) en D91 y

SUM(D77:D78) en D92. Utilizando de nuevo el comando COPIAR - TODO - RELATIVA, copiamos el área D85:D92 sobre el área E85:E92 con lo que nos aparece en pantalla la repercusión de cada grupo de productos en el índice general. La variación de los precios en cada grupo se obtiene como media ponderada de las variaciones de los productos del grupo. La variación de los precios del grupo ALIMENTACION se calcula fácilmente introduciendo la fórmula E85*100/D85 (con formato numérico) en la celda C85. Para el resto de los grupos se copia dicha celda al área C86:C92 relativamente. Por fin, el índice general se obtiene introduciendo en la celda E94 la fórmula SUM(E85:E92).

Mediante el comando FORMATO - ATRIBUTOS establecemos la precisión de las celdas de la columna C a 1 decimal y las de las columnas D y E a 3 decimales. La celda E94 se define con 1 decimal de precisión.

Después de estas definiciones, la hoja debe quedar como sigue:

NUCLEO:	HOE
UTI:	0022
PAG.:	4

Evolución del IPC en 1984
##########################

PRODUCTO	% VARIACION	PESO en %	REPERCUSION
ALIMENTACION			
Cereales	11.9	1.739	0.207
Pan	7.3	1.964	0.143
Vacuno	13.9	2.403	0.334
Cordero	12.0	1.358	0.163
Cerdo	1.0	1.015	0.010
Pollo	20.1	1.647	0.331
Otras carnes	8.9	3.191	0.284
Pescado fresco	9.3	2.753	0.256
Preparados de pescado	12.5	1.309	0.164
Huevos	10.0	1.420	0.142
Leche	9.2	2.478	0.228
Productos lácteos	11.2	1.009	0.113
Aceites y grasas	9.1	2.077	0.189
Frutas frescas	5.4	2.185	0.118
Otras frutas	15.2	0.309	0.047
Hortalizas frescas	-5.6	1.304	-0.073
Preparados hortalizas	9.0	0.978	0.088
Patatas	-7.6	0.768	-0.058
Café, té y similares	18.8	0.840	0.158
Azúcar y derivados	10.1	1.059	0.107
Otros alimentos	16.6	0.229	0.038
Bebidas no alcohólicas	9.8	0.541	0.053
Bebidas alcohólicas	9.8	2.000	0.196
Tabaco negro	3.2	0.906	0.029
Tabaco rubio	16.3	0.368	0.060
VESTIDO			
Telas	10.4	0.346	0.036
Prendas exteriores varón	11.0	3.027	0.333
Prendas exteriores mujer	11.3	2.575	0.291
Prendas interiores varón	9.9	0.394	0.039
Prendas interiores mujer	10.1	0.584	0.059
Complementos	11.6	0.457	0.053
Calzado caballero	14.2	0.880	0.125
Calzado señora	11.8	0.551	0.065
Calzado niño	13.2	0.459	0.061
Reparación de calzado	9.8	0.122	0.012
VIVIENDA			
En alquiler	5.4	1.751	0.095
Gastos	7.9	3.797	0.300
En propiedad	5.2	8.481	0.441
MENAJE			
Muebles	8.6	3.349	0.288
Textiles	9.8	1.061	0.104
Aparatos domésticos	5.3	1.057	0.056
Menaje y repuestos	10.0	0.390	0.039
Artículos de limpieza	14.2	1.582	0.225
Servicio del hogar	11.9	0.756	0.090
MEDICINA			
Medicina	9.0	2.211	0.199
Farmacia	12.7	0.963	0.122

TRANSPORTE			
Privado	6.7	7.701	0.516
Público urbano	11.5	1.513	0.174
Público interurbano	7.4	0.568	0.042
Comunicaciones	7.1	0.563	0.040
CULTURA			
Objetos recreativos	8.0	1.388	0.111
Publicaciones	9.2	0.902	0.083
Esparcimientos	7.8	2.128	0.166
E.G.B.	10.2	1.275	0.130
B.U.P. y F.P.	10.9	0.651	0.071
Universidad	17.0	0.471	0.080
OTROS			
Uso personal	7.5	3.933	0.295
Turismo	11.5	8.261	0.950

****

RESUMEN

PRODUCTO	% VARIACION	PESO en %	REPERCUSION
ALIMENTACION	9.3	35.851	3.327
VESTIDO	11.4	9.397	1.074
VIVIENDA	6.0	14.029	0.836
MENAJE	9.8	8.195	0.802
MEDICINA	10.1	3.174	0.321
TRANSPORTE	7.5	10.345	0.772
CULTURA	9.4	6.814	0.641
OTROS	10.2	12.194	1.245

INDICE DE PRECIOS AL CONSUMO EN EL AÑO 1984 .. 9.0

Utilización de la hoja.

1. Supongamos que una familia distribuye su presupuesto anual de la siguiente manera:

Alimentación:	44%
Vestido:	10%
Vivienda:	15%
Menaje:	6%
Medicina:	3%
Transporte:	15%
Cultura:	4%
Otros:	3%

No podemos modificar directamente estos pesos en el resumen de la tabla y recalcular, pués la variación de los precios de este resumen está calculada en función de los pesos de cada producto de la parte superior de la tabla. Debemos por tanto modificar cada uno de estos pesos antes de recalcular. Introducimos entonces los siguientes pesos en la columna D:

Cereales	2.4	Viv. en alquiler	10.0
Pan	2.1	Gastos	5.0
Vacuno	2.8	Viv. en propied.	0
Cordero	1.7		
Cerdo	1.8	Muebles	2.0
Pollo	3.2	Textiles	1.0
Otras carnes	2.3	Aparat. domésticos	0.5
Pescado fresco	1.2	Menaje y repuestos	0.5
Prep. de pescado	3.1	Artic. limpieza	1.7
Huevos	2.5	Servicio hogar	0.3
Leche	2.3		
Productos lácteos	1.9	Medicina	1.0
Aceites y grasas	.1.8	Farmacia	2.0
Frutas frescas	1.7		
Otras frutas	1.2	Transp. privado	10.0
Hortalizas frescas	1.6	Público urbano	3.0
Prep. hortalizas	1.7	Público interurb.	1.5
Patatas	1.6	Comunicaciones	0.5
Café, té y sim.	1.2		
Azúcar y derivados	1.3	Objetos recreat.	0.8
Otros alimentos	1.9	Publicaciones	0.5
Bebidas no alcoh.	0.6	Esparcimientos	1.0
Bebidas alcohol.	2.1	E.G.B.	1.7
Tabaco negro	0	B.U.P. y F.P.	0
Tabaco rubio	0	Universidad	0

Telas	Ø.4	Uso personal	Ø.4
Prendas ex. varón	2.8	Turismo	2.6
Prendas ex. mujer	2.8		
Prendas in. varón	Ø.4		
Prendas in. mujer	Ø.5		
Complementos	Ø.5		
Calzado varón	Ø.8		
Calzado mujer	Ø.8		
Calzado niño	Ø.8		
Repar. calzado	Ø.2		

Ahora recalculamos toda la tabla para estos nuevos pesos observando que el índice resultante es de 9.3%.

2. Supongamos que el gobierno hubiera subido la gasolina 1Ø pts. el litro y que ésto hubiera supuesto las siguientes variaciónes anuales de los precios del transporte:

Privado	9.7
Público urbano	13.5
Público interurb.	9.Ø

Introducimos estos valores en las celdas C63, C64 y C65 respectivamente y recalculando de nuevo, obtenemos para esta familia un incremento en el índice general de 9.7%.

PRACTICAS

1. Ordenar los productos de la tabla en orden creciente a su peso.

2. Ordenar los productos de la tabla en orden creciente a su variación de precios en 1984.
(Observación: Para realizar este proceso debemos usar el comando ORDENAR; sin embargo, éste no funciona si existen celdas en blanco. Por tanto, debemos borrar mediante el comando BORRAR, las filas que contienen celdas en blanco en las columnas D y C. Para recuperar el modelo original utilizamos el comando TERMINAR/GUARDAR - SELECCIONAR - SIN ALMACENAR.)

3. Calcular el I.P.C. anual de 1984 en el supuesto del ejemplo 2 anterior con los pesos oficiales.

4. El gobierno alteró los pesos oficiales de cada producto en 1985. El IPC de este año aumentó en un 8.1 % con estos nuevos pesos. Calcular cuál hubiera sido este índice de haber mantenido los pesos de 1984.

5. Realizar una encuesta entre distintos sectores de la población (distintas clases sociales, ámbito urbano o rural, familias con o sin hijos, familias con coche propio o no, etc) para obtener la distribución del presupesto anual y por tanto, los pesos que deben ser asignados a cada producto y calcular en cada caso la variación del I.P.C. en el año 1984.

6. Confeccionar una hoja similar a ésta sobre la calidad de vida en las distintas provincias de España obteniendo de alguna fuente (Anuario de

El País, I.N.E., Banco de España, etc) los datos que se consideren interesantes relacionados con aquella: Renta per cápita, densidad de población, número de camas hospitalarias por habitante, etc., asignar pesos determinados a cada parámetro y obtener un índice de calidad de vida por provincia y global comparando los resultados de una provincia a otra.