

amaestra tu dragón



 **DRAGON**

amaestra tu dragón



 **DRAGON**

Editado por EUROHARD, S. A.

Prohibida la reproducción, grabación
o transmisión por cualquier medio
de esta publicación,
sin permiso del editor.

Diseñado por Diréctica, S. A.
Teléfono 234 94 36

Impreso por Artes Gráficas GALA, S. L.
Miguel Yuste, 36 - 28037 Madrid

Depósito Legal M. 14.196-1985

Indice

— Introducción	5
— Entra en casa del dragón. (PRINT, END, RUN, LIST.) ..	11
— Dialogas con el dragón. (NEW, CLS, INPUT, GOTO.) ..	17
— El dragón impone condiciones. (IF... THEN...)	31
— El dragón está algo mareado. (FOR... NEXT.)	39
— El dragón come datos. (DATA, READ, RESTORE.)	45
— El dragón hace los recados. (GOSUB... RETURN.)	51
— ¡Has amaestrado tu dragón! (Ejemplos.)	57
— Apéndice. Manejo del Cassette.	61
— Epílogo	64

introducción

El ordenador está en todas partes, ¡incluso en tu casa!, pero es importante saber para qué sirve, cómo funciona y cómo se usa.

Hay muchos tipos de ordenadores: grandes, pequeños, medianos, etc. Tú tienes un ordenador pequeño y por eso es un MICROORDENADOR.

Sirve para muchas cosas y algunas muy útiles: como llevar la contabilidad de tu casa o de una oficina, manejar el listín telefónico, estudiar matemáticas, física..., para conectar juegos y pasarlo fenomenal. Pero por encima de estas cosas hay una fundamental: sirve para todo aquello que tú seas capaz de enseñarle.



Claro está, que para que tú puedas enseñarle, primero tendrás que descubrir algunas cosas con mi ayuda y ha llegado la hora de que me presente:

yo soy
Dragon

Ya me conoces. Soy DRAGON, vivo dentro de un ordenador y lo conozco muy bien.

¡Entra dentro conmigo!

Imagínate que tu ordenador es un edificio de oficinas. Lee la siguiente historieta y verás cómo no es magia lo que hace tu DRAGON, sino fruto de un trabajo en equipo de sus distintas partes.



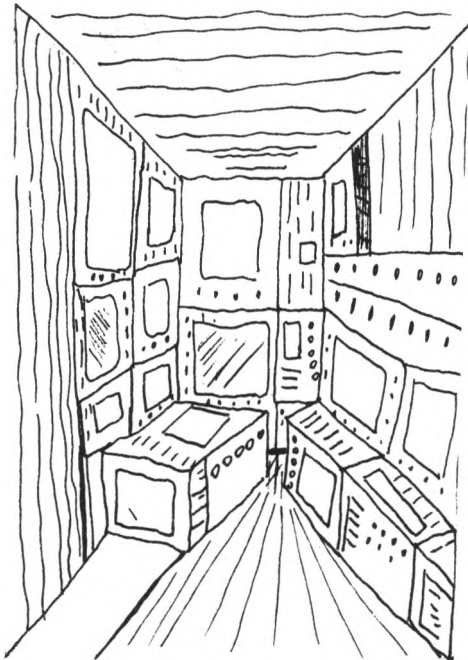
Este Sr. que entra por la puerta se llama MR. BASIC y lleva consigo un montón de cartas ordenadas que han de ser leídas y contestadas.



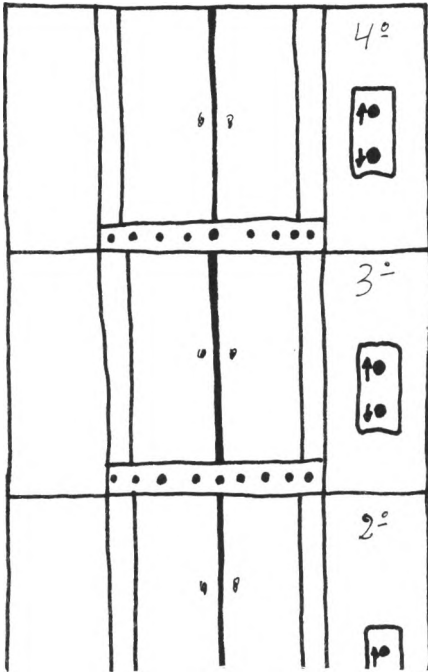
MR. BASIC entra en la gran oficina DRAGON y espera a ser llamado por el Director D. UNIDAD DE CONTROL.



El Director le ordena pasar e identifica todas las cartas para dar la orden de ejecución.



Como ves, la sala donde se hace lo que las cartas piden, tiene mucha maquinaria.



En los pisos se encuentran todos los datos necesarios para escribir las cartas; son como archivos.



Finalmente MR. BASIC sale contento con sus cartas contestadas y sus problemas resueltos.

En un ordenador, entonces podemos distinguir tres partes fundamentales:

- * La U.C.P. (Unidad Central de Proceso), que es la UNIDAD CONTROL y la UNIDAD ARITMETICO-LOGICA.
- * LA MEMORIA INTERNA.
- * LOS DISPOSITIVOS DE ENTRADA/SALIDA.

La UNIDAD CONTROL, que en nuestro edificio es el Director, rige todo lo que ocurre dentro del ordenador y la UNIDAD ARITMETICO-LOGICA, en nuestro edificio, la sala de máquinas, es la encargada de realizar las operaciones.

La MEMORIA, que son las distintas habitaciones u oficinas de los pisos, es la parte del ordenador donde se guardan todos los datos y órdenes que se necesitan.

Los dispositivos ENTRADA/SALIDA, la puerta del edificio, son los que nos permiten comunicarnos con el ordenador y que él se comunique con nosotros, como tu televisión (salida) o el teclado (entrada).

Bueno, ya has visto que el ordenador es como cualquier oficina de trabajo, pero es muy especial, cambia continuamente de oficio, dependiendo de las órdenes que tú le des.

Las órdenes deben ser muy detalladas y siempre deberás ir diciendo a tu MICROORDENADOR, paso por paso, lo que tiene que hacer. El conjunto de órdenes que le das para realizar algo se llama PROGRAMA.

¡Se me olvidaba! Tenemos un gran problema, el DRAGON habla un lenguaje extrañísimo «El Dragonés» y no hay quien lo entienda.

Por suerte, yo tengo un amigo que nos puede ayudar a comunicarnos con él, se llama BASIC. También habla un idioma especial pero es muy parecido al inglés y te lo voy a enseñar rápidamente.

El BASIC no es el único lenguaje que se dedica a transmitir fácilmente órdenes al ordenador. Existen otros como el COBOL, el FORTRAN, etc., se llaman lenguajes de programación y son los traductores entre un lenguaje parecido al nuestro y el suyo: «El Dragonés».

Ahora continúa leyendo, porque te voy a explicar el BASIC y cómo utilizarlo para que el ordenador te obedezca.

entra en la casa del dragón

aprende a escribir

Vamos a empezar a manejar nuestro DRAGON.

Primero aprendamos un poquito de vocabulario inglés: solamente cuatro palabras.

ESCRIBIR se dice PRINT, FIN se dice END, CORRER se dice RUN y por último en inglés LISTA se dice LIST.

Estas cuatro palabras, las entiende perfectamente tu DRAGON y si no te lo crees, escribe cualquier de ellas y pulsa la tecla (ENTER) que es la señal para decirle ¡Ahora obedece!

De todas formas, ninguna de ellas funciona como tú te imaginas. Pero no te preocupes, porque son muy sencillas.

Como comprenderás, no tiene sentido que le mandes escribir al DRAGON, si no le dices qué tiene que escribir. Entonces, cuando utilices PRINT, lo harás así: PRINT y a continuación, lo que quieras que escriba. Haz algunas pruebas.

PRINT 2

PRINT 4

PRINT HOLA

¿Qué dices? ¿Qué has escrito PRINT HOLA y él ha escrito Ø?

Pues bien, cuando quieras escribir palabras o cualquier cosa que no sea un número, ponlo entre comillas, esto es, teclea:

```
PRINT "HOLA"
```

De todas formas, escribir solamente números o frases es bastante poco. Hasta una calculadora es capaz de hacer más cosas.

Pero, ¿funcionará si a continuación de PRINT escribimos una operación matemática?

Vamos a probar:

```
PRINT 2+3
```

¡Ha respondido 5! ¡FUNCIONA!

Además, puede hacer otras operaciones: SUMAR (+), Restar (−), MULTIPLICAR (*), DIVIDIR (/) y ELEVAR A POTENCIAS (↑). También puede hacer cualquier combinación entre ellas y hasta poner paréntesis, igual que haces en el colegio.

Y ahora ya puedes empezar a hacer cosas, ¡oblígale a tu DRAGON a que te haga los deberes!

```
PRINT 18/6
```

```
PRINT 8 + 7 * 3
```

```
PRINT (5 + 3) / 2
```

```
PRINT 475 * 25
```

Como verás, el DRAGON es enormemente listo y tarda muy poco en responder tus preguntas, pero, sin embargo, nunca debes olvidar esto: la única ventaja que tiene el ordenador so-

bre tí es que es muchísimo más rápido que tú pero nunca será capaz de pensar por sí mismo.

Si quieres escribir bien, debes dominar los signos de puntuación y la mejor forma es practicar.

Nuestro amigo BASIC, que es un poco analfabeto, sólo conoce dos signos de puntuación, la COMA «,» y el PUNTO Y COMA «;».

Los signos de puntuación sirven para separar cada una de las cosas que quieres escribir y con ellos podrás imprimir varias cosas a la vez aunque unas sean números y otras frases.

La COMA, la vamos a utilizar cuando queramos que las cosas nos aparezcan en columnas, mira este ejemplo:

```
PRINT "DOS MAS DOS SON:" , 4  
PRINT "TRES MAS UNO SON:" , 4
```

Si utilizamos en lugar de la COMA, el PUNTO Y COMA el resultado saldrá a continuación de lo último que hemos escrito. Observa la diferencia escribiendo:

```
PRINT "YO ME LLAMO" , "DRAGON"  
PRINT "YO ME LLAMO" ; "DRAGON"
```

nuestro primer programa

Hasta ahora, sólo hemos introducido órdenes sueltas que si queremos repetir, tendremos que volverlas a escribir.

Pero yo sé una forma para que no tengas que reescribir todo; HACER UN PROGRAMA.

Un Programa, como ya sabes, es el conjunto de órdenes que hay que darle al ordenador, para que realice lo que queremos.

Pero este conjunto, para que el DRAGON lo entienda, debe estar en orden y la mejor forma de hacerlo es numerando las órdenes y es lo que vamos a hacer. De esta forma, el ordenador sabrá cuándo tiene que realizar cada cosa. Además, esta va a ser la diferencia entre un ORDEN DIRECTA (la que se ejecuta al momento) y una ORDEN QUE FORMA PARTE DEL PROGRAMA. Si delante de la orden hay un número, es que forma parte de un programa y si no, es una orden directa.

Haz un programa que escriba tu dirección completa. Para ayudarte voy a enseñarte cómo lo haría yo:

```
10 PRINT "SEÑOR DRAGON"  
20 PRINT "C / U C P"  
30 PRINT "MICROORDENADOR DRAGON"  
40 PRINT "CACERES"  
50 END.
```

¿Sabes por qué número las líneas de 10 en 10?

Muy fácil, porque soy muy despistado y a veces se me olvida alguna orden y si las numero 1, 2, 3, ... no puede intercalarla. Sin embargo, de 10 en 10 puedo ponerla como la línea 15, 17 ó 25 o donde quiera que esté.

Te aconsejo que tú también lo hagas.

La línea 50 es la más fácil de entender: la Orden END, que significa FIN, lo único que hace es informar al ordenador que su trabajo ha terminado, que no hay más órdenes para ejecutar.

Si ya has introducido el Programa, tienes que ordenarle que lo ejecute, esto es, que realice lo que le has escrito y en el orden que le has dado. Se hace tecleando RUN,, que significa "A CORRER EL PROGRAMA", "A EJECUTAR TODAS LAS ORDENES" ¡PRUEBALO!

Para acabar, sólo me queda una cosa por contarte. Cuando quieras ver el Programa que has escrito hay una Orden para que aparezca en pantalla, esta es LIST que significa «LISTame el Programa, ahora mismo».

Por último, prueba los siguientes Programas. Espero que a partir de ellos podrás imaginar infinidad de variantes y seguro que más originales.

```
1 10 PRINT "QUERIDOS REYES MAGOS"  
20 PRINT "QUIERO"  
30 PRINT " UN COCHE", "UNA MOTO"  
40 PRINT " UN RELOJ", "UNA PELOTA"  
50 PRINT  
60 PRINT "HE SIDO BUENISIMO," ; "BESOS A LOS TRES"  
70 PRINT " JUAN"  
80 END
```

```
2 10 PRINT "                x x x x x x"  
20 PRINT "      x                x x x x x x"  
30 PRINT "x x x x x      x x      "  
40 PRINT "      x x      x x x x x x x "  
50 PRINT "x x x  x  x  x x x x      "  
60 PRINT "  x      x x  x x x x x      "  
70 PRINT "  x      x      x x x      "  
80 PRINT "x      x x      x x      "  
90 PRINT "x  x x x x x x x x x x x x "  
100 PRINT "x  x x x x x x x x x x x x "  
110 PRINT "x                x x x x"  
120 PRINT "  x x x x x x x x x x x      x x"  
130 PRINT "    x x x x x x x x x x x x x "  
140 PRINT "      x  x                x  x "  
150 PRINT "    x  x                x  x  x"  
160 END
```



has aprendido

1. Con la Orden PRINT puedes escribir lo que quieras en la pantalla. También se puede utilizar para cálculos aritméticos $+$, $-$, $*$, $/$, \uparrow .
2. Las frases que tiene que escribir el ordenador irán entre comillas.
3. Al pulsar la tecla ENTER se ejecuta la Orden que hayas escrito.
4. Cuando el ordenador ha de ejecutar un Programa, le diremos RUN.
5. Para ver el Programa que hemos escrito, escribiremos LIST.
6. El final de un Programa es END.
7. Las líneas de un Programa te aconsejo que las enumeres de 10 en 10. ¡Que no se te olvide después de introducir cada línea de Programa pulsar ENTER!

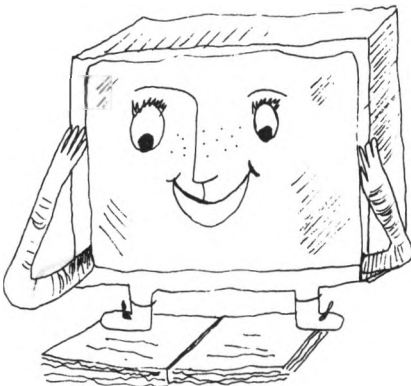
dialogas con el dragón

¡Hola! Me imagino que ya manejas perfectamente la instrucción PRINT y que habrás hecho tus propios ejemplos. Ahora avanzaremos otro poco.

Como vamos a empezar a escribir nuevos Programas, antes debes ordenar al DRAGON que se olvide de lo que tenga para que no se mezcle con lo «NUEVO» que le vamos a dar. Se hace con la Orden NEW que significa «NUEVO».

En la pantalla hay cosas escritas que no influyen en el nuevo Programa, pero que nos molestan. Para borrar la pantalla se utiliza la Orden CLS que es la abreviatura en inglés de «CLEAR SCREEN», «BORRAR LA PANTALLA».

Recuerda que CLS sólo borra la pantalla. No borra ni tus datos ni tus Programas, esto lo hace la Orden NEW.



la capital de Francia, es Paris; la capital de Italia, Roma...; que fácil!

Una ventaja fundamental de los ordenadores es que tienen MEMORIA, es decir, son capaces de recordar los datos y los Programas que tú le hayas dado. ¡Hasta tu libro de Geografía! y mucho, mucho, mucho más.

Ya sabemos cómo escribir Programas pero nos falta descubrir cómo introducir los datos en tu ordenador.

El ORDENADOR tiene unos núcleos de ferrita que son unos elementos biestables y que...

— ¡Oye que no entiendo nada!

— Tienes razón, vamos a consultar a nuestro BASIC, a ver qué nos cuenta.



DRAGON: BASIC, ¿Cómo podemos meter los datos en el Ordenador, para que no se le olviden?

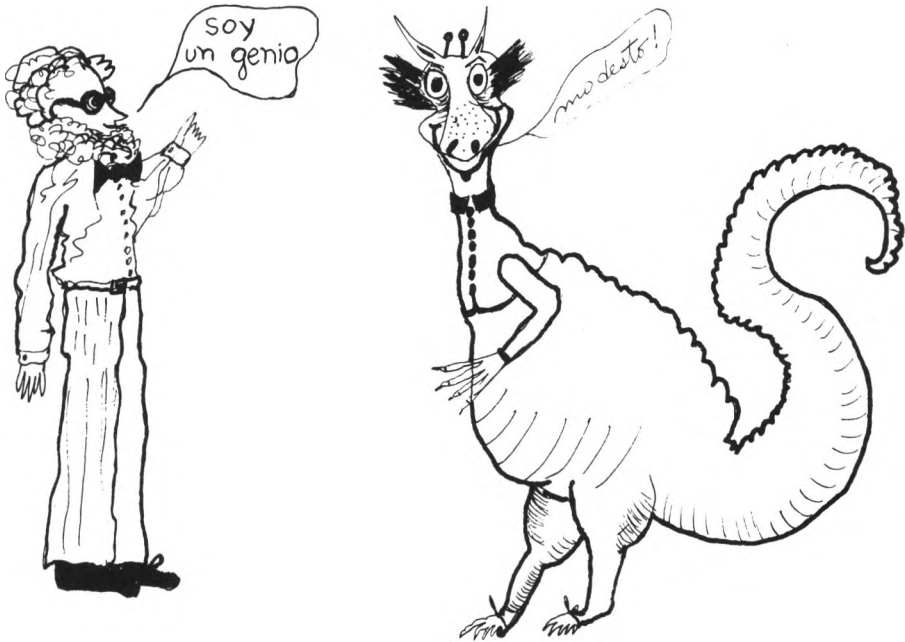


BASIC: Mmmm ...! Déjame pensar.

$$\frac{5x^2 + 3M}{2N\sqrt{5}} = \sqrt{4bg(x+1)}$$



BASIC: ¡Ya está! A cada dato le damos un nombre PEPE, ARBOL, COCHE..., el nombre que quieras y para saber qué dato contiene un nombre, vamos a utilizar el signo =
PEPE = 2
guardará el dato 2, con el nombre PEPE.



El BASIC es fenomenal, nos ha resuelto el problema de una forma sencillísima, observa

```
10 PEPE = 1  
20 PRINT PEPE  
30 END
```

Si tecleas RUN, te saldrá 1. Tradúzcanos el Programa al español:

Línea 10: Guarda en un dato que se llame PEPE el valor 1.

Línea 20: Escribe el número que tenga el dato PEPE.

Cualquier nombre que represente un dato, se llama variable, porque puede variar o cambiar de valor.

Prueba a ejecutar el siguiente Programa:

```
10 A = 8
20 A = 7
30 PRINT A
40 END
```

Como verás, has obtenido un 7, esto es porque el ordenador solamente guarda el último valor que le has dado a la variable y se olvida de todos los anteriores. Si quieres guardar varios valores, tendrás que utilizar varias variables.

```
10 ROSA = 5
20 PELOTA = 7
30 PRINT ROSA + PELOTA
40 END
```

Los nombres de variables pueden ser, como ya te he dicho, cualquiera, pero respetando dos reglas:

1. El primer carácter tiene que ser una letra (de la A a la Z).
2. El ordenador sólo reconoce los dos primeros caracteres, con lo que para él es lo mismo PELOTA que PEPITO.

Para introducir valores en las variables, como nos ha dicho BASIC, vamos a utilizar el signo « = ».

Debes tener una cosa muy clara. El signo « = » no tiene el significado que te enseñan en el colegio. Para nosotros sirve para dar valores a las variables. Mira este ejemplo y lo entenderás estupendamente.

Vamos a calcular el cuadrado de un número:

Primero tecleamos NEW para borrar los datos que hubiera. Te recuerdo, que cuando acabes de escribir una línea tienes que pulsar la tecla ENTER, para que te obedezca.

Ahora, escribamos el Programa paso por paso.

Introducimos la primera línea que borra la pantalla.

```
10 CLS
```

Necesitamos una variable que contenga el número que vamos a elevar al cuadrado, por ejemplo: PAPA y tenemos que darle el valor del primer número que tiene que elevar al cuadrado: el 7.

```
20 PAPA = 7
```

Calculamos el cuadrado del número y lo guardamos por ejemplo en MAMA

```
30 MAMA = 7 ^ 2
```

Ahora escribimos en pantalla un mensaje que diga: "EL CUADRADO DE 7 es 49", esto es:

```
40 PRINT "EL CUADRADO DE"; PAPA; "ES"; MAMA
```

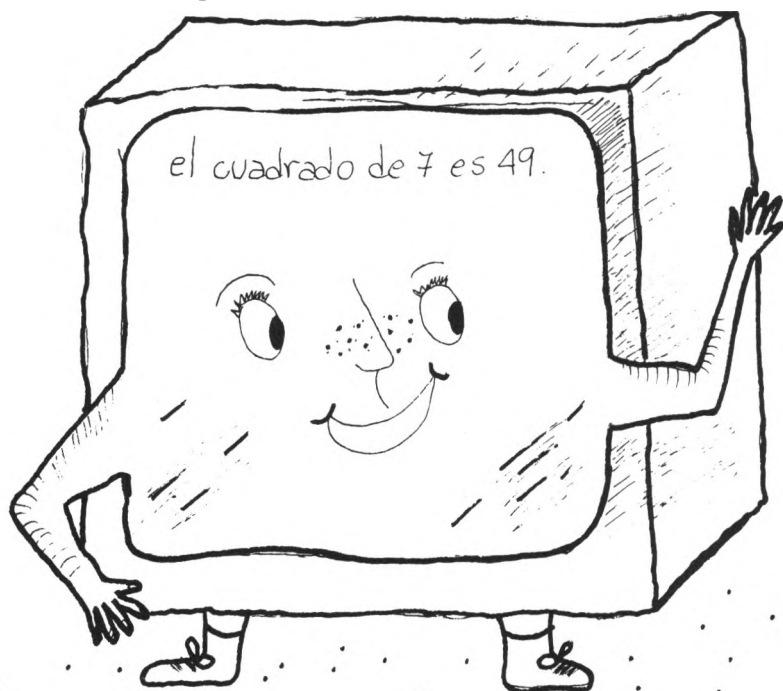
Puedes dar el valor de la variable de muchas formas. La más sencilla, es poniendo el número que debe contener como en

la línea 20, o una operación matemática, cuyo resultado es el valor de la variable, como hemos visto en la línea 30.

Como antes te he dicho, el signo “=” no funciona como hasta ahora estás acostumbrado, funciona así:

Siempre a la izquierda del signo «=» está el nombre de una variable y a la derecha el valor que debe tomar en ese momento.

Teclea RUN, ¿qué ocurre?



Hemos hecho un Programa que nos calcula el cuadrado de 7. Si quieres calcular el cuadrado de 8 o de otro número, sólo tienes que volver a escribir las líneas 20 y 30 poniendo 8 en lugar de 7.

—¿Cómo que «sólo»? Para eso, cojo mi calculadora de bolsillo y acabo antes.

Bueno pues tienes toda la razón en quejarte. Vamos a modificar un poco el Programa para que sea más sencillo.

En la expresión que se pone al lado derecho del igual, también se pueden poner nombres de variables y el DRAGON las sustituirá por su valor. Cambia la línea 30 por:

```
30 MAMA = PAPA ↑ 2
```

El ordenador guardará en MAMA, el valor que tenga PAPA, elevado al cuadrado.

Así, cuando quieras saber el cuadrado de un número sólo tendrás que cambiar la línea 20.

¿Estás contento?

— ¡No! Me quedo con la calculadora.

La verdad, no sé qué más quieres.

— Pues quiero, que el DRAGON me pregunte "qué número quieres" y yo le diga: "8" y él me conteste "EL CUADRADO DE 8 ES 64" y yo no tenga que andar modificando el Programa cada "dos por tres".

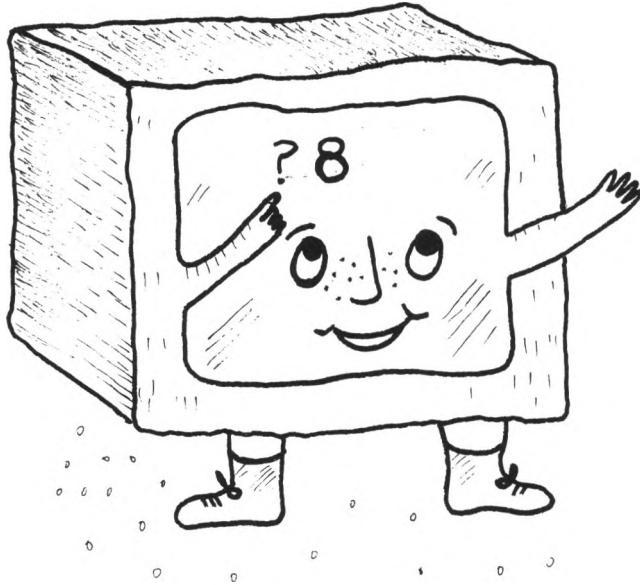
¡Qué exigente eres! Pero no te creas que me has pillado, porque ayer estuve cenando con BASIC y me contó cómo hacerlo.

Cambia la línea 20 y escribe

```
20 INPUT PAPA
```

Teclea RUN. Verás que sale un signo de interrogación en la pantalla y el cursor parpadeando al lado. La razón, es que el DRAGON está esperando que teclees el número que deseas guardar en PAPA y pulses ENTER.

Teclea 8 y pulsa ENTER. Te ha respondido.



Es lo mismo que si hubieses escrito

```
20 PAPA = 8
```

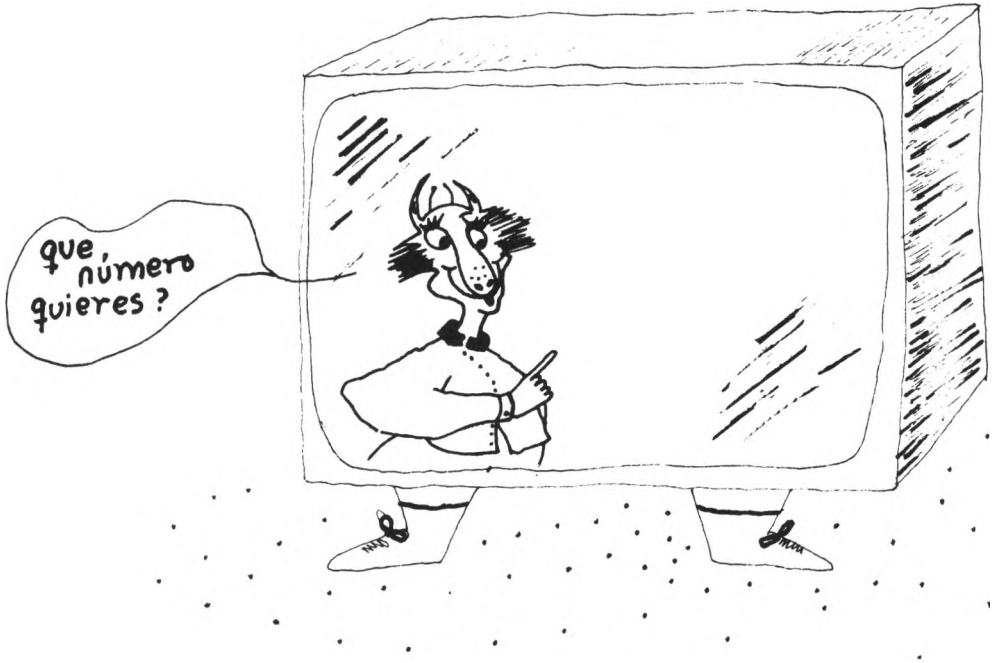
INPUT, es una instrucción estupenda que sirve para darle el valor a una variable en lugar de desde dentro del Programa, desde el teclado y así dialogas con él.

Así, cada vez que ejecutes el Programa, la variable PAPA no tendrá un valor fijo sino que te preguntará su valor.

Claro que normalmente, una pregunta no es sólo el signo “?” sino que tiene un texto, por eso, en la instrucción INPUT puedes escribir una frase de la siguiente forma.

```
20 INPUT "QUE NUMERO QUIERES"; PAPA
```

De esta forma, en lugar de escribir solamente el signo de interrogación, escribe la frase entera acabada en “?”



Hasta ahora, sólo hemos utilizado variables que contenían números, pero también pueden contener frases.

Funcionan igual, la única diferencia está en el nombre. Para indicar que una variable va a contener frases, en lugar de números, al final del nombre se le añade el signo «\$». Por lo demás, como si fueran números.

¡Oye! no intentes hacer operaciones con variables que contengan frases, que todavía no se ha inventado cómo multiplicar o dividir por letras.

Las variables que contienen frases, se llaman variables alfanuméricas (porque pueden contener cualquier carácter) o variables de cadena (porque contienen una cadena de caracteres).

Mira qué ejemplo más divertido

```
10 CLS
20 A$ = "y pulsa ENTER"
30 PRINT "PIENSA UN NUMERO", A$
40 INPUT B$
50 PRINT "SUMALE 2", A$
60 INPUT B$
70 PRINT "EL RESULTADO MULTIPLICALO POR 2", A$
80 INPUT B$
90 PRINT "RESTA 7 AL RESULTADO", A$
100 INPUT B$
110 PRINT "SUMALE EL NUMERO QUE HAS PENSADO", A$
120 INPUT B$
130 PRINT "DIVIDELO POR 3", A$
140 INPUT B$
150 PRINT "SUMALE 2", A$
160 INPUT B$
170 PRINT "RESTALE EL NUMERO QUE PENSASTE", A$
180 INPUT B$
190 CLS
200 PRINT "          TATACHAN!!"
210 PRINT
220 PRINT "MUCHA ATENCION"
230 PRINT "SOY UN GRAN ADIVINO"
240 PRINT "EL RESULTADO ES:          1"
250 PRINT
260 PRINT "APLAUSOS POR FAVOR!"
```

¿Te has fijado en que pongo después de cada frase INPUT B\$? Esto lo hago, porque cuando he pensado el Programa, me ha parecido bien que después de decirle cada operación que debe hacer, haya una parada para darte tiempo a calcular y la mejor

forma es la que he utilizado (¡modestia aparte!). Escribo INPUT B\$ y así le pido un dato, me da igual qué dato me dé, pero hasta que no me dé algo no continúo.

Además he utilizado otro truco. Para continuar jugando tienes que pulsar ENTER. Recuerda que conviene que en todos los Programas, quien los usa sepa en todo momento lo que debe hacer, por lo que después de cada frase hay que escribir «y pulsa ENTER». Es muy aburrido estar todo el rato escribiendo lo mismo, por eso guardo «y pulsa ENTER» en la variable A\$ y le digo que escriba A\$. Es mucho más cómodo y rápido.

un gran salto

Vamos ahora a hacer un Programa, que nos resuelva el típico problema de calcular el área de un triángulo.

Para poder resolverlo, lo primero que necesitamos es conocer la base y la altura y la mejor forma es preguntártelo a ti:

```
10 CLS
20 INPUT "BASE"; A
30 INPUT "ALTURA"; B
```

Ahora calculamos el área y escribimos el resultado en la pantalla;

```
40 S = (A * B) / 2
50 PRINT "EL AREA ES"; S
```

Así pues, de ahora en adelante sólo escribiremos RUN, y respondiendo a los datos, tendremos rápidamente el área del triángulo.

— ¡Oye! y si tengo que calcular el área de muchos triángulos, ¿es necesario que teclee RUN cada vez?

Pues, tal como tenemos el Programa, no tienes otro remedio. Pero hay una solución muy fácil.

Vamos a introducir una nueva línea que diga ¡y ahora vuelve a empezar desde la línea 20! es decir, «vete a 20» y como vete en inglés se dice GOTO, pondremos

```
60 GOTO 20
```

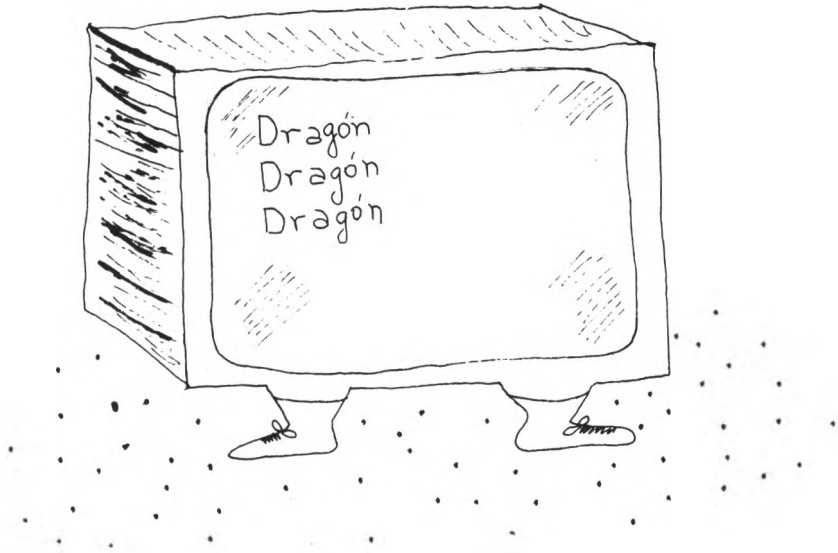
Ejecuta tu Programa. Ya ves, automáticamente después de darte el resultado, vuelve a pedirte la base y así indefinidamente.

Si no has entendido todavía la instrucción GOTO, te voy a poner otro ejemplo más fácil.

Voy a hacer un Programa que escriba tu nombre muchas, muchas veces.

```
10 CLS  
20 INPUT "DIME TU NOMBRE"; A$  
30 PRINT A$  
40 GOTO 30
```

Te pedirá el nombre, lo guarda en A\$, lo escribe en pantalla (línea 30) y vuelve a escribir tu nombre y así indefinidamente.



Si quieres que lo haga a «dos columnas», sustituye la línea 30 por

```
30 PRINT A$,
```

Y si quieres que te lo escriba todo seguido, escríbele

```
30 PRINT A $;
```

Elige la forma que más te guste ¡ah! y si no te acuerdas para que sirve la coma y el punto y coma, consulta la lección anterior.



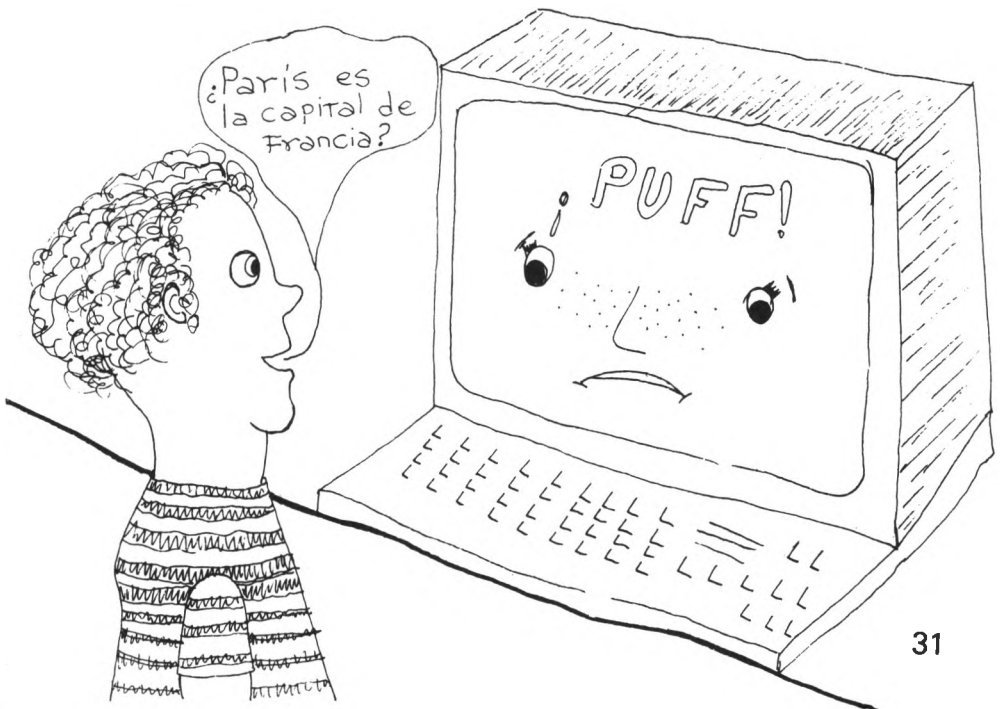
has aprendido

1. Para borrar la pantalla utilizaremos la palabra `CLS` y para borrar la memoria, `NEW`.
2. Cualquier nombre que represente un dato se llama variable, pero recuerda que el nombre ha de empezar por una letra, de la A a la Z.
3. Si el dato es una palabra o frase, es decir, es una cadena, la variable tiene que terminar con el signo `$`.
4. Para dar el valor a una variable, utilizaremos el signo « = ».
5. El valor de una variable lo podemos dar desde el teclado, mientras el Programa se está ejecutando, con la instrucción `INPUT`.
6. La orden `GOTO`, que significa «IR A», hace saltar directamente a la línea cuyo número acompaña a la orden.

el dragón impone condiciones

Hasta ahora, el Ordenador nos ha hecho preguntas a nosotros. Pero ya es hora de que se las hagamos nosotros a él. Para que nos pueda responder, es necesario que sea capaz de tomar alguna decisión, aunque sea pequeñita. Por eso, te voy a explicar una nueva instrucción: IF... THEN...

Pero para que se pueda decidir, necesita algo sobre lo que pueda optar, es decir, necesita alguna frase que pueda ser verdad o mentira. Ahora bien, como en el fondo él es bastante tonto, la frase no puede ser muy complicada.



Se me han ocurrido frases que son muy sencillas y en las que el Dragón no va a tener ningún problema para respondernos.

Solamente le vamos a preguntar ¿Es 2 mayor que 3? o ¿Es 5 igual a 8? y cosas de este estilo.

Acompáñame y te daré una explicación detalladísima.

Todas las preguntas que le hagamos al Dragón, van a tener esta forma:

¿ES VARIABLE $\left\{ \begin{array}{l} \text{Mayor} \\ \text{Menor} \\ \text{Igual} \\ \text{Mayor o igual} \\ \text{Menor o igual} \\ \text{Distinto} \end{array} \right\}$ que $\left\{ \begin{array}{l} \text{variable} \\ \text{número} \end{array} \right\}$?

Como escribir «mayor», «menor», etc... es muy largo, vamos a aprovecharnos de las matemáticas que tienen ya signos especiales. Mira, estos son:

Mayor: >

Menor: <

Igual: =

Mayor o igual: >=

Menor o igual: <=

Distinto: < >

Donde he escrito VARIABLE, quiere decir que tienes que poner el NOMBRE de una variable.

Igualmente donde pone $\left\{ \begin{array}{l} \text{variable} \\ \text{número} \end{array} \right\}$ tienes que poner el nombre de una variable o un número.

Vamos a ver muchos ejemplos para que te quede muy claro:

– Tienes guardado en X, la nota que has sacado en «tracas» y quieres saber si has aprobado. Le preguntarías al ordenador:

¿Es X mayor o igual que 5? que es lo mismo que preguntarle si vale 5, 6, 7...

Y esto lo escribirías en su lenguaje así:

$$X \geq 5$$

– Uno muy fácil. Pregunta si X es distinto de Y.

$$X \neq Y$$

– Ahora pregúntale si eres menor de edad. Suponte que tu edad está en la variable EDAD.

$$EDAD < 18$$

Bueno pues ya sabemos qué preguntas hacer al Dragón.

Ha llegado entonces la hora de que te cuente cómo funciona la instrucción IF... THEN... y la mejor forma es traducírtela al español:

IF pregunta	THEN instrucción.
SI pregunta	ENTONCES instrucción

Si la pregunta es cierta entonces ejecuta la instrucción. Si no continúa en la línea siguiente.

Y ahora a practicar mucho.

Vamos, en primer lugar, a hacer un Programa muy sencillo, que nos diga cuál es la capital de Italia y de Francia.

Este Programa, tú lo puedes ampliar para que te diga la capital de los países que quieras.

Como siempre, veámoslo paso por paso:

Empezamos pidiendo el nombre del país que queremos saber su capital.

```
10 CLS  
20 INPUT "DIME UN PAIS"; A$
```

Ahora, mira estas líneas atentamente:

```
30 IF A$ = "ITALIA" THEN GOTO 70  
40 IF A$ = "FRANCIA" THEN GOTO 90  
50 PRINT "NO CONOZCO SU CAPITAL"  
60 GOTO 20  
70 PRINT "SU CAPITAL ES ROMA"  
80 GOTO 20  
90 PRINT "SU CAPITAL ES PARIS"  
100 GOTO 20
```

En la línea 30 hemos escrito:

«Si el país que han introducido es ITALIA vete a la línea 70»

Y en la línea 70, como es de esperar, contestamos que la capital es ROMA y volvemos a la línea 20 para pedir otro país.

De la misma forma, en la línea 40 pedimos que si el país es FRANCIA, se dirija a la línea 90 e imprima que la capital es PARIS, para volver de nuevo a pedir otro país (GOTO 20).

Presta mucha atención:

Si el país es ITALIA se va a la línea 70 y si es FRANCIA a la línea 90, entonces, el Programa tomará el camino de la línea

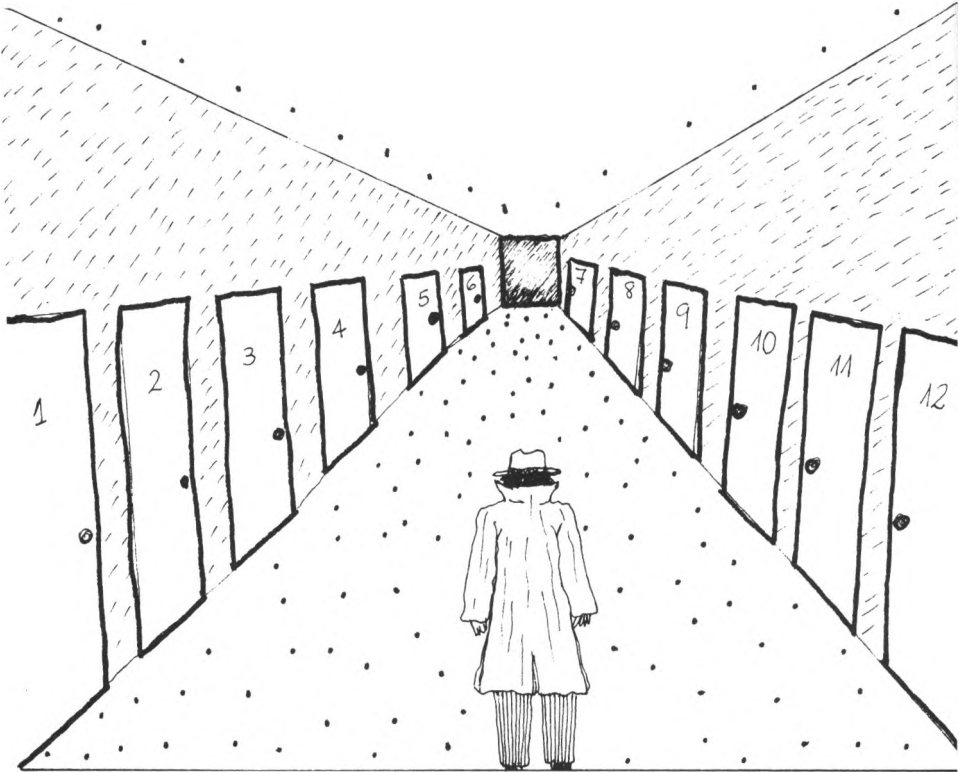
50, sólo en el caso de que tecleemos un país distinto de ITALIA y FRANCIA y como no hemos programado más países, escribimos que no conocemos la capital y volvemos a pedir otro país.



Y por último, te planteo un Programa de detectives. Antes de mirar la solución intenta hacerlo tú.

«Eres James Bond, el agente secreto 007. Estás en un gran pasillo con doce habitaciones numeradas del 1 al 12. Una potencia enemiga, ha escondido en una de ellas, un potente explosivo que debes desactivar, abriendo la puerta de la habitación donde esté. Tienes tres intentos para conseguirlo. Por suerte,

un agente amigo ha dejado en cada habitación una pista, que indica si la bomba está en una habitación con número mayor o menor. ¡Buena suerte!



```
10 CLS
20 X=7
30 I=0
40 IF I=3 THEN GOTO 130
50 INPUT "QUE PUERTA QUIERES ABRIR"; A
60 I=I+1
70 IF X=A THEN GOTO 110
80 IF X<A THEN PRINT "BUSCA EN UNA HABITACION
  MAS PEQUEÑA"
```

```
90 IF X>A THEN PRINT "BUSCA EN UNA HABITACION
  MAS GRANDE"
100 GOTO 40
110 PRINT "ENHORABUENA, LO CONSEGUISTE!!"
120 END
130 PRINT "BOOOOOMN!!!"
140 PRINT "HAS FALLADO"
150 PRINT "ESTALLO LA BOMBA"
160 PRINT "Y HAS MUERTO"
170 END
```

Vamos a comentar algunas instrucciones:

- En la línea 20, hemos guardado en X el número de la puerta donde está la bomba.

Si quieres que el ordenador elija la habitación, consulta la función RND en tu manual de referencia.

- En la variable I, guardamos el número de intentos que hemos hecho, por eso, en la línea 30 le decimos que vale 0 (todavía no hemos hecho ningún intento) y en la línea 40, que cuando ya hemos utilizado los tres intentos, explote la bomba (GOTO 130).
- En la línea 50, introducimos la puerta que queremos abrir, y como ya hemos gastado un intento, sumamos 1 a I (línea 60).
- En la línea 70, miramos si tenemos desactivada la bomba y si es así, nos vamos a la línea 110, que son las felicitaciones.
- En las líneas 80 y 90 damos la pista y la 100 volvemos a la 40, para un nuevo intento.

Divertido, ¿verdad?



has aprendido

1. La instrucción IF... THEN... se usa para tomar decisiones en los Programas. Significa SI... ENTONCES...
2. Su formato es IF pregunta THEN instrucción.
3. Si la pregunta es cierta, entonces ejecuta la instrucción. Si no continúa en la siguiente línea.
4. Las preguntas han de ser:

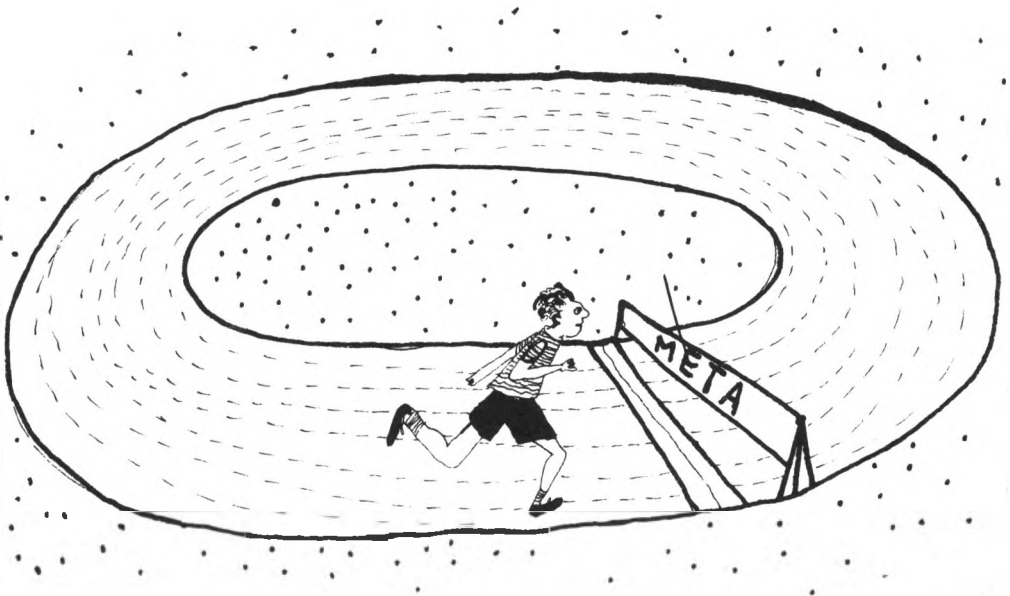
¿ES VARIABLE $\left\{ \begin{array}{l} > \\ < \\ > = \\ < = \\ < > \end{array} \right\}$ que $\left\{ \begin{array}{l} \text{variable} \\ \text{número} \end{array} \right\}$?

el dragón está algo mareado

En esta lección vamos a aprender qué es un bucle y cómo manejarlo con tu ordenador.

Imagínate que participas en una carrera que consiste en dar 10 veces la vuelta a un estadio. Te pondrás a correr y llevarás mentalmente la cuenta de cuántas vueltas has dado. Así, si has dado tres vueltas y vuelves a pasar por la línea de meta pensarás: "Ya he dado cuatro vueltas."

Cuando veas que has dado 10 vueltas entonces dejarás de correr y descansarás.



Bueno, pues quizá sin saberlo, estás realizando un bucle.

Un bucle es un conjunto de instrucciones que repites un número de veces. En nuestro ejemplo, la instrucción «DAR UNA VUELTA AL ESTADIO» lo has repetido 10 veces y por eso tienes el bucle «DAR 10 VUELTAS AL ESTADIO».

Vamos con otro ejemplo.

Cuando te pones a jugar al escondite con tus amigos te tapas los ojos y cuentas hasta 100.

Estás otra vez haciendo un bucle.

La instrucción que repites 100 veces es «DECIR EL NUMERO SIGUIENTE», si has dicho «DOS», dices «TRES» y después «CUATRO», etc... y el bucle es decir consecutivamente todos los números del 1 al 100.

Veamos cómo podemos hacer que nuestro ordenador cuente por nosotros.

Como el ordenador no puede hablar, haremos que escriba el número en la pantalla.

El Programa es muy fácil. Consiste en imaginar que una variable, por ejemplo la variable I, es el número que nos toca decir y entonces, empezando desde 1, lo escribimos en la pantalla y después lo sumamos 1. Estas son las dos instrucciones que debemos repetir 100 veces.

Para saber si hemos acabado ponemos una línea que diga «si I es menor que 100 vuelve a repetir las instrucciones del bucle».

Y ahora, vuélvete a leer estos últimos párrafos pero mirando el listado del Programa:

```
10 CLS
20 I = 1
```

```
30 PRINT I
40 I = I + 1
50 IF I < 100 THEN GOTO 30
60 END
```

Los bucles son importantísimos y se utilizan muy a menudo, por eso el BASIC tiene una instrucción especial para manejarlos.

Fíjate en todos los bucles hay una serie de pasos que son comunes:

1. Necesitamos una variable (I) para controlar el número de veces que hemos de repetir el bucle.
2. Necesitamos el primer valor que debe tomar la variable (1).
3. Evidentemente necesitamos también el valor final que debe tomar la variable, 100.
4. Cada vez que se realice un bucle la variable I aumenta en 1.

Bueno pues conociendo estas cosas, el BASIC ha ideado una forma más fácil y cómoda de escribir los bucles. Fíjate:

```
10 CLS
20 FOR I = 1 TO 100
30 PRINT I
40 NEXT I
50 END.
```

Es decir:

Línea 10: BORRAR LA PANTALLA

Línea 20: DESDE QUE I VALE 1 HASTA QUE VALE 100 REPITE

Línea 30: ESCRIBE EL VALOR DE I.

Línea 40: PASA AL SIGUIENTE I.

Línea 50: FIN DEL PROGRAMA.

La instrucción FOR/NEXT sirve para controlar los bucles. Vamos a descubrirla un poco más.

Como sabes, un bucle es un conjunto de instrucciones que se repite un número de veces. Pues bien, la instrucción FOR... TO... indica el principio de las instrucciones que se repiten y la sentencia NEXT... el final.

Ten presente que siempre que pongas una sentencia FOR... TO... debes poner una sentencia NEXT.

Como verás es fácil. Con la instrucción FOR... TO... la variable que eliges toma el primer valor que la has dado y ejecuta las instrucciones siguientes hasta que encuentra una instrucción NEXT en la que se le suma 1 a la variable y se vuelve a ejecutar el bucle. Esto se hace hasta que la variable toma el último valor que has dado.

Y para que te quede muy claro, vamos a hacer otro ejemplo.

Supongo que te acuerdas lo pesado y aburrido que era aprenderse la tabla de multiplicar en colegio.



Vamos a hacer un Programa que nos cuente la tabla de multiplicar de un número para tener una «chuleta» cuando no nos acordemos.

Veamos cuáles son los pasos que tenemos que realizar.

Primero introducimos el número del que queremos la tabla de multiplicar y después escribimos el número por 1, por 2..., hasta 10. Es decir, tenemos que repetir 10 veces el mismo proceso: «Escribir el número que hemos elegido por la variable del bucle.»

Fíjate que cómo la variable cuenta las veces que estamos haciendo el bucle vale 1, luego 2, 3, ... y entonces estamos multiplicando el número por 1, 2, 3, ... tal y como queremos.

Mira el Programa, ejecútalo y analiza cómo funciona.

```
10 CLS
20 INPUT "QUE NUMERO QUIERES"; A
30 CLS
40 FOR I = 1 TO 10
50 PRINT A; "POR"; I; "ES"; A*I
60 NEXT I
70 END
```

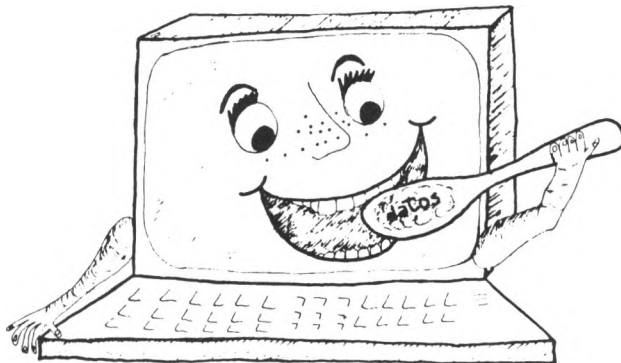


has aprendido

1. Un bucle es un conjunto de instrucciones que repites un número de veces.
2. Los bucles comienzan con la instrucción FOR... y terminan con la orden NEXT.
3. La instrucción FOR... TO..., que se escribe FOR variable = número TO $\left\{ \begin{array}{l} \text{variable} \\ \text{número} \end{array} \right\}$ sirve para contar el número de veces que vamos a realizar el bucle.
4. Recuerda que siempre que utilices una sentencia FOR necesariamente deberás poner una sentencia NEXT, para indicar el fin del bucle.

el dragón come datos

A estas alturas que ya casi eres un experto Programador, te habrás dado cuenta de que lo más importante, para el Ordenador, son los datos. Es lógico, son su comida y sin ellos no puede hacer nada.



Por eso, no te debe extrañar que existan tres instrucciones que juntas se utilizan para manejar muchos datos.

Estas instrucciones son READ (leer), DATA (datos) y RESTORE (restaurar).

Mira este ejemplo y verás qué fácil es:

```
10 DATA "ENERO", 31, "FEBRERO", 28, "MARZO", 31,  
"ABRIL", 30, "MAYO", 31, "JUNIO", 30, "JULIO", 31,  
"AGOSTO", 31, "SEPTIEMBRE", 30, "OCTUBRE", 31,
```

"NOVIEMBRE", 30, "DICIEMBRE", 31

```
20 CLS
30 FOR I = 1 TO 12
40 READ A$
50 READ A
60 PRINT A$; "TIENE"; A; "DIAS"
70 NEXT I
80 END
```

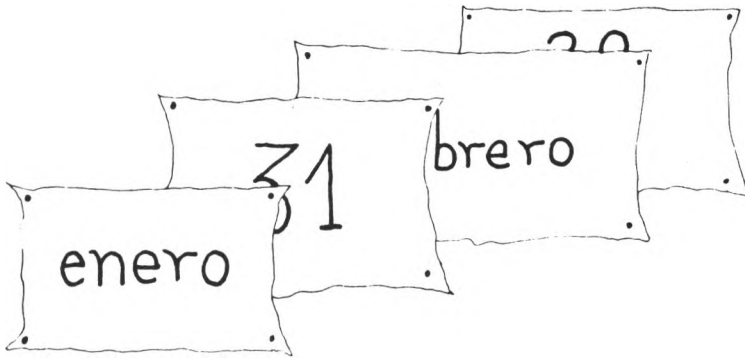


Este Programa te ayudará a recordar cuántos días tiene cada mes. Pero veamos cómo funciona.

Cuando tú ejecutas un Programa, nuestro amigo BASIC busca a ver si encuentra alguna SENTENCIA DATA y si la encuentra, coge un montón de hojas en blanco y hace lo siguiente:

Empieza por el principio del Programa a buscar las sentencias DATA y, cada vez que encuentra una, escribe los datos que contenga en las hojas de papel pero de una forma muy particular. Como los datos van separados por comas, escribe en cada hoja un solo dato, ¡acuérdate! UNO SOLO y lo hace con todas las sentencias DATA que encuentra, sin importarle dónde están situadas en el Programa.

Después de esto, tiene un montón de hojas cada una con un dato y puestas correlativamente, según las ha encontrado.



Como ya ha cogido todos los datos, ya no volverá a hacer caso a ninguna instrucción DATA.

Ahora ya el funcionamiento es muy sencillo.

Cada vez que encuentre una sentencia READ (que lleva detrás una variable) leerá el dato de la hoja que tenga encima y lo guardará en la variable que acompaña a READ, pasando después la hoja.



Y esto lo hará con todas las hojas.

Ten cuidado con una cosa, si le mandas leer un dato cuando ya ha dado la vuelta a todas las hojas, te dará un mensaje de error.

¿Fácil verdad?

La instrucción READ, te permite que pongas detrás varias variables separadas por comas y te leerá varios datos (por supuesto pasando tantas hojas como datos lea).

Por eso, las líneas 40 y 50 de nuestro Programa se podrían haber escrito en una sola línea de esta forma:

```
40 READ A$, A
```

Si has entendido esto, la instrucción RESTORE es de lo más simple. Lo único que hace es colocar todas las hojas al derecho con lo que el siguiente dato que leerá, cada vez que encuentre esta instrucción, será otra vez el primero.



Vamos ahora a hacer un ejemplo para que veas cómo se utiliza.

```
10 DATA "ENERO", 31, "FEBRERO", 28, "MARZO", 31,  
"ABRIL", 30, "MAYO", 31, "JUNIO", 30, "JULIO", 31,  
"AGOSTO", 31, "SEPTIEMBRE", 30, "OCTUBRE", 31,  
"NOVIEMBRE", 30, "DICIEMBRE", 31  
20 CLS  
30 INPUT "DIME UN MES"; A$
```

```
40 C = 1
50 READ B$, A
60 IF A$ = B$ THEN GOTO 120
70 C = C + 1
80 IFC < 12 THEN GOTO 50
90 PRINT A$; "NO ES NINGUN MES"
100 RESTORE
110 GOTO 30
120 PRINT A$; "TIENE"; A; "DIAS"
130 RESTORE
140 GOTO 30
```

En la línea 30, introduces el mes que quieres saber cuántos días tiene.

En la línea 40 inicializas la variable C a 1, que la vas a utilizar para controlar cuántos meses has leído.

En la línea 50, lees el mes que te toca y el número de días que tiene.

Si el mes que has leído coincide con el mes que has introducido (la línea 60), saltas a la línea 120, escribes los días que tiene, colocas los datos para empezar a leer de nuevo desde el mes de enero (el RESTORE de la línea 130) y vuelves a pedir otro mes.

Si no coinciden los meses, como has hecho una lectura, aumentas tu contador en 1 (línea 70) y si no has leído los doce meses (línea 80) te vas a leer el siguiente mes (GOTO 50).

Si has leído los 12 meses y ninguno de ellos coincide con el que tú has metido, es que ese mes no existe y entonces sigues el camino de la línea 90, te preparas para leer enero (RESTORE) y pides otro mes.

Mira el Programa varias veces y verás que enseguida le coges el truco y lo manejas estupendamente.



has aprendido

1. Cuando queramos que nuestro amigo DRAGON coma muchos datos, es decir, los guarde para cuando los necesitemos, utilizaremos la sentencia DATA.
2. Los datos se escriben detrás de la instrucción DATA y separados por comas.
3. Para leer un dato lo haremos con la sentencia READ y a continuación una variable.
4. La instrucción RESTORE obliga a que el próximo dato a leer sea otra vez el primero.

el dragón hace los recados

En esta lección, vamos a estudiar las SUBROUTINAS y por supuesto antes de empezar tengo que explicarte qué son.

Una subrutina es una parte del Programa que se utiliza varias veces en el mismo, es decir, un subprograma.

Como vas a ver, son muy útiles y te permiten que los Programas sean más cortos y más fáciles de entender.

Vamos a ver un ejemplo:

```
10 CLS
20 PRINT "ESTE ES UN EJEMPLO"
30 PRINT
40 INPUT "PULSA ENTER"; A$
50 CLS
60 PRINT "PARA ENTENDER COMO FUNCIONAN"
70 PRINT
80 INPUT "PULSA ENTER"; A$
90 CLS
100 PRINT "LAS SUBROUTINAS"
110 PRINT
120 INPUT "PULSA ENTER"; A$
130 CLS
140 END
```

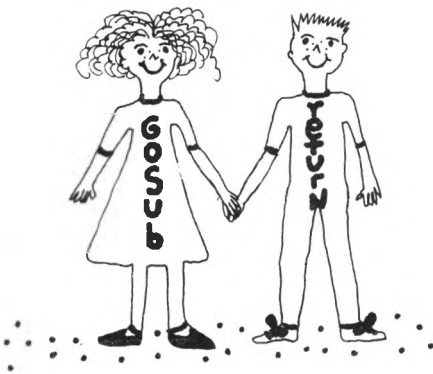
Hemos hecho un Programa en el que después de cada frase, para continuar, hay que pulsar ENTER.

Fíjate en estos grupos de líneas:

30, 40, y 50
70, 80, y 90
110, 120, y 130

Son exactamente iguales ¿verdad? Pues bien, esto es una subrutina, que en nuestro ejemplo se ha utilizado tres veces.

Nuestro amigo BASIC, que no nos podía fallar, nos ha preparado una instrucción, para utilizar las subrutinas con facilidad. Se llama GOSUB y va siempre acompañada de RETURN, igual que FOR y NEXT.



GOSUB funciona de forma muy parecida a como lo hacía la sentencia GOTO.

Cuando nosotros escribimos «20 GOSUB 100», continúa ejecutando la línea 100, igual que si hubiésemos puesto «20 GOTO 100».

Pero aquí viene la diferencia, cuando se encuentre con una línea que diga «RETURN», que significa VOLVER, continuará ejecutando la línea que haya después de la sentencia GOSUB, en nuestro caso la 30.

Mira cómo haríamos el Programa anterior.

```
10 CLS
20 PRINT "ESTE ES UN EJEMPLO"
30 GOSUB 90
40 PRINT "PARA ENTENDER COMO FUNCIONA"
50 GOSUB 90
60 PRINT "LAS SUBROUTINAS"
70 GOSUB 90
80 END
90 PRINT
100 INPUT "PULSA ENTER"; A$
110 CLS
120 RETURN
```

Sigamos el Programa para ver qué camino toma.

Línea 10: Borra la pantalla.

Línea 20: Escribe «ESTE ES UN EJEMPLO».

Línea 30: Salta a la subrutina que empieza en la línea 90.

Línea 90: Imprime una línea en blanco.

Línea 100: Escribe «PULSA ENTER» y espera a que pulses ENTER.

Línea 110: Borra la pantalla.

Línea 120: Sentencia de vuelta, como se llamó a la subrutina en la línea 40.

Línea 40: Escribe «PARA ENTENDER COMO FUNCIONAN».

⋮

¿Ves qué fácil?, GOSUB 100, manda a ejecutar la subrutina que empieza en la línea 100 y cuando encuentre una sentencia RE-

TURN, vuelve a la línea siguiente a la sentencia GOSUB que la llamó.

Continuemos con otro ejemplo. Fíjate, que las subrutinas, normalmente se escriben detrás de la sentencia END.

Tu profesor de Matemáticas, que el otro día vino de muy mal humor, os ha puesto el siguiente problema: "Cuántos días tiene ENERO y NOVIEMBRE juntos."

Tú, que eres un chico listo, se te ha ocurrido utilizar el Programa que hiciste en la lección 5, para que te responda, aun cuando decidan cambiarte los meses ¿verdad que eres capaz de hacerlo?

Inténtalo antes de mirar la solución.

```
10 DATA "ENERO", 31, "FEBRERO", 28, "MARZO", 31,
  "ABRIL", 30, "MAYO", 31, "JUNIO", 30, "JULIO", 31,
  "AGOSTO", 31, "SEPTIEMBRE", 30, "OCTUBRE", 31,
  "NOVIEMBRE", 30, "DICIEMBRE", 31
20 CLS
30 INPUT "PRIMER MES"; A$
40 GOSUB 110
50 X = A
60 INPUT "SEGUNDO MES"; A$
70 GOSUB 110
80 X = X + A
90 PRINT "LOS DOS JUNTOS TIENEN"; X "DIAS"
100 END
110 C = 1
120 RESTORE
130 READ B$; A
140 IF A$ = B$ THEN RETURN
150 C = C + 1
160 IF C <= 12 THEN GOTO 130
```



```
170 PRINT "MES EQUIVOCADO"  
180 END
```

Como verás es sencillo, la única duda que te puede surgir está en la línea 140. Si A\$ es igual a B\$, entonces salimos de la subrutina para volver al Programa. Recuerda que en A tenemos el número de días, pues lo hemos leído en la línea 130.



has aprendido

1. Una subrutina es una parte del Programa que se ejecuta en varios sitios.
2. Para ejecutar la subrutina que empieza en la línea *r*, se escribe `GOSUB r`.
3. El final de la subrutina es la línea `RETURN`.
4. Cuando se encuentra la sentencia `RETURN`, el Programa vuelve a la línea siguiente a la que llamó a la subrutina.
5. Las subrutinas, se escriben normalmente después de la sentencia `END`.

has amaestrado a tu dragón

En esta lección te propongo algunos ejemplos de lo que puedes hacer con tu Ordenador.

¡Que te diviertas!

Primer Programa

Este Programa te calcula la nota media que llevas en tus estudios.

```
10 CLS
20 INPUT "CUANTAS ASIGNATURAS TIENES"; A
30 FOR I = 1 TO A
40 PRINT "NOTA"; I;
50 INPUT B
60 S = S + B
70 NEXT
80 S = S/A
90 PRINT "LA NOTA MEDIA ES"; S
100 END
```

Segundo Programa

Vamos a hacer un reloj digital con nuestro Ordenador.

```
10 CLS
20 INPUT "DIME LA HORA"; H, M, S
30 FOR I = H TO 23
40 FOR J = M TO 59
```

```

50 FOR K = S TO 59
60 CLS
70 PRINT I; J; K
80 FOR L=1 TO 23 : NEXT L
90 NEXT K
100 S = 0
110 NEXT J
120 M = 0
130 NEXT I
140 H = 0
150 GOTO 30

```

Si se adelanta o atrasa, cambia la línea 80 y pon 25 ó 21 en lugar de 23.

Tercer programa

Te desafío a que intentes salir del laberinto del DRAGON.

```

10 CLS
20 PRINT "ESTAS EN UN LABERINTO"
30 PRINT "INTENTA SALIR DE EL"
40 PRINT "CON EL MINIMO DE MOVIMIENTOS"
50 PRINT "RESPONDE SOLAMENTE CON EL"
60 PRINT "PUNTO CARDINAL AL QUE QUIERES IR"
70 PRINT "PUEDES IR: AL NORTE, AL ESTE"
80 INPUT A$
90 IF A$ = "NORTE" THEN GOTO 130
100 IF A$ = "ESTE" THEN GOTO 70
110 PRINT "NO PUEDES IR EN ESA DIRECCION"
120 GOTO 80
130 PRINT "PUEDES IR AL SUR, AL OESTE"
140 INPUT A$
150 IF A$ = "SUR" THEN GOTO 70
160 IF A$ = "OESTE" THEN GOTO 190

```

```

170 PRINT "NO PUEDES IR EN ESA DIRECCION"
180 GOTO 140
190 PRINT "PUEDES IR: AL OESTE, AL NORTE, AL SUR"
200 INPUT A$
210 IF A$ = "OESTE" THEN GOTO 130
220 IF A$ = "NORTE" THEN GOTO 70
230 IF A$ = "SUR" THEN GOTO 260
240 PRINT "NO PUEDES IR EN ESA DIRECCION"
250 GOTO 200
260 PRINT "POR FIN LO CONSEGUISTE"
270 PRINT "ENHORABUENA"
280 END

```

Cuarto Programa

Y por último vamos a hacer un Programa un poco más serio, para que puedas llevar tu contabilidad.

```

10 CLS
20 INPUT "CUAL ES TU ASIGNACION SEMANAL"; A
30 CLS
40 PRINT "QUE DESEAS"
50 PRINT "1. INTRODUCIR GASTOS"
60 PRINT "2. SITUACION ECONOMICA"
70 PRINT "3. LISTA DE GASTOS"
80 PRINT
90 INPUT "PULSA 1, 2 ó 3"; B
100 IF B = 1 THEN GOSUB 140
110 IF B = 2 THEN GOSUB 280
120 IF B = 3 THEN GOSUB 460
130 GOTO 30
140 CLS
150 PRINT "EN QUE HAS GASTADO?"
160 PRINT "1. EN PIPAS"
170 PRINT "2. EN CHICLES"
180 PRINT "3. EN CANICAS"

```

```

190 PRINT "4. GASTOS VARIOS"
200 PRINT
210 INPUT "PULSA 1, 2, 3 ó 4"; B
220 INPUT "CUANTO HAS GASTADO"; C
230 IF B=1 THEN PIPAS = PIPAS + C
240 IF B=2 THEN CHICLES = CHICLES + C
250 IF B=3 THEN CANICAS = CANICAS + C
260 IF B=4 THEN VARIOS = VARIOS + C
270 RETURN
280 SI = A - PIPAS - CHICLES - CANICAS - VARIOS.
290 CLS
300 PRINT "TU SITUACION ECONOMICA ES"
310 IF SI > A/2 THEN GOTO 410
320 IF SI > A/3 THEN GOTO 390
330 IF SI > 0 THEN GOTO 370
340 PRINT "MOROSO"
350 PRINT "DEBES"; SI; "PESETAS"
360 GOTO 430
370 PRINT "POBRE"
380 GOTO 420
390 PRINT "PELIGROSA"
400 GOTO 420
410 PRINT "BUENA"
420 PRINT "TE QUEDAN", SI, "PESETAS"
430 PRINT
440 INPUT "PULSA ENTER"; A$
450 RETURN
460 CLS
470 PRINT "HAS GASTADO EN PIPAS"; PIPAS
480 PRINT "HAS GASTADO EN CHICLES"; CHICLES
490 PRINT "HAS GASTADO EN CANICAS"; CANICAS
500 PRINT "GASTOS VARIOS"; VARIOS
510 PRINT
520 PRINT "TOTAL GASTOS"; PIPAS + CHICLES + CA-
    NICAS + VARIOS
530 PRINT
540 INPUT "PULSA ENTER", A$
550 RETURN

```

apéndice. manejo del cassette

Vamos a aprender cómo manejar el cassette para poder guardar tus Programas.

Lo primero que debes hacer es consultar tu manual de referencia, para conectar correctamente tu cassette. A continuación toma una cinta virgen, que podamos grabar.

Antes de introducirla en tu cassette, asegúrate de pasar la parte del principio que es de plástico y sobre la que no se puede grabar.

Escribe este Programa para experimentar:

```
10 PRINT "PROGRAMA DE PRUEBA"  
20 PRINT "PARA VER SI"  
30 PRINT "HE APRENDIDO"  
40 PRINT "COMO MANEJAR"  
50 PRINT "EL CASSETTE"  
60 END
```

Para manejar el cassette, tienes que aprender dos cosas: cómo grabar tus Programas en la cinta y cómo recuperarlos.

En una cinta puedes grabar muchos Programas y para poderlos identificar, las vamos a dar un nombre, por supuesto, el que nosotros queramos.

A este Programa le vamos a llamar "DRAGON".

Para grabarlo tenemos el comando CSAVE.

Fijate que es un comando y por lo tanto no se utiliza con número de línea.

Es muy fácil de utilizar: pulsa las teclas PLAY y RECORD de tu cassette escribe

CSAVE "DRAGON"

y pulsa ENTER.

Entonces, te grabará el Programa que tenemos en Memoria, en el cassette con el nombre de DRAGON.

Ahora teclea NEW y pulsa ENTER.

Así has borrado tu Programa.

Rebobina la cinta al principio.

Vamos a recuperar el Programa que hemos grabado, mediante el comando CLOAD.

Por supuesto, después del comando tendremos que escribir el nombre del Programa que queremos recuperar. Escribe pues:

CLOAD "DRAGON"

pulsa ENTER y después la tecla PLAY de tu cassette.

Espera un momento y verás que tu Ordenador te da un mensaje, diciendo que ha encontrado tu Programa DRAGON y lo está cargando en Memoria.

Cuando acabe, teclea LIST y comprobarás que has recuperado tu Programa.



has aprendido

1. Para grabar el Programa que tienes escrito en ese momento, se escribe CSAVE "nombre".
2. Para recuperar un Programa, primero teclea NEW y luego el comando CLOAD "NOMBRE", donde «NOMBRE» es el nombre del Programa que quieres recuperar.

Espero que este libro de Programación te haya dado los conocimientos suficientes como para que hagas tus primeros Programas.

De todas formas, seguro que te ha abierto las puertas del apasionante mundo de la Informática y de la Programación.

Ahora te toca hacer el trabajo más atractivo. Ya estás preparado para investigar tu Ordenador y descubrir hasta su último secreto.

Para ello, además de este libro, tienes un Manual de Referencia que aunque mucho más complejo, contiene todos los detalles técnicos de tu DRAGON.

No intentes leer el Manual de Referencia «de un tirón».

Posiblemente te desesperes.

Hazlo poco a poco y sobre todo, después de leer algo, inmediatamente compruébalo en tu Ordenador y experimenta con ello.





Eurohard

Españoleto, 25

Teléf. (91) 410 30 64 - 410 31 96 - 410 34 98

Télex 45845 ICSG E

28010 MADRID