



Elementos *software*

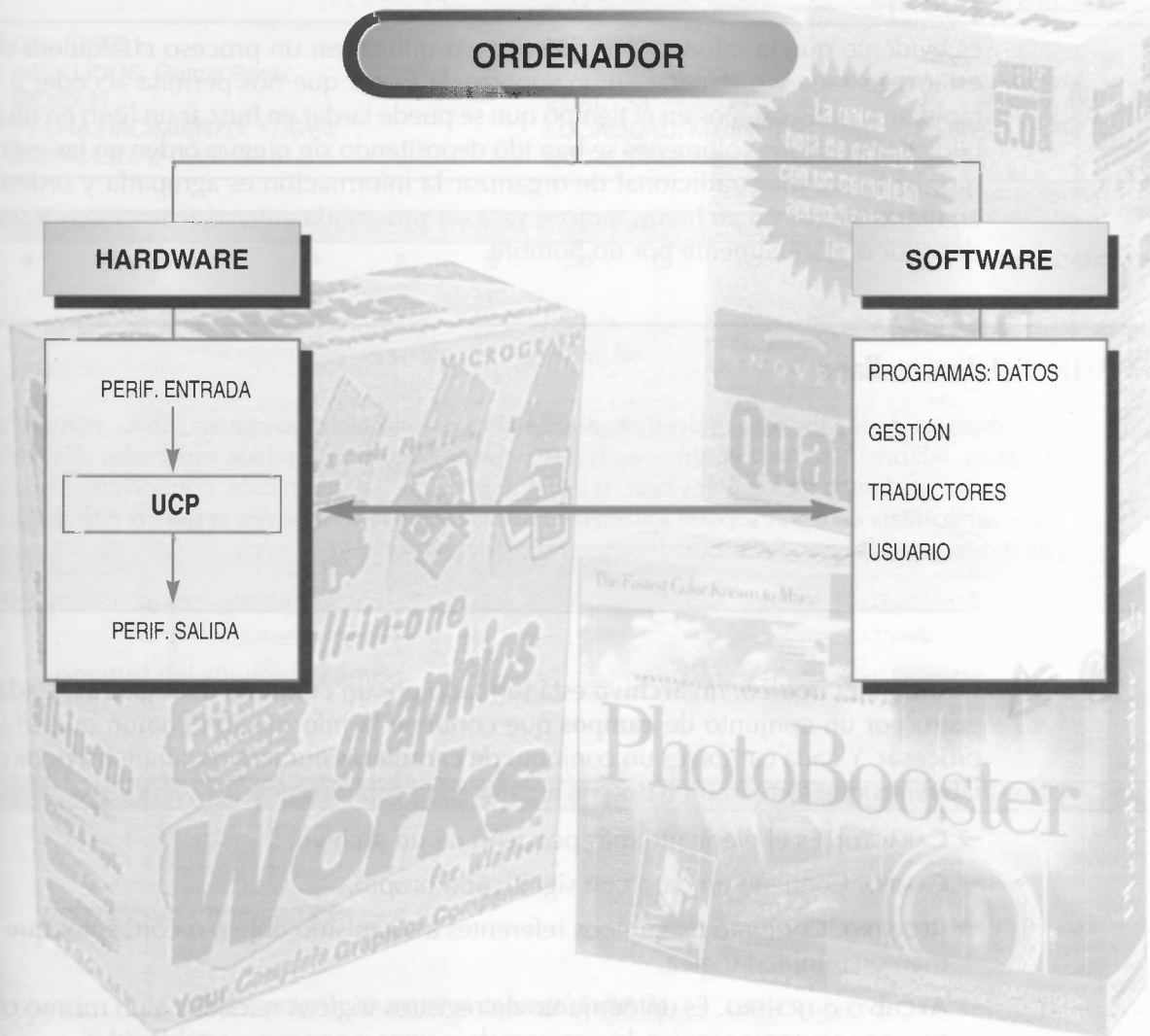
Cuando definíamos el concepto de *software* nos referíamos a la parte inmaterial del sistema informático que permite su funcionamiento de acuerdo a las necesidades de cada usuario, es decir, todos los programas que pueden ser ejecutados, tanto los que se suministran con la propia máquina (programas del sistema operativo, algunas aplicaciones, compiladores, ensambladores, etc.) como los que el propio usuario realiza con estas herramientas.

En los siguientes apartados veremos primero cómo se organizan los datos y a continuación, los diferentes **lenguajes de programación** y algunos **programas de aplicación**.

ESTRUCTURA DE CONTENIDOS



1. Concepto de archivo, registro y campo.
2. Tipos de registros y archivos.
3. Operaciones sobre los archivos y registros.
4. Concepto de programa.
5. Tipos de programas.
6. Lenguajes de programación.
7. Lenguaje máquina.
8. Lenguajes simbólicos.
9. Programas de usuario. Aplicaciones.
10. Aplicaciones comerciales.



ACTIVIDADES INICIALES

1. ¿Qué se entiende por campo, registro, archivo y programa?
2. ¿Qué diferencia hay entre registro lógico y físico?
3. ¿Qué entiendes por acceso secuencial y directo?
4. ¿Qué operaciones se pueden realizar sobre un archivo?
5. ¿Dónde se almacenan los programas para ejecutarse?
6. ¿En qué lenguaje deben estar almacenados?



Concepto de archivo, registro y campo

Es evidente que la información que se va a utilizar en un proceso cualquiera debe estar registrada y ordenada en un soporte de forma que nos permita acceder a ella rápidamente. Pensemos en el tiempo que se puede tardar en buscar un libro en una biblioteca donde los volúmenes se han ido depositando sin ningún orden en las estanterías. La forma más tradicional de organizar la información es agruparla y ordenarla en fichas y éstas en archivos, tanto si va a ser procesada automáticamente por un ordenador o manualmente por un hombre.



Actividad desarrollada

Por ejemplo, todos los datos referentes a cada libro de una biblioteca (autor, título, número de páginas, editorial, etc.) se registran en fichas y todas ellas en un archivo ordenadas alfabéticamente por autores o títulos. Pues bien, si esta información, así organizada, consideramos que ya está almacenada en algún soporte informático, a cada ficha la llamaremos **registro** y al conjunto de registros, **archivo** o fichero.



En orden jerárquico, un **archivo** está formado por un conjunto de **registros**. Cada registro por un conjunto de **campos** que contiene la información común que se va a procesar. Y cada campo es un conjunto de **caracteres** numéricos, alfanuméricos o alfabéticos que forman un **dato** con significado propio (nombre y valor).

- **CARACTER**. Es el elemento más pequeño de un archivo.
- **CAMPO**. Contiene un dato con significado propio.
- **REGISTRO**. Conjunto de campos referentes a un mismo objeto o concepto, que forman una unidad lógica.
- **ARCHIVO** o **FICHERO**. Es un conjunto de registros lógicos referidos a un mismo objeto o concepto y organizado con arreglo a unas normas preestablecidas.



Tipos de registros y archivos

Debemos distinguir dos conceptos cuando hablamos de registros:

- Registro lógico.
- Registro físico.



→ **REGISTRO LÓGICO**. Es el que se toma como una unidad de información homogénea y que se refiere a la misma entidad. Por ejemplo, los datos personales de un cliente. Estos registros vienen definidos por uno o varios campos fijos cuyo contenido es único dentro de una colección de registros (el DNI en un archivo de datos personales sería el campo identificativo o clave). Un registro debe tener definido el or-

den, la longitud (número de caracteres) y el tipo (numérico, alfabético o alfanumérico) de cada campo.

DNI: 539281	NIF: 539821H
APELLIDOS: García Pérez	NOMBRE: Ana
SEXO: H	EDAD: 33
FECHA NACIMIENTO: 17/6/64	LOCALIDAD: Madrid PROVINCIA: Madrid
DOMICILIO: c/ Gran Vía nº 5	LOCALIDAD: Madrid PROVINCIA: Madrid

DNI	APELL.	NOMBRE	SEXO	EDAD	FECHA	LOC.	PROV.	CALLE	LOC.	PROV.	LONG. CAMPOS
8	30	15	1	2	8	30	15	30	30	15	

Representación de un registro de longitud fija.

Los registros pueden ser de **longitud fija**, cuando el número de caracteres de cada campo es fijo, o **variable** en caso contrario.

8	DNI	15	APELL.	15	NOMB.	1	SEXO	2	EDAD	8	FECHA	30	LOC.	15	PROV.	30	CALLE	30	LOC.	15	PROV.	10	SALDO
---	-----	----	--------	----	-------	---	------	---	------	---	-------	----	------	----	-------	----	-------	----	------	----	-------	----	-------

↑ ↑
Longitud del siguiente campo

*	DNI	*	APELL.	*	NOMB.	*	SEXO	*	EDAD	*	FECHA	*	LOC.	*	PROV.	*	CALLE	*	LOC.	*	PROV.	*	SALDO
---	-----	---	--------	---	-------	---	------	---	------	---	-------	---	------	---	-------	---	-------	---	------	---	-------	---	-------

↑ ↑
delimitador entre campos

Representación de registros de longitud variable.

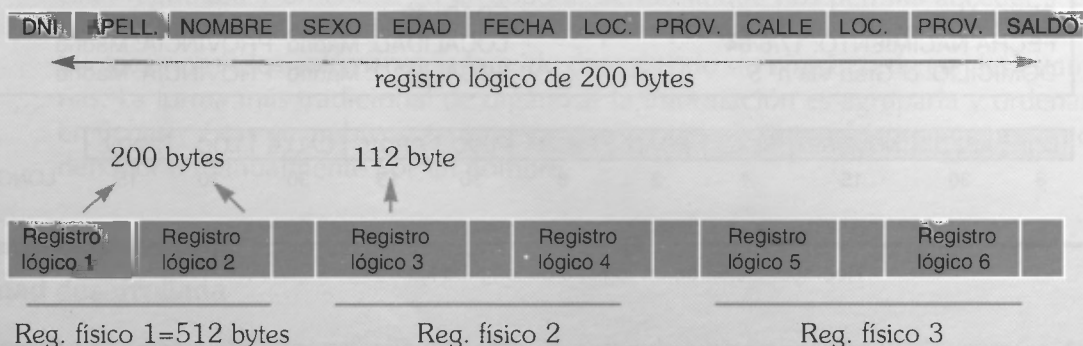
→ **REGISTRO FÍSICO.** Es la unidad de información que se transfiere desde un soporte de almacenamiento a otro mediante una sola operación de entrada o salida. El caso más típico es el traslado desde memoria principal a disco o cinta magnética o viceversa. El tamaño del registro físico depende de cada dispositivo y no tiene por qué guardar relación alguna con los registros lógicos. Es decir, su longitud puede ser menor, igual o mayor que la de un registro lógico.

{ A los archivos con información de entrada en un proceso, en general, se les llama **archivos de entrada**; a los que contienen los resultados, **archivos de salida**; y a los creados por el sistema mientras dura el proceso, **archivos temporales**. También existen archivos de *backup* o copias de seguridad que se deben obtener periódicamente de aquellos archivos con información importante para impedir que un accidente la destruya.



Actividad desarrollada

En el ejemplo de la figura vemos que el registro lógico tiene una longitud de 200 bytes y el registro físico, 512 bytes (valor típico de un sector de un disco), entonces, en cada registro físico caben dos registros lógicos, quedando vacíos 112 bytes por cada registro físico, esto es, 2 registros por bloque



Operaciones sobre los archivos y registros

Debe haber una interacción entre los archivos y el usuario, de tal forma que el sistema pueda procesar los datos que contienen y, por otro lado, el usuario pueda tener acceso a sus archivos para recuperar su información o actualizarla cuando se produzca algún cambio. Y en el caso de que no hayan sido creados, el usuario debe tener la posibilidad de crear nuevos archivos. Son siete las operaciones que pueden realizarse sobre un archivo:

- **CREAR UN ARCHIVO.** Donde se definen los registros y campos.
- **AÑADIR REGISTROS.** Ej: cuando aparece un nuevo cliente se añade un nuevo registro con los nuevos datos al archivo de clientes.
- **BORRAR REGISTROS.** Ej. cuando se da de baja a un cliente.
- **MODIFICAR REGISTROS.** Ej: un cambio de domicilio de un cliente.
- **CLASIFICAR.** Los registros de un archivo deben estar ordenados para simplificar su búsqueda según algún criterio determinado.
- **BORRAR UN ARCHIVO.** Cuando la información contenida en el archivo deja de ser válida por cualquier razón.
- **FUSIÓN.** Cuando se unen dos archivos o más para formar uno solo.
- **DIVISIÓN.** Cuando se obtienen dos archivos distintos a partir de uno.
- **CONSULTA.** Consiste en acceder a uno o varios registros para conocer su contenido o ciertas características relativas a todos los registros; por ejemplo, para calcular la edad media de los clientes.



Concepto de programa

Ya sabemos que un programa es un **conjunto de instrucciones** que se suministran al ordenador, el cual, al ejecutarlas en un determinado orden, es capaz de resolver un problema automáticamente. Para ello el ordenador debe almacenar las instrucciones dentro de su memoria principal, por lo que se le llama programa almacenado. ~~También~~ existen programas que están cableados mediante circuitos electrónicos formando parte del **hardware (Firmware)**. Estos programas realizan electrónicamente las mismas funciones que un programa en lenguaje máquina almacenado en memoria. Sólo se suelen usar en circunstancias especiales, como en el caso del arranque del sistema.

A la persona encargada de escribir las instrucciones de forma detallada se la denomina **programador**. Debe escribir todos los pasos necesarios para ejecutar una tarea sin omitir nada y de forma ordenada y lógica. El programador puede escribir las instrucciones utilizando alguno de los muchos lenguajes de programación existentes, siempre que la máquina donde va ejecutarse sea capaz de entenderlo.



Tipos de programas

1 ■ Según su estructura y orden en que se ejecutan las instrucciones:

- **PROGRAMA LINEAL**: las instrucciones se ejecutan una detrás de otra, según el orden en que fueron escritas.
- **PROGRAMA CÍCLICO**: cuando, al ejecutarse linealmente algún grupo de instrucciones, se repite hasta que se cumple determinada condición.
- **PROGRAMA ALTERNATIVO** cuando se ejecutan unas partes del programa y otras no, en función de determinadas opciones tomadas de los datos del programa.

2 ■ Según su contenido:

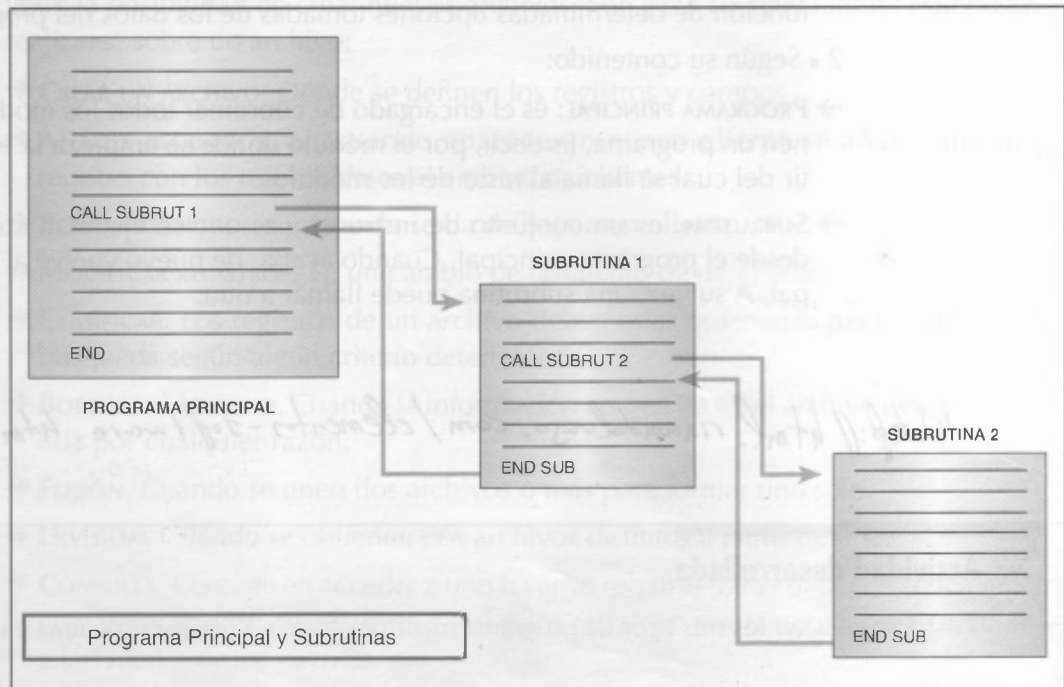
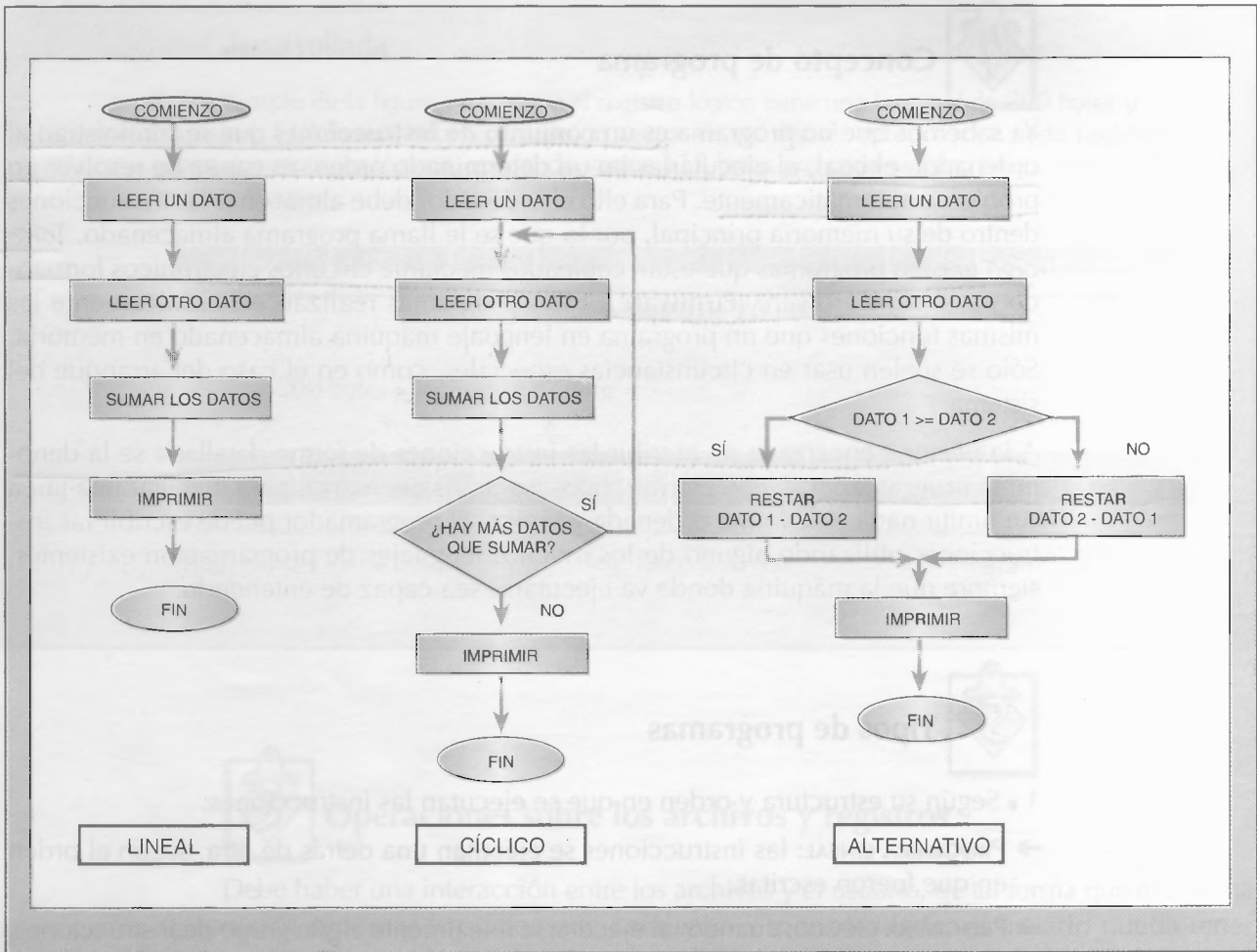
- **PROGRAMA PRINCIPAL**: es el encargado de coordinar todos los módulos que componen un programa. Es decir, por el módulo donde se empieza la ejecución y a partir del cual se llama al resto de los módulos.
- **SUBROUTINA**: es un conjunto de instrucciones que se ejecutan cuando es llamada desde el programa principal. Cuando acaba, de nuevo vuelve al programa principal. A su vez, una subrutina puede llamar a otra.

Por eventos
<http://html.roncalvaso.com/elementos-software.html>



Actividad desarrollada

Vamos a ver los tres tipos de programa mencionados para realizar una suma y una resta.





Lenguajes de programación

Un lenguaje de programación es el **enlace entre el hombre y el hardware** del ordenador. Es decir, para que el hombre y el ordenador puedan comunicarse tienen que emplear el mismo lenguaje considerando a éste con su propio alfabeto (conjunto de símbolos), sintaxis (orden de los símbolos) y semántica (significado de las frases).

Por tanto, el programador utiliza el lenguaje para escribir las órdenes que representan el algoritmo que resuelve el problema y así indicar al *hardware* las acciones que debe iniciar en cada momento. Desde este punto de vista consideraremos que **un programa es un conjunto de sentencias o instrucciones que representan un algoritmo**, donde **algoritmo** es un conjunto de reglas bien definidas para resolver un problema en un número determinado de operaciones, y donde una frase es una cadena de símbolos de un alfabeto formada de acuerdo con ciertas reglas sintácticas y construidas de forma que tengan sentido. *(reglas semánticas)*

Podemos establecer dos grupos de lenguajes dependiendo de si están orientados al ordenador o al hombre:

- Lenguaje máquina.
- Lenguaje simbólico.
 - * De alto nivel.
 - * De bajo nivel.



Lenguaje máquina

El lenguaje máquina o código máquina, es el primero que se utilizó en la programación cuando aparecieron los ordenadores. Además, es el único que entienden los ordenadores de forma que, aunque se utilicen otros más cercanos al hombre para escribir programas, al final habrá que traducirlos a código máquina para que el *hardware* entienda lo que tiene que hacer. El código máquina consiste en la representación de las instrucciones como una combinación de unos y ceros que representan cada una de las instrucciones de un programa.



Actividad desarrollada

Por ejemplo, la instrucción SUM A, 65 (suma al contenido del acumulador el valor 65) se representaría como:

SUM	65	LENGUAJE BAJO NIVEL
00100000	0110101	CÓDIGO MÁQUINA
Cod. Instrucción	Operando	



Lenguajes simbólicos

Éstos pueden estar orientados a la máquina o al hombre, en cuyo caso se les denomina lenguajes de bajo nivel o de alto nivel, respectivamente.

→ **LINGUAJES DE BAJO NIVEL.** Las instrucciones escritas en estos lenguajes guardan una cierta analogía con las instrucciones de lenguaje máquina en que posteriormente serán traducidas; las principales características de este tipo de lenguajes son:

a) El código de las instrucciones es de tipo mnemotécnico:

SUM = sumar

SUB = restar

MOV = mover

b) Las direcciones de los operandos o datos pueden ser también simbólicas, es decir, se le atribuye un nombre a cada dato y la instrucción hace referencia a ese nombre: CANTIDAD1, EDAD,...

c) Cada instrucción en lenguaje de bajo nivel origina, al traducirse, una sola instrucción en lenguaje máquina. A los lenguajes de este tipo se les llama lenguajes *ensambladores*.



Actividad desarrollada

Escribimos un programa en ensamblador para calcular el área de un triángulo sabiendo que la fórmula es la siguiente:

$$\text{SUPERFICIE} = \frac{\text{BASE} \times \text{ALTURA}}{2}$$

Donde BASE y ALTURA son variables cuyos valores son introducidos al principio del programa y deben mantenerse todo el tiempo.

ETIQUETA	MNEMÓNICO DE INSTRUCCIÓN	COMENTARIOS
DOS	EQU 2	Asigna el número 2 constante al nombre DOS.
	ORG 0	Los siguientes datos se almacenan a partir de la posición de memoria 0.
BASE	BLOCK 1	Se reserva la primera posición de memoria para almacenar el dato correspondiente a la base del triángulo y a esta posición se la llama BASE.
ALTURA	BLOCK 1	Se reserva la siguiente posición (la segunda) de memoria para almacenar el dato correspondiente a la altura del triángulo y a esta posición se la llama ALTURA.
SUPERF.	BLOCK 1	Se reserva la siguiente posición de memoria para almacenar el dato correspondiente al resultado del cálculo y a esta posición se la llama SUPERF.
	ORG 100	Las instrucciones de programa comienzan a almacenarse a partir de la posición 100 de memoria (origen = 100).

	MOVE BASE,A	Lleva el dato desde la posición de memoria que contiene la longitud de la base al acumulador de la UAL.
	MUL ALTURA,A	Multiplifica el contenido de la posición de memoria llamada ALTURA por el contenido del acumulador y el resultado de la multiplicación lo deja en el acumulador.
	DIV A,DOS	Divide el contenido del acumulador por el valor constante que hemos llamado DOS y el resultado lo deja en el acumulador.
	MOV A, SUPERF	Lleva el dato desde el acumulador a la posición de memoria que hemos llamado SUPERF.
	END	Finaliza el programa.

*Interpretado
compilado*

→ **LENGUAJES DE ALTO NIVEL.** Estos lenguajes permiten escribir instrucciones orientadas al problema que se quiere resolver, y no al ordenador que va a ejecutar el programa correspondiente. Por tal motivo, dichos lenguajes emplean terminología fácilmente comprensible y más o menos aproximada al propio lenguaje humano. Las instrucciones escritas con estos lenguajes son muy potentes, ya que una sola instrucción produce varias en código máquina al ser traducida. Además, son universales en el sentido de que los lenguajes evolucionados más comunes pueden utilizarse para diferentes ordenadores, siempre y cuando éstos posean el traductor adecuado.



Actividad desarrollada

El programa del ejemplo anterior podría escribirse en BASIC (lenguaje de alto nivel) con sólo tres instrucciones:

DOS = 2

SUPERF = BASE * ALTURA / DOS

END

Los lenguajes de programación también han evolucionado de forma que se pueden diferenciar cinco generaciones: las dos primeras se corresponden a la aparición del lenguaje máquina y al lenguaje ensamblador respectivamente. En la tercera generación aparecen los lenguajes de alto nivel como BASIC, lenguaje muy fácil; FORTRAN para resolver fórmulas matemáticas; COBOL, lenguaje para gestionar gran volumen de datos; ALGOL (resuelve algoritmos); PASCAL, C, lenguajes estructurados. En la cuarta generación aparecen los Sistemas de Manejo de Bases de Datos –DBMS: Data Base Management System– capaces de mantener independientes los datos de los programas.

5º RAD y Eventos



Programas de usuario. Aplicaciones

NO

Para resolver un problema específico con un ordenador, es necesario utilizar uno o varios programas que, al ser ejecutados, obtengan los resultados perseguidos. Al conjunto de estos programas se le denomina **aplicación informática**. Los programas que componen las aplicaciones no forman parte del sistema operativo.

Existen aplicaciones estándar en el mercado que resuelven problemas generales, es decir, un usuario puede comprar un paquete de programas (aplicación) de forma que

para resolver su problema sólo tiene que introducir, de forma guiada por la aplicación, sus datos y ejecutar el programa. Es típico en la aplicación de nóminas o de declaración de la renta, entre otros muchos.

Por otro lado, el usuario puede diseñar su propia aplicación mediante la edición y ejecución de una serie de programas de acuerdo a sus necesidades.

En cualquier caso, para implantar una aplicación en un sistema, es necesario analizar una serie de puntos y tomar una serie de decisiones antes de desarrollar la aplicación. Será preciso hacer un **análisis previo** del problema para valorar su viabilidad. Una vez tomada la decisión de proseguir con el proyecto, se comenzará por concretar el sistema o equipos que habrá que implantar, basándose en los estudios realizados en la fase inicial. A esta segunda fase se la denomina **análisis funcional**. Por último, se debe realizar un **análisis orgánico**, donde se determina la estructura y organización de la información y se especifican las aplicaciones que se desarrollarán o comprarán caso de que existan en el mercado, en función de la entrada y salida de archivos detallados en la fase anterior y dependiendo de la configuración del ordenador que se ha decidido utilizar.

Después de estas fases de estudio, comienzan las fases puras de desarrollo y depuración hasta llegar a la explotación de la aplicación:

- **PROGRAMACIÓN.** Si se ha decidido utilizar una aplicación que se puede comprar no será necesaria esta fase. Si el desarrollo lo realiza el usuario, lo normal es que los programas se repartan entre varios programadores. Una vez que ha terminado su programa, debe verificar que funciona con los datos de prueba y entregarlo al analista. Éste se encarga de probar la aplicación completa y cuando está seguro de que funciona, se entrega a explotación con la documentación apropiada.
- **EXPLOTACIÓN.** A pesar de todas las pruebas existen riesgos de que la aplicación no funcione correctamente; por ello, siempre que se empieza a explotar una aplicación, se debe hacer el proceso paralelo con el procedimiento que se estuviera empleando antes. Así podremos comparar resultados y corregir errores si los hubiera.
- **DOCUMENTACIÓN.** Para dar por terminada una aplicación, debe quedar documentado todo lo que se ha hecho durante el desarrollo y la forma de manejar la aplicación de cara al usuario final.



Aplicaciones comerciales

En general, todos los programas comerciales se acompañan de un manual con las instrucciones para su uso, y la mayoría presentan *tutoriales* o programas de aprendizaje guiado, que facilitan al usuario sus primeros contactos con el programa.

A continuación veremos las aplicaciones más utilizadas y que se pueden encontrar en el mercado.

- **PROCESADORES DE TEXTO.** Son programas que sirven para presentar documentos escritos centrando, encuadrando, moviendo párrafos y modificando el texto a gusto del usuario. Algunos también permiten la realización de *mailings*, incluyendo datos variables en un texto tipo.

Ejemplos de procesadores de texto son **WordPerfect**, de Borland y **Word** de MicroSoft, –cuyas últimas versiones son exclusivamente para uso con entorno Windows, también muy conocido. Otros procesadores menos difundidos en la actualidad son: **WordPro** de Lotus, **Writing Assistant** de la serie Assistant de IBM y **Wordstar** de Micropro.

Un apartado especial en los procesadores de texto, son aquellos que permiten un diseño conjunto de texto y gráficos, que son conocidos como editores, o **programas de autoedición**. Son ejemplos: Page Maker de Aldus y Ventura Publisher.

- **BASES DE DATOS**. Se conoce bajo este nombre a programas que gestionan las bases de datos. Éstas pueden ser **relacionales** y documentales. En las relacionales, los registros están formados por campos, tal y como hemos descrito anteriormente. En las **documentales**, los registros son documentos completos.

Son gestores de bases de datos muy conocidas los **dBase IV** y **dBase V** de Borland. Son también conocidas **Fox Pro** de Microsoft, **Paradox** de Borland, **Approach** de Lotus y **Access** de Microsoft.

- **HOJAS DE CÁLCULO**. Son programas que se utilizan para realizar cálculos basándose en unos datos que se incluyen en una cuadrícula a modo de tabla. Generalmente, incluyen funciones matemáticas, estadísticas y de confección de gráficos.

Muy difundidas son **Lotus 1-2-3** de Lotus, **Excel** de Microsoft y **Quattro Pro** de Borland.

- **GRÁFICOS Y ESTADÍSTICOS**. Son programas que permiten realizar todo tipo de gráficos a partir de unos datos, ya sean tabulados o de otro tipo. Muy conocido es **Harvard Graphics** de S.P.C., **PowerPoint** de Microsoft o **Freelance** de Lotus.

- **DISEÑO GRÁFICO**. Son utilizados para diseño de todo tipo de gráficos, más generales que los anteriores. Trazan gráficos realizando operaciones geométricas sobre las figuras, y en general, son usados en diseño industrial y arquitectura. Muy conocidos son **AutoCad** y **Corel Draw** de Corel.

- **PROGRAMAS PARA COMUNICACIÓN**. Permiten las comunicaciones entre usuarios, gestionan los módem o las redes locales.

- **ENTORNOS GRÁFICOS**. Son programas que sirven para facilitar al usuario el manejo del sistema operativo. El más conocido es el programa **Windows**.

- **UTILIDADES**. Son programas utilizados como herramientas para los procesos propios del sistema, por ejemplo, las copias de seguridad. Son representativos de este grupo **PC-Tools** de Central Point, **Norton Utilities** de Symantec, etc.

- **ANTIVIRUS**. Un virus es un programa que se reproduce a sí mismo y que se introduce de forma no deseada en los ordenadores, a través de disquetes que han sido utilizados en un ordenador infectado. Algunos, simplemente, resultan molestos pues introducen imágenes que entorpecen la visión de la pantalla. Otros borran algunos datos del disco y otros realizan en la unidad en la que se encuentran reorganizaciones tan profundas que pueden originar costosas pérdidas de información e, incluso, la inutilización de la unidad.

Para subsanar estos problemas se comercializan vacunas que detectan estos programas destructivos y los borran, corrigiendo los daños que hayan podido producirse. Una vertiente importante de estos programas antivirus es la detección. Antes de introducir cualquier disco en nuestro ordenador, es una sana costumbre *pasarle* un antivirus que nos informe sobre su estado.

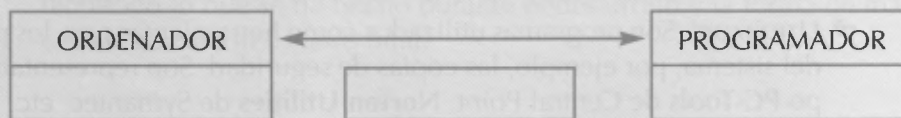
Existen multitud de antivirus en el mercado, de los cuales los más conocidos son **Central Point**, **Scan**, **Anyware** o **Panda**.

Actividades de Enseñanza-Aprendizaje



Actividades de refuerzo

1. ¿Qué es un archivo? Define los elementos de un archivo en orden jerárquico.
2. ¿Qué es un registro?
3. Si tienes en un archivo los discos de tu colección, ¿cómo serían los registros? ¿Qué campos definirías?
4. Di si es verdadero o falso:
 - Un programa es un conjunto de registros.
 - Un registro es un conjunto de archivos.
 - Todos los campos de un registro tienen igual importancia.
5. ¿En qué se diferencia un archivo lógico de uno físico?
6. Señala los diferentes tipos de organización de los registros dentro de un archivo, características y tipo de soporte típico.
7. Señala la diferencia entre acceso secuencial y directo a un archivo.
8. Define el concepto de programa.
9. ¿Qué se entiende por *firmware*?
10. ¿Cómo se llaman las personas que realizan los programas?
11. Define lenguaje de programación.
12. ¿Cuáles son las diferencias entre lenguaje máquina, lenguaje de alto nivel y lenguaje de bajo nivel?
13. Señala los diferentes lenguajes utilizados en cada generación.
14. Completa el siguiente esquema con los tipos de lenguaje que utilizan:



15. ¿Qué son Basic, Cobol y Pascal?
16. ¿Cuál de los siguientes no es un lenguaje de programación?:
Fortran, Cobol, C-Asa
17. ¿Cuál es un lenguaje de gestión?:
APL, Fortran, Algol, Cobol
18. ¿Qué es una aplicación informática?
19. ¿Cuáles son los análisis que hay que realizar antes de desarrollar una aplicación?
20. ¿Qué se entiende por usuario en Informática?
21. ¿Por qué crees que se comercializan tantos programas?

22. ¿Qué es un procesador de textos? Cita alguno.

23. ¿Qué es una hoja de cálculo?

24. ¿Qué es Windows?

Actividades de ampliación

25. Haz una lista de lenguajes de alto nivel y organízalos en los siguientes apartados:

- Lenguajes de propósito general.
- Lenguajes orientados al problema.

26. Define *virus informático* y haz una clasificación de los mismos.

27. Busca folletos sobre *software* y elabora un pequeño catálogo con los programas existentes en el mercado, sus precios, características y casas que los elaboran. Organízalo según los tipos que hemos estudiado.

28. De los folletos anteriores, intenta deducir lo que es una *actualización*.

29. Estudia los folletos anteriores e intenta extraer las características que debes mirar en un programa, además de las prestaciones que te ofrece.