Curso Autodidáctico

INFORMATICA

BASIC

LA VERSION PEDAGOGICA

CONTIENE

Microordenador, Manual de instrucciones. Curso de introducción "Manejo de ordenadores". "Lenguaje Basic". "Evaluaciones periódicas". "Diploma del curso". (Autorizado por M. E. y C.)

ATV DRAGON

Curso Autodidáctico DE INFORMATICA-BASIC

ATV

ATV DRAGON

IMPRESO EN ESPAÑA
PRINTED IN SPAIN

ISBN: 84 - 398 - 4342 - 9

Depósito Legal: MA. 499

DISEGNO, J.P. s.a. c/ Larios, 4 - 3a Planta, Ofic. 304
Tel.: 22 62 01 - MALAGA-29005

Curso Autodidáctico

INFORMATICA BASIC

el curso ATV de
Informática-Basic
ha sido realizado por:
EQUIPO ATV.

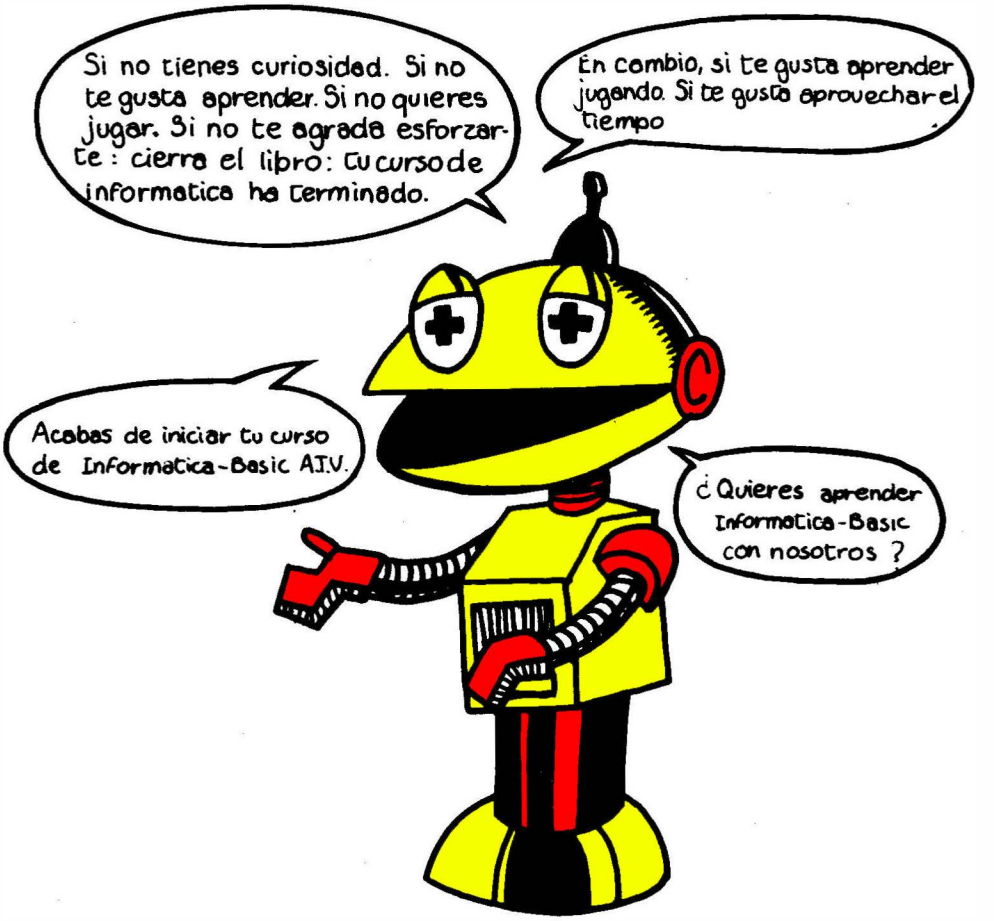
Juan A. Ferrández
Juan F. Rojas
Antonio Fernández
José L. Poveda
Juan F. Aguilar

Voces:
Francisco Linares
Mayka Lozano

Dirección:
Salvador Segura

ATV

Paseo de la Farola, 25
Telf. 22 81 79
29016 - MALAGA


Si no tienes curiosidad. Si no te gusta aprender. Si no quieres jugar. Si no te agrada esforzarte : cierra el libro: Tu curso de informatica ha terminado.
En cambio, si te gusta aprender jugando. Si te gusta aprovechar el tiempo
Acabas de iniciar tu curso de Informatica-Basic A.I.V.
¿ Quieres aprender Informatica-Basic con nosotros ?

Presentación

MEMORIA DESCRIPTIVA DE LAS ENSEÑANZAS A IMPARTIR

El Curso de Informática-Basic, es un Curso de Enseñanza Audio-Tacto-Visual. Con estas tres últimas palabras se pretende definir el modo de acceso al alumno, de comunicación, que se imparte en los cursos A.T.V., y en este caso en el de Informática-Basic.

El aprendizaje es una tarea para la que todos los seres vivos están capacitados. Aprende el que modifica su conducta, después de haber sido sujeto de una vivencia. De este modo A.T.V, **teniendo en cuenta las tres más importantes vías de acceso de la información a nuestro cerebro: vista, oído y tacto, parte del supuesto de que un curso de enseñanza Audiovisual y de autoestudio, conseguirá el máximo rendimiento en el aprendizaje, fomentando la participación del alumno en la tarea.**

La informática es una parcela "Misteriosa" del conocimiento, que se ha dado en considerar como novedosa, y que se ve envuelta en un halo de hermetismo, solo apto para iniciados. A.T.V., consciente de que ese halo ha envuelto cualquier novedad que se haya producido históricamente en el mundo del conocimiento, de la técnica, **pretende,** mediante un desenfado no exento de rigor académico, **dar a la Informática, y al ordenador, el valor de una herramienta que como la máquina de escribir o el martillo, cualquiera puede manejar, estando los resultados en función del interés personal, la curiosidad, la motivación.**

Desprovista la Informática de su carácter de mito moderno, convertida en la herramienta que "sirve para", no resulta difícil comprender que el ordenador no es más que una máquina que hay que manejar con pericia, y que la informática no es más que la ciencia que prepara los ordenadores para ser utilizados.

Para manejar un martillo no es imprescindible conocer las teorías de NEWTON. . . aunque tampoco estorbarían.

Por último, partiendo de la base de que cualquiera está capacitado para aprender Informática, para manejar un ordenador, puesto que el tratamiento de datos es tan antiguo como la vida, y teniendo en cuenta que el ordenador es una copia de la estructura lógica del cerebro, A.T.V. hace hincapié en esa cualidad: El tratamiento de datos no es una novedad, sí lo es por medio de ordenadores, por eso, y en definitiva, el curso de Informática-Basic A.T.V. pretende que el alumno comprenda la relación entre Informática, lógica y Ordenador. Nada más.

En cuanto a la mecánica del curso, se compone éste de doce cintas magnéticas, cada una de las cuales contiene una lección. Las cintas son la parte audiovisual del curso. Contienen la voz del profesor con las explicaciones y nociones teóricas, que se van ilustrando por medio de imágenes, de modo sincronizado y automático. El alumno asiste a cada clase como si estuviera viendo un programa de televisión, en el que se le diera la oportunidad de participar por medio del teclado, siempre bajo las instrucciones del profesor "Bitillo", cicerone en el viaje hacia el universo de la Informática.

Antes y después de cada lección el alumno acude al Libro A.T.V. de informática donde, por un lado se le pone en antecedentes de la lección que va a afrontar; por otro se repasa y amplía la lección que ya ha dado; por tanto, cada lección ha de ser considerada como la suma de lección audiovisual, mediante cinta magnética, y lección tradicional mediante el Libro A.T.V.

Al final de cada lección el alumno hallará un pequeño repaso de la lección, que inmediatamente habrá de resolver mediante un cuestionario de respuestas múltiples. En el libro los ejercicios se diversifican, y, desde dibujar a mano un cubo hasta diseñar un programa, se pretende en todo momento enriquecer al alumno con su propia participación.

En todo momento el alumno hallará la solución de ejercicios y cuestionarios con lo que consigue una constante retroalimentación que se ve reforzada en el hecho de que, al manejar alternativamente cinta magnética y Libro, el alumno aprende conceptos y técnicas que ya "le suenan" porque le han sido esbozados en alguno de los dos instrumentos. De este modo se facilita su labor de estudio.

En cuanto a las lecciones audiovisuales, respetan el siguiente índice:

Lección 1ª :

Conocimiento del teclado. Técnica de carga. Cloadm. Enter.

Lección 2ª :

Informática. Ordenador. Dato-Información. PRINT. Conocimiento del teclado II.

Lección 3.ª :

Informática. NEW. RUN. Partes del ordenador: UCP, UC, UAL. Memoria. Byte. Kilobyte. Bit. Lenguaje máquina. Traductor.

Lección 4ª :

Hardware. Software. Periféricos: Caset, Unidad de disco. Teclado, Pantalla, Impresora. Terminal. Memoria interna: RAM, ROM. Memoria externa. Lenguajes de alto nivel.

Lección 5.ª :

Repaso. El Basic. Programa. Línea. Instrucción. Número de línea.

Lección 6.ª :

Variable: Numérica, de cadena. Asignación. LET.

Lección 7.ª :

La calculadora PRINT. Punto. Coma. Los operadores aritméticos. Paréntesis. La impresora PRINT. Punto y coma.

Lección 8.ª :

PRINT AT. INPUT.

Lección 9.ª :

Algoritmo. Diagrama de flujo-Organigrama. GOTO. Programa. IF . . . . THEN. Operadores lógicos.

Lección 10.ª :

Repaso. Programa.

Lección 11.ª :

FOR. . . . . NEXT.

Lección 12.ª :

GOSUB . . . . . RETURN.

Cada una de estas lecciones termina con un repaso cuestionario, después del cual el alumno puede jugar, por el puro placer de jugar, con el juego que, al efecto, se le sitúa al final de cada lección.

El índice anterior refleja la pretensión general del curso, y actúa de modo sincronizado con el libro A.T.V., en que se repasa, se hacen las prácticas, y se amplían conocimientos y materias.

Introducción al Curso

FUNCIONAMIENTO DEL CURSO

A lo largo del curso irás encontrando aclaraciones acerca de cómo debes desenvolverte con las lecciones.

El curso de divide en dos parte fundamentales:

–LECCIONES AUDIOVISUALES.

–MODULOS DE REPASO Y ACTIVIDAD.

Cada lección audiovisual tiene su correspondencia en el Módulo del Libro de mismo número.

FORMA CORRECTA DE HACER EL CURSO

El primer día debes dedicarlo, fundamentalmente, a hacer correctamente las conexiones de los aparatos. Este paso es muy importante pues, si falla, fallará el resto: todo debe quedar bien conectado. El primer día debes también ojear el prólogo del Libro ATV de Informática-Basic.

Es conveniente que no intentes hacer todo el curso en un solo día; es imposible. El aprendizaje es una tarea amena, pero que requiere un esfuerzo y una constancia por tu parte. Cada día debes hacer, a ser posible a la misma hora y en el mismo sitio, una lección audiovisual (las del caset). Sólo cuando tengas un dominio absoluto de lo que se explica en la lección, sólo entonces, debes

pasar a hacer los módulos. El ciclo lectivo idóneo sería: Día 1, haces la leccion audiovisual; Día 2, la repites; Día 3, haces el módulo. Si este ciclo lectivo se ve incrementado, es decir, si tardas más tiempo en hacer las lecciones, no te preocupes: lo más importantes es aprender bien, no, aprender pronto.

LAS LECCIONES AUDIOVISUALES

Tienen una duración aproximada de entre 25 y 45 minutos, siendo más largas las cinco primeras. En las lecciones debes limitarte a seguir, al pie de la letra, las instrucciones que te da el profesor, prestando la mayor atención a sus explicaciones. No te impacientes: en todo momento el profesor te dirá qué tienes que hacer, con el suficiente tiempo como para que lo hagas bien.

En cada lección te ponemos algún pasatiempo, que debes resolver en un cuardeno. Las soluciones las encontrarás en el Libro ATV o en la misma lección.

Al final de cada lección, completamente al final, después de que el profesor se despide y se calla. . . al final del todo hay. . . una sorpresa.

En cada lección, también hay una serie de ejercicios que tienes que responder, atinadamente, todos los días. Son muy fáciles, solo tienes que prestarles atención. Cuando hayas terminado todos los ejercicios el profesor te dirá tu puntuación:

NO PASES NUNCA A LA LECCION SIGUIENTE SIN HABER RESUELTO A LA SATISFACCION TODOS LOS EJERCICIOS. TODOS.

LOS MODULOS DE REPASO Y ACTIVIDAD

Son las lecciones del Libro ATV. En ellos damos un repaso de lo visto en las lecciones, y hacemos prácticas de los sectores más interesantes de las lecciones. Bien, esto es el principio. ATREVETE.

(Se me olvidaba decirte que en la página 172 encontrarás las evaluaciones del curso. La primera de estas evaluaciones la tienes que hacer después del módulo 5, la segunda después del módulo 10, y la tercera y última, después del módulo 12. ¡Suerte!).

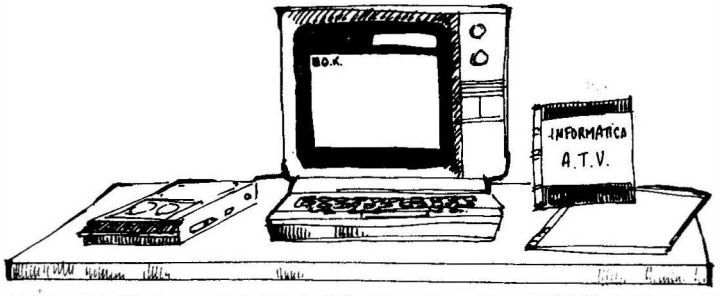
CONTENIDO DEL PAQUETE

1.– Ordenador A.T.V. Dragón versión Pedagógica.
2.– Cable de conexión para la red eléctrica.
3.– Cable de conexión para el televisor.
4.– Cable de conexión para el caset.
5.– Manual de manejo del A.T.V. Dragón.
6.– Libro de Informática-Basic A.T.V.
7.– 12 cintas-caset que contienen las lecciones.

Para hacer el curso de Informática-Basic A.T.V. precisas:

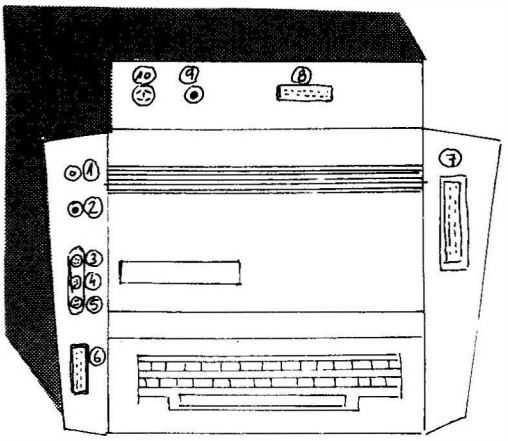
- El ordenador A.T.V. Dragón.
- Un televisor provisto de UHF. No importa que sea en blanco y negro, pero es preferible que sea de color.
- Un caset de calidad media.
- Este libro.
- Lápiz y papel.

También será muy útil una mesa donde ponerlo todo . . .



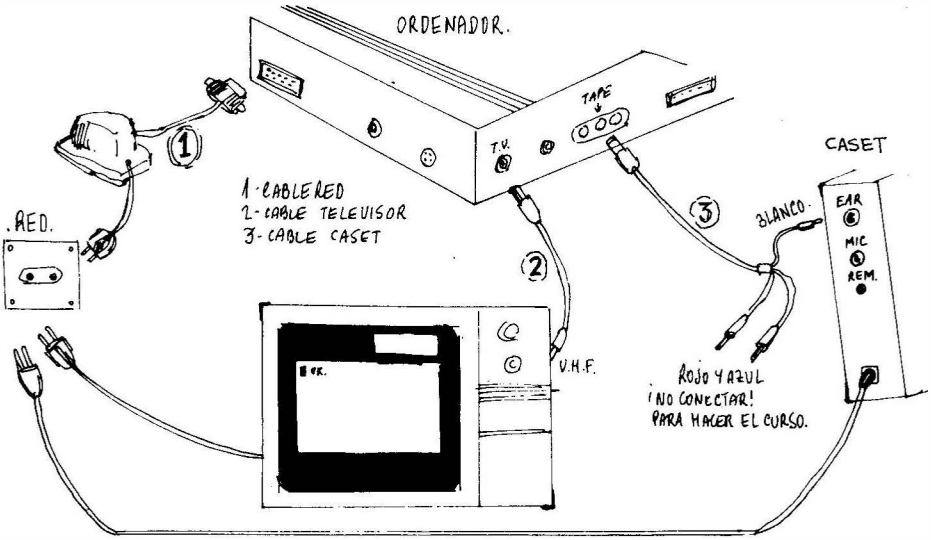
CONEXIONES

Una vez que está todo sobre la mesa, y la mesa cerca de un enchufe, o cerca de un prolongador, procedemos a conectar los aparatos. Es una tarea importante. Debes hacerla despacio y con cuidado.



Tomas del Ordenador

1.– TOMA PARA TELEVISOR
2.– Botón de Reset
3.– Toma para Joystick izquierdo. (Mando para juegos)
4.– TOMA PARA CONECTAR CASET
5.– Toma para Joystick derecho. (Mando para juegos)
6.– Conexión para impresora
7.– Toma para cartuchos o para unidad de disco.
8.– TOMA PARA LA RED ELECTRICA
9.– Interruptor DE ENCENDIDO ON/OFF
10.– Toma para el monitor



Vamos a Conectar

1. Toma el cable de conexión para el televisor y conecta ordenador y televisor como se ve en la figura 3.
2. Toma el cable de conexión al caset (fig. 3) y realiza la conexión como aparece en la figura (NO CONECTES la clavija ***REM*** para hacer nuestro curso. Ni la clavija ***MIC)***.
3. Acopla al ordenador el cable de conexión a la red eléctrica (1) y POSTERIORMENTE, enchúfalo a la RED.
4. Pon el volumen del televisor al mínimo y enchufa televisor y caset a la red.

En este momento todo debe estar interconectado, y enchufado a la red.

El siguiente paso será. . . . . .

SINTONIZACION

Vamos a SINTONIZAR el canal por el que vamos a ver las lecciones. Algunos televisores tienen un canal especial para ordenadores. Si el tuyo carece de él, da lo mismo, también sirve.

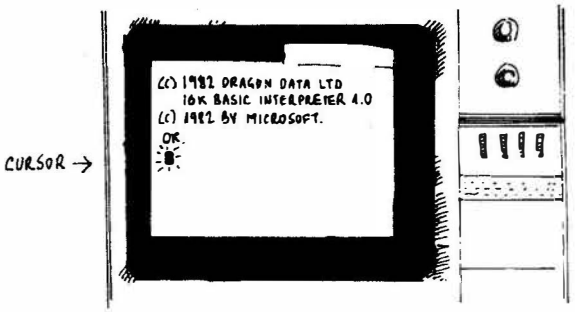
1. Enciende el televisor (que ya debe estar conectado al ordenador).

2. Enciende el Microordenador, mediante el interruptor de encendido ***ON/OFF*** (10), que debe quedar pulsado hacia dentro ***(ON).***

   En unos segundos la pantalla se llenará de puntitos o de imágenes raras, o incluso puede que te aparezca el programa que están poniendo por la tele. Vamos, pues, a sintonizar.

   En estos momentos tu Ordenador es una emisora de Televisión, así que, vamos a sintonizar con Tele-A.T.V.

3. Acude a los mandos de sintonización de tu televisor, los canales, selecciona un canal que no emplees normalmente para ver la tele, y hazlo girar hasta que en tu pantalla apareza la figura (4).

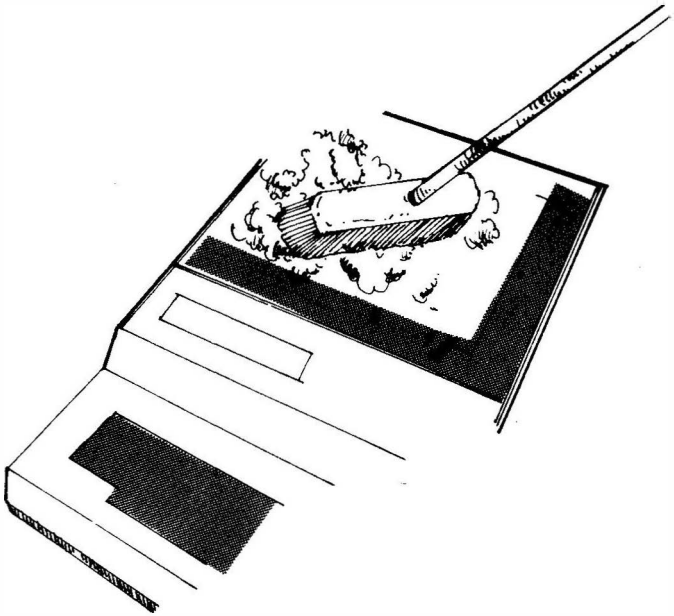


Si tu televisor es de color la pantalla será verde. No debes dejar de sintonizar hasta que veas las letras totalmente nítidas.

Regula también el color, el brillo, y el contraste. Una vez que veas las letras nítidas, sitúa el volumen del televisor un poco por debajo de la mitad: No tiene que oírse ningún ruido de fondo, y a lo sumo un leve ruidillo. Si el ruido es molesto tienes que seguir sintonizando.

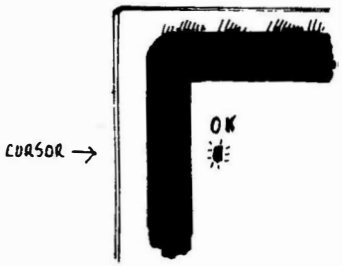
ON/OFF. CURSOR OK

¿Lo tienes ya? Apaga y enciende un par de veces el ordenador con el interruptor de encendido ***ON/OFF***. Esa operación sirve para limpiar el ordenador por dentro.



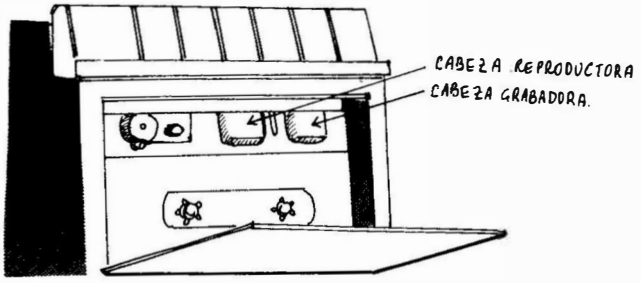
Ya están conectados el ordenador y la Pantalla, listos par actuar. Siempre que en la PANTALLA aparezca

quiere decir que el ordenador está listo para actuar. El cuadradito negro se llama **CURSOR.**



EL CASET

Vamos ahora al ***caset.*** En primer lugar vamos a realizar la operación de carga. El ***caset*** que vas a utilizar debe tener las cabezas limpias, y no vendría mal que, aunque tú consideres que lo están, las limpies con un algodón humedecido en alcohol (fig. 7).



(Tengo que decirte que si quieres un ***caset*** especialmente diseñado para la reproducción de cintas de ordenador, no tienes más que rellenar el pedido que te adjuntamos, y remitírnoslo). Bien, una vez que el ***caset*** está en condiciones. . .

1. Toma la cinta núm. 1, que contiene la lección 1 por ambas caras. Introdúcela en el ***caset***, y pulsa el mando ***REWIND*** hasta que toda la cinta quede almacenada a un lado.
2. Sitúa el volumen del ***Caset*** justo en el máximo.
3. Si tu ***caset*** tiene regulador de graves y agudos, o es estéreo, sitúa todos los mandos en la posición intermedia.
4. A continuación, comprueba que el volumen del televisor esté situado un poco menos de la mitad. Probablemente luego tendrás que regularlo más a tu gusto.

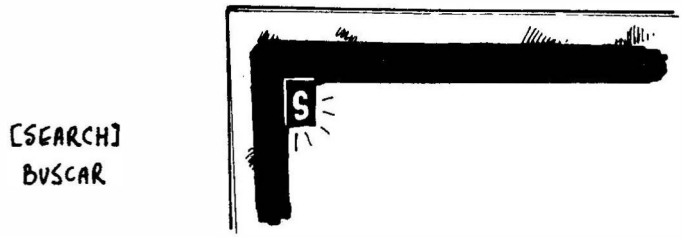
LA CARGA

¿Estás sentado cómodamente delante del ordenador? ¿Ves bien la pantalla? Si la pantalla es muy grande no es conveniente que estés muy cerca de ella. ¿Tienes a mano todo lo necesario para iniciar el curso? ¿Tienes lápiz y papel?

Si es así, empecemos: DESDE ESTE MOMENTO TE HALLAS EN UNA ACADEMIA.

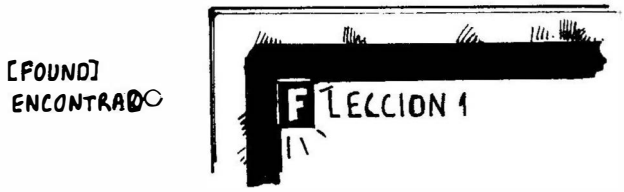
1. Teclea en el ordenador **CLOADM,** y luego pulsa la tecla **ENTER.** (Si al teclear te equivocas, puedes corregir con la tecla).

   En la Pantalla debe aparecer. . . .

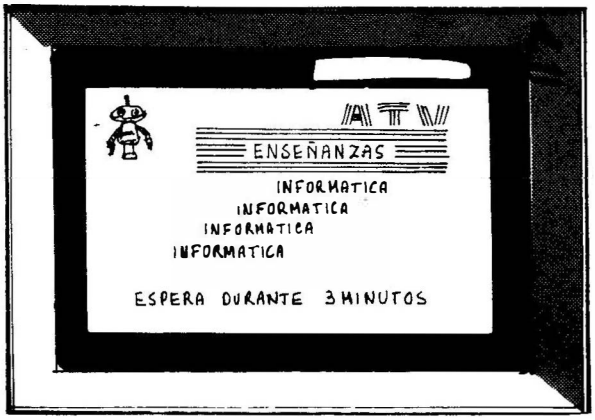


   Y luego oirás un clik!. . . .

2. Pulsa el ***PLAY*** del ***caset.*** Después de unos segundos aparecerá. . . .

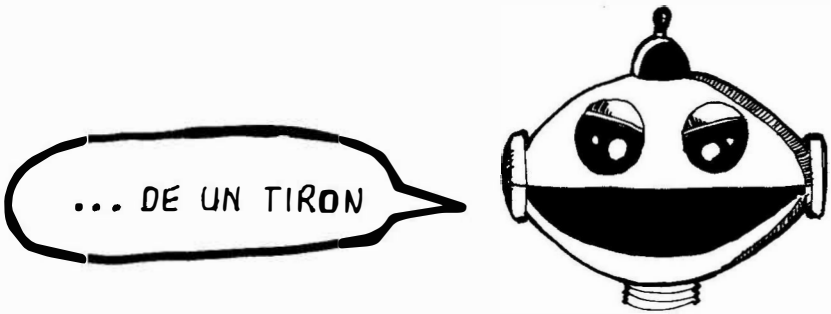


   Luego se pondrá la pantalla en blanco y. . . A continuación, y por espacio de 3 minutos, verás en tu pantalla la carátula de presentación del curso:



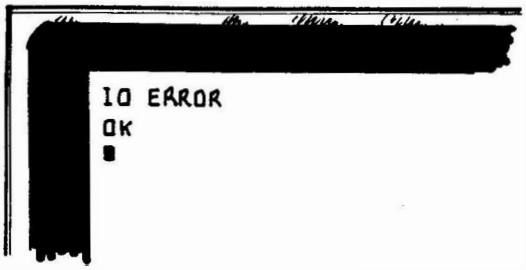
Ahora, durante unos minutos, hasta que escuches mi voz, te diré, que una vez iniciada la lección, y así en lo sucesivo, **NO DEBES INTERRUMPIRLA HASTA TERMINARLA. NO DEBES PARAR EL CASET BAJO NINGUN CONCEPTO.**

Esto es un curso de Informática, y de tu propio interés depende del éxito del aprendizaje, de hecho, la metodología de nuestro curso aconseja seguir estrictamente todos los consejos que te demos, y uno de ellos es: **"HAZ LAS LECCIONES SIN INTERRUPCIONES".**



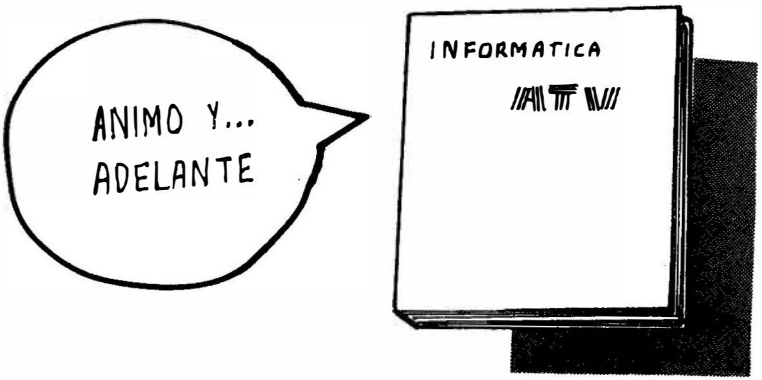
MUY IMPORTANTE:

En caso de que el ordenador se detenga, por algún motivo desconocido para ti, y aparezca en pantalla algún mensaje, como por ejemplo.



Tendrás que rebobinar la cinta, y regular el volumen del *caset*. Es decir, tendrás que subirlo o bajarlo un poco, teclear **CLOADM, ENTER**; PLAY del *caset*. Si volviera a repertirse ***IO ERROR*** o ***FM ERROR***, habrás de probar nuevamente con el volumen. Una vez que hayas dado con un volumen ideal, déjalo siempre así, y si tienes que bajar el volumen o subirlo para escuchar mejor la voz, regúlalo con el volumen del televisor, **NUNCA CON EL CASET. NUNCA CON EL CASET.**

Y ahora atento a la Pantalla. Cuando acabes la lección, vuelve aquí, por favor, que tenemos cosas interesantes que decirte, y que tú debes saber. Esto es el principio.



Si una vez empezada la lección se te interrumpiera por cualquier motivo, habrás de empezar nuevamente desde el principio.

INDICE

Modulo 1

¿Cómo te fue la 1ª lección? Si aún no la has hecho, por favor, deja de leer, y vuelve a intentarlo, ahora, o en otro momento en que estés más relajado.

Una vez que hayas hecho la primera lección, primero te felicitamos, porque todos los principios son difíciles, luego, ¡te aconsejamos que la repitas! Sí, que la repitas. Aunque hayas hecho bien todos los ejercicios. No consiste en hacer las lecciones por obligación, sino en hacerlas con aprovechamiento. "NO DEBES PASAR A LA LECCION SIGUIENTE SIN CONOCER A FONDO LA ANTERIOR".

También debemos darte un consejo: Inicia hoy mismo un cuaderno en el que puedes ir anotando, una vez terminada cada lección, lo más importante, lo más difícil, lo más útil.

CLOAD es el **Comando** mediante el cual pasamos a la memoria del ordenador los programas que están almacenados en el caset.

CLOAD sirve para. . . . . un programa en el ordenador.

LOAD en inglés significa cargar. C LOAD es cargar del C aset. CLOAD **M** significa cargar del caset algo que está escrito en un código especial que ya veremos: El Código **Máquina.** Nosotros empleamos **C LOAD M** porque cargamos del C aset y porque las lecciones están escritas en Código Máquina.

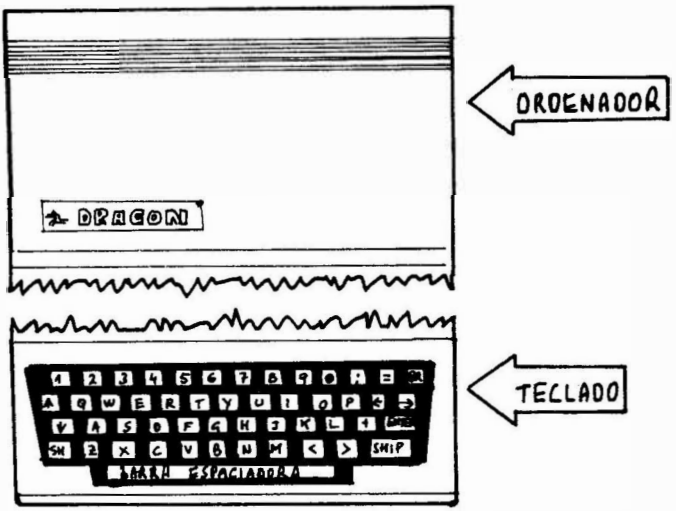
CLOADM. . . ENTER. . . ¡a la carga!

Un **COMANDO** es una palabra, una instrucción mediante la cual ordenamos al ordenador que haga algo.

ENTER es una de las teclas de control del ordenador. ENTER se pulsa después de cualquier COMANDO para que este se ejecute.

En la primera lección, fundamentalmente hemos visto el teclado del ordenador. El ordenador es una máquina que se comunica con nosotros por medio de la pantalla.

La Pantalla no forma parte del ordenador, pero es necesario conectar el ordenador a la Pantalla para que ésta pueda comunicarse con nosotros. Del mismo modo, el teclado no forma parte del ordenador.



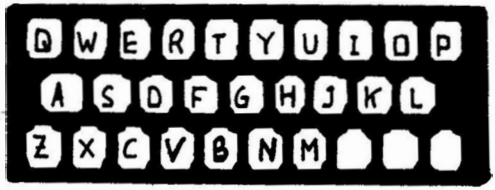
TECLADO

El ***teclado*** sirve para manejar el ordenador, y para comunicarnos con él. Precisamente por eso, es imprescindible conocer el ***teclado***. ¡Atento! No queremos decir que sea necesario que teclees rápido: ¡NO! Lo que sí es necesario es que tomes algunas precauciones para teclear en el ***teclado*** de un ordenador.

El ***teclado*** sirve para. . . . . con el ordenador.

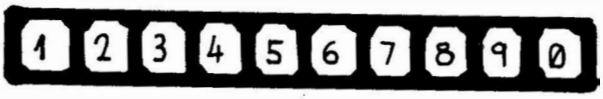
El ***teclado*** de un ordenador es parecido al de una máquina de escribir. Las letras se disponen según el ***sistema QWERTY***, más conocido como ***teclado universal***. Con el resto de las teclas hay que tener mucho cuidado. Podemos dividir las teclas por grupos:

LETRAS:



Falta la Ñ

NUMEROS:



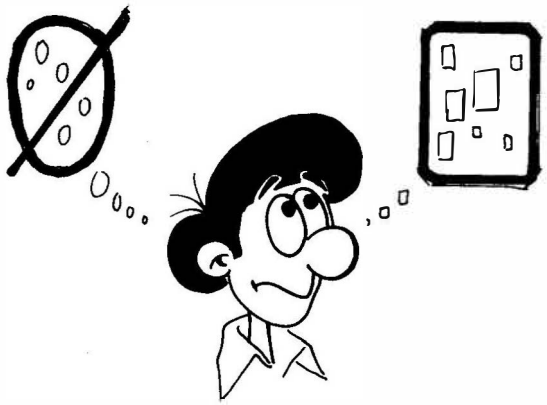
Esos son los números. Todas las teclas hay que pulsarlas con precaución, y, luego de pulsarlas, hay que verificar que han sido pulsadas, mirando la pantalla. Especial cuidado hay que tener con dos teclas:

La de 1 y la de Ø

El 1 sólo se puede escribir con la tecla de 1. La tecla de L sirve solamente para escribir la letra L. La de 1 para escribir el número 1.

Por otro lado, la tecla de Ø sirve para escribir el número Ø, la tecla de ◘ para la letra ◘ .

NO	SI
10053	10053
30900	30900
400	400
1L780	11780
50	50



Otras teclas a las que vamos a prestar especial interés son los ***operadores aritméticos,*** o ***signos aritméticos.***

Operadores aritméticos:

Multiplicado por	*
Dividido por	/
Mas	+
Menos	—
Potenciación	↑

El signo. . . es multiplicado por
El signo. . . es potenciación
El signo. . . es dividido por

Estos son los signos que emplearemos para hacer las ***operaciones aritméticas.*** El más + y el menos — los conocías. En cambio, puede que no te fueran familiares el multiplicado por * y el dividido por / . A partir de ahora ten en cuenta que esos son los signos que emplea el ordenador. Otro signo que seguro que desconocías es el de la potenciación ↑ : como es lógico, sirve para potenciaciones:

$2 \uparrow 2 = 2^2$
$5 \uparrow 3 = 5^3$
$8 \uparrow 7 = 8^7$
$1 \uparrow 10 = 1^{10}$

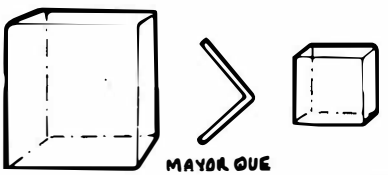
Son los ***operadores aritméticos,***

Otros signos que sirven para hacer operaciones, son los ***operadores lógicos.***

< > =

El signo. . . . . es "mayor que"
El signo. . . . . es "menor que"

Menor que < , Mayor que > , e igual = . Sin estos signos el BASIC se vería en serios apuros. Los operadores lógicos sirven para hacer operaciones de comparación, por ello también se les llama signos de comparación.



9 < 45
MENOR QUE

CARAMELO=CARAMELO
IGUAL QUE

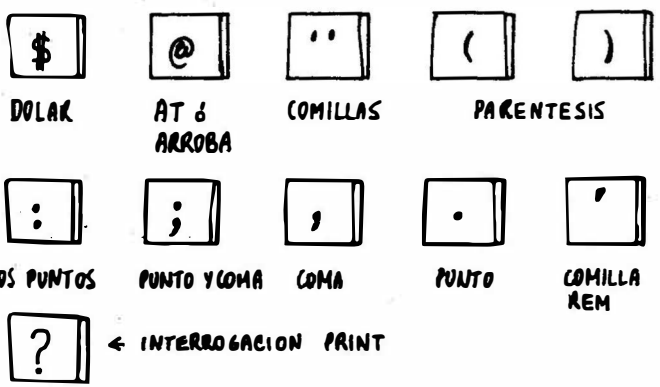
CASA<>TROMPETA
DISTINTO QUE

6>2
MAYOR QUE

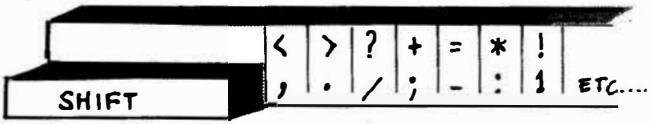
La finalidad de este curso es que inicies tu alfabetización informática. El ordenador es sólo una parte de esta alfabetización. Nos estamos aproximando al ordenador. Teclea despacio. Corrige si te equivocas.

Al final de la lección haremos algunas prácticas.

¿Otros signos de interés?



Como dijimos en la lección 1, la tecla SHIFT es un escalón para subir a la parte de arriba de las teclas que contienen dos signos.

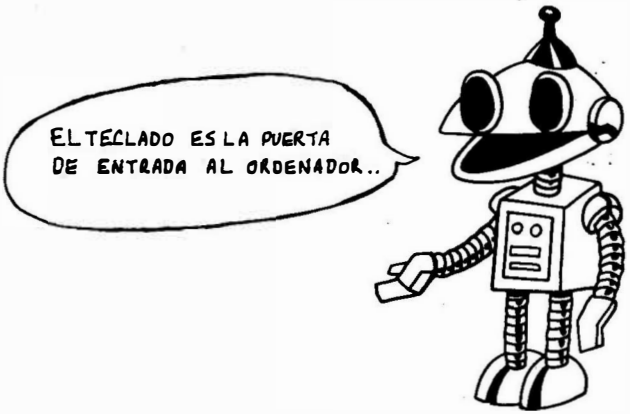


La tecla **SHIFT** es la tecla de mayúscula. El Microordenador siempre escribe en mayúscula. (Excepto cuando lo hace mediante impresora).

¿No hay entonces ningún modo de distinguir mayúsculas de minúsculas? Sí lo hay. Se pueden distinguir mediante la ESCRITURA EN NEGATIVO, que se activa por medio de **SHIFT Ø** pulsados simultáneamente.

SHIFT Ø pone en. . . . . la escritura (letras)

ESCRITURA EN NEGATIVO



Todo esto lo has practicado en la lección 1ª.

BREAK		ENTER	CLEAR
SHIFT @		SHIFT ←	SHIFT Ø

No así las teclas de control, que nosotros vamos a considerar **BREAK, ENTER, CLEAR, SHIFT @** , , y **SHIFT ←, SHIFT Ø.**

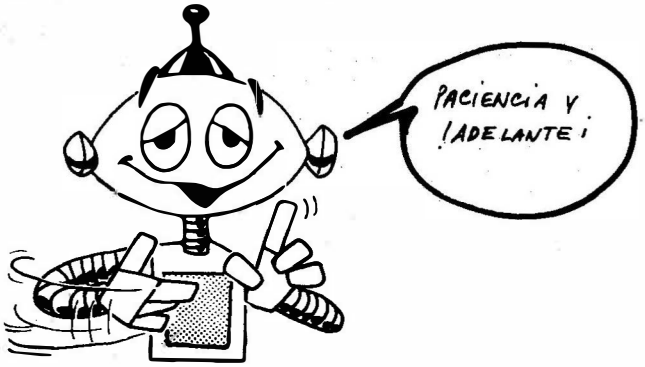
Cada una de estas teclas ejerce una importante función sobre el teclado.

ENTER	ya la conoces.
	borra hacia la izquierda, letra a letra.
SHIFT ←	borra una línea entera.
CLEAR	borra toda la pantalla.
SHIFT Ø	activa la escritura en negativo (también sirve para desactivarla).
SHIFT @	detiene la ejecución de un programa. Pulsando cualquier tecla excepto SHIFT y BREAK reanuda la ejecución. Con **SHIFT @** se frena.
BREAK	rompe un programa por alguna línea en concreto. Con **BREAK** se para. Después de pulsar **BREAK,** si quieres que se reanude el programa, tienes que teclear **RUN** y luego pulsar **ENTER.**

Si en algún momento encuentras en una lección algo que no comprendes, no te preocupes, porque te lo explicamos más adelante, o quizá... es que no prestaste bastante atención a la lección anterior.

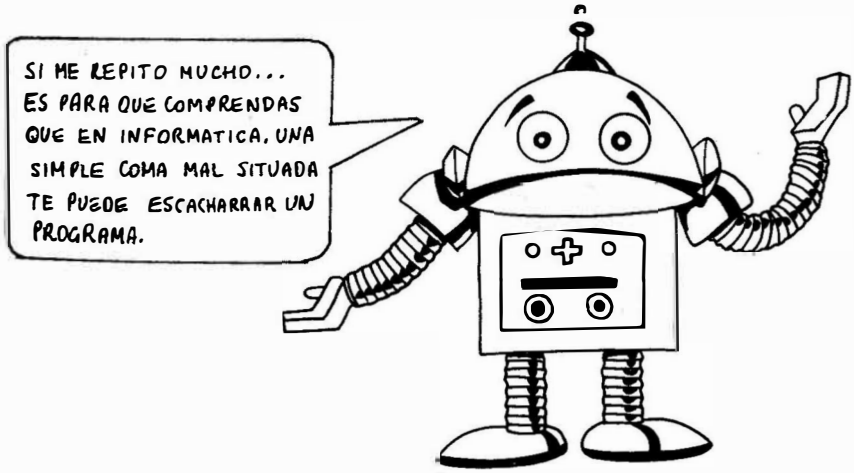
? SN ERROR

Más de una vez, después de pulsar **ENTER**, te aparecerá este mensaje en la pantalla. Esto se debe a que has tecleado algo mal. Tendrás que copiar nuevamente la línea en que te sale el error.



```
10 TIMER=0:A=RND(58)+31:A$=""
20 PRINT CHR$(A);" ";
30 IF TIMER/50<3 THEN 40 ELSE 60
40 A$=INKEY$
50 IF A$="" THEN 30
60 IF CHR$(A)=A$ THEN PRINT A$:GOTO 10
70 PRINT A$:PRINT "FALLASTE"
80 GOTO 10
```

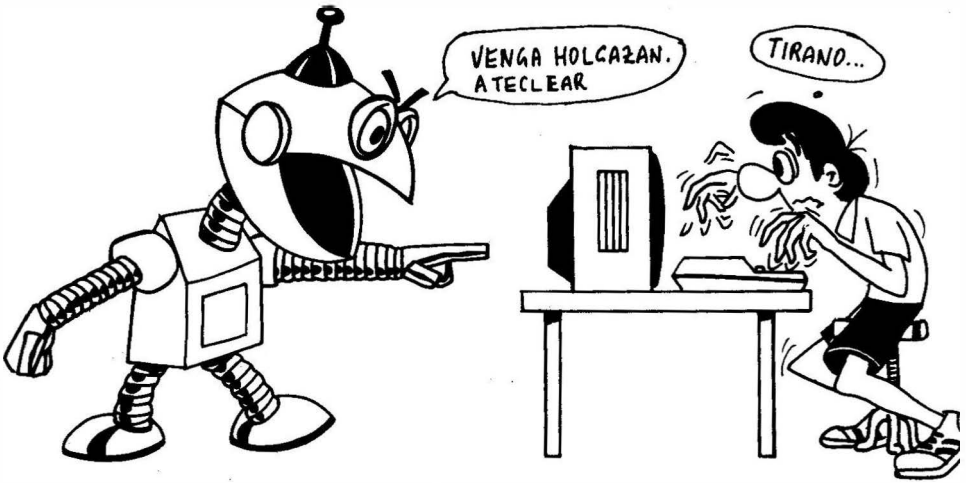
¿Tienes a punto tu ordenador? ¿Está conectado? Desconéctalo con ***ON/OFF***, y vuélvelo a conectar. Ahora copia ese programa, cuidadosamente, tal y como está escrito en el recuadro. Al terminar cada línea, debes pulsar **ENTER**. Detente antes de teclear **RUN**. Hasta luego, pero recuerda: Copia el programa, cuidadosamente, tal y como aparece en el recuadro.



¿Lo has copiado tal y como aparece en el recuadro? Ahora pulsa la tecla **CLEAR**, y teclea

R U N

La práctica va a consistir en que vayas copiando lo que te vaya apareciendo en la Pantalla. ¿De acuerdo? Pues bien, practica durante un par de minutos, y luego sigue con el libro.



pulsa **ENTER.**

¿Has terminado? Si ha ido muy rápido para ti, prueba a copiar esta línea, pero antes, para interrumpir el programa, pulsa la tecla **BREAK** varias veces, y luego copia esta línea.

```
30 IF TIMER/50<6 THEN 40 ELSE 60
```
ENTER

En cambio, si ha ido muy lento, prueba con esta otra línea, aunque antes, para detener el programa, pulsa **BREAK.**

```
30 IF TIMER/50<10 THEN 40 ELSE 60
```
ENTER

Si sólo quieres practicar letras, copia ésta.

```
10 TIMER=0:A=64+(RND(26)):A$=""
```
ENTER

Si sólo quieres copiar números y signos, prueba con esta otra.

```
10 TIMER=0:A=32+(RND(32)):A$=""          ENTER
```

Como verás, en todos los casos, lo que hemos hecho ha sido modificar algún número.

MODIFICACIONES

De momento, cuando quieras modificar alguna línea de algún programa, tienes que copiar la misma línea entera con la modificación que desees.

Esto es un curso de enseñanza. Estás en un academia: contamos con tu esfuerzo, tu colaboración, y tu curiosidad.

BREAK interrumpe el programa.
Si lo quieres reanudar teclea RUN y pulsa ENTER

Ahora copia y ejecuta el siguiente programa:

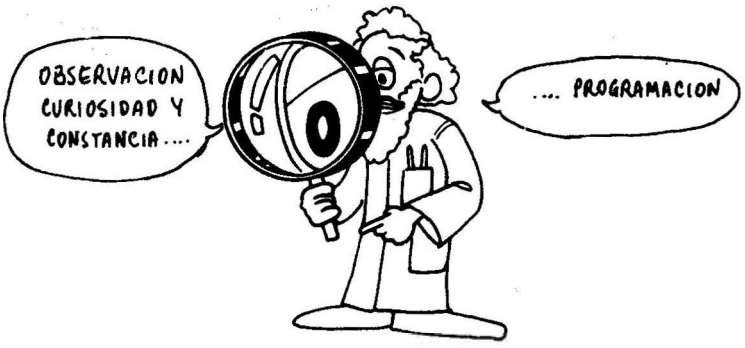
```
10 PMODE 3:PCLS:SCREEN 1,1
20 FOR X=6 TO 250 STEP 20
30 FOR Y=6 TO 190 STEP 20
40 CIRCLE (X,Y),10,2
50 NEXT Y,X
60 GOTO 60
```

(Cambiando los números de las líneas 20, 30 y 40, modificarás el dibujo).

Con este programa harás prácticas con **BREAK** y **SHIFT** @ . Cuando lo hayas copiado teclea **RUN** y pulsa **ENTER**, y cuando el dibujo vaya por la mitad pulsa, al mismo tiempo, **SHIFT** @ . El dibujo se detendrá: para que siga, pulsa cualquier letra, excepto **SHIFT** y **BREAK**.

Con ese programa puedes hacer prácticas con **SHIFT** @ y **BREAK**.

Ser curioso y observador, y, sobre todo, constante, es una garantía para tu aprendizaje.



Te esperamos en el MODULO 2, después de que hayas hecho la lección 2ª del Curso.

Por favor, no pases a la lección siguiente sin dominar la lección y el módulo anteriores. No te importe repetir una lección cuantas veces sea necesario. Al final tu serás el más beneficiado.

REPASO

1.- INTERRUPTOR ON/OFF:
Sirve para encender o apagar el ordenador. Limpia la memoria del ordenador.

2.- **OK**:
Ordenador listo para actuar.

3.- ■
CURSOR

4.- **LOAD**:
Sirve para cargar un programa en la memoria del ordenador.

5.- **C LOAD M;**

Carga de un C caset, algo que está escrito en **M** ***Código Máquina.***

6.- **ENTER:**

Ejecuta órdenes.

7.- ***COMANDO:***

Palabra mediante la cual se da una instrucción al ordenador. Ejemplos: **CLOADM, RUN, PRINT, CLS.**

8.-

0 = CERO
□ = LETRA O
1 = UNO
L = LETRA ELE

9.- ***OPERADORES ARITMETICOS:***

* multiplicado por
/ dividido por
\+ mas
— menos
↑ potenciación

10.- ***OPERADORES LOGICOS:***

< Menor que < = menor o igual que
> Mayor que > = mayor o igual que
= Igual que <> diferente

11.- **SHIFT**

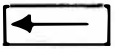
Sirve para escribir las teclas dobles.

12.- **SHIFT Ø**

ESCRITURA EN NEGATIVO

13.- **BREAK** – **SHIFT** @

Detiene la ejecución del programa.

14.- 

Borra carácter a carácter.

SHIFT

Borra líneas enteras.

15.- La tecla **CLEAR**

Borra la pantalla.

16.- **RUN**

Echa a andar un programa

Solución al **Jeroglífico**:

M ANTE QUILLA.

EJERCICIOS DEL MODULO 1

1) Qué instrucción utilizamos para cargar un programa en la memoria del ordenador. (Subraya la respuesta correcta).

   a) **ENTER**
   b) **CLOADM**
   c) **SHIFT Ø**

2) Los operadores lógicos o signos de comparación son:

   a) = > <
   b) = * /
   c) 0 + –

3) Escribe, utilizando los operadores lógicos:

   A es mayor que B
   B es menor que C
   A es igual que C

4) El signo de multiplicar en el ordenador es:

   a) X
   b) .
   c) *

5) El signo de dividir en el ordenador es:

   a) :
   b) /
   c) –

6) Podemos modificar una línea de programa:
   a) Copiándola otra vez igual.
   b) Copiándola otra vez con la modificación.
   c) No se puede.

7) Con **SHIFT @**
   a) Se escribe en negativo.
   b) Se detiene el programa.
   c) Se borra una línea.

8) Con **BREAK**
   a) Se borra la pantalla.
   b) Se escribe en negativo.
   c) Se "rompe" el programa.

9) Con **RUN**
   a) Se detiene el programa.
   b) Se borra la pantalla.
   c) Se echa a andar el programa.

10) Con **CLEAR**
   a) Se borra la pantalla.
   b) Se rompe el caset.
   c) Se limpia la memoria

11) Cuando aparece **SN ERROR?** es que te has equivocado al copiar. Entonces tendrás que:
   a) Copiar nuevamente la línea equivocada.
   b) Apagar el ordenador.
   c) Desesperarte!

SOLUCIONES:

B; A; A>B; B<C; A=C; C; B; B; B; C; A; A.

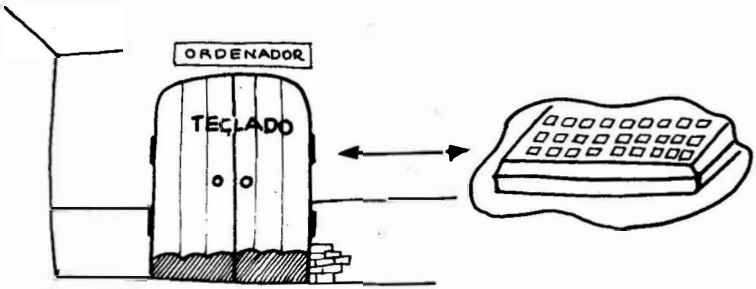
Módulo 2

información automática: ***INFOR – MATICA***

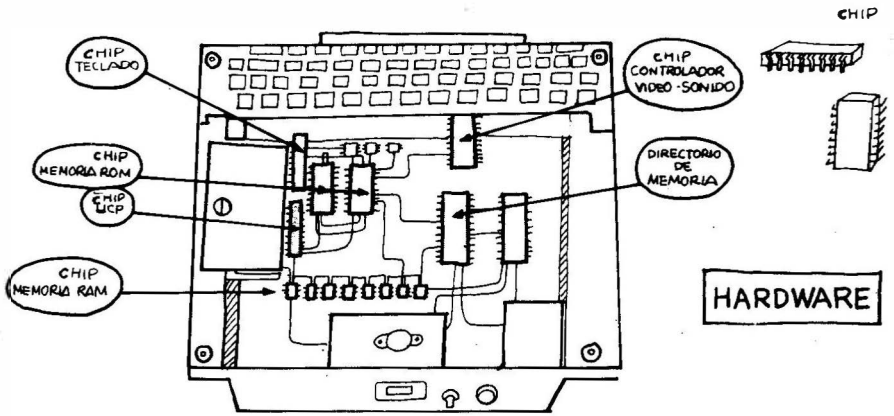
Un ordenador es una máquina electrónica que recibe datos y después de trabajar con ellos los devuelve convertidos en información. Un ordenador es una calculadora lista, una calculadora que habla, que oye, que pregunta.

Los datos entran en el ordenador por una puerta: el teclado·

La puerta por la que entran los datos es . . . .



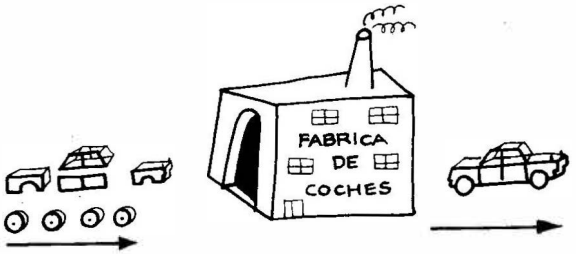
por eso hay que conocerlo bien.

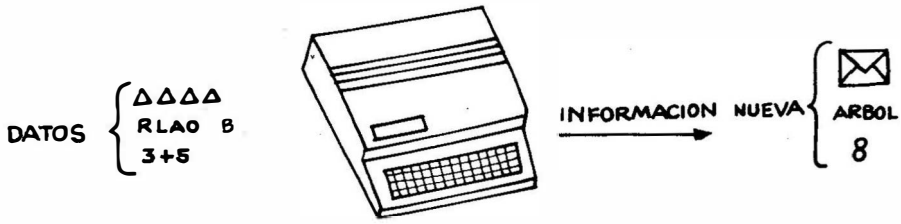


El ordenador es como una fábrica: por la puerta, que es el teclado, van entrando muchas piezas, que son los datos. Dentro del ordenador están la ***unidad de control, la unidad aritmético lógica*** y ***la memoria.*** Estas tres piezas cogen los datos, los manejan, los tratan, y después de este proceso, producen algo nuevo, producen información.

El ordenador es una máquina electrónica que recibe datos, y produce información.

Es como una fábrica, que recibe piezas, las ensambla, y luego produce coches. Los coches serían la información.

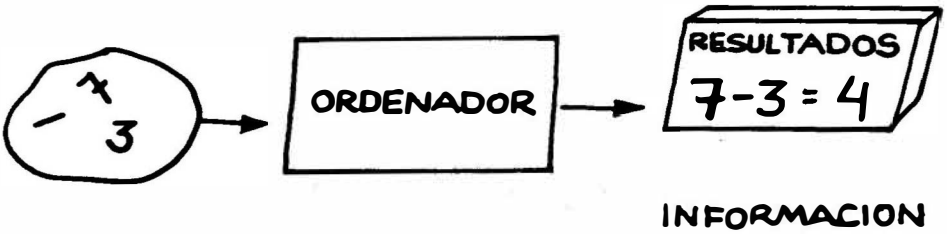




Informática es la ciencia que se encarga de estudiar ese proceso. Estudia el proceso en que un ordenador recibe datos, los procesa, y los convierte en información.

Igual que tú tomas con tus manos un trozo de plastilina, el ordenador toma datos.

Con sus piezas, ***UC, UAL, MEMORIA*** los trata, y los convierte en información.



Pero el ordenador es una máquina. Al ordenador hay que decírselo todo. Hay que enseñárselo todo. El ordenador es una herramienta electrónica que te ayuda a estudiar, a jugar, a trabajar, pero hay que decirle las cosas con orden.

1. Toma el número 7
2. Toma el número 3
3. Resta el número 3 al número 7
4. Dame el resultado

Al ordenador hay que decirle las cosas con . . .

Pero ni siquiera así lo entenderá, porque no está bastante claro para él.

Hay que decírselo en ***BASIC***. Teclea en tu Microordenador ese programa, pero antes, teclea **NEW**, y luego pulsa **ENTER.**

```
10 CLS
20 INPUT "DAME UN NUMERO";X
30 INPUT "DAME OTRO NUMERO";Y
40 R=X-Y
50 PRINT @ 235,R
60 END
```

¿Lo has copiado exactamente igual? En ***BASIC*** hay que teclear muy despacio. Si no, teclea **NEW-ENTER**, vuelve a copiar. Los programas hay que copiarlos exactamente igual que aparecen en el libro.

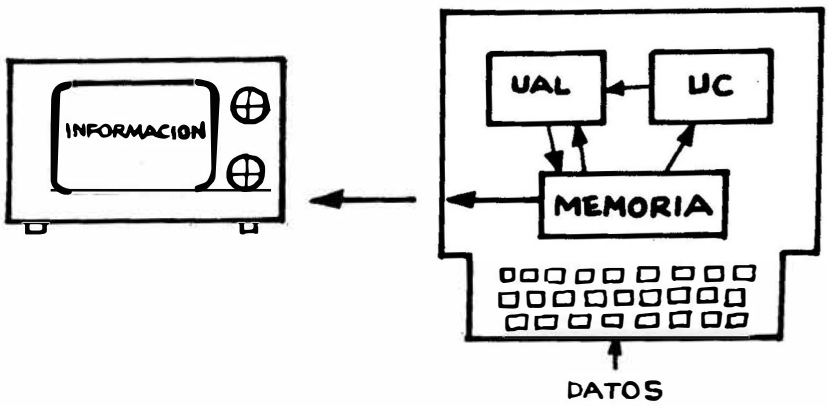
Ahora pulsa **CLEAR**, teclea **RUN** y luego pulsa **ENTER.** ¿Ves? Teclea el número que quieras. . . y **ENTER.** Ahora te pide otro número. Teclealo y . . . **ENTER.**

Te ha dado el resultado.

PROGRAMA

Eso que has copiado es un programa; un conjunto de instrucciones y datos. Tú se lo introduces al ordenador por el teclado. El hace los cálculos en su cerebro electrónico, su **UCP**, y luego te da la respuesta por la pantalla.

El ordenador hace los cálculos en su . . . .



Al ordenador hay que enseñarle todo. Lo primero que le vamos a enseñar es a copiar, a escribir en la Pantalla. Teclea **NEW,pulsa ENTER.** Con esto hemos borrado todo lo que hay en la memoria del ordenador. Ahora pulsa **CLEAR** para borrar la pantalla.

Copia ese programa:

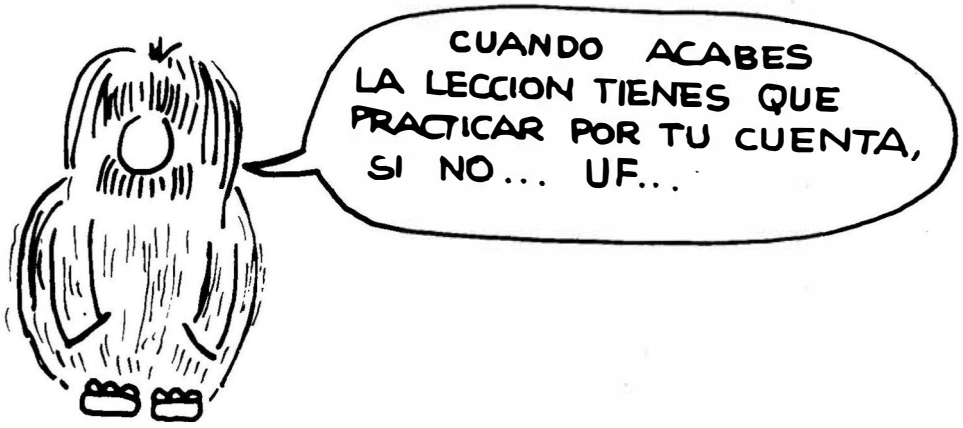
(Haz un dibujo con **PRINT**)

```
10 PMODE 3:SCREEN 1,1:PCLS
20 FOR N=1 TO 20
30 COLOR (RND(3)+5)
40 GOSUB 70
50 NEXT N
60 GOTO 60
70 LINE (RND(225),RND(191))
   -(RND(255),RND(191)),PSET,B
80 RETURN
```

Ejecútalo: Teclea **RUN** y pulsa **ENTER**.

¿Ha salido un dibujo? Si es así, todo va bien. Si no, todo va mal. Si por la Pantalla sale **SN ERROR?** todo va mal. Si te sale ese mensaje, es que te has equivocado al copiar.

* * *



¿Quieres aprender algo más?

NEW borra la memoria del ordenador. La limpia. El interruptor ***ON/OFF*** también. El botón **RESET**, que está a la izquierda, al lado del cable del televisor, limpia la pantalla, y la memoria, aunque no la borra. Púlsalo para hacer la prueba. (Recuerda que con **CLEAR** se puede limpiar la pantalla).

Ahora teclea

LIST y ENTER

con . . .	se limpia la pantalla

¡EH! Ha reaparecido el programa.

LIST te saca por la pantalla el ***LISTADO*** de los programas que hay en la memoria del ordenador. **LIST** te saca por la pantalla la lista de todos los programas que hay guardados en la memoria del ordenador.

LIST	. . .	los programas.

RESUMEN

INFORMATICA:	Ciencia que trata la información mediante ordenadores.
HARDWARE:	Las partes mecánicas o físicas de un ordenador.
ORDENADOR:	Máquina electrónica que procesa datos.
PROGRAMA:	Conjunto de instrucciones y datos.
NEW:	Borra la memoria del ordenador.
CLEAR:	Borra la pantalla del ordenador.
SN ERROR?	Se produce cuando se teclea algo mal.
LIST:	Saca por la pantalla el listado del programa que hay en la memoria.

Para terminar esta lección, sólo te vamos a decir

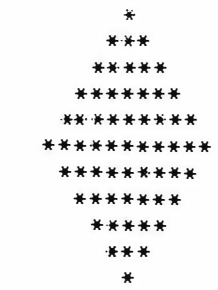
PRINT "lo que sea"

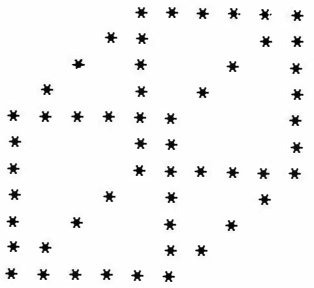
Imprime en Pantalla.

```
10 PRINT "*"
20 PRINT "**"
30 PRINT "***"
40 PRINT "****"
50 PRINT "*****"
60 PRINT "******"
70 PRINT "*******"
80 PRINT "********"
90 PRINT "*********"
100 END
```

```
10 PRINT "ESTO ES"
20 PRINT "UN EJEMPLO"
30 PRINT "DE COMO FUNCIONA"
40 PRINT "EL COMANDO PRINT CON COMILLAS"
50 END
```

Ahora, intenta por tu cuenta hacer con **PRINT** los siguientes dibujos.





Copia este programa

```
10 PRINT "+++++OOOOO+++++"
20 PRINT "+++++OOOOO+++++"
30 PRINT "+++++OOOOO+++++"
40 PRINT "OOOOO*****OOOOO"
50 PRINT "OOOOO*****OOOOO"
60 PRINT "OOOOO*****OOOOO"
70 PRINT "+++++OOOOO+++++"
80 PRINT "+++++OOOOO+++++"
90 PRINT "+++++OOOOO+++++"
```

EJERCICIOS DEL MODULO 2

1) Informática es:
   a) Recepción de datos.
   b) Proceso de resultados.
   c) Tratamiento de datos.

2) Un ordenador es:
   a) Una máquina que piensa.
   b) Una máquina que calcula.
   c) Una máquina que procesa datos.

3) Un dato es:
   a) Algo que introducimos en el ordenador por medio del teclado.
   b) Las letras que salen por la pantalla.
   c) Una palabra entre comillas.

4) Información es:
   a) Cualquier número.
   b) Todo lo que podemos percibir.
   c) Lo que vemos en la pantalla.

5) **NEW** sirve para:
   a) Limpiar la pantalla.
   b) Borrar la memoria del ordenador.
   c) Borrar la pantalla.

6) Un programa es:
   a) Una palabra.
   b) Una instrucción.
   c) Un conjunto de instrucciones y datos.

7) La instrucción **LIST** sirve para:
   a) Sacar por la pantalla el listado del programa.
   b) Echar a andar el programa.
   c) Imprimir en pantalla.

8) Si se produce un **SN ERROR?** Tienes que:
   a) Copiar todo el programa.
   b) Copiar la línea en que se ha producido.
   c) Apagar el ordenador.

9) Después de copiar una línea de programa:
   a) Se teclea **RUN.**
   b) Se pulsa la barra espaciadora.
   c) Se pulsa **ENTER.**

10) La instrucción que "echa a andar el programa" es:
   a) **RUN.**
   b) **LIST.**
   c) **PRINT.**

SOLUCIONES

C; C; A; B; B; C; A; B; C; A.

Solución a los Pasatiempos

1º. Respuesta:
Es el número 7.

2º. Respuesta:
Los números 1 y 11

Módulo 3

¿Te diviertes con los juegos? Seguro que es la parte de las lecciones que más te gusta.

Los programas que te ponemos en cada lección del libro, son menos vistosos, pero cumplen una importante función: debes hacerlos, primero para divertirte. En segundo lugar para aprender, y en tercer lugar, y es lo más importante, debes intentar modificarlos.

MODIFICACIONES

Un programa de juegos, o cualquier otro programa se puede modificar de muchos modos. Hoy te voy a explicar el modo más fácil de introducir una modificación en un programa.

Lo primero, antes de nada, limpiar la memoria del ordenador. ¿Recuerdas cómo se hacía?

ON/OFF o NEW – ENTER

N. DE LINEAS

Ahora copia este programa. (No olvides que después de copiar cada línea hay que pulsar **ENTER**).

```
10 CLS 0
20 PRINT"QUE EDAD TIENES"
30 INPUT C
40 PRINT"ES DECIR, QUE TIENES";C*365;"DIAS"
```

(Prescindimos de los bisiestos).

Como verás hay 4 líneas: Desde la 1Ø a la 4Ø.

¿No te llama la atención que las líneas se numeren de 1Ø en 1Ø? ¿Sabes por qué se hace así?

Pulsa **CLEAR**. Teclea **RUN**. Pulsa **ENTER**, y luego te lo explico.

El ordenador te pregunta "Qué edad tienes". Contéstale con tu edad escrita en números y pulsa **ENTER**.

El ordenador te ha dicho cuántos días tienes. Supon-te que ahora quieres añadir algo a ese programa. Por ejemplo, quieres que la Pantalla, en vez de negra, sea de otro color. ¿Cómo lo haces? Muy sencillo: Pulsa **BREAK**. Pulsa **CLEAR**. Ahora **LIST - ENTER**. Ahí tienes otra vez el programa. Copia la siguiente línea.

```
10 CLS 3
```

Al copiar esa línea, la 1Ø, aunque esté al final del programa, el ordenador la coge, y la pone en su sitio, es decir, al principio del programa.

Vuelve a rodar el programa. **RUN ENTER.**

¿Ves?

Ese es un modo de modificar un programa: Se copia nuevamente la línea que se quiere modificar, introduciéndole el cambio deseado. Cuando se copia una línea con el mismo número de otra línea que ya está en el programa, la última se superpone a la primera, y la "machaca".

La nueva línea que se ha tecleado, se superpone a la antigua, y la borra.

CLS

Prueba ahora a poner otros colores. Los colores que puedes hacer con tu micro son:

Ø– NEGRO	1– VERDE	2– AMARILLO
3– AZUL	4– ROJO	5– BEIGE
6– TURQUESA	7– MAGENTA	8– NARANJA

Y la instrucción para cambiar el color de Pantalla es:

C L S N (N = número del 0 al 8).

¿Has hecho los cambios?, Pulsa **CLEAR,** teclea **LIST (ENTER).**

Vamos a hacer otro, pero ahora, entre líneas. Entre la línea 1Ø y la 2Ø vamos a meter otra, la 15, que será la siguiente, y cambiamos la línea 4Ø.

```
15 PRINT"CUAL ES TU NOMBRE":INPUT N$
4Ø PRINT N$;"TIENES";C*365;"DIAS"
```

¿Cómo hemos introducido esta línea en el programa? Pues igual que la otra: Tecleas la línea 15, y el ordenador se encargará de ponerla entre la 1Ø y la 2Ø.

```
1Ø CLS 3
15 PRINT "Cual es tu nombre": INPUT N $
2Ø PRINT "Que edad tienes"
```

Si las líneas no fueran de 1Ø en 1Ø, sino de una en una:

```
1 CLS 3
2 PRINT "Qué edad tienes"
```

no podrías meter ninguna línea entre la 1 y la 2, porque entre 1 y 2 no hay ningún número (los decimales no valen como número de línea). En cambio, entre 1Ø y 2Ø caben: 11, 12, 13, 14, 15, 16, 17, 18, 19.

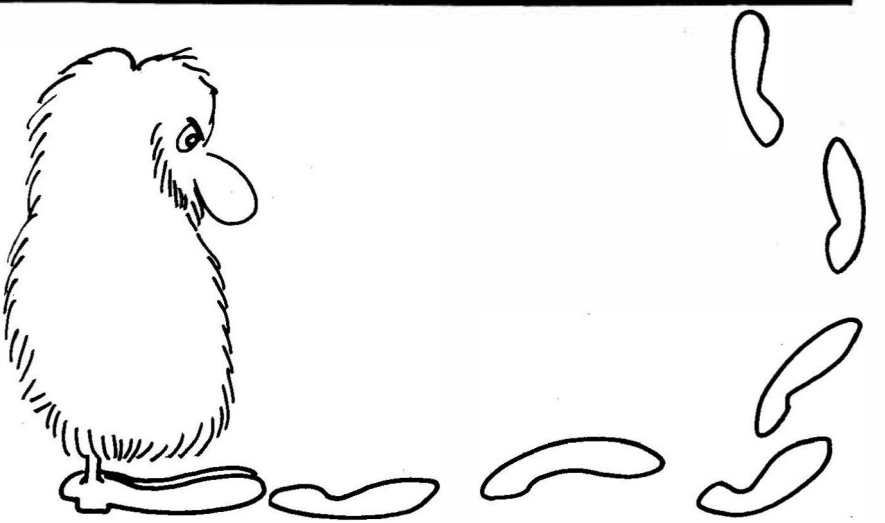
¡Puedes meter 9 líneas más! Si es que quieres modificar el programa.

Esa es una forma de modificar programas. En otras lecciones te explicaremos otras.

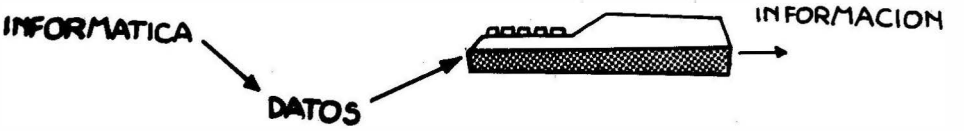
RUN ECHA A ANDAR UN ***PROGRAMA***.
NEW BORRA LA ***MEMORIA*** INTERNA.
LIST ***LISTA UN PROGRAMA***.

Vamos ahora a dar un pequeños repaso a lo que has visto en la lección audiovisual.

REPASO



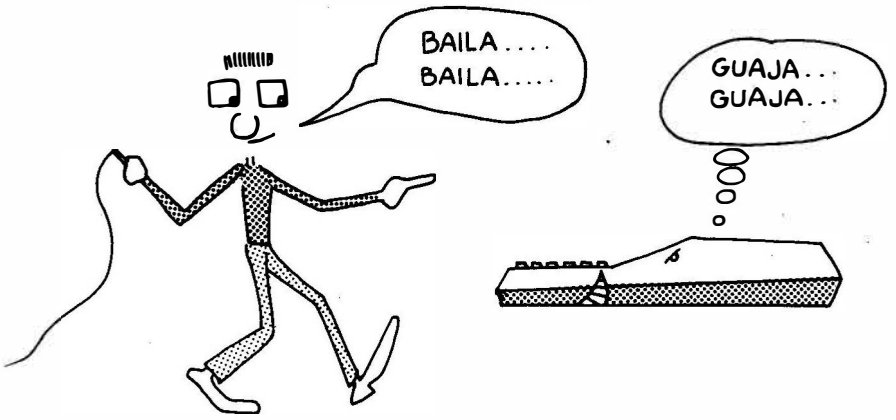
La Informática es el tratamiento de la información mediante ordenadores.



La información se introduce en el ordenador en forma de datos: los datos son números, palabras, signos, letras.

DATOS

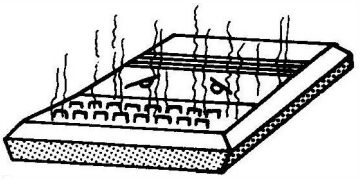
Pero el ordenador es una máquina, y no entiende las cosas que se le dicen así como así.



A tu ordenador hay que decirle las cosas en un lenguaje llamado ***BASIC*** y hay que decírselo todo de un modo claro.

> Al ordenador se le dicen las cosas en lenguaje . . . .

Una vez que tú le has introducido los datos por el teclado, dentro del ordenador se produce un proceso: El ordenador se pone a trabajar con los datos.



* * *

Dentro del ordenador hay unas piezas: ***la Memoria, la Unidad de Control, la Unidad aritmético lógica*** etc. Estas piezas son ***CHIPS,*** y cada una tiene una misión.

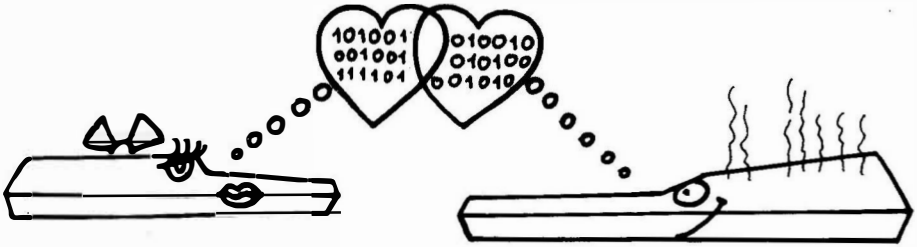
Una de esas piezas es la ***memoria.*** ***La memoria,*** el ***CHIP*** de ***memoria,*** tiene dentro miles de casilleros donde se van almacenando los datos.

Pero en un ***CHIP*** no se pueden meter los datos tal y como se pueden escribir en un papel.

Dentro del ordenador hay un ***traductor*** que se encarga de traducir tus instrucciones y datos al lenguaje que habla el ordenador en su interior:

AL ORDENADOR SE LE HABLA EN ***BASIC***
EL ORDENADOR HABLA EN ***LENGUAJE MAQUINA***
EL ***TRADUCTOR*** TRADUCE DE ***BASIC***
A ***LENGUAJE MAQUINA***

* * *



Los ordenadores hablan en ***lenguaje máquina.*** En este lenguaje todo se escribe con combinaciones de unos y ceros.

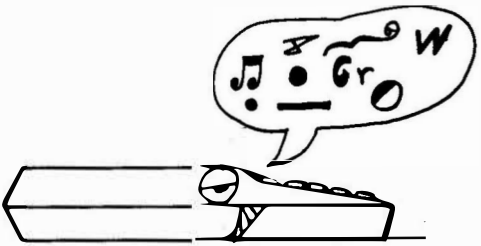
0	=	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
85	=	Ø	1	Ø	1	Ø	1	Ø	1
119	=	Ø	1	1	1	Ø	1	1	1
224	=	1	1	1	Ø	Ø	Ø	Ø	Ø
255	=	1	1	1	1	1	1	1	1

Cada uno de esos ceros y unos es un

B I T

El *BIT* es la unidad mínima de información.

¿Te acuerdas cuándo escuchaste cómo hablaba el ordenador? Pues aquello que oiste era



el ordenador hablando en lenguaje máquina. Cada ruidito era un ***BIT***. Cada 8 ***BIT*** forman un ***BYTE***. Cada ***BYTE*** es un carácter: una letra, un número, un signo. El abecedario español contiene de la A a la Z. El alfabeto del ordenador ¡contiene 255 caracteres!: Las letras, los números, los signos, y otros muchos más caracteres que sirven para dibujar, hacer música etc.

La unidad mínima de información es . . . .

Si quieres ver el código que contiene todo los caracteres del alfabeto del ordenador, ejecuta el siguiente programa.

```
10 FORX=1 TO 255
20 PRINTX,CHR$(X)
30 PRINT
40 FORC=1 TO 100:NEXT C
50 NEXT X
60 END
```

RUN ENTER

Si quieres que se detenga pulsa **SHIFT** **@** . Para seguir, pulsa cualquier carácter.

Habrás visto que cada número sale acompañado de un caracter: una letra, un número, un cuadradito, un signo.

CODIGO	CARACTERES
65	A
66	B
75	K
94	↑
48	Ø
60	<

El número de la izquierda representa el número que tiene cada uno de los caracteres en un código de escritura para ordenadores que se denomina:

A S C I I

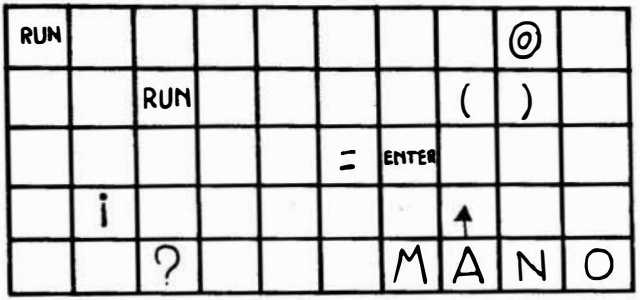
A la derecha ves los caracteres.

Nada de esto es necesario que lo aprendas de memoria. Basta con que lo leas, para que te suenen palabras como

BIT ***BYTE*** ***LENGUAJE DE MAQUINA*** ***CODIGO BINARIO*** ***CODIGO ASCII***

Bien. Todas las instrucciones, todos los caracteres se almacenan en la ***memoria. La memoria*** es la parte del ordenador en que se almacenan las instrucciones, datos y resultados. La ***memoria*** es como un gran cajón, lleno de casilleros, en cada uno de los cuales cabe un ***BYTE:*** una letra, un número, una instrucción etc.

Memoria



Cada uno de esos cuadraditos es un ***BYTE***.

En las ***memorias*** de los ordenadores caben miles de ***BYTES***. Cada mil ***BYTES*** forman un ***KILOBYTE***.

KILOMETRO	KILOGRAMO	*KILOBYTE*
↓	↓	↓
1.000 Metros	1.000 Gramos	1.000 (1.024) BYTES

Cada mil BYTES forman un . . . .

La capacidad de ***memoria*** de los ordenadores se mide en ***KILOBYTES (K's)*** de ***memoria.***

Un ordenador puede tener 16 ***K's***, 32 ***K's***, 64, 200, 10.000 ***K's*** etc. Si tiene un millón de ***BYTES*** tiene un

MEGABYTE

Mil ***k's*** de ***memoria*** son un ***MEGABYTE***.

La capacidad de memoria se mide en . . . .

Bien.

Programas que debes hacer para practicar:

Con este programa puedes convertir números del sistema decimal en números del sistema binario.

NEW. ENTER. CLEAR.

```
10 CLS
20 C$="":A$="":PRINT"DIME UN NUMERO DEL SISTEMA
   DECIMAL";
30 INPUT ND
40 IF ND=0 THEN A$="0":GOTO 80
50 IF ND=1 THEN A$=A$+"1":GOTO 80
60 IF (ND/2)=INT(ND/2) THEN A$=A$+"0":ND=ND/
   2:GOTO 40
70 A$=A$+"1":ND=(ND-1)/2:GOTO 40
80 FOR X=LEN(A$) TO 1 STEP -1
90 C$=C$+MID$(A$,X,1)
100 NEXT X
110 PRINT"SU CORRESPONDIENTE EN SISTEMA
    BINARIO ES --->";C$
120 GOTO 20
```

Con este programa puede convertir KBYTES en BYTES y BITS.

NEW, ENTER, CLEAR

```
1Ø CLS
2Ø INPUT "CUANTOS KBITES QUIERES CONVERTIR EN
   BYTES Y BITS";A
3Ø B=A*1Ø24
4Ø C=B*8
5Ø PRINT A;" KBYTES SON ";B;" BYTES Y ";C;" BITES"
6Ø PRINT "PARA ACABAR PULSA BREAK"
7Ø GOTO 2Ø
```

RESUMEN

CLS n	BORRA LA PANTALLA SEGUN EL COLOR n (número de 0 a 8).
LIST	LISTA EL PROGRAMA QUE SE ENCUENTRA EN LA MEMORIA.
BIT	UNIDAD MINIMA DE INFORMACION.
BYTE	8 BITS.
KBYTE	1.000 (1.024 BYTES).
MEGABYTE	1.000.000 BYTES.
ASCII	CODIGO DE ESCRITURA PARA ORDENADORES.
CODIGO BINARIO	LENGUAJE DE LOS ORDENADORES FORMADO POR 1 Y 0.
K'S	MEDIDA DE CAPACIDAD DE MEMORIA.

EJERCICIOS
DEL MODULO 3

1) Las líneas de los programas se suelen numerar:
   a) De ningún modo.
   b) De 1 en 1.
   c) De 10 en 10.

2) **CLS** sirve para:
   a) Limpiar la pantalla y cambiarle el color.
   b) Cambiarle el color y limpiar la memoria.
   c) Limpiar la memoria y la pantalla.

3) Una línea se modifica:
   a) Nunca.
   b) Copiándola con el número de la línea anterior
   c) Copiándola con el mismo número de la línea.

4) El Código ***ASCII*** contiene:
   a) 290 caracteres.
   b) 256 caracteres.
   c) 150 caracteres.

5) El ***traductor*** traduce:
   a) De español a inglés.
   b) De inglés a ***Basic.***
   c) De ***Basic*** a ***lenguaje máquina.***

6) Un ***CHIP*** es:
   a) Una parte de la ***memoria.***
   b) Una pieza.
   c) Un ***programa.***

7) Un ***BIT*** es:
   a) Una pieza.
   b) La unidad mínima después de 2.
   c) La 1/8 parte de ***BYTE.***

8) Un ***BYTE*** es:
   a) 8 ***BITS.***
   b) 9 ***Bits.***
   c) 8 Caracteres.

9) Un ***Kbyte*** contiene:
   a) 1.000 – 1.024 ***Bytes.***
   b) 32.000 – 64.000 ***Bytes.***
   c) 255 ***Bytes.***

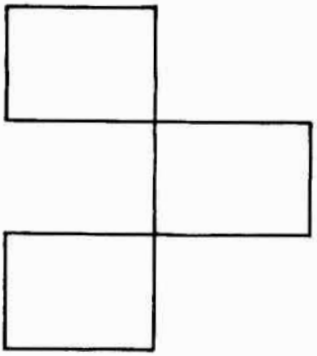
10) Un ***Megabyte*** contiene:
   a) 10.000 ***Kbytes.***
   b) 1.000.000 de ***Bits.***
   c) 1.000 ***K's.***

SOLUCIONES

Soluciones: C; A; C; B; C; B; C; A; A; C.

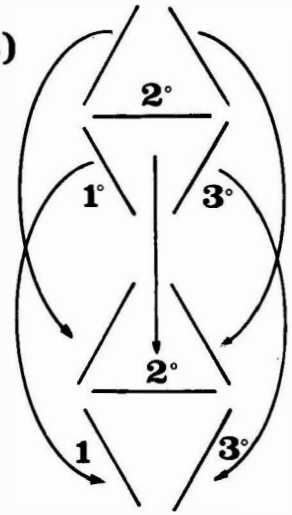
Solución al Pasatiempo

1)



Solución al Pasatiempo

2)



Módulo 4

El ordenador, como todos los instrumentos, sólo se puede dominar si se maneja con frecuencia.

Hay que practicar constantemente, pero antes de practicar hay que tener claros algunos conceptos. Para enfrentarse al micro hay que conocer los ***comandos*** de ***BASIC,*** y algunos otros puntos de interés.

ORDENADOR HARDWARE

Ya sabes que un ordenador es una máquina electrónica que recibe datos, los procesa, y los devuelve convertidos en información.

El ordenador es una máquina formada por piezas que son el ***HARDWARE.*** Dentro del ordenador hay unas hormiguitas que se llaman ***CHIPS.***

En el CHIP, de ***memoria*** se almacenan los datos, instrucciones y resultados.

La ***memoria*** es como una gran cuadricula. En cada casillero de la ***memoria*** cabe un carácter o ***BYTE***. Cada ***BYTE*** está formado por 8 ***BITS***.

Tú le hablas al ordenador en ***BASIC,*** y el traductor traduce tus instrucciones a ***Código Máquina.*** El abecedario del ***Código*** o ***lenguaje Máquina*** tiene 256 caracteres, y cada uno de ellos se representa por una combinación de unos y ceros.

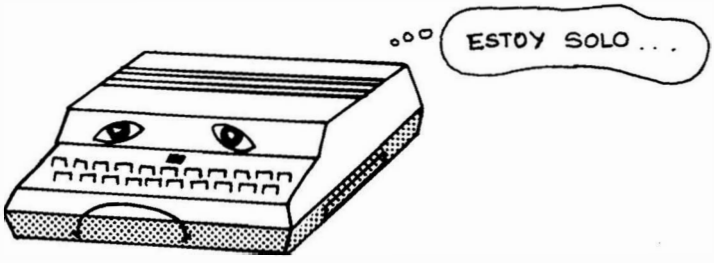
* * *

PERIFERICOS HARDWARE

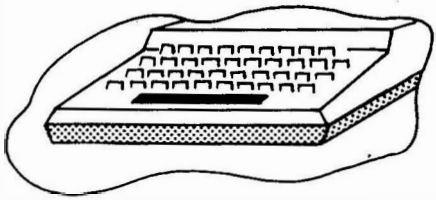
Ya sabes qué es ***HARDWARE***. También sabes que ***SOFTWARE*** son los programas: las instrucciones y datos que tú das al ordenador por medio del teclado.

Las instrucciones y datos que das al ordenador se denominan. . . . . .

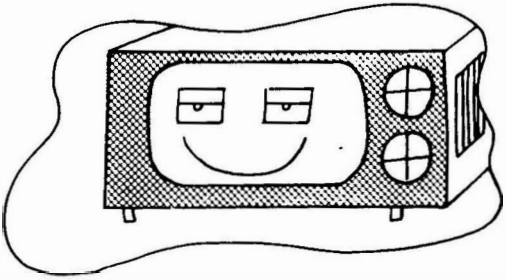
Bien. El ordenador es una herramienta. Pero el ordenador SOLO, no sirve para nada.



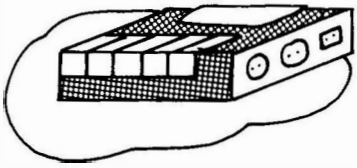
Si el ordenador no tuviera ***teclado*** no se le podrían introducir datos: No nos podríamos comunicar con el ordenador.



Si al ordenador no se le conectara una ***Pantalla*** no podríamos ver los programas, las operaciones, los resultados: el ordenador no podría comunicarse con nosotros.



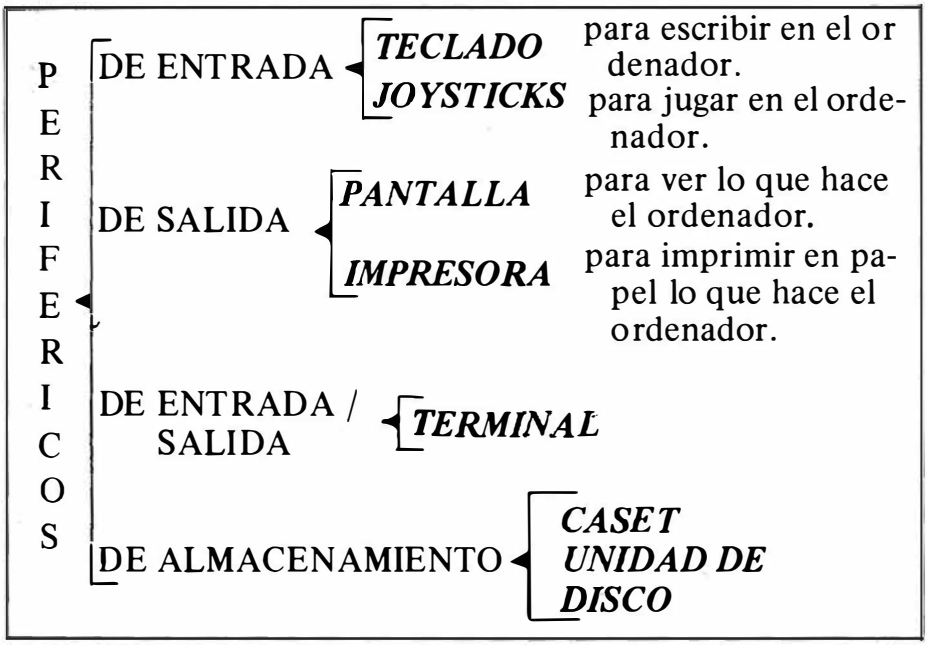
Si al ordenador no pudiéramos conectar un ***caset,*** no podríamos, por ejemplo, ver las lecciones audiovisuales.



Pues bien, a todos los aparatos que se conectan a un ordenador se les llama ***PERIFERICOS.***

Los ***PERIFERICOS*** son aparatos que se conectan al ordenador, y le ayudan a desempeñar sus tareas.

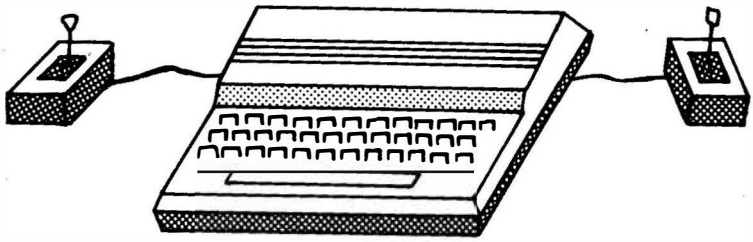
Los ***periféricos*** más utilizados son:



La pantalla, el teclado y el caset son . . . . . .

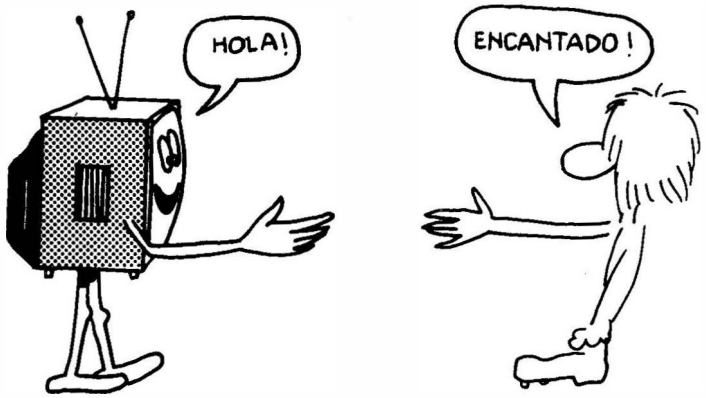
El Teclado, ya sabes, sirve para introducir datos al ordenador.

Los ***JOYSTICKS*** son los ***PERIFERICOS*** que se conectan al ordenador y sirven para jugar.

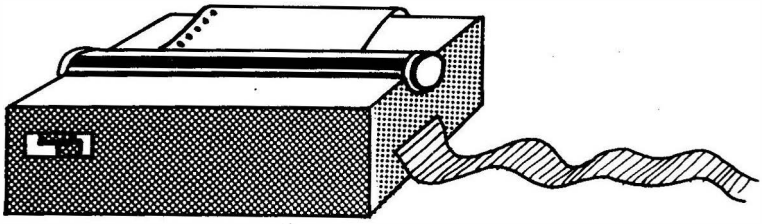


Los ***JOYSTICKS*** son mandos de juego. Muchos de los juegos que pueden comprarse en las tiendas de informática, funcionan con ***JOYSTICKS.***

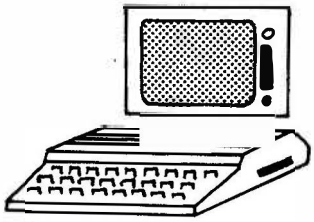
La Pantalla ya la conoces:



En cuanto a la ***Impresora,*** es como una máquina de escribir, pero sin teclado, que se conecta al ordenador, y que puede imprimir textos, programas, dibujos. La principal ventaja de la ***Impresora*** es que salen los datos y resultados impresos, y esto es de gran utilidad para facturas, recibos, escritos, etc.



Un ***terminal*** es la suma de un ***teclado*** y una ***Pantalla.*** Un ***terminal*** no es un ordenador. Un ***terminal*** es un ***teclado*** unido a una ***Pantalla.***



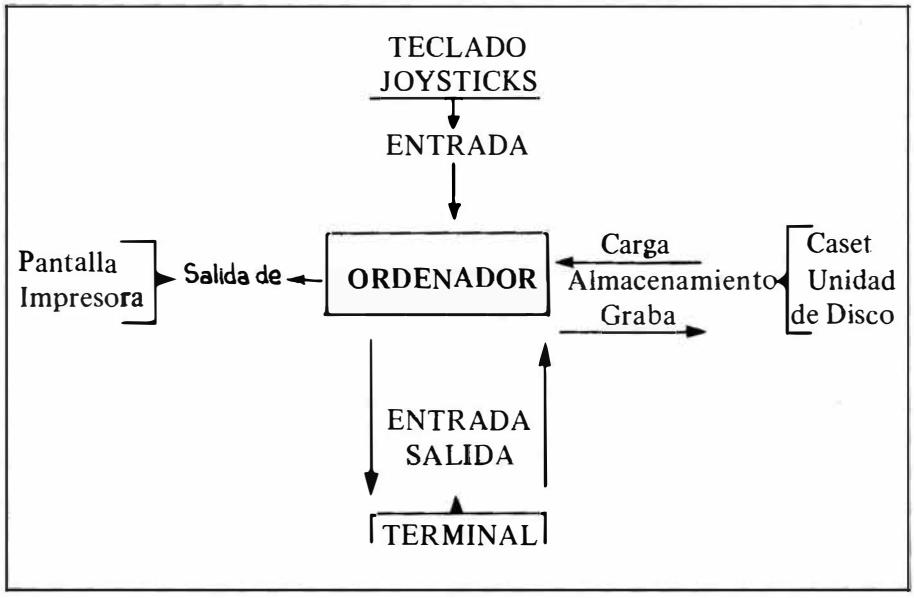
Los ***terminales*** se conectan a un ordenàdor, que es el que hace las operaciones.

Los ***terminales*** sirven para meter datos, por el ***teclado,*** y luego ver los resultados, por la ***Pantalla.***

Otra ventaja de los ***terminales*** es que, simultáneamente, varios ***terminales*** pueden estar conectados a un mismo ordenador: pueden compartir el tiempo, pueden trabajar al mismo tiempo. Eso es muy importante en las empresas, los Bancos, las universidades, los ministerios. Los ***terminales*** permiten un rápido y fácil acceso a la información de un ordenador central.

Una pantalla y un teclado forman un. . . .

Los ordenadores a los que se conectan los ***terminales*** suelen ser muy grandes, muy caros, y con miles de ***K's*** de ***memoria.***



La capacidad de ***memoria*** de los ordenadores es muy importante. Como ya sabes, la capacidad de ***memoria*** de un ordenador se mide en ***K's.*** Mientras más ***K's*** tiene un ordenador, más capacidad de trabajo tiene; más complejos y más rápidos son los programas y los cálculos que se pueden hacer.

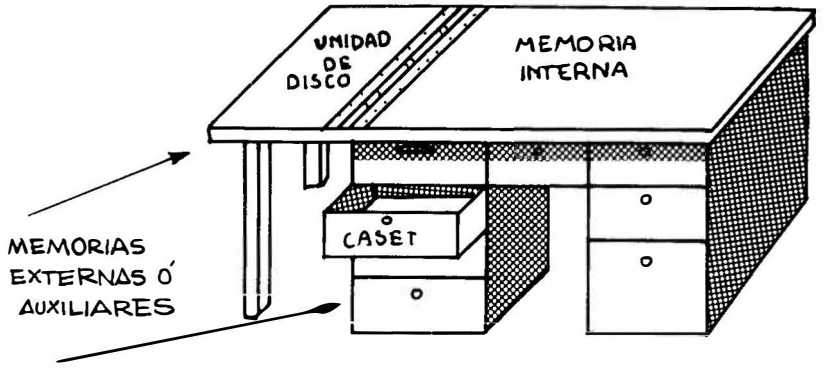
Los ***CHIPS*** de ***memoria*** de un ordenador se llaman ***memoria interna. Memoria Interna*** es la que está dentro del ordenador y, claro, si hay ***memoria interna,*** también tiene que haber ***memoria externa.***

¿Cuál será la ***memoria externa?***

La ***memoria externa*** es la que está fuera del ordenador. ¿Dónde? En el caset, o en la unidad de disco. Por ejemplo: las lecciones audiovisuales son programas que están guardados en un caset, porque no caben en la memoria de tu ordenador. Cuando tú haces **CLOADM** estás traspasando los programas que están en la ***memoria externa,*** en el caset, a la ***memoria interna*** del ordenador.

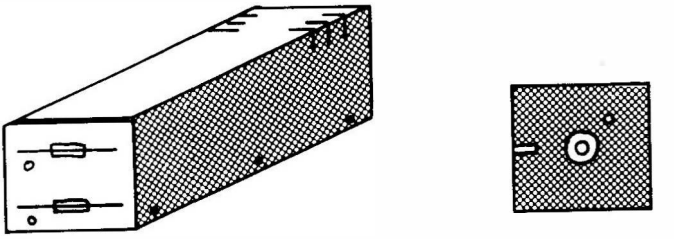
La memoria que está
fuera del ordenador se llama.....

Como recordarás, la ***memoria interna*** es como una mesa donde se van poniendo los datos, las instrucciones, todo, para trabajar. Cuando el tablero de la mesa está lleno y ya no caben más cosas, tenemos que guardar las cosas en los cajones. Los cajones serían la ***memoria externa*** o auxiliar.



En una cinta-caset tú puedes guardar los programas que vayas haciendo. Luego te explico cómo.

En cuanto a la ***Unidad de Disco***, también llamada ***DRIVE*** es también una ***memoria auxiliar***.



El ***diskette*** o ***Floppy-disk***, es como un disco de los de música, pero más pequeño. El ***diskette*** es muy delicado, y va siempre dentro de su funda para que no se estropee. Sí el caset es como un cajón donde se guarda lo que no cabe en el tablero de la mesa, o donde se guarda

lo que no se va a utilizar de momento, la ***Unidad de Disco*** es como un tablero auxiliar que se pone al lado del tablero principal, y que se puede utilizar al mismo tiempo que se está utilizando la ***memoria interna.*** Mientras que el caset es solo un sitio donde guardar programas, el ***Diskette*** sirve para guardar programas, pero también sirve para utilizarlos al mismo tiempo que los de la ***memoria interna.*** La ***unidad de disco*** es una ampliación de la ***memoria.***

En definitiva, ***caset*** y ***unidad de disco,*** son dos ***periféricos*** de almacenamiento, ***dos memorias auxiliares.***

NEW, ENTER, CLEAR. (Al final de cada línea pulsa **ENTER**).

```
5 CLS 2
10 REM REPASILLO
20 PRINT
30 PRINT"REM NO SALE EN EL PROGRAMA "
40 PRINT
50 PRINT"PRINT SIN NADA SALTA UNA LINEA "
60 PRINT
70 SOUND 80,32
80 SOUND 160,16
90 SOUND 240,8
100 PRINT"CON SOUND HACEMOS SONIDOS "
110 END
```

RUN, ENTER.

Lee despacio el repaso de la pantalla. Como verás, en la Pantalla no aparece ningún **REM**, y es que la Instrucción **REM** sirve para poner comentarios, pero dentro del **LISTADO** del Programa.

CLEAR. LIST. ENTER.

SOUND, es la Instrucción que sirve para hacer sonidos. **NEW. ENTER. CLEAR.** y Copia.

```
10 REM PROGRAMA PARA HACER SONIDOS
20 SOUND 1,8
30 SOUND 50,8
40 SOUND 100,8
50 SOUND 150,8
60 SOUND 200,8
70 SOUND 250,8
80 SOUND 255,32
90 PRINT "SI QUIERES HACERLO OTRA VEZ,
   TECLEA RUN.ENTER"
100 END
```

RUN. ENTER.

Sound nota,duración

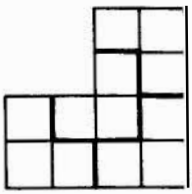
Sound 53, 16 (16 = 1 segundo)

* * *

Respuestas a los Pasatiempos de la lección 4ª

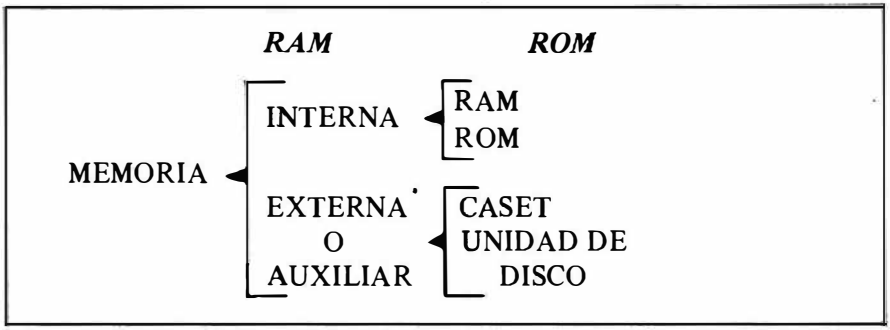
1º El Hexágono tiene 114 triángulos.

2º



3º "Más vale pájaro en mano que ciento volando"

Estábamos hablando de la ***memoria.*** Hemos visto los ***Periféricos.*** La ***memoria interna*** puede ser de dos tipos:



La memoria ***RAM*** es donde se almacenan los programas: instrucciones y datos. Se borra con **NEW**, con **ON/OFF.**

Con **NEW** se borra todo el contenido de la memoria. Con **DEL** se pueden borrar las líneas que se deseen: Por ejemplo **DEL** 50, borra la línea 50.

DEL – 50 borra todo hasta la línea 50.

DEL 50 – borra desde la línea 50 hasta el final.

Por otro lado, la memoria ***ROM,*** contiene unas instrucciones, unos programas, que se graban en la fábrica del ordenador. La ***ROM*** contiene los programas necesarios para el funcionamiento del ordenador. La ***Memoria ROM*** nunca se borra, no se puede borrar.

Por ahora hemos terminado con la ***memoria.***

Ahora voy a explicarte cómo se pasa un programa del ordenador al ***caset***.

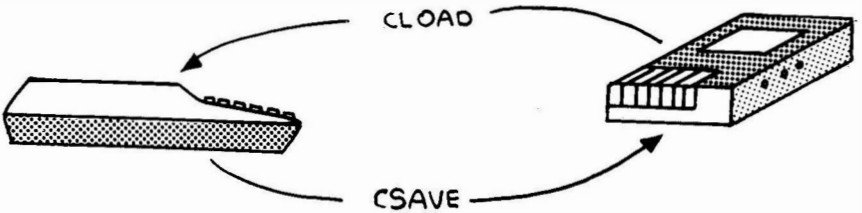
CLEAR. LIST. ENTER.

Si en tu pantalla no aparece ningún programa, copia nuevamente el de **SOUND.**

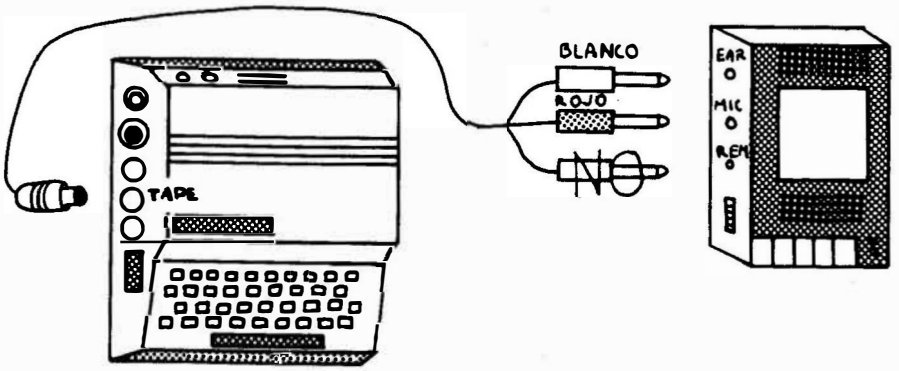
La memoria de fábrica se llama. . . . . .

La memoria donde se almacenan los programas se llama. . . . . .

El ***caset*** sirve para guardar programas. Con **CLOAD** se pasan los programas del ***caset*** al ordenador.



Pues bien, **CSAVE** sirve para archivar programas en el ***caset***. ¿Cómo se hace?



1) Conecta el ***caset*** al ordenador como aparece en el dibujo.
2) Introduce una cinta en el ***caset.***
3) Rebobina la cinta hasta el principio, luego pulsa el Play del ***caset,*** cuenta, despacio hasta 5, y para el ***caset.***
4) Teclea **LIST ENTER** para verificar que hay algún programa en la ***memoria*** del ordenador. Luego **CLEAR.**
5) Regula el volumen del ***caset*** al máximo.
6) Teclea **CSAVE** "PROGRAMA" (o el título que tú quieras, siempre que no pase de 8 letras).
7) Pulsa RECORD–PLAY del ***caset*** y luego **ENTER.**
8) Cuando te aparezca. . .

Y escuches CLICK!

Puedes parar el ***caset.*** El programa está grabado. (OK siempre indica que el ordenador está listo para actuar).

LIST. ENTER.

Con la instrucción . . . . . grabamos los programas en cinta.

¿Has visto?

Para pasar del ***caset*** a la ***memoria*** del ordenador tendrás que hacer como al principio de cada lección, pero en vez de:

CLOADM

sólo pondrás **CLOAD** (sin **M**).

Bien. Para probarlo, una vez que hayas grabado en una cinta un programa, teclea **NEW**, y procede a pasarlo de la cinta al ordenador. Hoy no vamos a ponerte más ejercicios, así que dedica el resto del tiempo a hacer prácticas con

CSAVE "PROGRAMA" y
CLOAD

RESUMEN

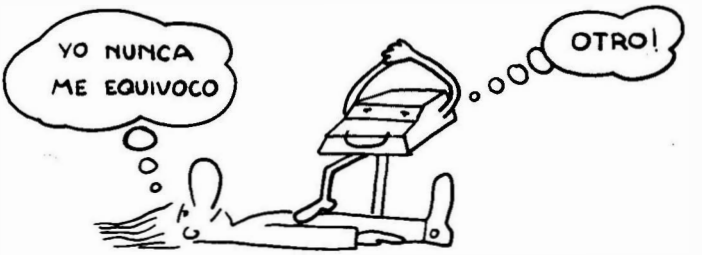
HARDWARE	Parte mecánica o física del ordenador.
CHIPS	Pieza de la memoria del ordenador (circuito integrado).
PERIFERICOS	Aparatos que se conectan al ordenador.
SOFTWARE	Parte lógica del ordenador (no física).
TERMINAL	Periférico del ordenador compuesto por pantalla y teclado.
MEMORIA INTERNA	Memoria que se encuentra en el interior del ordenador.
MEMORIA EXTERNA	Memoria utilizada por el ordenador, que se encuentra en los periféricos.
MEMORIA ROM	Memoria inalterable del ordenador, que contiene los programas necesarios para su funcionamiento.
MEMORIA RAM	Memoria donde se almacenan los programas, instrucciones y datos que introduce el programador.
CLOAD	Carga un programa en la memoria.
CSAVE	Guarda en cinta un programa que está en la memoria.

Modulo 5

¿Has tenido algún problema con las lecciones anteriores? Espero que sí. De lo contrario es que no le estás sacando partido al curso. Si al enfrentarte a un ordenador no tienes problemas, es que el ordenador te está ganando.

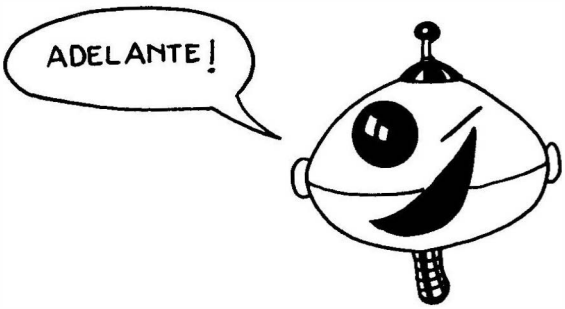
En informática, si algo puede salir mal, saldrá mal. De los errores se aprende, pero sólo podrás equivocarte si haces cosas, si manejas el ordenador.

Confiamos en tu responsabilidad y en tu coraje.



No tenemos ninguna prisa: lo importante es aprender: tenemos que ganarle la partida al ordenador.

Animo y



LENGUAJES

El ***Basic*** es un ***lenguaje de alto nivel*** mediante el cual podemos comunicarnos con un ordenador.

Del mismo modo que para comunicarnos con una persona, tenemos que emplear algún idioma que esa persona entienda, para comunicarnos con un microordenador, tenemos que emplear el ***Basic***. ¿Por qué es el ***BASIC*** un ***lenguaje de alto nivel?***

Un lenguaje informático es de alto nivel cuanto más se parezca a un lenguaje humano: es decir, cuando tiene palabras, entendibles por una persona.

Lenguajes de alto nivel son: ***BASIC, COBOL, FORTRAM, RPG, LOGO, PASCAL,*** etc.

Lenguaje de bajo nivel, es el que entiende el ordenador en su interior: ***lenguaje máquina.***

A este ordenador se le habla en ***BASIC.***

El ***BASIC*** es un conjunto de palabras (Comandos, funciones y reglas), para comunicarnos con el ordenador, de modo que el ordenador nos pueda ser útil.

La diferencia que hay entre un ***lenguaje de alto nivel*** y ***uno de bajo nivel*** es. . . . . .

Con el ordenador nos podemos comunicar en ***BASIC*** de dos modos: Por ejemplo, tecleamos una instrucción y algún dato: (ejecuta los ejemplos).

NEW. ENTER. CLEAR

```
CLS 3   (ENTER)
CLS 5   (ENTER)
PRINT "ESPANTA", "PAJAROS" (ENTER)
SOUND 78, 30  (ENTER)
SOUND 250, 50  (ENTER)
```

Como has podido comprobar, cada vez que has pulsado **ENTER**, después de la instrucción y el dato, la instrucción se ha ejecutado inmediatamente. A este modo de trabajar con el ordenador, se le llama

MODO INMEDIATO O DIRECTO

MODOS DE PROCESO

En un Modo Inmediato no se pueden hacer programas, sólo se pueden ejecutar instrucciones aisladas, como las del ejemplo, o, a lo sumo, ejecutar varias instrucciones.

NEW. ENTER. CLEAR.

CLS 4:

PRINT "ESTO ES": PRINT "UNA PRUEBA": PRINT "del funcionamiento": PRINT "del ordenador": PRINT "en ***modo inmediato***":

SOUND 200, 16: SOUND 50, 20:
SOUND 190, 16 (ENTER)

Hemos ejecutado varias instrucciones, en modo inmediato, separándolas con "dos puntos" (:). Pero este modo es poco práctico.

Qué Nº de ***línea*** tiene el ***modo inmediato***. . . . . .

El otro modo de trabajar con el ordenador es en modo programa o diferido.

MODO PROGRAMA O DIFERIDO

MODO PROGRAMA es cuando el ordenador ejecuta un conjunto de instrucciones y datos agrupados en líneas.

Copia el siguiente ejemplo.

NEW. ENTER. CLEAR.

```
10 CLS 0
20 FOR X=1 TO 62
30 FOR Y=1 TO 30
40 SET (X,Y,RND(8))
50 NEXT Y
60 SOUND X,INT(31/4)
70 NEXT X
80 END
```

RUN. ENTER

¿Has visto qué bonito?

Eso que has copiado, eso que has ejecutado, es un ***PROGRAMA***.

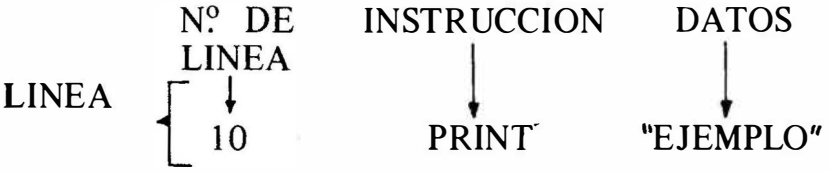
Un ***PROGRAMA*** es ***un conjunto*** de instrucciones y datos, pero, dispuestos en forma de líneas.

Un ***PROGRAMA*** es. . . . . .

En otras palabras: un programa ***BASIC*** es un conjunto de líneas, y una línea es un número seguido de instrucciones y datos.

Teclea **LIST. ENTER**

En cada línea irá alguna instrucción y algún dato:



El número de línea es muy importante, ya que el ordenador ejecutará las instrucciones en el orden que tú le digas: hará primero la línea con número más bajo, luego la siguiente, y así sucesivamente hasta llegar a la última. Al ordenador no le importa que tu teclees: **(NEW. ENTER. CLEAR.)**

```
40 PRINT"DE ACUERDO?"
10 CLS 3
5 REM NUMERO DE LINEA
20 PRINT"EL NUMERO DE LINEA"
30 PRINT"ES MUY IMPORTANTE"
50 PRINT"**==**"
```

porque él leerá las líneas en orden creciente: empezará SIEMPRE, por el número más bajo, y terminará siempre por el más alto. El ordenador lo hace todo en orden: por eso se llama ordenador.

Tú eres un programador. Estás haciendo este curso para aprender a diseñar tus propios programas. El ordenador ordena lo que tú le digas, pero tienes que enseñarle todo, y se lo tienes que enseñar todo en orden.

Por qué ***línea*** empieza a ejecutar el ***PROGRAMA***. . . . .

Ahora copia este programa:

NEW. ENTER. CLEAR.

```
50 PRINT"EL TRON AC";
90 PRINT"QUE EL ORDENADOR"
10 PRINT"ESTO ES UN"
30 PRINT"EJECUCION DE UN"
70 PRINT"PARA QUE VEAS"
60 PRINT"TIVADO"
20 PRINT"EJEMPLO DE"
40 PRINT"PROGRAMA CON"
100 PRINT"EJECUTA EL PROGRAMA"
80 PRINT"EL ORDEN EN"
110 END
```

RUN. ENTER.

TRON – TROFF

Ahora, teclea **TRON. ENTER.** Esta instrucción sirve para ver el orden que sigue el ordenador para ejecutar la líneas.

RUN. ENTER.

Ese programa era muy complejo. Tú lo has tecleado desordenadamente, y el ordenador lo ha hecho en orden, en su orden interior.

Ahora teclea **TROFF. ENTER. TRON** sirve para ver en qué orden va ejecutando el ordenador las líneas. **TROFF** desconecta la instrucción **TRON.**

* * *

Vamos a cambiar de tema. Ya conoces algunas instrucciones **BASIC:**

PRINT, CLS, TRON, TROFF, CLOAD, LIST, RUN, NEW, SOUND, CSAVE, END, REM. CLOADM

TRON sirve para. . . . . .

TROFF desactiva. . . . . .

EDITOR

Ahora vamos a ver la instrucción que, en **BASIC,** sirve para hacer las correcciones de los programas.

Ya sabes que un programa se puede corregir cambiando una línea por otra, pero eso es poco práctico. Hay una instrucción de **BASIC** que sirve para corregir los errores que se pueden cometer en los programas.

Se trata de **EDIT**. ¿Cómo funciona **EDIT?**

NEW. ENTER. CLEAR.

Copia el siguiente programa, tal y como aparece en el recuadro. Cópialo exactamente igual que viene en el recuadro, aunque te parezca que hay errores.

```
10 CLS 18
20 PRINTR "EJEMPLO PARA EL EDITOR CON D"
30 PRINY "EJEMPLO PARA EL EDITOR CON C"
40 PRINT EJEMPLO PARA EDITAR CON I"
50 PRINT"EJEMPLO DE EDITOR CON X VAYA HOMBRE!
```

Ahora pulsa **CLEAR** y teclea **LIST. ENTER**. Ahí tienes el programa que has copiado. En cada una de las líneas hemos introducido un error, que ahora vamos a proceder a corregir con **EDIT**.

Queremos corregir, en principio, la línea 10. Como esta línea es cortita, la corregiremos como ya sabes: vuelve a copiarla, pero en vez de **CLS 18** le pones **CLS 2.**

10 CLS 2 (ENTER)

Para qué sirve **EDIT**. . . . . .

Ahora vamos a la línea 20.

Teclea **EDIT** 20. **ENTER.**

```
2Ø PRINT  R  "Ejemplo para editar con D"
2Ø ■
```

Eso es lo que te aparece al final de la Pantalla. Es decir, te aparece la línea que vas a corregir, y debajo una línea en blanco, que es donde vas a hacer la correción. Lo que queremos es suprimir la R que hay después de **PRINT**. Bien: Ve pulsando la ***barra espaciadora*** hasta que R quede ***bajo el CURSOR***. Ahora pulsa la letra **D**. Ahora **ENTER**. Ya está corregido. **EDIT** con **D** borra lo que haya debajo del **CURSOR**.

D se usa. . . . . .

Ahora **EDIT** 30 **ENTER**

```
3Ø PRINY "ejemplo para editar con C"
3Ø ■
```

Lo que está mal en esa línea es la Y que hay en lugar de la T de **PRINT**. Ve pulsando la ***barra espaciadora*** hasta situar el ***CURSOR*** sobre la Y. Ahora pulsa C. y ahora teclea T. **ENTER**.

Ya has corregido la línea 30. Ahora limpia la Pantalla con **CLEAR**.

C se usa. . . . . .

EDIT 40 **ENTER**

```
4Ø PRINT "Ejemplo para editar con I"
4Ø ■
```

Esta vez el error consiste en que nos hemos olvidado de poner el texto, entre comillas. Para insertar las comillas, ve pulsando la ***barra espaciadora***. Cuando estés justamente debajo de la E de ejemplo, pulsa la tecla **I**, luego pon las comillas, y luego **ENTER**. **I** sirve para insertar algo que se nos ha olvidado.

I sirve para . . . . . .

Ahora **EDIT** 50 **ENTER**.

```
5Ø PRINT "Ejemplo de editar con X"
5Ø ■
```

En esta ocasión hemos puesto letras de más al final de la línea, y, a parte, se nos ha olvidado poner las comillas del final. Bien, pulsa la tecla X: como verás el ***CURSOR*** se ha ido al final de la línea. Ahora, con la tecla borradora o de retroceso (←) borra hasta la X, luego pon las comillas, y **ENTER**.

CLEAR

```
10 CLS 2
20 PRINT"EJEMPLO PARA EL EDITOR CON D"
30 PRINT"EJEMPLO PARA EDITAR CON C"
40 PRINT"EJEMPLO PARA EDITAR CON I"
50 PRINT"EJEMPLO DE EDITOR CON X "
```

```
10 PRINT"EDIT:CON EDIT CORREGIMOS LAS LINEAS"
20 PRINT"D:BORRA LO QUE HAY BAJO EL CURSOR"
30 PRINT"C:CAMBIA UN CARACTER POR OTRO"
40 PRINT"I:INSERTA UN CARACTER EN MEDIO DE OTRO
50 PRINT"X:VA AL FINAL DE LA LINEA Y DESDE
   ALLI SE PUEDE BORRAR"
```

Cuál es la tecla borradora. . . .

Por hoy, no está mal. Ahí te dejo unos programas para que los ejecutes.

```
10 CLS
20 PRINT"CAMBIA a Y b  EN EL PROGRAMA
   POR LOS VALORES QUE QUIERAS"
30 PRINT
40 PRINT"PARA CALCULAR LA SUMA DE LOS
   CUADRADOS DE a Y b"
50 PRINT
60 PRINT"DESPUES HAS run 80"
70 END
80 PRINTA↑2+B↑2
```

```
10 PRINT1234
20 PRINT
30 PRINT 1234
40 PRINT
50 PRINT;1234
60 PRINT
70 PRINT,1234
80 PRINT
90 PRINT1,2,3,4
100 PRINT
110 PRINT"1234"
120 PRINT
130 PRINT1;2;3;4
```

```
5 REM ORGANILLO
10 A$=INKEY$
20 IF A$="" THEN 10
30 A=ASC(A$)
40 PRINT A$;
50 SOUND A*2,1
60 GOTO 10
```

Este programa te va a encantar. Cuando acabes el curso tú sabrás hacer programas como éste. Primero, antes de ejecutarlo, quiero hacerte una advertencia: Este programa te sirve tanto para divertirte como para aprender. Tu ordenador se convierte en una caja musical. Puedes teclear en minúsculas o en Negativo, todos los caracteres del teclado, excepto **BREAK**. Cada vez que pulses **BREAK** romperás el programa. Te ayudará mucho a conocer el valor de cada tecla, ejecutar antes de este programa, el programa en que aparecían los caracteres del Código ***ASCII***.

Tampoco tengo que decirte que debes guardar todos los programas en tu caset, con un nombre distinto cada uno.

RESUMEN

Lenguaies de alto nivel son los más parecidos al lenguaje humano: el Basic, el Cobol, etc.

Lenguaje de bajo nivel es el que trabaja el ordenador internamente.

Un ***lenguaje (Basic, Cobol)*** es un conjunto de comandos, funciones y reglas para comunicarnos con el ordenador.

Cuando queremos pedir un resultado inmediato al ordenador se lo pedimos en ***modo inmediato,*** es decir, sin número de línea. Cuando necesita procesos complejos lo hacemos en ***modo diferido*** o a través de un programa con número de línea.

Un programa Basic es un conjunto de líneas y una línea es un número seguido de instrucciones y datos.

Tron activa el rastreo del programa.

Troff desactiva el **Tron.**

Edit nos edita una línea para corregir algún dato erróneo.

Solución al Pasatiempo

Es blanco porque sólo en el Polo Norte puede darse esa circunstancia.

EJERCICIOS DEL MODULO 5

1) Cuál es correcta:
   a) Sound 15, 32
   b) Sound 290, 16
   c) Sound 150, 290

2) Cuando se apaga el ordenador se borra:
   a) La memoria ROM
   b) La memoria REM
   c) La memoria RAM

3) Las líneas se corrigen con:
   a) No se puede
   b) EDIT
   c) TRON

4) Qué línea es incorrecta:
   a) 40 RUN
   b) 110 PRINT A
   c) 60, PRINT A+1

5) La función X de EDIT:
   a) Borra una letra.
   b) Te lleva al final de la línea
   c) Sirve para insertar palabras

6) La función I de EDIT sirve para:
   a) Nada
   b) Borrar letra a letra
   c) Insertar

7) La función C de EDIT sirve para:
   a) Borrar la línea de golpe
   b) Borrar la línea letra a letra
   c) Cambiar un carácter por otro

8) La función D de EDIT sirve para:
   a) Borrar un carácter
   b) Borrar un línea
   c) Ninguno

9) Qué instrucción usamos para ver en qué orden ejecuta el ordenador las líneas:
   a) CLS
   b) TRON
   c) TROFF

10) Para borrar algunas líneas se utiliza:
   a) RUN 100
   b) EDIT X
   c) DEL

SOLUCIONES:

A; C; B; C; B; C; C; A; B; C.

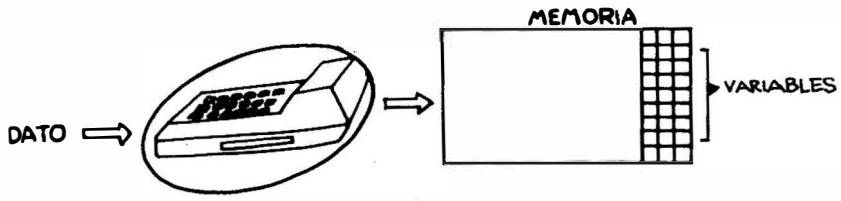
Modulo 6

La informática es la ciencia que procesa datos. Un ***ordenador*** recibe ***datos,*** trabaja con ellos, los procesa, y emite una ***información*** que antes desconocíamos. El ordenador trabaja con datos. Un ordenador sirve para manejar datos, sin datos no sirve para nada. Para que un ordenador funcione hay que introducirle datos.

INFORMATICA es. . . . . .

LAS VARIABLES

Los datos se introducen al ordenador por medio del teclado; desde el teclado pasan al interior del ordenador, van a la memoria, pero no se meten en cualquier parte de la memoria, se meten en las ***variables,*** que son unos compartimentos especiales.



Bueno, no todos los datos van a las variables.

Los datos pueden introducirse directamente al lado de una ***instrucción,*** y siempre que ejecutes esa ***instrucción*** el ***dato*** aparecerá en el mismo sitio, de un modo ***constante.***

Dato Constante es aquel que, como su nombre indica,permanece constante a lo largo de todo el programa, y siempre aparecerá en el mismo lugar.

```
10 PRINT "Hola"
20 END
```

Cada vez que tu ruedes el programa anterior, aparecerá HOLA en la pantalla y en el mismo sitio. HOLA es un dato constante, no varía; siempre aparecerá en el mismo lugar.

En cambio, un dato que se guarda en una ***variable,*** puede utilizarse muchas veces a lo largo del programa y si el dato es un número puede cambiar su valor.

El ***dato constante*** solo se puede usar una vez en el programa.

El ***dato*** que se guarda en la ***variable*** se puede utilizar varias veces a lo largo del ***programa***.

```
10 LET A=0
20 A=A+10
30 CLS RND(9)-1
40 PRINT@ 238,A;
50 SOUND A,3
60 IF A=250 THEN 60
70 GOTO 20
```

Es una ***variable,*** y por ello se puede utilizar varias veces, y por ello su valor varía en cada ***línea:*** cada vez que se ejecuta el ***programa*** te sale un color, un número distinto y un sonido distinto.

La ***instrucción*** que se utiliza para dar valor a las ***variables*** es

L E T

Se pone:

10 LET A = 48

o

10 LET CA = A * H

o

10 LET N3$ = "15 años"

o

10 LET S$ = " * "

En las variables se guardan datos. La instrucción que sirve para guardar datos en las variables es L E T.

Cada ***variable*** tiene un nombre. **El nombre de una variable siempre empieza por una letra** y es conveniente que no sea muy largo.

NOMBRES DE VARIABLES

A	HR$
CASA	C1$
TR	P2$
P1	R2D2$
N2	SI$

Cada variable debe tener un nombre distinto.

Las variables pueden ser de dos tipos:

NUMERICAS ⇓ NUMEROS PARA HACER OPERACIONES

DE CADENA (o ALFA-NUMERICAS) ⇓ CADENAS DE CARACTERES

En las variables NUMERICAS se guardan números para hacer operaciones, o letras que representan a números.

```
10 A=5
20 B=10
30 C=A*B
40 PRINT C
```

A, B, C son VARIABLES NUMERICAS
son letras que representan
a números

En las ***variables*** numéricas guardamos. . . .

En las variables de CADENA $ se guardan "Cadenas" de caracteres. Lo que se guarda en las variables de cadena tiene que ir entre comillas, y al final del nombre de la variable tiene que ir el signo $.

```
10 A$="ESTO ES"
20 B$="UN EJEMPLO"
30 C$="DE VARIABLE EN CADENA"
40 PRINT A$;B$
50 PRINT C$
```

En las ***variables*** alfanuméricas guardamos. . . . . .

Si te equivocas y guardas ***palabras*** en una ***variable numérica,*** o no le pones ***comillas*** al contenido de una ***variable de cadena,*** el ordenador te advertirá que te has equivocado con

TM ERROR

Para comprobarlo copia 100 LET A = "observa"

Y ahora, **RUN** 100 **ENTER**

Como puedes observar, te da error **TM** en la línea 100, porque has querido guardar una cadena en una ***variable*** numérica. **RUN** 100 ejecuta desde la ***línea*** 100.

Ahora **DEL** 100 y **ENTER**, con ello borras la línea 100.

El **ERROR TM** nos indica. . . . . .

Dijimos que el ***comando*** **LET sirve para guardar datos en las variables.** Pues bien: el ordenador se sabe de memoria el comando **LET,** y cuando quieras dar valor a una ***variable,*** no hace falta (si quieres) poner **LET;** ejemplo:

```
10 LET A$="COMO SABES,25*10 ES IGUAL A"
20 LET B=25
30 LET C=10
40 PRINT A$;B*C
```

```
10 A$="COMO SABES,25*10 ES IGUAL A"
20 B=25
30 C=10
40 PRINT A$;B*C
```

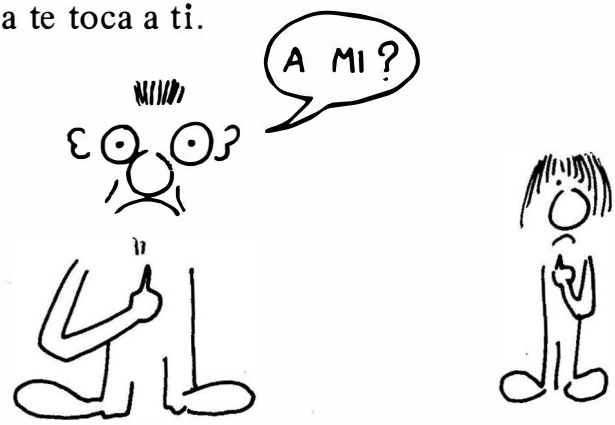
Se puede poner:

LET A$ = ...

o

AS$ = ... etc.

Ahora te toca a ti.



Sí, a ti.

Debes copiar y ejecutar los siguientes ***programas:***

```
10 REM"RESOLUCION DEL TRIANGULO"
20 PRINT"DIME UN CATETO"
30 INPUT A
40 PRINT"DIME OTRO CATETO"
50 INPUT B
60 LET C=(A↑2)+(B↑2)
70 LET C=C↑(1/2)
80 PRINT"LA HIPOTENUSA VALE";C
90 END
```

```
10 REM"AREA DE UN TRIANGULO"
20 PRINT"DIME LA BASE"
30 INPUT B
40 PRINT"DIME LA ALTURA"
50 INPUT H
60 LET A=(B*H)/2
70 PRINT"EL AREA VALE";A
80 END
```

```
10 REM"AREA DEL CIRCULO"
20 PRINT"DIME EL RADIO"
30 INPUT R
40 LET A=3.141592*(R↑2)
50 PRINT"EL AREA DEL CIRCULO VALE";A
```

Una vez que hayas ejecutado los programas, guárdalos en tu caset. Intenta hacer algo parecido, aunque no te salga, con otras fórmulas.

RESUMEN

La **INFORMATICA** es la ciencia que procesa datos por medio de máquinas.

Una **VARIABLE** es un casillero de la memoria donde guardamos un dato.

La **INSTRUCCION LET** se utiliza para dar valor a una Variable.

El nombre de una Variable tiene que empezar SIEMPRE por una letra y no ser muy largo.

Las **VARIABLES NUMERICAS** sirven para guardar NUMEROS.

En las **VARIABLES ALFANUMERICAS** guardaremos Cadenas de caracteres.

El **ERROR TM** se producirá cuando queramos guardar un Número en una Variable Alfanumérica o lo contrario.

EJERCICIOS DEL MODULO 6

1) Una variable es:
   a) Un casillero de memoria.
   b) Un dato.
   c) Un número.

2) El nombre de las variables empieza por:
   a) Un número.
   b) Un número o una letra.
   c) Una letra.

3) El nombre de las variables es conveniente que sea:
   a) Largo
   b) Corto.
   c) Es igual.

4) Las variables pueden ser:
   a) Numéricas, o alfagenéticas.
   b) De cadena, o numeradas.
   c) Numéricas, o de cadena.

5) Cuál no es correcta:
   a) LET A = "cadena".
   b) LET A$ = "1.980".
   c) A = A + 1

6) Cuál es la correcta:
   a) A = C + A
   b) 5A = 5A
   c) P3 = "alfa".

7) Para asignar valor usamos la instrucción:
   a) TRON
   b) PRINT
   c) LET

8) Si aparece TM error:
   a) No pasa nada.
   b) Hay que borrar el programa.
   c) Hay error de asignación.

9) DEL sirve para:
   a) Borrar líneas concretas.
   b) Limpiar la pantalla.
   c) Borrar la ROM.

10) RUM 190 ejecutará el programa:
   c) Desde le principio.
   b) Desde la línea 190.
   c) Hasta la línea 190.

SOLUCIONES

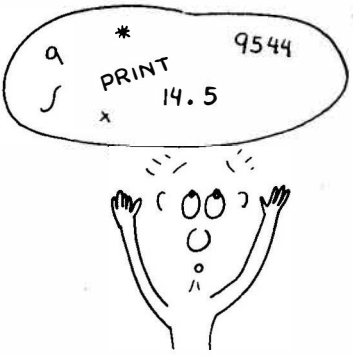
A; C; B; C; A; A; C; C; A; B.

Pasatiempos

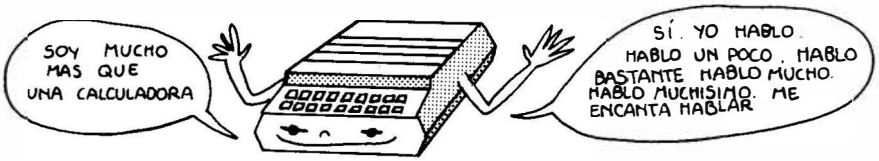
SOLUCION:

El número de posibles caminos entre A y B es de 70.

Modulo 7

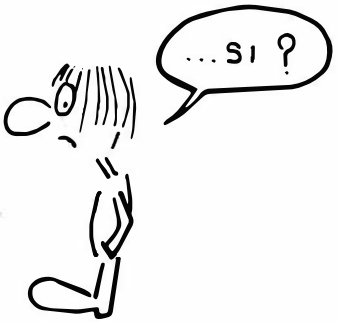


Hoy es el día de los números. El ordenador es, ante todo, una calculadora, una calculadora que habla.



El ordenador maneja, ***procesa*** los ***datos*** que tú le introduces. Unos son letras, otros son números. ¿Cómo ***se convierte*** el ordenador en una calculadora?

Ya lo sabes.



Claro. El ordenador se convierte en calculadora tecleando **PRINT** (o **SHIFT** ? que es lo mismo). Luego basta plantear cualquier operación.

Por ejemplo:

```
10 A=30
20 B=7
30 C=5
40 PRINT A↑2
50 PRINT A*C
60 PRINT A/B
70 PRINT A+B+C
80 PRINT A-B-C
90 END
```

Con las tres ***variables*** de las ***líneas*** 10, 20 y 30, en las líneas 40 a 80 hemos hecho distintas operaciones.

¿Has ejecutado el ***programa?*** Si no lo has hecho, hazlo ahora.

* * *

PRINT con los ***operadores aritméticos*** convierte al ***micro*** en una calculadora, que puede hablar.

Ejecuta el siguiente ***programa:***

```
10 REM PROGRAMA PARA CALCULAR TANTOS POR CIENTOS
20 INPUT"DIME EL NUMERO AL QUE QUIERES CALCULAR
   EL %";A
30 PRINT
40 INPUT"DIME EL % QUE QUIERES CALCULAR";B
50 PRINT
60 PRINT"EL";B;"%DE";A;"ES";A*B/100
70 PRINT"*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*"
80 RUN
```

Para "escapar" de ese programa, utiliza **BREAK**.

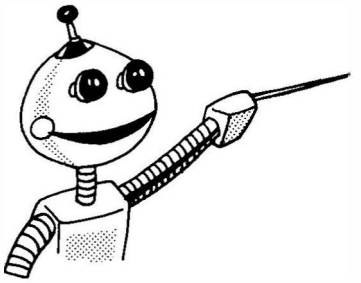
En ese programa has calculado tantos por ciento. Por si no lo has estudiado aún, el tanto por ciento de un número se halla multiplicando el número, por el tanto por ciento, y luego se divide por 100. Por ejemplo, el 10 % de 50 se halla multiplicando 10 * 50 y dividiéndolo por 100.

$$\frac{10 * 50}{100} = 5$$

* * *

Como recordarás, de la lección audiovisual, los ***operadores aritméticos*** tienen un orden de ejecución.

Si en una operación te aparecen varios operadores, se ejecutan en el siguiente orden.



1º	↑	POTENCIACION
2º	*	MULTIPLICACION
	/	DIVISION
3º	+ −	SUMA Y RESTA

La primera categoría y orden corresponde a la potenciación. Es lo primero que hace el ordenador.

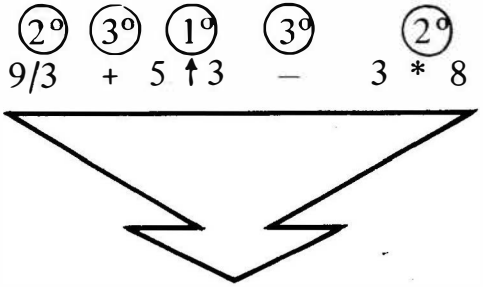
(1º)

5 ↑ 3 (= 125)

En segundo lugar hará las multiplicaciones y divisiones, empezando por la izquierda.

$$\overset{2^o}{9/3} \quad (=3) \qquad\qquad \overset{2^o}{3 * 8} \quad (=24)$$

En tercer lugar, hará lo más fácil: las sumas y restas, también de izquierda a derecha.



$$3 + 125 - 24 = 104$$

El ordenador le da mucha importancia al orden para hacer las operaciones, porque no es lo mismo

$$(15 - 4) * 3 = 33$$

que

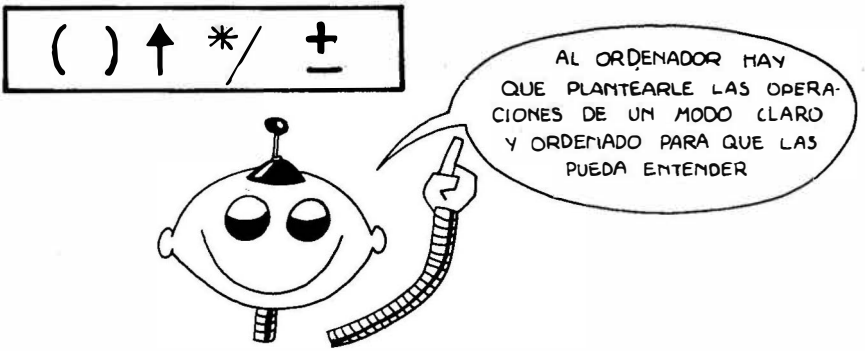
$$15 - (4 * 3) = \mathbf{3}$$

Por eso, para hacer las operaciones es conveniente separarlas mediante paréntesis.

> El ordenador empieza por la . . . . . .
>
> cuando tienen igual categoría.

Cuando el ordenador se encuentra una operación para realizar, lo primero que hace es lo que hay entre paréntesis ( ). Si hay varios paréntesis empieza por los de la izquierda, a continuación hace las potenciaciones ↑ , a continuación hace las multiplicaciones * y divisiones / , empezando también por la izquierda. Para terminar, el ordenador hace las sumas + y restas − .

En definitiva, el ordenador hace primero lo difícil y luego lo fácil.



Ahora, muy despacio, resuelve esos ejercicios: primero sin ***ordenador***. Luego, para comprobar las respuestas, introduce los datos en la calculadora **PRINT** y ejecútalos.

Estamos enseñando aritmética al ordenador.

Adelante.

1º) 5 ↑ 3 − 8 * 4 + 6/2 =
2º) 3 * 9 * 8 − 3 − 1 =
3º) 5/1 * 2 ↑ 3 / 5+(3 * 9) =
4º) 4 * 3 − (5 * 2) * (7 + 8) / 10 =

Con la práctica irás adquiriendo mayor soltura y te será más facil plantear las operaciones.

Vamos a hablar ahora de varios puntos importantes, muy relacionados con **PRINT**.

.	,	;	:
PUNTO	COMA	PUNTO Y COMA	DOS PUNTOS

El BASIC es un lenguaje de programación mediante el cual nos comunicamos con el ordenador. Los lenguajes tienen palabras, gramática, ortografía. En el ***BASIC*** los ***signos de puntuación*** son muy importantes.

El punto se utiliza para los decimales. En ***BASIC*** los números enteros no llevan ni punto ni coma.

1000 14328 116531 1944709

En cambio sí se les pone punto a los decimales.

4.3 77.8 .9 165.72

En ***BASIC*** los decimales se ponen con punto.

● PUNTO

* * *

En cambio la coma , y el punto y coma ; se utilizan para otras cosas.

PRINT tiene dos utilidades fundamentales en ***BASIC:***

1) Imprimir en Pantalla (lo que sea entre comillas).

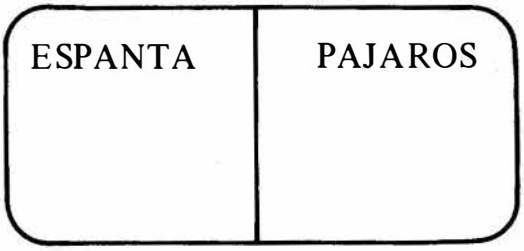
2) Actuar como calculadora (con números).

La coma SEPARA ,

El punto y coma UNE ;

```
5 CLS
10 PRINT"ESPANTÁ","PAJAROS"
20 PRINT
30 PRINT"CORTA";
40 PRINT"PLUMAS"
50 END
```

Como habrás visto la coma divide la pantalla en dos, escribiendo en el lado izquierdo lo que estaba a su izquierda, y en el lado derecho de la pantalla lo que estaba a su derecha.



El punto y coma hace todo lo contrario: une lo que tiene antes con lo que tiene después.

No te vamos a repetir más esto, porque es un tema que verás en casi todos los programas que hagas. Lo que sí te quiero advertir es que muchos de los

S/N ERROR

que te aparezcan pueden venir motivados porque hayas colocado una coma, un punto o un punto y coma en lugar indebido, o porque no los has puesto, o porque los hayas puesto de más.

* * *

En cuanto a los dos puntos : tienen una importante misión. Cada línea de un programa contiene una instrucción. **Si quieres meter dos instrucciones o más, en una línea, tienes que separarlas por: dos puntos.**

```
10 CLS 2:LET A=2:PRINT@ 230,"EJEMPLO DE:";
20 SOUND 1,A:SOUND 50,A:SOUND 100,A
30 RUN
```

Rueda ese programa. (Para escapar de él: BREAK) Como verás en la línea 10 hemos metido 3 instrucciones, y en la 20 hemos metido otras 3.

Podemos meter varias instrucciones en una línea, separándolas por (:) dos puntos.

* * *

SHIFT ←——	Borra líneas enteras.
DEL	Borra líneas aisladas de la memoria.
SHIFT @	Detiene la ejecución de programa.
RUN N [número de línea]	Ejecuta desde la línea que tú le digas.

Haz pruebas con **RUN**, **RUN** 10 **RUN** 20, **RUN** 30, **RUN** 40, **RUN** 60, **RUN** 70, **RUN** 80, **RUN** 90.

NEW. ENTER. CLEAR

```
5 CLS 4
10 PRINT"LINEA 10";:SOUND 1,8
20 PRINT"SEGUNDA LINEA":SOUND 35,8
30 PRINT"LINEA TERCERA":SOUND 70,8
40 PRINT"LINEA 40":SOUND 105,8
50 END
60 CLS 3
70 PRINT"OTRO PROGRAMA":SOUND 250,4
80 PRINT"YES?":SOUND 255,4
90 PRINT"CON RUN":SOUND 190,4
100 END
```

Solo algunos comandos y controles que ya conoces.

←	Borra retrocediendo.
SHIFT ←	Borra de una vez un renglón entero.
SHIFT @	Detiene la ejecución de un programa.
RUN	Con un número de línea detrás empieza a ejecutar el programa desde la línea que tú le has dicho.

Para practicar estos controles y comandos, copia el siguiente programa y ejecútalo: ← y **SHIFT** ← utilízalos si te equivocas. **RUN** N.º de línea.

RUN N (N = Nº de líneas desde donde quieras que empiece el programa)

Pulsando **SHIFT** y @ simultáneamente se detiene el programa. Para reanudarlo pulsa @ u otra tecla cualquiera. ¡Adelante!

Contador

```
10 CLS 6
20 A=0
30 A=A+1
40 PRINT@ 237,A;
50 GOTO 30
```

Haz pruebas con SHIFT @ y con BREAK

dibujo de colores

```
10 CLS0
20 FOR Y=0 TO 31
30 FOR X=0 TO 63
40 C=C+1:IF C>8 THEN C=0
50 SET(X,Y,C)
60 NEXT X:NEXT Y
70 GOTO 70
```

Prueba y modifica X de 0 a 62, Y desde 0 a 31, y C, desde 0 a 8

Juego de rapidez con SHIFT @

```
10 REM CONCURSO DE RAPIDEZ
20 PRINT"QUIEN SE ACERCA MAS AL NUMERO QUE
   DIGA EL OTRO, PULSANDO SHIFT@";
30 SOUND 200,32:SOUND 222,16:SOUND 1,2
40 A=0
50 A=A+1
60 PRINT@ 237,A;
70 GOTO 50
```

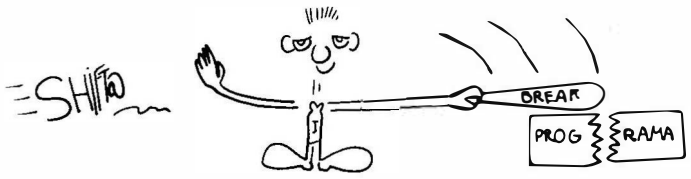
Con **SHIFT** @ se detiene el ***programa.***

En cambio, si en vez de pulsar **SHIFT** @ pulsas **BREAK**; no sólo se detiene el programa, sino que se rompe en la línea que está ejecutando en ese momento, y te lo dice:

BREAK X

Para reanudarlo tienes que teclear **RUN. ENTER.**

Después de **SHIFT** @ , cualquier tecla.
Después de **BREAK, RUN (ENTER)**



Bien por hoy es bastante, aunque se me olvidaba que nos queda por ver un comando: **PRINT** @

P R I N T @

¿Para qué sirve? **PRINT** sirve para imprimir, **PRINT** @ sirve para imprimir en un lugar determinado de la Pantalla.

NEW. ENTER. CLEAR

```
10 PRINT@ 30,CHR$(143);
20 PRINT@ 60,CHR$(159);
30 PRINT@ 90,CHR$(175);
40 PRINT@ 120,CHR$(191);
50 PRINT@ 150,CHR$(207);
60 PRINT@ 180,CHR$(223);
70 PRINT@ 210,CHR$(239);
80 PRINT@ 240,CHR$(255);
90 PRINT@ 290,"ES UN EJEMPLO DE PRINT@";
100 GOTO 100
```

CLEAR RUN. ENTER

RETICULA PRINT@

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351
352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383
384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447
448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479
480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511

¿Cómo se hace?

PRINT @ N, "ALGO"

NUMERO ENTRE 0, 511

Esa es la **RETICULA PRINT** @ que representa la pantalla. Cada uno de esos cuadraditos es una posición de la pantalla. Con **PRINT** @ podemos posicionarnos en la Pantalla. Esa **RETICULA PRINT** @ tiene 512 cuadritos, desde 0 hasta 511, ambos incluídos.

Para posicionarnos en la Pantalla se pone

PRINT @ N [número del casillero de la retícula Print @]

, "lo que se quiera imprimir"

Haz pruebas en modo inmediato o directo. Por ejemplo:

NEW. ENTER. CLEAR

PRINT @ 100, "Ejemplo de Print @"

[no olvides poner la coma, y las comillas]

Haz otros, (Puedes hacer dibujos, etc.)

* * *

Para **eso sirve PRINT** @

* * ⁘

Con PRINT @ , " " se posicionan "Cadenas" en la Pantalla.

Para posicionarnos en la Pantalla, usamos. . . . .

Pero ¿y si en vez de cadenas de caracteres quieres imprimir cuadritos para hacer un dibujo? Esos cuadritos se llaman *"caracteres gráficos"* del Código ***ASCII***

	VERDE	AMARILLO	AZUL	ROJO	BEIGE	TURQUESA	MAGENTA	NARANJA
	128	144	16Ø	176	192	2Ø8	224	24Ø
	129	145	161	177	193	2Ø9	225	241
	13Ø	146	162	178	194	21Ø	226	242
	131	147	163	179	195	211	227	243
	132	148	164	18Ø	196	212	228	244
	133	149	165	181	197	213	229	245
	134	15Ø	166	182	198	214	23Ø	246
	135	151	167	183	199	215	231	247
	136	152	168	184	2ØØ	216	232	248
	137	153	169	185	2Ø1	217	233	249
	138	154	17Ø	186	2Ø2	218	234	25Ø
	139	155	171	187	2Ø3	219	235	251
	14Ø	156	172	188	2Ø4	22Ø	236	252
	141	157	173	189	2Ø5	221	237	253
	142	158	174	19Ø	2Ø6	222	238	254
	143	159	175	191	2Ø7	223	239	255

Pues bien, con esta instrucción, complementaria,

C H R $, (N)

↓

[N.º de Código ASCII, entre 0 y 255]

de **PRINT** puedes imprimir, en vez de letras, cuadritos de color para hacer dibujos.

PRINT @ N, CHR$ (N)

Ejecuta el ejemplo:

```
10 CLS
20 FOR A=128 TO 255
30 PRINT@ 105,CHR$(A);:PRINT@ 118,CHR$(A);
40 PRINT@ 393,CHR$(A);:PRINT@ 406,CHR$(A);
50 PRINT@ 206,CHR$(A);:PRINT@ 209,CHR$(A);
60 PRINT@ 302,CHR$(A);:PRINT@ 305,CHR$(A);
70 PRINT@239,CHR$(A),:PRINT@ 240
80 PRINT@271,CHR$(A),:PRINT@ 272,CHR$(A);
90 SOUND RND(255),16
100 NEXT
```

RUN (ENTER)

Por hoy está bien, ahora, intenta hacer algún dibujo distinto, modificando el programa anterior mediante **EDIT**. Utiliza los caracteres gráficos del Código ***ASCII***

(Te será fácil cambiando los números que hay al lado de @).

RESUMEN

El Orden en que el Ordenador realiza las operaciones aritméticas es el que sigue.

1º **Los Paréntesis.**

2º **Las Potenciaciones.**

3º Después las **Multiplicaciones y las Divisiones.**

4º Y por último las **Sumas y las Restas**

En caso de haber varias de un mismo nivel de dificultad, empieza por la Izquierda.

El Punto nos indica los Decimales.

La Coma divide la pantalla en dos partes.

El Punto y Coma une dos impresiones (PRINT).

Los dos Puntos indican que es otra Instrucción.

PRINT @ sirve para imprimir en una posición específica de la pantalla. Su rango va desde 0 hasta 511.

La función **CHR$** (X) nos permite ver el carácter del Código ASCII que corresponde a X.

Pasatiempos

SOLUCION:

El parentesco tan complejo que une a las dos mujeres con los dos hombres es debido a que los dos hombres son viudos y cada uno está casado con la hija del otro, que son las dos mujeres que comentan el parentesco.

EJERCICIOS DEL MODULO 7

1. La retícula PRINT @ va de:

   a) 0 a 510
   b) 1 a 255
   c) 0 a 511

2. Con CHR$ (...) imprimimos los caracteres del:

   a) Basic
   b) Código binario
   c) Código ASCII

3. El Código ASCII va desde:

   a) 0 a 255
   b) 0 a 511
   c) 1 a 256

4. PRINT @ :

   a) Te ayuda a posicionarte en la pantalla
   b) Te saca un número entre 0 y 500
   c) Detiene la ejecución del programa

5. SHIFT @ :

   a) Es lo mismo que PRINT
   b) Es lo mismo que PRINT @
   c) Detiene el programa

6. BREAK:

   a) Es lo mismo que SHIFT @
   b) Detiene el programa
   c) Interrumpe el programa

7.- Cuál es la correcta:

a) 30 PRINT 1 : PRINT A
b) 190 PRINT " @ " : 200 PRINT
c) 40 REM 60 : END 40

8.- Cada número del Código ASCII es igual a:

a) 0
b) Un carácter
c) Un cuadradito

9.- El botón RESET es:

a) Igual que ON/OFF
b) Igual que CLEAR
c) Como BREAK y CLEAR juntos

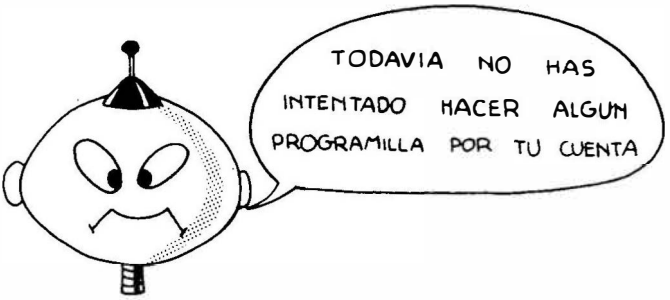
10.- Cuál es incorrecto:

a) ? 5, 6, * 3
b) ? 4/3, 1
c) ? .3 * . 4

SOLUCIONES

C; C; A; A; C; C; A; B; C; A;

Modulo 8



El universo de la Informática está vedado para aquellos que no se atreven.

Para obtener gran rendimiento de tu micro, para aprender informática – BASIC, no basta con que hagas las lecciones audiovisuales y las del libro : NO es suficiente. **Tienes que atreverte a hacer tus propios programas.** En este momento tienes conocimientos suficientes para intentarlo. Cada uno de los programas que copies del libro son susceptibles de modificación por parte tuya.

PRINT imprime o hace operaciones

LET asigna valor a los variables

I N P U T

Ahora vamos a ver una importante instrucción.

I N P U T

El ordenador funciona con datos. Los datos se pueden introducir en el ordenador de dos modos:

1º Directamente con una instrucción . Ejemplos:

```
80  CLS 4          (dato)        1010 PRINT @250, "ejemplo'   (datos)
17  GOTO 58        (dato)        70  PRINT 7 + 3              (datos)
```

2º Mediante asignación con **LET**, es decir, metiendo el dato en una variable numérica o de cadena. Ejemplos.

```
70  LET A= 5          |    10  LET H = 250
80  CLS  A            |    20  LET B$ = "Elegante"
150 CLS A             |    1010 PRINT @ H , B$
                      |    1500 PRINT B$
```

Sólo con prácticas podrás establecer cuando te interesa guardar datos en variables; y cuando no (Para las instrucciones **PRINT IF. . . . THEN, FOR. . . . NEXT, INPUT,** siempre es conveniente asignar variables con **LET**).

```
LET  A = 308
LET B$ = "Programación"
```

LET asigna valor a una variable

INPUT también sirve par asignar valor a una variable, pero a diferencia de **LET**, el valor se introduce desde fuera del programa.

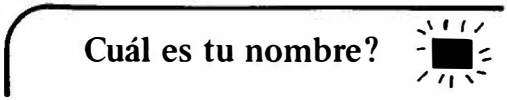
Si tú ejecutas el siguiente programa, observarás que **LET** asigna un valor a las variables, y es el valor que **PRINT** imprime cada vez que se rueda el programa.

```
10 CLS 4
20 LET A$="PEPITO"
30 LET B$="ES MI AMIGO"
40 PRINT@ 250,A$
50 PRINT@ 299,B$
60 END
```

Ejecútalo un par de veces más. Ese programa sólo servirá para Pepito. En cambio, ejecuta el siguiente, tecleando **RUN** 100 (para saltarnos el programa anterior) y **ENTER**.

```
100 CLS 5
110 INPUT"CUAL ES TU NOMBRE";A$
120 LET B$="ES MI AMIGO"
130 PRINT@ 200,A$
140 PRINT@ 300,B$
150 END
```

INPUT es una instrucción que sirve para dar valor a las variables desde fuera del programa. El ordenador va ejecutando las líneas una a una por orden numérico; cuando se encuentra con un **INPUT** se detiene y pide un dato por la pantalla.



Y hasta que no se le introduzca ese dato al programa no podrá continuar. El programa de **LET** solo servía para Pepito, éste vale para todos tus amigos.

INPUT detiene la ejecución del programa, y éste no se reanuda hasta que tú no le introduces un número o una cadena. INPUT te dice: Por favor dame datos o no podremos continuar.

El formato de **INPUT** es muy sencillo.

INPUT V (V = variable)

80 INPUT J | 110 INPUT P$

REDO

INPUT sirve para dar valor a variables desde fuera del programa: en las variables numéricas guardaremos números, y en las de cadenas guardaremos cadenas.

NEW. ENTER. CLEAR

```
10 CLS
20 INPUT A
30 PRINT:PRINT A
40 END
```

Ejecuta ese programa, poniendo tu nombre.

```
? REDO
```

REDO es un error que consiste en guardar números en una variable de cadena, o cadenas en una variable numérica.

Cámbiale al programa anterior A por A$, y ejecútalo introduciendo tu nombre.

¿Ves ahora si funciona?

Para terminar con **INPUT**, te diré que tiene otra característica muy útil.

```
550 INPUT "DIME TU NOMBRE"; A$
1409 INPUT "DAME UN NUMERO"; X
```

A INPUT se le puede poner un mensaje, un título, para que se sepa qué es lo que pide; como puedes ver en los ejemplos anteriores. El mensaje tiene que ir entre comillas.

Lo último que te quiero decir de **INPUT** es que, en un mismo **INPUT**, en una misma instrucción, se pueden poner varias variables mediante comas.

30 INPUT "Dime tu nombre, edad, peso"; A$, B, C

Como ves son tres datos los que se pueden guardar con un solo **INPUT**.

* * *

Sólo practicando aprenderás BASIC. Copia esos programas con la memoria que tiene, luego velos ejecutando uno a uno: **RUN**, **RUN** 200, **RUN** 300. Adelante.

```
10 REM REGLA DE TRES
20 CLS
30 INPUT"DAME EL PRIMER TERMINO";A
40 INPUT"DAME EL SEGUNDO TERMINO";B
50 INPUT"DAME EL TERCER TERMINO";C
60 PRINT:PRINT
70 PRINT A;"ES A";B;"COMO";C;"ES A";C*B/A
```

Prueba a hacer por tu cuenta la Regla de tres inversa.

[Aplicación]

```
200 REM CONVERSION DE MONEDAS EXTRANJERAS
210 CLS 3
220 INPUT"NOMBRE Y VALOR DE LA MONEDA";A$,X
230 INPUT"CUANTAS QUIERES CAMBIAR";C
240 PRINT C;A$;"EQUIVALE A";C*X;"PTAS"
```

Prueba a hacer otro para convertir pesetas en monedas extranjeras.

Haz programas para calcular áreas, volúmenes, alturas, etc. de figuras geométricas. (La raíz cuadrada se puede hacer elevando un número a .5. Ejemplo raíz cuadrada de 25 = 25 ↑ .5).

Con **PRINT** e **INPUT** podrás hacer todos esos programas.

Y ahora con **CLS**, **INPUT**, **SOUND** tienes que hacer un programa en que el ordenador le pregunta a tus amigos su nombre, luego su edad, y su lugar de nacimiento. Cada dato debe estar bien situado en la pantalla, y las preguntas en negativo, y al final de programa se tiene que escuchar una musiquilla y el ordenador tiene que saludar a tus amigos ¡Suerte!

RESUMEN
DEL MODULO 8

INPUT detiene la ejecución del programa, y ésta no se reanuda hasta que se le introduce valor a la variable de **INPUT.**

REDO es el error que se produce cuando se asigna una cadena, mediante INPUT a una variable numérica.

Con **RUN N** (siendo N un número de línea) se puede empezar la ejecución del programa desde el número de línea que acompañe a RUN.

La RAIZ CUADRADA se puede hacer elevando a .5 cualquier número. Ejemplo A = 25 ↑ . 5 A = 5

EJERCICIOS DEL MODULO 8

1.- Cuál es incorrecta:

a) 100 RUN
b) 110 RUN 70
c) 300 RUN A .2

2.- Cuál es correcta:

a) 10 INPUT, N
b) 100 INPUT, N1
c) 40 INPUT B$

3.- Cuál es incorrecta:

a) 10 INPUT A, B, $
b) 10 INPUT A1, B1, B2
c) 10 INPUT A10

4.- REDO se produce cuando:

a) Se asigna una cadena a una variable numérica.
b) Se asigna un número a una variable numérica.
c) No se asigna valor a la variable.

5.- REDO se produce con:

a) LET
b) INPUT
c) PRINT

6.- Cuál borra de la línea 150 en adelante:

a) DEL 150
b) DEL 150–
c) DEL –150

7.- Cuál es incorrecta:

a) LISTA
b) LIST 150–
c) LIST 150

8.- Qué instrucción sirve para grabar un programa en el caset:

a) CLOADM
b) CSAVE
c) CLOAD

9.- RUN puesto en la última línea de un programa hace que éste:

a) Termine.
b) Se vuelva a ejecutar desde el principio.
c) Se autodestruya.

10.- Escribe todas las instrucciones que conoces de BASIC

SOLUCIONES:

B; B; A; B; B; A; A; C; C.

Pasatiempos Módulo 8

Solución al pasatiempo de la leccion 8.ª

La progresión sería de, (siempre en sentido de las agujas del reloj), tres cuartos de hora:

Media hora:

Un cuarto de hora:

y falta por supuesto
una hora completa:

Modulo 9

¿Qué es un programa?

¿Cómo que no lo sabes? Un ejemplo de programa es esto:

```
10 REM EJEMPLO DE PROGRAMA
20 CLS
30 INPUT "DAME UN NUMERO";A
40 INPUT "DAME OTRO NUMERO";B
50 PRINT
60 PRINTA;"MULTIPLICADO POR";B;"DA";A*B
70 END
```

PROGRAMA

Un programa es un conjunto de instrucciones y datos que se dan a un ordenador para que los ejecute. Los programas están formados por líneas. Las líneas empiezan por un número. Es conveniente numerar las líneas de 10 en 10 por si en medio hay que meter más líneas. No importa el orden en que vayas situando las líneas, porque el ordenador las ejecutará de menor a mayor.

En Basic las cosas se hace en orden.

Un programa es una relación detallada de instrucciones y datos que se da a un ordenador para que los ejecute. Pero esa relación detallada debe ser en el orden correcto para que el ordenador la entienda. Por eso, antes de hacer un programa, es conveniente que tomemos lápiz y papel para que aclaremos qué es lo que queremos hacer.

ALGORITMO

Un Algoritmo es un conjunto de instrucciones para hacer en orden una tarea completa.

Un programa es un conjunto de instrucciones y datos para hacer una tarea.

EJEMPLO PROGRAMA

Un programa se hace cuando es necesario, cuando hace falta. Para sumar 2 + 2 no hay que hacer un programa. En cambio, imagínate que tienes un Bar y quieres hacer un programa para calcular el precio de los resfrescos. Como tú sabes, los bares y las tiendas compran los artículos a un precio y los venden un poco más caros para ganarle algo. Imagínate que tienes un Bar y quieres hacer un programa para calcular el precio de los productos que vendes.

ALGORITMO

1.– Programa para calcular precios de venta.

Eso es lo primero que tienes que saber: ***¿Qué quieres hacer?*** En segundo lugar tienes que saber ***los datos*** con que cuentas.

2.– DATOS

2.1.– Precio de compra de los refrescos.

2.2.– Cantidad o % en que se incrementa cada artículo para venderlo.

Una vez que se conocen los datos hay que hacer ***las operaciones.***

3.– OPERACIONES.

3.1.– Calcular el tanto por ciento de incremento y sumárselo al precio de coste de los refrescos.

Por último hay que presentar ***los resultados*** de un modo bonito.

4.– Presentación de resultados.

5.– Fin.

¿Has visto? Los pasos a seguir:

1.– Saber qué vas a hacer.

2.– Introducir los datos.

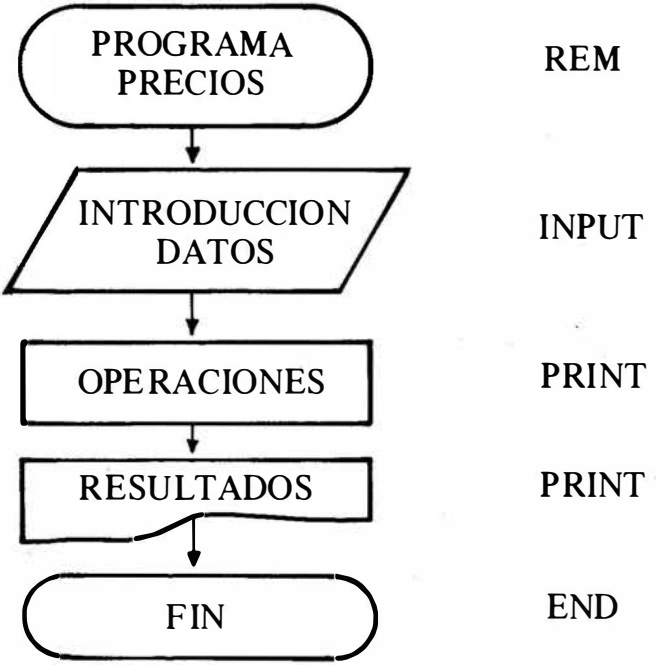
3.– Hacer las operaciones.

4.– Presentar los resultados.

Ahora basta con traducir ese algoritmo a BASIC.

```
10 REM "PROGRAMA PRECIOS"
     1- PROGRAMA PARA CALCULAR PRECIOS
20 INPUT"DIME PRECIO DE COSTE";A
     2- PRECIO DE COMPRA DE LOS REFRESCOS
30 INPUT"DIME % A INCREMENTAR";B
     3- CANTIDAD (%) DE INCREMENTO
40 REM "OPERACION CALCULO"
     4- OPERACIONES
        (PRECIO COSTE* % INCREMENTO)/100
50 LET C=A+A*B/100
     5- PRESENTACION DE RESULTADOS
60 PRINT"EL PRECIO DE VENTA ES";C
     6- FIN
70 END
```

Eso es un programa. El paso previo al programa es el algoritmo, y todavía hay otro paso previo al programa que es conveniente hacer: ***el organigrama.***



NO. Aún no sabes qué es un programa: un programa es un conjunto de instrucciones y datos puestos por ti en orden para que funcionen. **Un programa sirve para algo, y tiene que funcionar.**

* * *

Hoy vamos a ver dos instrucciones de gran importancia en ***BASIC:*** **GOTO** y **IF . . . . THEN**.

GOTO

GOTO ya la conoces bien. Cuando el ordenador se encuentra su **GOTO**, va a la línea que **GOTO** le señala, y sigue desde allí.

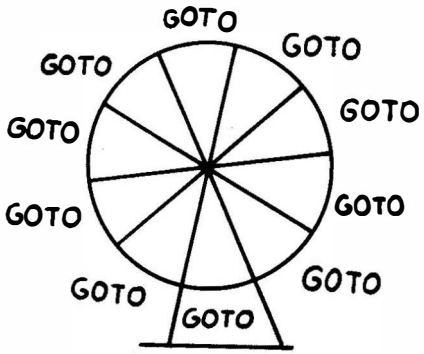
```
10 PRINT @200,CHR$(128);
20 CLS 2
30 GOTO10
```

Ejecuta la línea 10 sola.
Luego la línea 20 sola.
Luego las líneas 10 y 20.
Ahora ejecuta el programa entero.
GOTO aquí es una rueda sin fin.
Ejecuta este programa.

```
10 A=0
20 A=A+1
30 PRINT A
```

Luego ejecútalo poniéndole esta línea.

```
40 GOTO 20
```



GOTO aquí es una noria, una rueda sin fin. En el programa anterior el ordenador seguirá contando hasta que ya no le quepan más números en la pantalla. Tengo que hacerte la siguiente aclaración: Cuando el ordenador hace operaciones con números muy largos, emplea lo que se llama **NOTACION CIENTIFICA**. Teclea PRINT 50 ↑ 10. En la pantalla aparecerá 9.76562503E+16. Esto quiere decir, que el número es tan grande, que en vez de escribirlo entero, se escribe mediante una fórmula. Sería: 9.76562503 multiplicado por 10, elevado a 10.

La **Notación Científica** se utiliza cuando los números son tan grandes, que más que el número en sí, nos interesa su tamaño aproximado, para hacernos una idea.

* * *

Volviendo con el **GOTO**, siempre hemos dicho que el ordenador ejecuta las líneas empezando por la primera y terminando por la última, pues bien, **GOTO** interrumpe esa orden. **GOTO** es un SALTO, cuando el ordenador llega a un **GOTO**, el control del programa queda preso del **GOTO**, y no puede escapar, a no ser que reciba ayuda. El ordenador no puede escapar del círculo de **GOTO** a no ser que reciba ayuda, en este PROGRAMA.

Añádele al programa anterior estas líneas.

```
25 IF A=1ØØ THEN GOTO 5Ø
4Ø GOTO 2Ø
5Ø PRINT "ESCAPARNOS DEL BUCLE"
7Ø END
```

Ahora, ejecuta el programa.

¿Cómo hemos escapado del **GOTO**?

Con **IF. . . . THEN.** Esta instrucción quiere decir: SI se cumple la condición que yo quiero, ENTONCES haz lo que yo te diga.

Ya hemos escapado del **GOTO**

RENUM

Teclea ahora **RENUM. ENTER.**

Lista el programa. Con **RENUM** se renumeran las líneas de 10 en 10, con ello si hay líneas que no acaben en 0, todas quedan emparejadas de 10 en 10.

Bien ya hemos escapado de **GOTO.**

IF . . . . THEN

IF . . . . THEN sirve para hacer comparaciones.

Las comparaciones posibles son:

IF A >B THEN	mayor
IF A< B THEN	menor
IF A>=B THEN	mayor o igual
IF A<=B THEN	menor o igual
IF A = B THEN	igual
IF A<>B THEN	distinto

IF. . . . THEN es la instrucción fundamental de los juegos de aventuras, y de las más importantes del BASIC, porque permite al ordenador elegir entre varios caminos, dependiendo de si se cumple o no la condición. Analiza y ejecuta los siguientes ejemplos:

```
10 REM"EJEMPLO DE MAYOR QUE"
20 A=0
30 A=A+2
40 PRINT A
50 IF A>100 THEN PRINT "FIN":END
60 GOTO 30

10 REM"EJEMPLO DE MENOR QUE"
20 A=100
30 A=A-2
40 PRINT A
50 IF A<0 THEN PRINT"FIN":END
60 GOTO 30

5 REM "EJEMPLO DE MAYOR O IGUAL QUE"
10 A=0
20 A=A+2
30 PRINT A
40 IF A>=100 THEN PRINT"FIN":END
50 GOTO 20

10 REM"EJEMPLO DE MENOR O IGUAL QUE"
20 A=100
30 A=A-2
40 PRINT A
50 IF A<=0 THEN PRINT"FIN":END
60 GOTO 30

10 REM "EJEMPLO DE IGUAL QUE"
20 REM "ESTE PROGRAMA NO FUNCIONARA POR QUE
   (A) NUNCA LLEGA A VALER 100"
30 A=1
40 A=A+2
50 PRINT A
60 IF A=100 THEN PRINT"FIN":END
70 GOTO 40
```

```
10 REM"EJEMPLO DE DISTINTO DE"
20 INPUT"INTRODUCE EL NOMBRE DE UN AMIGO";A$
30 INPUT"INTRODUCE CUALQUIER NOMBRE";B$
40 IF A$<>B$ THEN PRINT"NO SE TRATA DE LA
   MISMA PERSONA":GOTO 20
50 PRINT "SON LA MISMA PERSONA":GOTO 20
```

El comando **IF. . . . THEN** se puede enriquecer me diante las siguientes ampliaciones.

IF . . . . THEN ELSE
IF AND THEN
IF OR THEN
IF NOT THEN

```
10 REM"EJEMPLO DE ELSE"
20 INPUT"INTRODUCE UN NUMERO";A
30 IF A>100 THEN PRINT A"ES MAYOR QUE 100"
   :GOTO 50
40 IF A<100 THEN PRINT A"ES MENOR QUE 100"
   ELSE PRINT A"ES IGUAL A 100"
50 END
```

```
10 REM"EJEMPLO DE AND"
20 A=0:B=0
30 A=A+1:B=B+2
40 PRINT A,B
50 IF A>100 AND B>100 THEN PRINT"FIN":END
60 GOTO 30
```

```
10 REM"EJEMPLO DE OR"
20 A=0:B=0
30 A=A+1:B=B+2
40 PRINT A,B
50 IF A>100 OR B>100 THEN PRINT"FIN":END
60 GOTO 30
```

```
10 REM"EJEMPLO DE NOT"
20 INPUT" DIME UN NUMERO";A
25 IF A=100 THEN PRINT A"ES IGUAL A 100":GOTO 100
30 IF A>100 THEN PRINT A"ES MAYOR DE 100":GOTO 120
40 IF NOT A>100 THEN PRINT A"ES MENOR QUE 100"
100 END
```

RESUMEN

Un **PROGRAMA** es un conjunto de instrucciones y datos, estructurados en líneas, y que sirven para ejecutar una tarea concreta, utilizando el lenguaje de programación Basic. (También se pueden hacer programas en otros idiomas de programación).

ALGORITMO es un conjunto de instrucciones para realizar de un modo ordenado una tarea concreta.

GOTO es la instrucción que nos manda a otra línea dentro del programa. GOTO es un bucle, una rueda sin fin. También se puede considerar un "salto" en el programa, pues transfiere el control a una línea, saltándose otras.

IF. . . . THEN es una instrucción que sirve para elegir entre más de una opción. Cuando el control del ordenador llega a IF. . . THEN se encuenta con que puede tomar uno entre varios caminos, dependiendo de que se cumpla, o no, la condición que acompaña a IF.

IF. . . . THEN funciona con los operadores lógicos, o signos de comparación:

= igual que	<> distinto que
< menor que	<= menos o igual que
> mayor que	<= mayor o igual que

IF. . . . THEN se puede enriquecer con **ELSE, AND, OR, NOT.**

EJERCICIOS
DEL MODULO 9

1) Qué línea es incorreta
   a) 100 GOTO 100
   b) 100 GOTO 10
   c) 100 GOTO B

2) Cuál es incorrecta
   a) 10 IF A = < THEN 60$
   b) 50 IF A$<>B$ THEN PRINT A$
   c) 70 IF A>= B THEN 20

3) GOTO es
   a) UN RETROCESO
   b) UN PASEO
   c) UN SALTO

4) GOTO manda el control del programa
   a) AL FINAL
   b) AL PRINCIPIO
   c) A UNA LINEA CUALQUIERA

5) IF. . . . THEN se puede completar con
   a) DEL 50
   b) ELSE
   c) END, OR, NOT

6) IF. . . . THEN es

a) UN SALTO

b) UN BUCLE

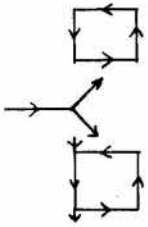
c) UN BIFURCACION

7) Cuál es IF. . . . THEN

a)

b)

c)



8) Cuál es incorrecta

a) IF A <> B THEN 10

b) IF A = B THEN 10

c) IF A = ( ) THEN 10

9) Cuál es incorrecta

a) IF A >< B THEN 10

b) IF A <> B THEN 10

c) IF A < = B THEN B

10) Un programa es

a) INSTRUCCIONES Y RESULTADOS.

b) DATOS Y OPERACIONES

c) CONJUNTO DE LINEAS

SOLUCIONES:

C; A; C; C; B; C; B; C; C; C;

Pasatiempos del Módulo 9

SOLUCION:

4	9	2
3	5	7
8	1	6

Modulo 10

En estos momentos, si has seguido el curso con aprovechamiento, estás en condiciones de hacer programas. Todavía no eres un malabarista del ***BASIC:*** te faltan muchos, muchos días de práctica.

La lección de hoy ha consistido en un repaso general de qué es un programa.

Un programa es un conjunto de instrucciones y datos.

Antes de hacer un programa hacemos un *ALGORITMO,* que es una lista de los pasos más lógicos para hacer cualquier tarea.

Antes de hacer un programa también es conveniente hacer un ***diagrama de flujo*** para ver en qué orden tiene el ordenador que ejecutar nuestras órdenes.

Y es que antes de hacer un programa, lo más importantes es saber:

1) Qué queremos hacer y para qué lo vamos a hacer
2) Con qué medios contamos
3) Cómo lo hacemos

Un programa no tiene sentido si no SIRVE PARA ALGO. Un programa se hace PARA ALGO.

Un programa se hace PARA resolver problemas, para jugar, para ayudarnos en el trabajo.

El ordenador no es una máquina maravillosa, que lo hace todo con solo pulsar una tecla.

NO. El ordenador es un instrumento, una herramienta, como un pico, como un martillo, como una cuchara, o como un bolígrafo. ***Un herramienta es algo que "sirve para"***. Tú harás los programas según tus necesidades: Primero tienes que saber para qué te va a servir el programa, y luego hacerlo de modo que te sirva, que te sea útil.

Tienes que saber tus necesidades para ver en qué te pueden ayudar el Ordenador y el ***BASIC***.

Bien:

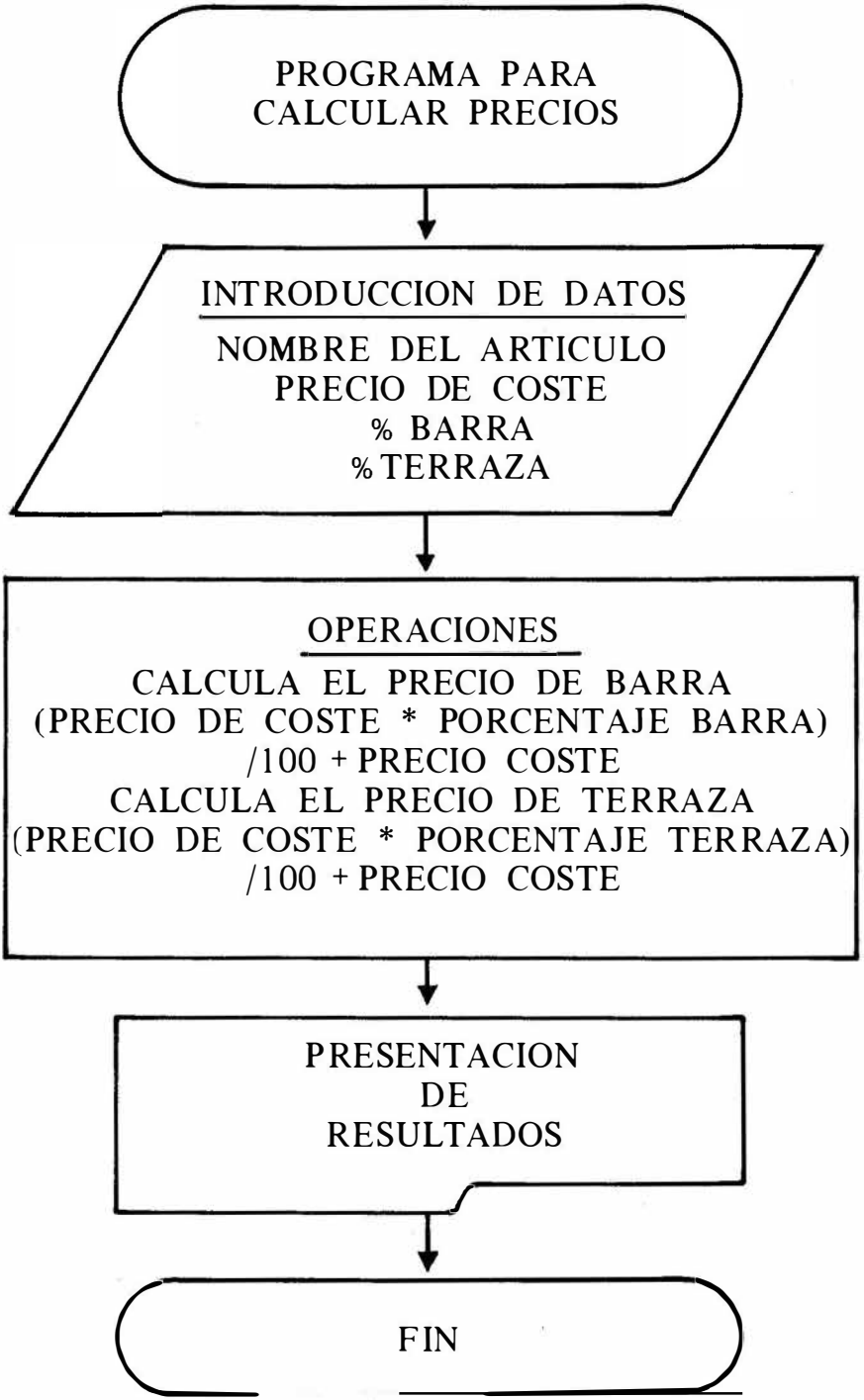
Recordarás que nuestro amigo Paco tiene un bar. Su problema es que vende muchos artículos: refrescos, zumos, bocadillos, etc., y cada vez que va a los almacenes a comprar artículos para venderlos en su bar, tiene que coger los artículos, uno a uno, y calcular a qué precio los tiene que vender. La operación es muy fácil, lo que ocurre es que tiene que hacerlo muchas veces, porque además, el precio de barra es distinto del de terraza.

Solución a su problema:

ALGORITMO

1) Programa para calcular precios
2) Introducir el nombre del artículo
3) Introducir el precio de coste
4) Introducir el porcentaje de incremento en Barra
5) Introducir el porcentaje de incremento en Terraza
6) Hacer las operaciones
7) Presentación de resultados
8) Fin

Estos son los pasos que hay que seguir para, sin ordenador, resolver el problema, la necesidad, de nuestro amigo Paco. Ahora vamos a hacer el programa en forma de ***organigrama***, o ***diagrama de flujo***.


PROGRAMA PARA
CALCULAR PRECIOS
INTRODUCCION DE DATOS
NOMBRE DEL ARTICULO
PRECIO DE COSTE
% BARRA
% TERRAZA
OPERACIONES
CALCULA EL PRECIO DE BARRA
(PRECIO DE COSTE * PORCENTAJE BARRA)
/100 + PRECIO COSTE
CALCULA EL PRECIO DE TERRAZA
(PRECIO DE COSTE * PORCENTAJE TERRAZA)
/100 + PRECIO COSTE
PRESENTACION
DE
RESULTADOS
FIN

Ahora vemos cómo sería en ***BASIC***

```
10 REM "PROGRAMA PARA CALCULAR PRECIOS"
20 CLS
30 INPUT"NOMBRE DEL ARTICULO";NA$
40 INPUT"DIME EL PRECIO DE COSTE";PC
50 INPUT"DIME EL % DE INCREMENTO EN BARRA";PB
60 INPUT"DIME EL % DE INCREMENTO EN TERRAZA";PT
70 CLS
80 PRINT NA$
90 PRINT"precio de barra=";PC+(PC*PB/100)
100 PRINT"precio de terraza=";PC+(PC*PT/100)
110 GOTO 30
```

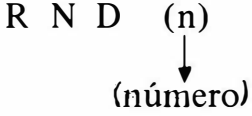
Una vez que se ha hecho un programa, lo más importante es ver si funciona. Adelante.

FUNCIONES

Hoy vamos a aprovechar que la lección ha sido fácil y corta, para explicarte tres funciones que vas a entender rápido, y que son de gran utilidad.

En ***BASIC*** al ordenador se dan las órdenes mediante ***instrucciones*** o ***comandos;*** pero a parte de comandos o instrucciones, hay otras muchas palabras que contienen instrucciones especiales: se trata de las ***FUNCIONES***.

Una ***función*** es una instrucción que realiza varias operaciones en una. Vamos a ver dos funciones y así lo comprenderás mejor.



Seguramente te habrás fijado en esta función a lo largo de las lecciones anteriores. ¿Qué hace **RND?** Pues, sencillamente, elige un número al azar entre 1 y el número que tú le digas. Ejecuta el siguiente programa.

```
10 LET A=RND(10)
20 PRINT A
```

El valor de A en la línea 20 es un número al azar, un número por casualidad, entre 0 y 10. Ejecútalo otra vez a ver qué número sale.

Ahora, en vez de 10 prueba con 1000.

¿Quieres que el ordenador te haga una quiniela?

```
10 LET C=0
20 LET A=RND(3)-1
25 IF A=0 THEN PRINT " X":GOTO 40
30 PRINT A
40 LET C=C+1
50 IF C=14 THEN 70
60 GOTO 20
70 END
```

En la línea 20 ponemos (3) -1 para que el ordenador nos de un número al azar entre 0 y 2: Es decir 0, 1, ó 2.

¿Quieres conocer otra función interesante?

INT (V)

↓

variable numérica

Esta función es muy importante para hacer cálculos matemáticos y juegos. Lo que hace **INT** es quitarle los decimales a un número.

```
10 A=7.3285
20 PRINT A
```

Ejecútalo. Y ahora con la siguiente modificación.

```
15 A=INT(A)
```

Como has podido comprobar, INT se limita a quitar los decimales a un número.

Haz pruebas.

Vamos a ver una tercera e interesante función ¿Sabes hacer raíces cuadradas con el ordenador? Con la FUNCION **SQR** (N). Puedes hacerlas.

SQR (N)

↓

número

```
10 A=SQR(928)
20 PRINT A
```

Otro modo de hacer raíces cuadradas es elevando el número en cuestión a 0'5.

```
10 A=928↑0.5
20 PRINT A
```

Bien.

RND
INT
SQR
→ FUNCIONES

CHR$
ASC (X$)
→ FUNCIONES

Vamos a ver otra función.

ASC (C)
↓
carácter

¿Recuerdas para qué servía **CHR$** (N) (N n. código ASCII)? **CHR$** convertía el número N en el carácter correspondiente del ***Código ASCII.*** Pues bien, ASC (C) hace lo contrario, convierte un carácter en el número correspondiente del ***Código ASCII.***

```
10 INPUT A$
20 PRINT "ASCII ";A$ ;ASC(A$)
30 GOTO 10
```

Por hoy está bien. . . de palabras. Vamos a practicar.

```
1Ø CLS
2Ø LET A=RND(Ø)
3Ø PRINT@3ØØ,A
4Ø GOTO 2Ø
```

```
1Ø A=RND(9)-1
2Ø CLS A
3Ø SOUND 1+(A*25),8
4Ø GOTO 1Ø
```

```
5 CLS
7 C=Ø
1Ø A=RND(51Ø)
3Ø PRINT@A,CHR$(128+RND(127));
35 SOUND RND(255),1
37 C=C+1
38 IF C>2ØØ THEN 5Ø
4Ø GOTO 1Ø
5Ø END
```

```
1Ø CLS
2Ø INPUT "INTRODUCE UN NUMERO:";N
3Ø INPUT "INTRODUCE OTRO NUMERO:";N2
4Ø PRINT N/N2
5Ø IF N/N2=INT(N/N2) THEN PRINT "LA DIVISION
   DE ESTOS DOS NUMEROSNO TIENE DECIMALES":END
6Ø PRINT"LA DIVISION DE ESTOS DOS NUMEROS
   TIENE DECIMALES":END
```

```
1Ø CLS
2Ø INPUT "INTRODUCE UN NUMERO:";NU
3Ø RC=SQR(NU)
4Ø PRINT"LA RAIZ CUADRADA ES: ";RC
5Ø INPUT"QUIERES SEGUIR [TECLEA SI O NO Y
   ENTER]";A$
6Ø IF A$="SI" THEN 1Ø
7Ø IF A$="NO" THEN END
8Ø GOTO 2Ø
```

```
1Ø CLS
2Ø FOR R=1 TO 255
3Ø PRINT@225,"DECIMAL";R;"CHR$(";R;")";CHR$(R)
4Ø FOR B=1 TO 3ØØ:NEXT B
5Ø NEXT R
6Ø END
```

* * *

RESUMEN
DEL MODULO 10

ALGORITMO es un conjunto de instrucciones para realizar de un modo ordenado una tarea concreta.

ORGANIGRAMA, junto con el DIAGRAMA DE FLUJO es la representación gráfica de los pasos a seguir para ejecutar una tarea.

FUNCION es un instrucción que realiza varias operaciones en una, facilitando de este modo el trabajo.

RND es la función que te da un número aleatorio, dentro de los márgenes que tú le des.

INT es la función que sirve para quitarle los decimales a un número.

SQR es una función que sirve para hacer raíces cuadradas.

ASC es un función que te da el valor, en el código ASCII de un carácter.

EJERCICIOS

1.– INPUT es un comando que. . .
   a) Multiplica variables
   b) Da valor a las variables numéricas
   c) Da valor a todas las variables

2.– El Comando **CSAVE**. . .
   a) Limpia la memoria ROM
   b) Limpia la memoria RAM
   c) Graba programas en cinta

3.– **CLOAD** lo usamos para. . .
   a) Guardar programas en caset
   b) Leer programas del caset
   c) Guardar programas en cinta

4.– **PRINT** es un . . .
   a) Comando doble
   b) Instrucción, o, comando
   c) Comando simple

5.– **PRINT** es una instrucción que. . .
   a) Imprime en pantalla en negativo
   b) Limpia las variables
   c) Imprime en pantalla y además nos posiciona

6.– Qué memoria no podemos borrar?

a) La interna o RAM
b) La ROM
c) La externa o ROM

7.– La unidad de disco es. . .

a) Una parte del ordenador
b) Memoria RAM
c) Un periférico

8.– La memoria externa es la . . . .

a) Unidad de disco
b) RUM
c) RAM

9.– Un Kbyte son. . .

a) 1024 Megabyte
b) 1000 BITS
c) 1024 BYTES

10.– Diez MEGABYTES son. . .

a) 1.000.000 K's
b) 10.000.000 BYTE
c) 10.000.000 KBYTE

SOLUCION:

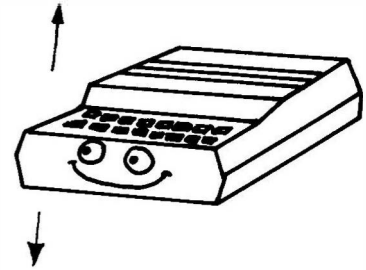
C; C; B; B; C; B; C; A; C; B.

Modulo 11

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. . .

ESTA CONTANDO

```
5 CLS
10 A=0
20 A=A+1
30 PRINT A
40 GOTO 20
```



ESTA CONTANDO

```
5 CLS
10 FOR A=0 TO 500
20 PRINT A
30 NEXT A
```

FOR . . . NEXT

Hoy vamos a ver la instrucción **FOR . . . NEXT**

1. FOR **V** = N TO N

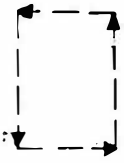
2. NEXT**V**

1. Donde aparezca Y cuenta desde N hasta N
2. Hazlo

Es una de las instrucciones más utilizadas en ***BASIC*** porque es muy útil

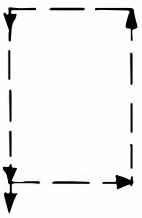
GOTO es una rueda sin fin. Un ***BUCLE*** sin fin.

```
10
20
30
40 GOTO 10
```



FOR. . . . NEXT es un ***bucle*** controlado, una rueda de la que se puede salir.

```
10
20 FOR. . .  TO. . .
30
40
50 NEXT
60
```



```
10 FOR A=1 TO 10
20 PRINT"EJEMPLO DE FOR...NEXT"
30 NEXT A
```

FOR. . . . NEXT cuenta desde donde-hasta donde tú le digas. **FOR . . . . NEXT** cuenta de uno en uno, pero también puede contar de 2 en 2, de 3 en 3, de 4 en 4, etc., añadiéndole **STEP.**

```
FOR . . .  TO. . . STEP. .
NEXT
```

```
5 CLS
10 FOR A=1 TO 100 STEP 2
20 PRINT @200,"EJEMPLO DEL STEP ";A
30 NEXT A
```

Quizás haya ido un poco rápido y no te haya dado tiempo de ver bien cómo el contador avanzaba de 2 en 2.

Vamos a remediarlo. Introduce esta línea en programa:

```
25 FOR T=1 TO 100:NEXT T
```

EJECUTALO.

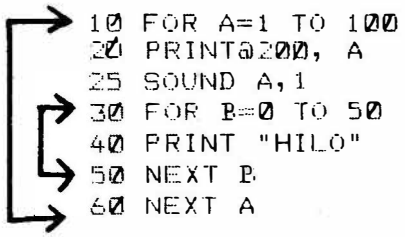
¿Lo ves ahora mejor? Ese Bucle **FOR. . . . NEXT** de la línea 25 se llama ***TEMPORIZADOR***, y sirve para que el programa en conjunto vaya más despacio. Es muy sencillo: se ejecuta la línea 10, la 20; al llegar a la 25 el ordenador cuenta de 1 a 20. Desde la línea 30 el ordenador manda el flujo de control a la línea 10, y así sucesivamente. ***El TEMPORIZADOR*** es un contador que cuenta en voz baja, y hasta que no termine de contar no permite que siga el programa. Cada vez que el flujo de control pasa por la línea 25, el ordenador cuenta, en voz baja, de 1 a 20.

Haz variaciones sobre ese pequeño programa.

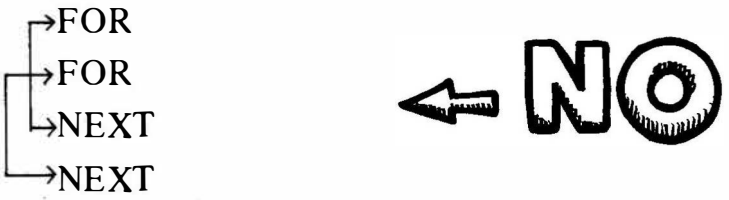
* * *

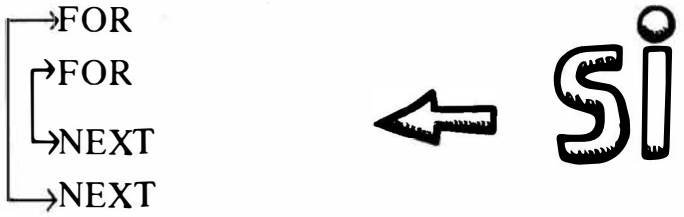
Tengo que hacerte notar que en un programa, como el anterior, pueden entrar varios **FOR. . . NEXT.**

Cuando en un mismo programa hay más de un bucle **FOR. . . NEXT,** hay que tener la precaución de que queden dispuestos del siguiente modo:



Es decir, un bucle debe quedar encajado dentro de otro: Los Bucles no se pueden cruzar entre sí.





Ejecuta el siguiente programa:

```
10 INPUT"TABLA DE MULTIPLICAR DEL";T
20 FOR M=0 TO 10
30 PRINT T;"*";M;"=";T*M
40 NEXT M
50 FOR D=1 TO 200:NEXT D
60 GOTO 10
```

Si quieres "escapar" de ese programa ya sabes que tienes que pulsar **BREAK**, o apagar el ordenador.

Copia el siguiente:

```
10 FOR A=1 TO 100
20 PRINT A
30 NEXT R
```

EJECUTALO.

Ese programa ha debido darte

NF ERROR?

¿Y qué quiere decir? Pues que el Bucle **FOR. . . NEXT** ha sido incorrectamente planteado: le sobra o le falta algo. Lista el programa. Efectivamente, has puesto **NEXT** R en vez de **NEXT** A. (No te preocupes, lo hemos hecho así par que te equivoques).

Ahora corrígelo.

¡Bien!

```
10 A$="ABRACADABRA"
20 FOR R=1 TO 11
30 FOR N=1 TO 200:NEXT N
40 PRINT MID$(A$,1,R)
50 NEXT R
```

Ha llegado el momento de las prácticas. Hasta luego.

```
10 PMODE4,1:PCLS:SCREEN1,0
20 A$="R2D2L2U2"
30 FOR R=10 TO 252 STEP 5
40 COLOR 1
50 DRAW"BM"+STR$(R)+",95;XA$;"
60 PCLS
80 NEXT R
90 GOTO 20
```

```
10 REM MUEVES CON LAS TECLAS DEL CURSOR
   (ARRIBA,ABAJO,IZQUIERDA,DERECHA)
20 CLS1
30 X=30:Y=10
35 B$=INKEY$
40 IF B$=CHR$(94) THEN Y=Y-1
50 IF B$=CHR$(10) THEN Y=Y+1
60 IF B$=CHR$(8) THEN X=X-1
70 IF B$=CHR$(9) THEN X=X+1
80 SET(X,Y,3)
90 GOTO 35
```

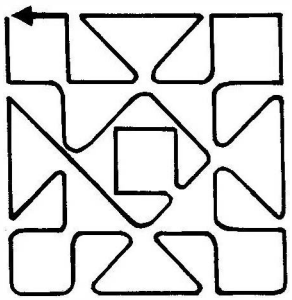
```
10 CLS
20 PRINT@0,"<<<<<<<LISTIN  TELEFONICO>>>>>>>";
30 PRINT@64,"TECLEA EL NOMBRE DE LA PERSONA
   DE LA QUE QUIERES CONOCER ELNUMERO
   DE TELEFONO Y PULSA       [ENTER]";
40 INPUT A$
50 IF A$="PEPE" THEN PRINT "TELEFONO:77.77.77"
60 IF A$="JUAN" THEN PRINT "TELEFONO:55.55.55"
1000 PRINT"OTRA CONSULTA S/N"
1020 IF INKEY$="S" THEN 10
1030 IF INKEY$="N" THEN CLS:END
1040 GOTO 1020
```

PARA UTILIZAR ESTE PROGRAMA CON TUS AUTENTICOS NUMEROS DE TELEFONO TIENES QUE INCLUIR ENTRE LA LINEA 50 Y LA 1000 TANTAS LINEAS COMO TELEFONOS QUIERAS CONSULTAR DE LA SIGUIENTE FORMA:

```
50 IF A$="NOMBRE"THEN PRINT"TELEFONO:NUMERO"
```

Pastiempo del Módulo 11

Solución al Laberinto:



RESUMEN
DEL MODULO 11

FOR. . . NEXT es una instrucción que sirve para contar. Es decir, se asigna a una variable un valor de inicio y otro de fin. Hasta que no se ejecuta ese valor, el control va de FOR a NEXT y viceversa. FOR NEXT es un bucle, una rueda, como GOTO, pero es un bucle controlado: tiene principio y fin. FOR NEXT es un contador.

STEP es un complemento de FOR NEXT. STEP determina el tamaño del paso con que FOR NEXT va a contar.

NF es el error que se puede producir cuando omitimos el FOR en la instrucción doble FOR NEXT.

EJERCICIOS

1.– Cuál es incorrecta
   a) FOR STEP NEXT
   b) FOR TO NEXT
   c) FOR TO STEP NEXT

2.– Qué error se relaciona con FOR NEXT
   a) SN ERROR
   b) RF ERROR
   c) NF ERROR

3.– Cuál es incorrecta
   a) NEXT
   b) NEXT 20
   c) NEXT A

4.– Un bucle sin fin es
   a) FOR NEXT
   b) GOTO
   c) FOR STEP

5.– FOR NEXT cuenta desde
   a) X hasta X
   b) 0 hasta 1.000
   c) 1 hasta 1.000

6.– Podemos formar un bucle controlado con
   a) PRINT E INPUT
   b) GOTO E IF THEN
   c) INPUT Y GOTO

7.– La variable del FOR NEXT tiene que ser
   a) De cadena
   b) Numérica
   c) Da igual

8.– Cuál es correcta
   a) FOR TO : FOR TO NEXT : NEXT
   b) FOR TO : FOR TO : NEXT : NEXT
   c) FOR TO : NEXT : NEXT : FOR TO

9.– Cuál es correcta
   a) FOR A = 0: NEXT A
   b) FOR A = O TO 10: NEXT
   c) FOR O TO 10: NEXT A

10.– FOR NEXT se puede utilizar como
   a) Posicionador en pantalla
   b) Temporizador
   c) INPUT con etiqueta

SOLUCIONES:

A; C; B; B; A; B; B; B; B; B;

Modulo 12

EL PROGRAMADOR

En este momento conoces las suficientes instrucciones de ***BASIC*** como para hacer programas. Los resultados que obtengas están en función de tus propios conocimientos, y del interés y esfuerzo que dediques cada día a conversar con tu micro. No debes medir la calidad de tus programas por la cantidad de líneas que tengan, ni por la complejidad que en ellos desarrolles. **El mejor programa es el que funciona.**

El mejor programa es aquel que te es útil, aquel que responde exactamente a tu necesidad, aquel que soluciona tu problema.

Copiar programas de un libro ayuda a aprender, pero el programador ***BASIC*** no es un mero copista de programas hechos por otra persona.

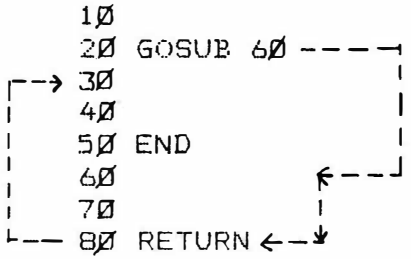
Un programador es una persona que partiendo de una necesidad y contando con unos datos y sus conocimientos, es capaz de confeccionar un conjunto de instrucciones que un ordenador puede entender y ejecutar.

Confeccionar un programa puede llevar horas, días, meses. . . pero no por ello debes rendirte ante ninguno. Antes de todo programa hay un paso previo: tu razonamiento. El ordenador más potente y más perfecto del mundo es tu cerebro. Todo lo que hace el ordenador puedes hacerlo tú, pero el ordenador sólo hará lo que tú seas capaz de enseñarle a hacer.

GOSUB. . . RETURN – SUBRUTINAS

Hoy vamos a ver una última instrucción, de gran importancia cuando se hacen programas largos, con muchas líneas: Se trata de **GOSUB. . . RETURN.**

GOSUB. . . RETURN quiere decir: "Ve a la línea que te mande **GOSUB**", y ejecutando el programa sigue desde allí. Cuando encuentres **RETURN**, retorna a la línea que hay después de la línea de **GOSUB.**



GOSUB transfiere el control del programa a una línea que hay después de **END**. Desde esta línea se sigue ejecutando el programa. Al encontrar **RETURN** el control vuelve a la línea que había después de **GOSUB**.

Y para qué sirve **GOSUB**; pues, para meter un programa secundario dentro de otro principal.

```
5 CLS
10 PRINT@100,"EJEMPLO"
20 PRINT@200,"DE SUBRUTINA"
30 PRINT@300,"CON GOSUB...RETURN"
40 PRINT@455,"LO HAS VISTO..."
50 STOP
```

PROGRAMA PRINCIPAL

GOSUB sirve para salir del programa principal. RETURN para retornar a él.

```
60 FOR T=0 TO 500
70 LET P=RND(255)
80 IF T=250 THEN SOUND P,8
90 NEXT T
```

PROGRAMA SECUNDARIO

GOSUB sirve para meter dentro de un programa principal uno secundario. Al programa secundario se le suele llamar ***SUBRUTINA***.

PROGRAMA PRINCIPAL	SUBRUTINA

Una *SUBRUTINA* es un programilla que se mete en otro programa más grande.

¿Has copiado el programa anterior desde la línea 5 a la 90? Ahora introdúcele las siguientes modificaciones: Con dos puntos (:) añade a las líneas 10, 20, 30 y 40 **GOSUB** 60. Y a la línea 90 añade: **RETURN.**

Si quieres ver cómo se van ejecutando las líneas, sabes que puedes hacerlo con **TRON.** Adelante.

Con **GOSUB. . . RETURN** se te puede producir un error.

? RG ERROR

Este error se produce cuando has planteado de forma incorrecta el puente **GOSUB. . . RETURN**: se te ha olvidado una de las instrucciones, o has invertido su orden, etc. No lo olvides, porque suele ocurrir con frecuencia.

* * *

INKEY$

Una instrucción, un función, que hemos dejado para el final, no porque no sea importante, sino por su relativa complejidad, es **INKEY$. INKEY$** quiere decir ***"hay alguna tecla pulsada?"*** o ***"cuál es la tecla que hay pulsada?".***

Es una ***función*** de gran utilidad, sobre todo para los juegos, pero también para otras muchas cosas.

El mejor modo de que veas como funciona, es verla funcionando. Copia y ejecuta el siguiente programa.

```
10 PRINT"QUIERES JUGAR DENUEVO?"
20 A$=INKEY$
30 IF A$="S" THEN RUN
40 IF A$="N" THEN END
50 GOTO20
```

INKEY$** es un detective, un sabueso, que rastrea el teclado buscando si hay alguna tecla pulsada. En el programa anterior, cada vez que tu has pulsado la letra "S", **INKEY$ se ha dado cuenta y ha cumplido la instrucción de la línea 30, es decir, "si la variable A$ es igual a la tecla que **INKEY$** ha rastreado, la "S", entonces empieza otra vez, **RUN**"

INKEY$ suele utilizarse con la instrucción doble **IF. . . . THEN**. Es fácil comprenderlo: SI SE CUMPLE **INKEY$** ENTONCES HAZ LO QUE SEA. Eso es lo que podemos observar en las líneas 30 y 40.

El modo de funcionamiento de **INKEY$**, la plantilla, para introducirlo en muchos programas sería.

```
→ VARIABLE$ = INKEY$
                    (TECLA)
→ IF VARIABLE$ = "T" THEN COMANDO

→GOTO L I
          (línea en que está INKEY$)
```

Veamos otro ejemplo. Ejecútalo.

```
1Ø A$=INKEY$
2Ø IF A$="Ø" THEN CLSØ
3Ø IF A$="1" THEN CLS1
4Ø IF A$="2" THEN CLS2
5Ø IF A$="3" THEN CLS3
6Ø IF A$="4" THEN CLS4
7Ø IF A$="5" THEN CLS5
8Ø IF A$="6" THEN CLS6
9Ø IF A$="7" THEN CLS7
1ØØ IF A$="8" THEN CLS8
11Ø GOTO1Ø
```

Ejecuta ese programa. Una vez que teclees **RUN**, pulsando los números del 0 al 8, la pantalla cambiará de color, con su número respectivo.

Bien, ahora te toca a ti. Vas a hacer un programa, que tendrá que ser largo (si lo haces corto mejor), para convertir el ordenador en un organillo. Cada letra y cada número tienen que poseer un sonido distinto. Te voy a dar una pista:

```
1Ø A$=INKEY$
2Ø IF A$="A" THEN SOUND 1,4
3Ø IF A$="B" THEN SOUND 5,4
4Ø......
1ØØØ GOTO 1Ø
```

```
1Ø A$=INKEY$
2Ø IF A$="" THEN GOTO 1Ø
3Ø A=ASC(A$):IF A=13 THEN GOTO 7Ø
4Ø GOSUB1ØØ
5Ø PRINTA$;
6Ø SOUND A*2,1:GOSUB1Ø
7Ø CLS:FORN=1 TO LEN(B$)
8Ø A=ASC(MID$(B$,N,1)):SOUND A*2,1
9Ø PRINTCHR$(A);:NEXTN:END
1ØØ X=X+1
11Ø IF X>99 THEN GOTO 13Ø
12Ø B$=B$+A$
13Ø RETURN
```

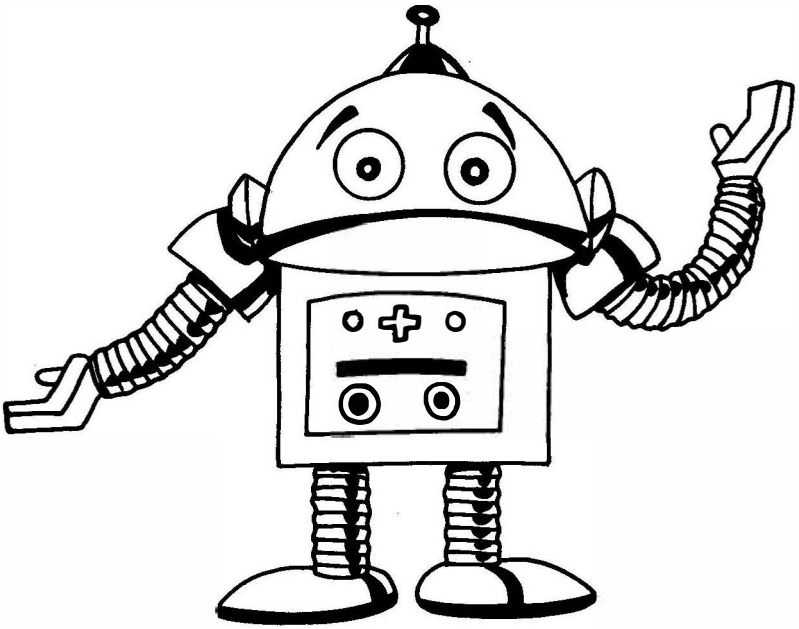
```
1Ø REM EL TECLEADOR MAS RAPIDO
2Ø CO$(1)="ERES UNA TORTUGA":CO$(2)="VAS
   MUY DESPACIO":CO$(3)="VAS MEJORANDO"
   CO $(4)="VALE"
3Ø CO$(5)="ERES RAPIDILLO":CO$(6)="ERES
   MUY RAPIDO":CO$(7)="ERES RAPIDISIMO":
   CO$(8)="ERES EL MAS RAPIDO":CO$(9)="...
   SOLO YO SOY MAS RAPIDO QUE TU"
4Ø CLS
5Ø PRINT" VEAMOS COMO ERES DE RAPIDO CON
   EL TECLADO."
6Ø PRINT
7Ø PRINT"   PULSE CUALQUIER TECLA"
8Ø PRINT
9Ø IF INKEY$="" THEN GOTO 9Ø
1ØØ TIMER=Ø
11Ø O$=CHR$(33+RND(57)):PRINTO$;
12Ø A$=INKEY$:IF A$=O$ THEN PO=PO+1:
   SOUND 2ØØ,1:GOTO 11Ø
13Ø IF TIMER<1ØØØ THEN GOTO 12Ø
14Ø PRINT
15Ø PRINT"EL RESULTADO ES:";PO:PO=INT(PO/3)
16Ø IF PO>9 THEN PO=9
17Ø PRINTCO$(PO):I$=INKEY$
18Ø FOR C=Ø TO 1ØØØ:NEXT C
19Ø PRINT"OTRA PARTIDA? PULSA S o N"
2ØØ I$=INKEY$:IF I$="" THEN GOTO 2ØØ
21Ø IF I$="S" THEN GOTO5Ø
22Ø END
```

```
<<<<<<PROGRAMA EJERCICIOS>>>>>>
================================


1Ø CLS:TIMER=Ø
2Ø PRINT@34,"EN QUE ORDEN SE REA
LIZAN LAS    SIGUIENTES OPERACIO
NES:"
3Ø PRINT@13Ø,"A)- PRIMERO LAS PO
TENCIAS"
4Ø PRINT@194,"B)- ANTES LAS SUMA
S"
5Ø PRINT@258,"C)- NO,PRIMERO SON
 LAS DIVISIONES"
6Ø GOSUB 15Ø
7Ø A$=INKEY$
8Ø IF TIMER/5Ø>4Ø THEN A$="C"
9Ø IF A$="A" THEN C=C+1:FOR T=1
TO 3ØØ:PRINT@13Ø,"X":PRINT@13Ø,"
A)- PRIMERO LAS POTENCIAS<<<<":N
EXT T: GOSUB 2ØØ:GOTO 13Ø
1ØØ IF A$="B" OR A$="C" THEN PRI
NT@352,"NO ES CORRECTO,DEBES SAB
ER QUE  PRIMERO SE REALIZAN LAS
POTEN-  CIAS":FOR T=1 TO 3ØØ:PRI
NT@13Ø,"X":PRINT@13Ø,"A)- PRIMER
O LAS POTENCIAS<<<<<":NEXT T:GOS
UB 3ØØ:GOTO 13Ø
11Ø GOTO 7Ø
13Ø IF TIMER/5Ø<45 THEN GOTO 13Ø
14Ø PRINT@32Ø,"SE TE ACABO EL TI
EMPO":END
15Ø PRINT@354,"PULSA LA TECLA QU
E CREAS         ACERTADA"
16Ø PRINT@32Ø,"----------------
--------------";
17Ø RETURN
2ØØ PRINT@354,"SE VE QUE APROVEC
HAS LAS         LECCIONES":RETURN
3ØØ PRINT@448,"HAS DE PRESTAR MA
S ATENCION A    LAS LECCIONES";:R
ETURN
```

TIMER: Se utiliza para mantener un control de tiempo sobre el ejercicio. La instrucción TIMER/50>X transforma el tiempo en segundos determinando la duración del ejercicio en los segundos dados por X.

Las subrutinas de las líneas 150, 200, 300 están realizadas para la posterior utilización de las mismas en caso de que se realizaran más ejercicios.



Bien. Hemos recorrido un largo camino. Has aprendido las instrucciones básicas de BASIC.

LET. INPUT. PRINT. GOTO. IF. . . THEN
FOR. . .NEXT. GOSUB. . . RETURN

Ten siempre presente que un ordenador es solo una máquina, una herramienta en tus manos. No olvides tampoco que para cualquier problema que tengas con tu aprendizaje, no tienes más que telefonear o escribir a

ENSEÑANZAS A. T. V.

Telf.: (952) 228179

Paseo de la Farola, núm. 25

29016 – MALAGA

Muchas gracias por tu compañía y por haber estudiado con nosotros. Esperemos que te haya divertido la experiencia, y te encontremos otra vez en el Curso de BASIC COMPLETO A.T.V. HASTA SIEMPRE.

RESUMEN

MODULO 12

EL MEJOR PROGRAMA es el que funciona.

PROGRAMADOR es la persona capaz de confeccionar un conjunto de instrucciones, que un ordenador puede entender y ejecutar.

GOSUB manda el control del programa a una Subrutina.

RETURN devuelve el control al programa principal.

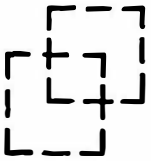
SUBRUTINA es un programa secundario que se mete dentro de otro, principal.

RG ERROR: se produce cuando se olvida situar el GOSUB, pero se pone el RETURN.

INKEY$: rastrea el teclado, buscando si hay alguna tecla pulsada. Cuando se da cuenta de que se pulsa una letra, actúa siguiendo tus instrucciones.

SOLUCION: Pasatiempos

1º. El resultado del movimiento en esta figura es:



2º. Luego los ordenadores no tienen imaginación, ni creatividad.

EJERCICIOS

1.– Un Bucle controlado es
   a) FOR NEXT
   b) GOTO
   c) GOSUB. . . RETURN

2.– Un Bucle sin fin es
   a) FOR NEXT
   b) GOTO
   c) GOSUB. . . RETURN

3.– Una subrutina es
   a) Un programa corto
   b) Un programa sin importancia
   c) Un programa dentro de otro

4.– Se produce RG ERROR
   a) Cuando falta GOSUB
   b) Cuando falta RETURN
   c) Cuando faltan GOSUB y RETURN

5.– INKEY$ es
   a) Un Comando
   b) Una función
   c) Una variable

6.– GOSUB. . . RETURN sirve para
   a) Hacer cadenas
   b) Utilizar las Subrutinas
   c) Rastrear el teclado

7.– INKEY$ suele funcionar con

a) PRINK @
b) SHIFT @
c) IF. . . THEN

8.– Cuál es incorrecta

a) A$ = INKEY$
b) B = $INKEY$
c) B1$ = INKEY$

9.– Cuál es correcta

a) GOSUB FOR RETURN
b) GOTO NEXT FOR
c) GOSUB. . . RETURN

10.– Cuál es correcta

a) IF. . . THEN. . . TO
b) IF. . . NOT . . . TO
c) IF. . . THEN. . . GOTO

SOLUCIONES:

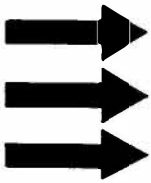
A; B; C; A; B; B; C; B; C; C;

1ª EVALUACION:
DESPUES DEL MODULO 5

2ª EVALUACION:
DESPUES DEL MODULO 10

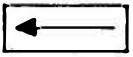
3ª EVALUACION:
DESPUES DEL MODULO 12

EVALUACIONES



1ª EVALUACION
INFORMATICA

1º Cuál no es tecla de control?

- ☐ a) BREAK
- ☐ b) SHIFT
- ☐ c) 
- ☐ d) ENTER
- ☐ e) 

2º Una instrucción o Comando ocupa

- ☐ a) Un BIT
- ☐ b) Un BYTE
- ☐ c) 32 BYTES
- ☐ d) 64 BYTES
- ☐ e) 12 BYTES

3º Qué Comando utilizamos para limpiar la Pantalla?

- ☐ a) CLEAR
- ☐ b) BREAK
- ☐ c) CLS
- ☐ d) NEW
- ☐ e) DEL–

4º Qué significa B.A.S.I.C.?

5º El lenguaje máquina opera en:

- ☐ a) Sistema Decimal
- ☐ b) Hexadecimal
- ☐ c) Binario
- ☐ d) ASCII
- ☐ e) BASIC

6º La instrucción para ver el programa que hay en la memoria es:

- ☐ a) RUN
- ☐ b) ENTER
- ☐ c) CLEAR
- ☐ d) LIST
- ☐ e) MEM

7º Qué Comando se utiliza para modificar programas

- ☐ a) COREG
- ☐ b) NEW
- ☐ c) RESET
- ☐ d) SOUND
- ☐ e) EDIT

8.º Cuántos BITS tiene un KBYTE?

- ☐ a) 1024
- ☐ b) 8192
- ☐ c) 1000
- ☐ d) 1000000
- ☐ e) 80000

9.º Binario es a dos como Hexadecimal es. . .

- ☐ a) Diez
- ☐ b) Seis
- ☐ c) Doce
- ☐ d) Dieciséis
- ☐ c) Ocho

10.º Por que solemos numerar las líneas de los programas de diez en diez

Rellena este cuestionario, recórtalo del libro y remítelo a:

ENSEÑANZAS A.T.V.

(Sección Informática Autoestudio)
Paseo de la Farola, núm. 25
29016 – MALAGA

SEGUNDA EVALUACION. BASIC

Amigo alumno, has superado con aprovechamiento las diez primeras lecciones de nuestro curso de introducción al BASIC. En estos momentos conoces varias instrucciones fundamentales, que te capacitan para la confección de programas, cuya complejidad se hallará en función de tus propios recursos: tu interés, tu curiosidad, tus conocimientos, tu edad.

Para superar esta evaluación, debes confeccionar un programa respetando las siguientes instrucciones:

1.– El programa, en ningún caso, puede tener más de 30 líneas.

2.– El programa, en ningún caso, puede contener más de 60 instrucciones.

3.– Sólo puedes utilizar las siguientes instrucciones: Cload, Run, New, List, Cls, Sound, Rem, Tron-Troff, Edit, Print, Print @ , Chr$, Asc, Input, Goto, If Then (Else, And, Or, Not), Rnd, Int, Sqr.

Naturalmente, también puedes emplear los operadores lógicos, los aritméticos y los signos de puntuación, y claro, las variables.

4.– No podrás utilizar otras instrucciones que las contenidas en la lista anterior, si bien sí podrás repetirlas, o utilizar sólo algunas de ellas.

5.– Dependiendo de tu edad, tendrás que resolver uno de los tres programas de uno de los siguientes grupos:

GRUPO 1.º : Alumnos de hasta 12 años de edad.
GRUPO 2.º : Alumnos de 13 a 16 años de edad.
GRUPO 3.º : Alumnos de 17 años de edad en adelante.

Para cada grupo de edad proponemos tres supuestos a resolver. De entre esos tres supuestos, **sólo tendrás que resolver y remitirnos uno,** que será el que evaluaremos en ATV. No obstante, podrás resolver y remitirnos el supuesto de otro grupo que no sea el tuyo, pero deberás hacerlo constar en el ejercicio.

6.– Conjuntamente con el programa, debes remitirnos un **algoritmo** del mismo, y un **breve comentario aclaratorio,** si lo crees necesario (ambos mecanografiados al igual que el programa).

7.– Por último, hacerte una aclaración: No es obligatorio que realices esta evaluación, y que la remitas a ATV, pero, si realmente estás haciendo este curso con interés, y eso lo damos por descontado, no eludas tu propia responsabilidad. (Por otro lado, sólo realizando las tres evaluaciones con aptitud, podrás obtener el Diploma, autorizado por el Ministerio de Educación y Ciencia, que al final de tu curso expide ATV).

Bien, basta de explicaciones. Lee detenidamente todos los supuestos, y elige de entre todos el que estimes adecuado para tu evaluación.

8.– Grupo 1º (Alumnos de 10 a 12 años de edad)

NIVEL 1

1) Hacer que el ordenador cuente desde 1 hasta 1000.
2) Este contador debe quedar en el centro de la pantalla, y en un color distinto al del fondo de la pantalla.
3) Cuando el ordenador haya contado hasta 100 la pantalla debe cambiar de color.
4) Cuando el ordenador haya contado hasta 200, deben aparecer ocho cuadraditos, cada uno de un color, y al mismo tiempo que van apareciendo se debe escuchar un pitido, de tres segundos de duración, con cada uno de ellos.
5) Al final del todo, debe aparecer, en el centro de la pantalla tu nombre y apellidos; debajo tu edad; debajo tu grupo y nivel.

NIVEL 2

1) El ordenador tiene que pedirte un número.
2) El ordenador contará hasta el número que le has dado, y luego se detendrá.
3) Cuando el ordenador acabe de contar debe aparecer en la pantalla algún dibujo de colores, al mismo tiempo que, durante unos segundos, suena una cortita composición musical.
4) Al final debe aparecer, del modo que tú creas más conveniente, tu nombre y apellidos, edad, grupo y nivel.

NIVEL 3

Debes hacer un programa en el que utilices, al menos, las siguientes instrucciones: PRINT, LET, INPUT, IF. . . . THEN, GOTO, CLS, SOUND, CHR$, RND, Este programa, en función de su resultado y vistosidad, obtendrá más puntuación que los otros dos del mismo grupo.

Grupo 2º (Alumnos de 13 a 16 años)

NIVEL 1

1) El ordenador te tiene que pedir un número hasta el cual va él a contar.
2) Luego te tiene que preguntar de cuánto en cuánto quieres contar.
3) El ordenador contará hasta el número indicado, y cuando llegue a él, contará hacia atrás, hasta quedar a cero, donde parará.
4) Al final tienes que poner tu nombre y apellidos, tu edad, tu grupo y tu nivel, todos en un color distinto al del fondo, y en línea separadas, y bien enmarcadas.

NIVEL 2

1) Tienes que hacer un programa para calcular el área de cualquier círculo, rectángulo, cuadrado o triángulo, dependiendo de lo que el usuario elija.
2) Ten en cuenta que este programa tiene que servir para que lo utilice cualquier persona, a la que el ordenador tendrá que pedirle todos los datos para solucionarle el problema.

NIVEL 3

Utilizando todas las instrucciones que se te han explicado hasta la lección 10, haz un juego, el que tú quieras.

Grupo 3º (Alumnos de 17 años en adelante)

NIVEL 1

Dibuja un coche que recorra la pantalla de izquierda a derecha. Te bastará con utilizar LET, PRINT @ , IF THEN, y GOTO.

NIVEL 2

Haz un programa que calcule los números primos.

NIVEL 3

Demuestra que has superado los niveles anteriores. Haz el programa que quieras. . . ¡sorpréndenos!

9.– Valoramos en la evaluación, en el programa que nos remitas,

1. Que funcione
2. Que se adapte a lo que te pedimos
3. Originalidad
4. Imaginación

10.– Una vez que hayas terminado el programa, que lo hayas probado, lo grabasen la cinta de prueba, y conjuntamente con el programa de la evaluación final, nos lo remites a

ENSEÑANZAS ATV
(Sección Informática Autoestudio)
Paseo de la Farola, núm. 25
29016 – MALAGA

11.– También tienes que remitirnos los programas (de esta evaluación y de la siguiente), en las hojas que al efecto te adjuntamos al final del libro.

SUERTE

TERCERA EVALUACION. BASIC

Grupo 1º

Código secreto. Tienes que hacer un programa para que al pulsar cualquier tecla, se imprima otra distinta, es decir, un código secreto que sólo tú conocerás. Este código debe activarse al teclear tú una palabra clave, y se desactivará al volverla a pulsar.

Grupo 2.º

El ordenador te pedirá un número; una vez que se lo hayas dado, debe decirte todas las combinaciones de tres números que den como resultado el anterior. Por ejemplo, el ordenador te da el número seis. Las combinaciones posibles de tres números que sumen 6 son:

0 0 6
0 1 5
0 2 4
0 3 3
etc ....

Grupo 3º

Tienes que hacer un frontón. Es decir: una raqueta que se mueva en sentido vertical, al lado izquierdo de la pantalla; una pelota, aunque sea cuadrada, que vaya hacia la pared de la derecha rebote en ella, y vuelva hacia la izquierda. Sencillamente, un juego.

En esta evaluación puedes utilizar todas las instrucciones que hemos explicado en el curso.

Evaluación: .....
Grupo: .....
Nivel: .....
Número: .....

Nombre: .....
1.er Apellido: .....
2.º Apellido: .....
Edad: .....
Dirección: .....
.....
Ciudad: .....

LISTADO DEL PROGRAMA

Evaluación: .....
Grupo: .....
Nivel: .....
Número: .....

Nombre: .....
1.er Apellido: .....
2.º Apellido: .....
Edad: .....
Dirección: .....
.....
Ciudad: .....

LISTADO DEL PROGRAMA

Sólo con tus manos. INFORMATICA-BASIC ATV pone a tu alcance hacer un viaje a través del maravilloso universo de la Informática y el BASIC. Tu ordenador es algo más que un juguete: es una herramienta; la Informática es una poderosa herramienta que tú puedes aprender a manejar con nosotros. No queremos enseñarte a matar marcianitos. . . vamos a enseñarte a programar.

¿Quieres aprender INFORMATICA–BASIC con nosotros?