

Talent



Creado por SISTEMAS LOGICAL

Retrocomputing



Autores:

José Luis Ferreyro

Julio G. Rosenberg Zurich

(C) Sistemas Logical S. R. L. — 1986
Esmeralda 561, piso 3°, oficina 'B'
1007 - Buenos Aires - Argentina
Tel.: 392 - 7928 — 393 - 7669
I.S.B.N.: 950 - 43 - 1081 - 2

Manual desarrollado en la Argentina bajo licencia de SISTEMAS LOGICAL

Talent MSX LPC es producido en la Argentina por Telemática S. A. bajo una licencia y puede utilizarse o copiarse únicamente de acuerdo a los términos de dicha licencia.

Hecho el depósito que marca la ley. Reservados todos los derechos del autor. Prohibida la reproducción total o parcial de los soportes que contiene el programa y de este manual, por cualquier medio o procedimiento y con cualquier destino.

Sólo podrá utilizarse el programa "LPC" para el comando de un ordenador por quienes se encuentren legitimados al efecto por un contrato de licencia. La utilización en ausencia o fuera de los límites de tal contrato constituye una infracción a los derechos de propiedad intelectual del titular de la obra.

Este manual no se vende al público. Se entrega sin cargo a quienes adquieren una licencia de utilización del programa "LPC".

INTRODUCCION

LPC (Lenguaje Programación en Castellano) es un lenguaje computacional de fácil aprendizaje, que permite introducir a niños y personas no especializadas, en el manejo de un microcomputador.

A través de LPC se pueden dar las instrucciones para que el microcomputador realice las operaciones deseadas.

Las instrucciones están dadas totalmente en español y no se requiere ningún conocimiento previo de computación; así, si se desea que el microcomputador escriba algo en el televisor, sólo tendrá que especificar la instrucción ESCRIBA y a continuación lo que desea que represente.

LPC mantiene una estructura de lenguaje que no admite bifurcaciones, o sea que a la vez que facilita el seguimiento de un programa, forma a quienes se inician con LPC dentro de una disciplina estructurada, dándole un manejo lógico muy útil para cualquier otro lenguaje de computación.

La comunicación con la computadora se logra mediante el ingreso de instrucciones a través del teclado.

Un programa es un conjunto de instrucciones que la computadora guarda en su memoria y que le indican paso a paso lo que debe hacer para resolver un problema.

Programar es fácil y divertido. Este manual lo irá introduciendo paso a paso en la programación de su computadora personal con LPC.

CAPITULO 1

EL INTERPRETE LPC

Qué es un Intérprete?

Todas las computadoras entienden internamente sólo combinaciones de ceros (0) y unos (1). Cada orden, cada letra o número que la computadora guarda en su memoria, lo hace en base a una combinación de ceros y unos.

Sería entonces muy difícil para Ud. recordar las distintas combinaciones que correspondan a cada una de las instrucciones que deba ejecutar la computadora.

Por lo tanto, para facilitar la tarea de programación de su computadora existen los intérpretes, cuya función es similar a la de una secretaria bilingüe que interpreta sus órdenes y las traduce al lenguaje de la máquina.

Así si Ud. desea comunicarse con la computadora en un lenguaje determinado, bastará con que previamente a introducir las órdenes cargue en la memoria de su computadora al intérprete adecuado.

Algunos intérpretes, como el BASIC, se cargan automáticamente en su computadora. Esto se debe a que está contenido en una memoria especial que no se borra cuando Ud. apaga a la computadora.

Ud. ha recibido al intérprete de LPC en un cartridge (Cartucho) y por lo tanto, antes de comenzar a trabajar, deberá transferirlo a la memoria de su computadora.

Cómo se carga LPC (CARTRIDGE)

Lo único que deberá hacer es tener su computadora conectada al televisor, y apagada.

Una vez conectados, ejecute los siguientes pasos:

- 1) Encienda el televisor.
- 2) Introduzca el cartridge en la ranura que tiene su computador (tapada), detrás de las teclas HOME INS y DEL ,con la etiqueta de LPC hacia adelante. (Asegúrese que quede bien conectado)
- 3) Encienda su computadora, espere unos segundos y deberá aparecer primero:

MSX System
versión 1. 0

luego de 2 ó 3 segundos aparecerá:

```
L.P.C. V 1.7 Versión MSX
Copyright (c) por Sistemas Logical
1986 Todos los derechos reservados
Creado: Julio/85 Revisado: Marzo/86
** L.P.C. **
] █
```

A partir de acá ya puede comenzar a trabajar con su intérprete L.P.C.

IMPORTANTE: Nunca quite o coloque un cartridge (cartucho) con su computadora encendida.

CAPITULO 2

Cómo pedir a la computadora lo que debe hacer

Encienda la computadora y cargue el intérprete LPC según lo explicado en el Capítulo 1.

Para lograr que haga algo, Ud. debe darle órdenes que entienda, por ejemplo:

CALCULE 3 + 3

aquí le está pidiendo que realice la suma $3 + 3$ y represente el resultado en la pantalla.

Una orden como ésta, que le indica a la computadora que haga algo en el mismo momento, se llama **COMANDO**. En el caso de aquellas órdenes que conforman los programas, la ejecución no es inmediata y se las denomina **INSTRUCCIONES**.

Cómo ingresar este comando:

1. Primero escriba **CALCULE** y deje un espacio a continuación. Esto es fundamental para que el intérprete pueda reconocer la palabra **CALCULE**; siempre a continuación de un comando o una instrucción deberá dejar un espacio.

] CALCULE

2. A continuación pulse el 3, deberá aparecer el 3 en la pantalla. En este caso no es necesario dejar un espacio a continuación, Ud. puede hacerlo igualmente (con o sin espacio), el intérprete entenderá la operación o cálculo que Ud. desea efectuar.

3. Ahora pulse **+**. Este caracter se obtiene apretando simultáneamente la tecla **SHIFT** y la que contiene el **+** en su parte superior. Imagínese que su computadora es una máquina de escribir, la tecla **SHIFT** corresponde a las mayúsculas y mientras la mantenga apretada podrá hacer aparecer en pantalla al pulsar una tecla el caracter que figura en su parte superior (como el **+**).

4. Luego vuelva a pulsar el 3 y la pantalla quedará:

] CALCULE 3+3

5. A continuación pulse **ENTER**, esto significa que Ud. ha completado la orden que desea ingresar a la computadora. Recuerde esto pues es fundamental, hasta que Ud. no pulse **ENTER** la computadora no escuchará lo que quiere que haga.

Al pulsar **ENTER** la computadora leerá la instrucción y en este caso la ejecutará instantáneamente (pues es un comando) y en la pantalla aparecerá:

] CALCULE 3+3

LPC - 7

6 es la respuesta a lo que Ud. le pidió, por supuesto que la computadora no es sólo una calculadora y Ud. no la compró para esto, pero paciencia, estamos en el Capítulo 2 y recién comienza Ud. a comunicarse con ella.

Veremos qué pasa cuando nos equivocamos, pulse en su computadora lo siguiente:

```
] CALCULE 3++3
```

Ahora pulse la tecla ENTER, veamos qué pasa.

```
CALCULE 3++3  
**MAL ESCRITO**
```

El intérprete LPC leyó la orden y detectó que es incorrecto, por lo tanto le avisa que está mal con el mensaje MAL ESCRITO. Cada vez que Ud. reciba este mensaje será porque ha ingresado algo que el intérprete LPC no puede comprender. En estos casos deberá ingresar nuevamente la orden tratando de hacerlo en forma correcta.

Puede ocurrir que Ud. se dé cuenta de un error antes de pulsar ENTER y quiera corregirlo. En este caso deberá retroceder el CURSOR hasta el carácter a corregir.

Qué es el CURSOR? El cursor es un cuadrado brillante que aparece en la pantalla para indicarnos el lugar donde aparecerá la próxima letra o carácter que pulsemos en el teclado.

Si Ud. observó detenidamente la pantalla, cada vez que pulsaba una tecla el carácter correspondiente aparecía en pantalla y el CURSOR avanzaba una posición.

Supongamos que Ud. deseaba realizar la suma $3 + 33$ y se equivocó, pero se dio cuenta antes de pulsar ENTER. En este caso bastará con que vuelva el cursor sobre el + que está repetido y pulse un 3 para reemplazarlo. Ud. tiene en su teclado cuatro flechas (que seguramente ha utilizado en distintos juegos), cada vez que Ud. pulse ← el cursor retrocederá una posición, por el contrario si pulsa → avanzará una posición.

Ahora pulse ← hasta lograr que el cursor se coloque sobre el + que sobra (2 veces). Si se anima pruebe de pasarse y regresar con el → .

```
] CALCULE 3+3
```

Ahora pulse 3 y verá como se reemplaza el + por el 3 y el cursor avanza a la siguiente posición. Si Ud. no desea efectuar otro cambio no hace falta que siga moviendo el CURSOR, bastará con que pulse ENTER.

```
] CALCULE 3+33
```

] CALCULE 3+33
36

Supongamos que lo que Ud. deseaba hacer era 3+3 y para ello debe eliminar el + repetido vuelva el CURSOR con la flecha ← hasta el caracter que desea eliminar.

] CALCULE 3+**B**

Ahora pulse la tecla DEL y verá cómo automáticamente desaparece + y lo que está escrito en la pantalla se corre a la izquierda ocupando el lugar del + eliminado.

] CALCULE 3+**B**

Pulse ENTER si no desea hacer más modificaciones y obtendrá el resultado.

] CALCULE 3+33
6

Ahora escriba lo siguiente (sin pulsar ENTER):

] CALCULE 3+3**B**

Supongamos que Ud. realmente desea hacer 3 +43, debe entonces insertar un 4 entre el + y el 3, para ello bastará con retroceder el CURSOR con la tecla ← hasta el carácter delante del cual desea insertar algo.

] CALCULE 3+**B**

Ahora pulse la tecla INS. Luego el caracter que Ud. pulse se insertará delante del 3 y éste se desplazará hacia la derecha tantos lugares como caracteres Ud. pulse. Entonces pulse 4 y obtendrá

] CALCULE 3+4**B**

Luego pulse ENTER y tendrá el resultado

] CALCULE 3+43
46

CAPITULO 3

Uso de la computadora como una calculadora

Encienda la computadora y cargue el intérprete LPC según lo explicado en el Capítulo 1.

Ahora puede utilizarla como una calculadora, siguiendo las indicaciones del Capítulo 2.

Escriba CALCULE, a continuación la operación que quiera calcular y después ENTER.

Como Ud. ya se imaginaría la computadora no sólo suma, sino que también resta, multiplica (utilizando el asterisco * como signo "por", lo que constituye una práctica común en computación) y divide (utilizando la barra / como signo de dividir). También puede calcular la potencia de un número utilizando la operación ^:

CALCULE 2^3(y ENTER)

Obtendrá 8 como resultado (2 elevado a la potencia 3, o el cubo de 2)

+, -, *, / y ^ son operaciones y los números que manejan son sus operandos.

CALCULE 2+3

CALCULE 4-2

CALCULE 5*3

CALCULE 7/2

CALCULE 3^3

Cada una de estas combinaciones de operaciones y operandos se llama expresión, en este caso una expresión aritmética o numérica porque la respuesta es un número. En general siempre que la computadora esté esperando que Vd. le facilite un número, puede darle una expresión en su lugar y ella misma calculará la respuesta.

Cada expresión consta de una sola operación, cuando desee hacer cálculos complicados deberá efectuarlos paso a paso, operación por operación.

También podrá escribir números con decimales, utilizando el punto en vez de la coma.

CAPITULO 4

PROGRAMANDO SU COMPUTADORA

Al fin! Ha llegado el momento de comenzar a escribir un programa.

Deberá acostumbrarse, antes de comenzar a ingresar un nuevo programa, a ejecutar el comando LIMPIE. El objetivo de este comando es "limpiar" la memoria de su computadora de cualquier instrucción que pudiera quedar de un programa anterior. El uso del comando LIMPIE produce que su computadora se comporte como si recién comenzara a trabajar y Ud. acabara de cargar el intérprete LPC. Recuerde, con LIMPIE Ud. pierde todo lo que tenía en la memoria de su computadora, cuidado.

Para ejecutar el comando LIMPIE bastará con escribirlo y luego pulsar ENTER

```
] LIMPIE
```

Al ejecutar LIMPIE el intérprete LPC también borra la pantalla y la deja de la siguiente manera:

```
** LPC **  
]
```

Luego escriba

```
10 ESCRIBA "HOLA" (y ENTER)
```

y la pantalla quedará así:

```
** LPC **  
] 10 ESCRIBA "HOLA"  
]
```

Esto es diferente a lo que ocurría con el comando CALCULE, dado que la orden ESCRIBA tiene un número 10 delante, el intérprete LPC no la ejecuta inmediatamente, sino que la guarda en su memoria.

10 es el número de línea asignado a la instrucción, el número de línea sirve para indicar el orden o secuencia en que deben ejecutarse las instrucciones.

El número de línea puede ser de 1 a 32.000.

El saludo entre comillas ("HOLA") se denomina texto o literal alfanumérico. Entre las comillas Ud. puede colocar cualquier caracter (letra, número o signo) salvo las comillas. En un capítulo próximo veremos más sobre el manejo de textos, por ahora nos interesa saber que cuando le ordenemos a la computadora que ejecute esta instrucción, escribirá HOLA en la pantalla.

Ahora escriba

```
20 ESCRIBA "SOY SU COMPUTADORA" (y ENTER)
```

y en la pantalla tendremos:

```
** LPC **  
] 10 ESCRIBA "HOLA"  
] 20 ESCRIBA "SOY SU COMPUTADORA"  
] █
```

Un conjunto de estas instrucciones almacenadas en la memoria de la computadora se llama programa.

Para lograr que la computadora ejecute estas órdenes, deberá escribir ARRANQUE y pulsar ENTER.

ARRANQUE es un comando que le ordena a su computadora que ejecute el programa que tiene almacenado en memoria.

Luego en la pantalla tendremos:

```
** LPC **  
] 10 ESCRIBA "HOLA"  
] 20 ESCRIBA "SOY SU COMPUTADORA"  
] ARRANQUE  
  
HOLA  
SOY UNA COMPUTADORA  
** FIN **  
]
```

El mensaje **FIN** indica que se ha terminado de ejecutar correctamente el programa.

Al ejecutar el programa ha aparecido en la pantalla el saludo deseado, pero se mezcla con las órdenes que Ud. había escrito. Para lograr un saludo más prolijo, le or-

denaremos que antes de escribir el saludo borre la pantalla. Para ello Ud. deberá ingresar la instrucción

5 BORRE (y ENTER)

Fíjese que aunque esta sea la última instrucción que Ud. ingresa, por tener el número de sentencia 5 (el menor del programa) será la primera instrucción en ejecutarse.

Ahora escriba ARRANQUE, pulse ENTER y obtendrá la siguiente pantalla:

```
HOLA
SOY UNA COMPUTADORA
** FIN **
]
```

Ha logrado que su computadora emita un saludo mucho más prolijo, pero ha perdido la representación o listado del programa que está en la memoria.

Para lograr nuevamente en pantalla el listado de su programa bastará ejecutar el comando LISTE.

Para hacerlo escriba LISTE y pulse ENTER, con lo cual obtendrá la siguiente pantalla:

```
HOLA
SOY SU COMPUTADORA
** FIN **

] LISTE
] 5 BORRE
] 10 ESCRIBA "HOLA"
] 20 ESCRIBA "SOY SU COMPUTADORA"
] █
```

Este es el listado de su programa, supongamos que Ud. desea cambiar algo en su programa. Bastará con posicionar el CURSOR en la línea correspondiente y reemplazar o agregar lo que desee según lo visto en el capítulo 2.

Para subir el cursor utilice la tecla ↑ y para bajar una línea la tecla ↓.

Intentaremos ahora que la computadora escriba SOY LA COMPUTADORA en lugar de SOY SU COMPUTADORA; para hacerlo suba una línea con ↑ y luego avance con → hasta la palabra SU, con lo cual tendremos la siguiente pantalla.

```
HOLA
SOY SU COMPUTADORA
** FIN **

] LISTE
] 5BORRE
] 10 ESCRIBA "HOLA"
] 20 ESCRIBA "SOY SU COMPUTADORA"
]
```

Ahora pulse LA y ENTER, con lo cual habrá corregido la instrucción y el cursor pasará nuevamente a la siguiente línea.

Luego escriba ARRANQUE, pulse ENTER y la pantalla quedará

```
HOLA
SOY LA COMPUTADORA
** FIN **
]
```

Ahora en un momento de distracción escriba:

```
12 ESCRIBA "HOLA" (y ENTER)
```

Ud. no desea que la computadora escriba dos veces HOLA, pero es lo que ocurrirá si le indica que ARRANQUE,

Veamos el programa, escriba LISTE y pulse ENTER.

```
HOLA
SOY LA COMPUTADORA
** FIN **

] LISTE
] 5BORRE
] 10 ESCRIBA "HOLA"
] 12 ESCRIBA "HOLA"
] 20 ESCRIBA "SOY SU COMPUTADORA"
]
```

Para eliminar la instrucción 12 bastará con que Ud. escriba 12 y luego pulse ENTER.

12 (ENTER)

Ahora liste su programa, ya sabe como hacerlo, y verifique que la instrucción 12 no exista más.

Recuerde, para eliminar cualquier instrucción bastará con ingresar sólo su número. Si desea reemplazarla por otra, simplemente ingrese la nueva instrucción con el mismo número de la que desea reemplazar.

Antes de terminar el capítulo, fíjese en la última línea de la pantalla, allí verá escritos los comandos:

LISTE	LIMPIE	LEA	GRABE	ARRANQUE
-------	--------	-----	-------	----------

Esta es una pequeña ayuda Para que escriba menos. Cada comando escrito corresponde a una tecla: F1, F2, F3, F4 y F5 según la posición que ocupan.

Así, al pulsar F1 será el equivalente a escribir LISTE y pulsar ENTER, pruébelo, verá que fácil.

Ahora F5 equivale a ARRANQUE y pulsar ENTER. Pulse F5 y verá que se ejecuta su programa.

Al pulsar F2 aparecerá LIMPIE en la pantalla y Ud. deberá pulsar ENTER, esto es para su seguridad y evitar que pulse F2 por error y se borre la memoria perdiendo el programa que Ud. había ingresado.

Otra ayuda es la tecla ESC., que le permitirá detener momentáneamente un listado o la ejecución de un programa, luego pulsando cualquier tecla, continuarán.

Los otros dos comandos los veremos en próximos capítulos.

CAPITULO 5

VARIABLES NUMERICAS

De la misma manera que una calculadora con memoria puede almacenar un número o un resultado y recordarlo más tarde, su computadora puede almacenar gran cantidad de datos para su posterior utilización a través del empleo de la sentencia USE.

Suponga que Ud. quiere que la computadora recuerde el precio del kilo de duraznos y lo muestre por pantalla. Entonces escribiremos el siguiente programa:

```
10 BORRE          (ENTER)  Borrará la pantalla
20 USE DURAZNO = 34 (ENTER)  Almacenará el precio del durazno
```

La computadora (cuando se ejecute el programa) va a reservar un sitio en su memoria donde Ud. podrá almacenar un número (en este caso 34) y le dará a ese sitio un nombre (DURAZNO) para que Ud. pueda localizarlo y utilizar el número almacenado. Esta combinación de sitio o espacio de memoria y nombre se la denomina variable.

En el momento de ejecutar al programa (ARRANQUE), la computadora guarda el número 34 en el lugar de memoria definido (DURAZNO), lo que también puede decirse como ha asignado el valor 34 a la variable DURAZNO.

Sigamos con nuestro programa, digite ARRANQUE y luego ENTER. Verá que su computadora sólo borra la pantalla, lo que ocurre es que la asignación de un valor a una variable es un proceso interno y no se refleja en la pantalla.

Ahora, si quiere ver el contenido de la variable en pantalla simplemente digite:

```
30 ESCRIBA DURAZNO
```

Ejecute el programa y vea que pasa. Ha aparecido el número 34 en la pantalla.

Compliquemos un poco el programa, trataremos de hacer una "lista de precios de DURAZNOS", para ello digite:

```
25 ESCRIBA "EL KILO DE DURAZNO" (ENTER)
40 ESCRIBA "EL MEDIO KILO" (ENTER)
50 ESCRIBA DURAZNO/2 (ENTER)
60 ESCRIBA "EL CAJON - 4 KILOS" (ENTER)
70 ESCRIBA DURAZNO*4 (ENTER)
```

Ejecute el programa y vea que pasa. (ARRANQUE y ENTER)

En la pantalla aparecen las siguientes cosas:

```

EL KILO DE DURAZNO
34
EL MEDIO KILO
17
EL CAJON - 4 KILOS
136

** FIN **
]
```

Fíjese que la palabra DURAZNO figura tanto en la instrucción 25 como en la 30, 30 y 70, aunque el resultado es totalmente distinto. Cuando se encuentra entre comillas (") la computadora simplemente transcribe las letras, pero cuando encuentra su nombre va en busca del valor que tiene almacenado con dicho nombre.

¿Qué pasa si le pedimos que escriba el contenido de una variable que nunca se usó?

Pruébalo, primero "limpie" la memoria y elimine el programa anterior (LIMPIE y ENTER), luego digite:

```

10 BORRE (ENTER)
20 ESCRIBA NUMERO (ENTER)
```

Ahora ejecute el programa.

```

0
** FIN **
]
```

Ha aparecido un cero en la pantalla, esto se debe a que cuando no hemos asignado un valor a una variable el intérprete LPC le asigna automáticamente un valor cero (0).

CAPITULO 6

VARIABLES ALFANUMERICAS (Textos)

Como adelantáramos en el Capítulo 4, su computadora puede manejar textos. Ingrese y ejecute el siguiente programa:

```
10 BORRE
20 ESCRIBA "HOLA – SOY UNA COMPUTADORA"
```

Ya Ud. debe recordar que antes de ingresar un nuevo programa debe ejecutar el comando LIMPIE, que al finalizar cada línea debe pulsar ENTER y que para ejecutar el programa deberá especificar el comando ARRANQUE.

El saludo que aparece en pantalla y que Ud. especificó entre comillas es un texto. Entre ambas comillas Ud. puede colocar cualquier combinación de caracteres (letras, números o signos) excepto una comilla.

En el capítulo anterior vimos como se combinaban distintas instrucciones ESCRIBA con textos y con variables numéricas, con lo cual se facilita la comprensión de los resultados de sus programas.

También así como en el capítulo anterior, Ud. puede asignar textos a variables alfanuméricas o variables texto (estas denominaciones se utilizan indistintamente). En este caso para indicar que se trata de una variable texto, su nombre debe comenzar con el caracter @. Toda variable que comience con este caracter contendrá textos y por lo tanto no podrá ser utilizada dentro de expresiones numéricas.

La sentencia que utilizamos es la ASIGNE, con la cual podemos almacenar un texto en una variable.

Por ejemplo:

```
10 BORRE
20 ASIGNE @SALUDO = "HOLA"
30 ESCRIBA @SALUDO
```

Al ejecutar este programa Ud, verá como aparece la palabra HOLA en pantalla.

Agregue al programa las siguientes sentencias:

```
15 ASIGNE @YO = "– SOY UNA COMPUTADORA"
40 ESCRIBA @YO
```

Ejecute el programa, tendrá el siguiente resultado:

```
HOLA
— SOY UNA COMPUTADORA
** FIN **
]
```

Ud. puede combinar o "sumar" dos textos y colocarlos en una sola variable. Sigamos con el programa anterior:

Elimine la sentencia 40

```
] 40
```

Agregue la sentencia 25

```
] 25 ASIGNE @TODO = @SALUDO +@YO
```

Y por último reemplace la sentencia 30

```
] 30 ESCRIBA @TODO
```

Para ver cómo quedó su programa use el comando LISTE (F1) y fíjese que debería quedar así:

```
10 BORRE
15 ASIGNE @YO = "— SOY UNA COMPUTADORA"
20 ASIGNE @SALUDO = "HOLA"
25 ASIGNE @TODO = @SALUDO + @YO
30 ESCRIBA @TODO
```

Ejecute el programa (F5) y obtendrá:

```
HOLA — SOY UNA COMPUTADORA
** FIN **
```

Esta operación de "suma" se llama concatenación, podemos concatenar dos variables, una variable y un texto o dos textos.

Pruébelo, verá que es divertido.

CAPITULO 7

MAS PROGRAMACION DE LA COMPUTADORA

Recuerde, antes de comenzar con un nuevo programa deberá "limpiar" la memoria de su computadora, para ello utilice el comando LIMPIE (F2).

Ahora ingrese el siguiente programa:

```
10 BORRE
20 INGRESE DATO NUMERO
30 USE CUADRADO = NUMERO ^ 2
40 ESCRIBA NUMERO
50 ESCRIBA CUADRADO
```

Ejecute el programa (F5) y vea que pasa. Se ha borrado la pantalla (instrucción 10) y luego apareció un signo de interrogación (?).

?

Esto se debe a la instrucción 20 (INGRESE DATO) que ha hecho que la computadora esté en "espera de entrada de datos".

La computadora se queda a la espera de que Ud. le ingrese un dato, un número, y no continuará hasta tanto Ud. escriba el número y pulse ENTER.

Una vez que Ud. escriba el número y pulse ENTER, será similar a que hubiese colocado

```
20 USE NUMERO = al número que Ud. escribió
```

Por lo tanto a partir de este momento se ha asignado a la variable NUMERO el valor que Ud. escribió en pantalla. Entonces pruebe, digite un número (por ejemplo 12) y luego pulse ENTER. Inmediatamente aparecerá nuevamente el 12 en pantalla (instrucción 40) y luego 144 (el cuadrado de 12) aparecerá en la siguiente línea (instrucción 50).

```
? 12
12
144
** FIN **
]
```

Ya tiene Ud. un programa que le permite calcular el cuadrado de distintos números, pruebe de ejecutarlo (F5) varias veces ingresándole distintos valores.

Trataremos ahora de "mejorar" el programa haciéndolo un poco más claro incorporándole textos.

Agregue las siguientes instrucciones:

```
15 ESCRIBA "CALCULO DEL CUADRADO DE UN NUMERO"
33 BORRE
36 ESCRIBA "UD. INGRESO"
45 ESCRIBA "EL CUADRADO ES"
```

Ejecútelo y vea el resultado, luego liste (F1) el programa para analizarlo más detenidamente.

También la instrucción INGRESE DATO tiene la posibilidad de agregarle un texto para poder aclarar qué se está pidiendo cuando aparece el ? en pantalla. Para ello reemplace la instrucción 20.

```
20 INGRESE DATO "ESCRIBA UN NUMERO" NUMERO
```

Ejecute el programa y vea qué pasa

```
CALCULO DEL CUADRADO DE UN NUMERO
ESCRIBA UN NUMERO ?
```

Luego de escribir el número y pulsar ENTER, se borra la pantalla y aparece:

```
UD. INGRESO
12
EL CUADRADO ES
144
** FIN **
]
```

Así como puede ingresar un número para asignarlo a una Variable, puede también ingresar un texto ingresando la instrucción INGRESE TEXTO.

Por ejemplo ingrese este nuevo programa (no se olvide de "limpiar" la memoria -F2-):

```
10 BORRE
20 INGRESE TEXTO @LINEA
30 ESCRIBA @LINEA
```

Ejecútelo (F5) y obtendrá también un ? que le indica que la computadora está en "espera de entrada de datos".

Pruebe ingresando HOLA !! y fíjese como le contesta

```
? HOLA !!
HOLA !!
** FIN **
]
```

Completemos un poco más el programa, agregue lo siguiente:

```
22 BORRE
24 ASIGNE @LINEA = "UD. INGRESO: " + @LINEA
```

Al ejecutarlo (F5) obtendrá un signo de pregunta, ingrese el saludo y vea qué pasa:

```
? HOLA !!!
```

```
UD. INGRESO: HOLA!!!
** FIN **
]
```

Recuerda el programa del capítulo 5 que le daba la lista de precios del durazno? Fíjese como puede hacer un programa para tener los precios de cualquier fruta: (No se olvide de limpiar la memoria -F2-)

```

10 BORRE
20 ASIGNE @TITULO = "LISTA DE PRECIOS"
30 ESCRIBA @TITULO
40 INGRESE TEXTO "DE QUE FRUTA " @FRUTA
50 INGRESE DATO "CUANTO CUESTA EL KILO" PRECIO
60 BORRE
70 ESCRIBA @TITULO
80 ASIGNE @UNO = "EL KILO DE " + @FRUTA
90 ASIGNE @MEDIO = "EL MEDIO DE " + @FRUTA
100 ASIGNE @CAJON = "EL CAJON -4KG- DE"+ @FRUTA
110 ESCRIBA "      "
120 ESCRIBA @UNO
130 ESCRIBA PRECIO
140 ESCRIBA @MEDIO
150 ESCRIBA PRECIO/2
160 ESCRIBA @CAJON
170 ESCRIBA PRECIO*4

```

Ahora ejecútelo cuantas veces quiera asignando distintos nombres y precios.

Si Ud. asigna un PRECIO impar (por ejemplo 17) obtendrá como valor del MEDIO un número con decimales (8,5), supongamos que desea tomar sólo la parte entera (despreciar los decimales). Para ello deberá utilizar la instrucción ENTERO. Efectúe los siguientes cambios en el programa:

```

145 ENTERO VALOR = PRECIO/2
150 ESCRIBA VALOR

```


CAPITULO 8

SI...

Hasta el momento hemos visto todos los programas que mantienen una secuencia fija, las instrucciones se ejecutan una tras otra, siempre en el orden de su número de secuencia y sin dejar ninguna de lado.

Pero en la resolución de un problema por lo general se plantea la necesidad de tomar una decisión, o lo que es lo mismo ejecutar un grupo u otro de instrucciones dependiendo que se dé o no una condición.

Por ejemplo, trataremos de que la computadora pueda reconocer si un número ingresado es par o impar:

```
10 BORRE
20 INGRESE DATO "INGRESE UN NUMERO" NUMERO
30 ENTERO VALOR = NUMERO/2
40 SI NUMERO = VALOR * 2 HAGA
50 ESCRIBA "EL NUMERO ES PAR"
60 OTRO
70 ESCRIBA "EL NUMERO ES IMPAR"
80 FIN
```

La sentencia SI tiene la siguiente estructura:

```
SI condición HAGA
sentencia 1
OTRO
sentencia 2
FIN
```

La condición es una expresión que la computadora calcula y que puede tener dos posibilidades: verdadero o falso. En el ejemplo NUMERO podrá ser igual o no a la expresión $\text{VALOR} * 2$. Si el contenido de NUMERO es igual al doble del contenido de VALOR se dice que la condición es verdadera, caso contrario se dice que es falsa.

Esto la computadora recién lo puede saber cuando se ejecuta el programa y después de que se haya ingresado el número.

Si la condición resulta verdadera, la computadora ejecuta las sentencias 1, o sea toda sentencia que esté entre el HAGA y el OTRO. Dicho de otra manera, toda sentencia cuyo número de secuencia sea mayor al del SI y menor al del OTRO.

En nuestro ejemplo, si la condición es verdadera ejecuta la instrucción 50.

Si la condición resulta falsa, la computadora ejecuta las sentencias 2, o sea toda sentencia que esté entre el OTRO y el FIN. Dicho de otra manera, toda sentencia cuyo número de secuencia sea mayor al del OTRO y menor al del FIN. Aquí hay que tener en cuenta que el FIN no indica fin de programa, sino el punto donde se retoma la secuencia normal de instrucciones. Lo que está antes del FIN se ejecuta según el resultado de una condición, lo que sigue al FIN se ejecuta siempre.

Las sentencias 1 y 2 pueden ser cualquier instrucción, incluyendo a otro SI.

Las condiciones comparan a dos expresiones numéricas, a una expresión y una variable o un número, o una variable y otra o un número. Las relaciones que se pueden establecer son las siguientes:

= igual a
 > mayor que
 < menor que
 <= menor o igual a
 >= mayor o igual a
 <> distinto de

Como habíamos visto en la introducción de este manual, LPC es un lenguaje estructurado, y esto es porque responde a las estructuras que nosotros usamos normalmente para pensar, es decir a las estructuras lógicas, y precisamente esta estructura de SI condición HAGA/OTRO/FIN es una estructura que nosotros también usamos en nuestra vida diaria.

En general, cuando nos encontramos ante una decisión a tomar, evaluamos esta y podemos seguir tres caminos:

- Si la condición es verdadera tomar 1 o más acciones
- Si la condición es falsa tomar 1 o más acciones
- Si la condición "no es verdadera" no hacer nada.

Para esto último es que en LPC la instrucción OTRO no es obligatoria.

En nuestro ejemplo, puede ser que solo interesa informar cuando el número es par, entonces eliminamos las instrucciones 60 y 70. Recordemos que para hacerlo bastará con escribir 60, dar RETURN y luego escribir 70 y dar RETURN. Si ponemos LISTE. el programa quedará de la siguiente forma:

```
LISTE
** L.P.C. **
10 BORRE
20 INGRESE DATO "INGRESE UN NUMERO" NUMERO
30 ENTERO VALOR = NUMERO/2
40 SI NUMERO = VALOR * 2 HAGA
50 ESCRIBA "EL NUMERO ES PAR"
80 FIN
```

Escriba ARRANQUE (F5) e ingrese números pares e impares. Notará que cuando el número es impar sólo aparece la palabra ****FIN****, es decir ante este caso la computadora no hace nada.

Vimos que otro de los caminos es que ante una decisión (ya sea Verdadera o Falsa) podemos tomar más de una acción. En nuestro ejemplo nos puede interesar, además de informar que el número es par, dar el cuadrado del mismo. Entonces al programa original agregamos:

```
53 ESCRIBA "EL CUADRADO DEL NUMERO ES"
55 ESCRIBA NUMERO^2
```

Notará que 53 y 55 quedan intercalados entre la instrucción 51 y la instrucción OTRO, es decir que Si el número es par el programa ejecutará las instrucciones 50, 53 y 55.

Una condición la podemos unir en forma lógica con otra condición. Para unir las LPC cuenta con 2 conectores lógicos, es decir dos términos que unen ambas condiciones y le dan un sentido según cual usemos; estos son "Y" y "O". Lo descripto anteriormente puede parecer un poco extraño o difícil de entender, pero no hay que asustarse ya que es algo que usamos todos los días.

Veamos esto en nuestro ejemplo.

Supongamos que queremos que el programa avise si un número es par o impar, pero sólo los números mayores de 100; es decir en el caso par 'Que el número sea par Y mayor que 100'. Entonces nuestra línea 40 será:

```
40 SI NUMERO = VALOR * 2 Y NUMERO > 100 HAGA
```

Entonces sólo hará la línea 50 si se cumplen ambas condiciones. Pruebe de ejecutarlo, ingrese valores y vea qué pasa.

Ahora queremos que el programa avise si el número es par O igual a 3, entonces ejecutará la línea 50 en caso de que NUMERO sea par o si el número ingresado es tres.

```
40 SI NUMERO = VALOR * 2 O NUMERO = 3 HAGA
```

Ejecute el programa, ingrese valores (al menos un tres) y vea qué pasa.

La diferencia entre el Y y el O es que:

- En el Y el programa hará lo pedido sólo si las dos condiciones se cumplen.
- En el O basta que se cumpla una de las condiciones, o ambas.

En nuestro ejemplo de Y, si NUMERO es menor que 100, aunque sea par, no hará nada, y si es impar y mayor que 100 tampoco hará nada. En cambio, en el ejemplo de O si el número es 3, sabiendo que 3 es impar, el programa ejecutará la instrucción 50, lo mismo yo cualquier par, que sabemos es distinto de 3. Pruebe distintas combinaciones y comprobará prácticamente la diferencia.

Un último ejemplo:

Queremos saber si un número es mayor o igual que 500 y menor o igual que 599, ingrese un nuevo programa (no se olvide de "limpiar" el anterior).

```
10 INGRESE DATO "INGRESE UN NUMERO" VALOR
20 SI VALOR >= 500 Y VALOR <= 599 HAGA
30 ESCRIBA "EL NUMERO INGRESADO PERTENECE A LA QUINTA
   CENTENA"
40 FIN
```

Ejecute este Programa y analice el resultado.

Puede ser que deseamos que el programa se detenga cuando se cumpla alguna condición, por ejemplo, nos interesa detectar si antes de hacer Una división el número por el cual vamos a dividir sea distinto de Cero, ya que si es cero daría error.

Para detener un programa antes que llegue al final tenemos la sentencia PARE. Veamos con un ejemplo lo dicho:

```
10 INGRESE DATO N
20 INGRESE DATO M
30 SI N = 0 HAGA
40 ESCRIBA "DATO MAL INGRESADO"
50 PARE
60 OTRO
70 ESCRIBA "RESULTADO M/N"
80 ESCRIBA M/N
90 FIN
```

Ejecute varias veces el programa con distintos valores, luego ingrese un cero, en cuyo caso se detendrá apareciendo la leyenda DATO MAL INGRESADO según está programado

CAPITULO 9

CICLOS

En el capítulo anterior vimos como alterar el orden de secuencia de un programa, haciendo que un conjunto de instrucciones se ejecutaran, dependiendo de una condición. Ahora veremos como hacer para que un conjunto de instrucciones se repitan una cierta cantidad de veces, o hasta que se cumpla una condición.

Supongamos que queremos escribir ocho veces el nombre PEDRO. Según lo que sabemos hasta ahora tendríamos que hacer:

```
10 ESCRIBA "PEDRO"
20 ESCRIBA "PEDRO"
30 ESCRIBA "PEDRO"
40 ESCRIBA "PEDRO"
50 ESCRIBA "PEDRO"
60 ESCRIBA "PEDRO"
70 ESCRIBA "PEDRO"
80 ESCRIBA "PEDRO"
```

Digamos que no es muy largo de escribir aunque es aburrido, si quisiéramos ahora que escriba 100 veces ya el programa no sólo sería aburrido sino un tanto largo de escribir; para eso tenemos una instrucción que nos permite repetir una o más instrucciones; es la instrucción ECO. Su formato es:

```
nn ECO | número |
        | variable |
        Sentencias
        ...
        ...
mm FIN
```

Veamos como hacemos para escribir 100 veces PEDRO

```
10 ECO 100
20 ESCRIBA "PEDRO"
30 FIN
```

Este conjunto de instrucciones se denomina CICLO, ya que si lo pusiéramos en un diagrama sería como un ciclo cerrado. Veamos como haría la máquina:

1. Pongo mi contador en 1
2. Escribo "PEDRO"
3. Sumo 1 a mi contador
4. El contador es mayor que 100
 - 4.1 No; entonces repito nuevamente desde el paso 2.
 - 4.2 Si; entonces termino.

Es decir, está realizando un ciclo entre 2 y 4 hasta alcanzar el valor 100 en un contador especial.

Además de números nuestra instrucción ECO nos permite poner variables. Supongamos que nosotros queremos decidir en cada momento cuantas veces escribiremos PEDRO. Entonces podemos hacer:

```
5 INGRESE DATO "CUANTAS VECES ESCRIBO" N
10 ECO N
20 ESCRIBA "PEDRO"
30 FIN
```

Esta instrucción nos sirve también por ejemplo para obtener una tabla de multiplicar.

Ejemplo:

```
10 INGRESE DATO "QUE TABLA QUIERE" TAB
20 ECO 10
30 USE VALOR = VALOR +1
40 ESCRIBA TAB * VALOR
50 FIN
```

Veamos que pasa.

1° VALOR vale originalmente 0

2° TAB vale un número ingresado por teclado, por ejemplo 5

3° VALOR vale 1 (30)

luego escriba 5 x 1 es decir 5

se fijó si ya repitió 10 veces

como no repitió

VALOR vale 2

luego escribe 5 x 2 es decir 10

y así va haciendo

5x3, 5x4, 5x5 hasta 5 x 10 y nos escribió

la tabla de 5

Si ahora quisiéramos hacer Una tabla cualquiera hasta un valor cualquiera bastará agregar:

```
15 INGRESE DATO "HASTA QUE NUMERO" NUM
20 ECO NUM
```

Complicuemos las cosas un poco; supongamos que queramos ahora saber varias tablas de multiplicar, sin necesidad de dar ARRANQUE para cada una. Entonces podríamos poner una condición de fin del programa, por ejemplo la última tabla que vamos a ingresar será la tabla del 5. En realidad esto suena complicado pero para hacer esto basta con agregar las instrucciones

```
1 ECO HASTA TAB = 5
y
60 FIN
```

¿qué hace esto? Muy simple: hace un ECO (es decir repite) HASTA que la variable TAB valga 5, como originalmente vale cero entra en el CICLO ECO HASTA y pide el valor de TAB, ejecuta las instrucciones y al final vuelve a 1 y pregunta cuánto vale TAB, ahora vale lo que nosotros ingresamos, y el programa terminará cuando en tab ingresemos un 5.

Si ahora hacemos LISTE tendremos:

```
1 ECO HASTA TAB = 5
10 INGRESE DATO "QUE TABLA QUIERE" TAB
15 INGRESE DATO "HASTA QUE NUMERO" NUM
20 ECO NUM
30 USE VALOR = VALOR + 1
40 ESCRIBA TAB * VALOR
50 FIN
60 FIN
```

Notará que hay dos instrucciones FIN. Bueno, el FIN de 50 corresponde al ECO de 20 de y el FIN de 60 corresponde al ECO HASTA de 1.

Como se habrán podido dar cuenta el formato de ECO HASTA es:

```
nn ECO HASTA Condición
Instrucciones
.....
.....
mm FIN
```

Cuando nos introduzcamos un poco más en la programación, descubriremos que este ECO HASTA es una herramienta muy poderosa, pero como toda herramienta poderosa es un poco peligrosa si no la sabemos usar.

Supongamos el siguiente programa:

```
10 BORRE
20 USE B = 5
30 ECO HASTA A = 10
40 ESCRIBA B
50 USE B = B+1
60 FIN
```

¿Qué puede pasar con este programa? Tendría que terminar si A alguna vez toma el valor 10, pero A siempre valdrá cero, entonces este programa va a escribir los sucesivos valores de B hasta que apaguemos la máquina, es decir nunca va a finalizar.

Si Ud. ya dio ARRANQUE al programa, no importa, no es necesario que apague su computadora y vuelva a cargar el intérprete, bastará con apretar juntamente las teclas SHIFT y ? y el programa se detendrá apareciendo un cartel en pantalla que dice PAUSA. Bueno, ahora ya sabemos como detener un programa mientras se está ejecutando. Cualquier programa, aún cuando éste fuera a terminar normalmente lo podemos detener con SHIFT y ?. Esto se denomina INTERRUPCION desde el teclado, y es uno de los tipos de INTERRUPCION que pueden ocurrir en un programa.

Otras interrupciones pueden ser por error (por ejemplo intentar dividir por cero, o poner un número de fila o columna en el dibujo —ya lo veremos— mayor que los permitidos), porque se apague la luz y algunas otras que ahora no tiene sentido mencionar.

CAPITULO 10

EL JUEGO DE CARACTERES

En este capítulo vamos a empezar a ver una parte del LPC que nos permitirá ir creando dibujos con la computadora, para lo que necesitaremos más que nada imaginación, ya que la forma de hacerlo es muy simple. Supongamos que tenemos una hoja de papel, un lápiz y una persona que sabe hacer algunos dibujos o formas, muy bien, con ese lápiz y esa hoja, tenemos además una cartilla con los dibujos o formas que la persona conoce.

¿Cómo le diríamos a la persona lo que queremos? Muy simple. Lo primero que le diríamos sería DIBUJE.


Entonces la persona nos preguntaría ¿Qué quiere que dibuje? (recordemos que sólo sabe dibujar los dibujos de la cartilla). Nosotros le mostraríamos las figuras que queremos.

Luego nos preguntaría en qué lugar de la hoja queremos que los dibuje.

Con estos datos, la persona ya puede dibujar, porque sabe qué tiene que hacer, con qué elementos lo tiene que hacer y dónde lo tiene que hacer. Bueno, la computadora funciona exactamente igual.

Si por ejemplo le decimos:

```
10 BORRE
20 DIBUJE  EN FIL #4 COL #7
```

Verán que en la instrucción 20 le digo que DIBUJE, es decir Qué hacer, le digo  es decir **Con Qué** y le digo **EN FIL #4 COL #7**, es decir **Dónde**.

Las figuras que están en el teclado se ponen en la pantalla pulsando simultáneamente la tecla GRAPH y las teclas donde están marcados los dibujos para los dibujos de la derecha, y las teclas SHIFT y GRAPH para los que están a la izquierda. Es decir, si pulsamos GRAPH y ; pondrá ♥ pero si pulsamos SHIFT, GRAPH y ; pondrá ♦ ya que al tecla es: ♦♥

Para decirle donde dibujar debemos saber que la pantalla tienen 22 filas o líneas horizontales y 36 columnas o líneas verticales, además ni el valor de FIL #(fila) ni el de COL #(columna) deben valer cero.

Aparte de usar números en las filas y columnas podemos usar variables. Existe una formación predefinida por el LPC que no está en las teclas, y es un cuadrado del mismo color del fondo que llamamos PIP. Ahora veremos como podemos usar este PIP y el hecho de que las filas y columnas las podemos variar.

Veamos el siguiente ejemplo:

```
10 USE C = 5
20 ECO HASTA C > 28
```

```

30 DIBUJE PIP EN FIL #6 COL #C
40 DIBUJE ♦ EN FIL #6 COL #C +1
50 USE C = C + 1
60 FIN

```

¿Qué pasó?

Si lo hicieron habrán visto que aparece Eo la pantalla y se va moviendo hacia la derecha. ¿Cómo pudo ser esto? La instrucción 20 dice ECO HASTA $C > 28$. Como C vale 5 ingresa en el bloque ECO HASTA y ejecuta la instrucción 30, entonces en la fila 6 columna 5 dibuja un PIP, es decir un cuadradito del color del fondo; luego la 40 dibuja ♦ en la fila 6 columna 6 y la 50 hace que C valga $5 + 1$, es decir 6. Como C no es mayor que 28 vuelve a entrar al bloque y dibuja PIP en columna 6, entonces desaparece el dibujo ♦ recién hecho, pero ahora se vuelve a dibujar en fila 6 columna 7, porque C vale 7 y esto se va repitiendo hasta que C es mayor que 28; por eso da la sensación de que el dibujo se mueve, en realidad lo que hicimos fue ir moviéndolo, pero borrándolo" de su anterior posición. Si eliminamos la instrucción 30 veremos que aparecen varios ♦, uno a continuación del otro.

Existe otra forma mucho más fácil de dibujar muchos ♦, uno al lado del otro y es con un programa con sólo dos instrucciones.

```

10 BORRE
20 DIBUJE (23 DERECHA) ♦ EN FIL #6 COL #5

```

Verá que este programa hace exactamente lo mismo que el anterior sin la línea 30. Además de a la derecha podemos repetir una figura hacia abajo. Entonces tenemos como formato del DIBUJE.


1. DIBUJE Caracteres gráficos EN FIL #Valor - 1 COL #Valor - 2
2. DIBUJE (Valor - 1 DERECHA) Caracteres EN FIL #Valor - 2 COL #Valor - 3
3. DIBUJE (Valor - 1 ABAJO) Caracteres EN FIL #Valor - 2 COL #Valor - 3

Donde Valor - 1 es la cantidad de veces que queremos que se repita el dibujo, Valor - 2 el renglón donde aparecerá y Valor - 3 la posición (o columna) dentro del renglón.

Además debemos decir que podemos poner hasta 10 caracteres gráficos en cada instrucción DIBUJE.

Veamos un último ejemplo:

```

10 USE C = 5
20 DIBUJE ♦ EN FIL #3 COL #25
30 DIBUJE (30 DERECHA) = EN FIL #18 COL #1
40 ECO HASTA C > 25
50 DIBUJE (5 DERECHA) PIP EN FIL #17 COL #C
60 DIBUJE  EN FIL #17 COL #C + 1
70 USE C = C + 1
80 FIN

```

CAPITULO 11

GRAFICOS

En el capítulo anterior vimos cómo hacer para que la computadora nos dibuje las figuras o formas que la máquina ya tiene predefinidas, pero nos puede interesar crear nuestras propias figuras. es decir ahora tenemos papel y una persona que nos puede dibujar cualquier cosa que le pidamos, pero esa persona no conoce ninguna forma y nosotros debemos indicarle cuál es la forma que queremos que dibuje; entonces le debemos pedir antes que dibuje, que FORME la figura que nosotros le vamos a indicar.

Primero le pedimos que FORME una casa, entonces debemos enseñarle cómo es una casa, ya que dijimos que no conocía ninguna figura.

A la computadora le decimos:

10 FORME CASA
y le damos ARRANQUE

Cuando hagamos esto se borrará la pantalla y aparecerá un cuadrado de 8 filas por 8 columnas, en donde la computadora nos pide que le mostremos como es una casa. Para dibujar tenemos tres teclas:

1: Pinta el punto donde está el cursor y avanza
0: No pinta y avanza
BS: Retrocede para borrar

Por ejemplo pulsamos 00011000 y en la primera fila tendremos



luego 00111100 y tendremos



y así vamos poniendo 1 y 0 hasta completar el dibujo.

Pulse la tecla RETURN para terminar y nuevamente ejecute el programa (ARRANQUE —F5—). Ahora es usted el dibujante, agudice su imaginación y trate de crear una linda CASA. Cuando termine pulse RETURN, si su dibujo llega a completar todo el cuadrado no le hará falta.

Una vez hecho esto, la computadora ya sabe como dibujar una casa, porque nosotros le enseñemos.

Ahora quisiéramos que la dibuje. Antes es necesario decir que la computadora en la pantalla trabaja de dos formas o modos:

1. MODO TEXTO
2. MODO GRAFICO

En MODO TEXTO es como está normalmente y es como la teníamos hasta ahora y en MODO GRAFICO debemos ponerla para que haga los dibujos que nosotros formamos. Para esto es la instrucción GRAFICOS; con esto le indicamos a la computadora que cambie a modo GRAFICO, ya que lo que viene a continuación son dibujos. Cuando el programa termina de ejecutar pasa automáticamente a MODO TEXTO, además pasa a MODO TEXTO cuando usamos la instrucción FIN—GRAFICOS.

Veremos un primer intento de que dibuje la casa.

```
10 GRAFICOS
20 DIBUJE CASA EN FIL +5 COL +10
y ARRANQUE
```

¿Qué pasó?

La casa apareció, desapareció rápidamente y nos salió un mensaje que dice:

```
FALTA FIN—GRAFICOS
```

¿Porqué?

Dijimos que cuando el programa termina pasa sólo a modo TEXTO, entonces nuestros dibujos que están en MODO GRAFICO desaparecen y el mensaje es porque como dijimos que éste es un lenguaje estructurado, y una estructura es la parte de dibujos, entonces la computadora nos avisa que ese bloque o estructura no está cerrado, en este caso no es grave, pero si nosotros siguiéramos con nuestro programa y tuviéramos instrucciones tales como ESCRIBA o INGRESE DATO, entonces sería grave porque no los ejecutaría, por eso el aviso.

Ahora veremos un pequeño truco para que el dibujo no desaparezca.

```
10 GRAFICOS
20 DIBUJE CASA EN FIL #5 COL #10
30 ECO 30
40 FIN
50 FIN—GRAFICOS
```

Ahora la casa se vio durante un tiempo y luego volvió a MODO-TEXTO, y el mensaje no apareció.

Otra forma sería realizando un ciclo que no termine nunca y detener el programa con SHIFT y ?

```
10 GRAFICOS
20 DIBUJE CASA EN FIL #5 COL #10
30 ECO HASTA A = 5
40 FIN
50 FIN-GRAFICOS
```

(No se olvide de SHIFT ? para poder salir del programa)

Por último veremos como dibujar un auto y hacer que se mueva, Pongamos:

```
1 FORME AUTO
25 DIBUJE (15 DERECHA) AUTO EN FIL #5 COL #11
   y ARRANQUE
```

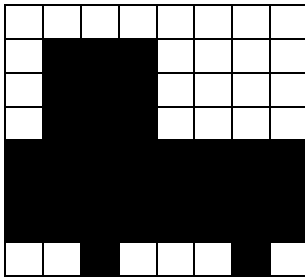
Aparece de nuevo el cuadrado de 8 x 8 y ahora nos pide que le enseñemos como es un auto.

Bueno, ahora jueguen con su imaginación y dibujen un auto.

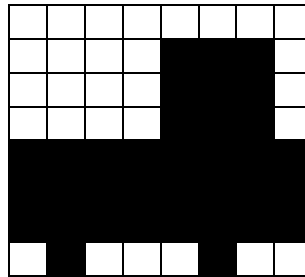
Una vez terminado el programa si queremos volver a ejecutarlo sin tener que dibujar de nuevo el AUTO, debemos eliminar la instrucción 1, recordemos que para esto bastará con tipear 1 y dar RETURN. Seguramente ya habrán pensado que si queremos que nuestro dibujo se mueva hacia abajo, en vez de DERECHA ponemos ABAJO. Por supuesto que también podemos hacer que se mueva como en el capítulo anterior, haciendo variar la fila o la columna, con esta forma (Derecha o Abajo) usamos menos instrucciones pero no nos permite hacer otras cosas mientras se mueve, como por ejemplo, que dos figuras se muevan al mismo tiempo o, como veremos en el próximo capítulo, que emita sonidos mientras avanza.

Ahora hagamos un choque de dos autos.

```
10 FORME AUTO1
20 FORME AUTO2
   ARRANQUE
```



AUTO1



AUTO2

Luego:

```

10 USE C1 = 5
20 USE C2 = 35
30 GRAFICOS
40 ECO 15
50 DIBUJE AUTO1 EN FIL #10 COL #C1
60 DIBUJE AUTO2 EN FIL #10 COL #C2
70 USE C1 = C1 +1
80 USE C2 = C2 - 1
90 FIN
100 ECO 20
110 FIN
120 FIN-GRAFICOS

```

CAPITULO 12

COLOR Y SONIDO

Ahora veremos como adornar un poco nuestros programas, sobre todo los que hacen dibujos, agregando el color y el sonido.

COLOR:

Nosotros podemos colorear tanto las formaciones como el fondo de la pantalla. Veamos el siguiente programa:

```
10 FORME PERRO
20 FORME CASA
ARRANQUE
```

Aquí los dejo jugar un poco con la imaginación. Luego hacemos:

```
10 (para borrar los FORME, cuando tengamos el
   PERRO y la CASA listos)
20
40 GRAFICOS
50 DIBUJE CASA EN FIL #6 COL #28
60 DIBUJE (25 DERECHA) PERRO EN FIL #6 COL #3
70 ECO 30
80 FIN
90 FIN-GRAFICOS
ARRANQUE
```

Ahora probemos a agregar las siguientes instrucciones

```
10 COLOR FONDO AMARILLO
20 COLOR PERRO NEGRO
30 COLOR CASA AZUL
y ARRANQUE
```

¿Qué pasó? El fondo dejó de ser azul y la casa y el perro dejaron de ser blancos. ¿De qué color son? Buenos, ni más ni menos que de los colores descriptos en 10, 20, 30; así de fácil. Entonces para colorear tenemos dos instrucciones:

```
COLOR FONDO color
COLOR figura color
```

donde figura es cualquiera de las formaciones hechas por nosotros y color son algunos de los siguientes:

NEGRO	VERDE	MAGENTA
AZUL	AMARILLO	GRIS
ROJO	CYAN	BLANCO

Lo único que debemos cuidar es que COLOR FONDO debe estar fuera del bloque GRAFICOS ya que si está adentro no funciona. El color de los gráficos o formaciones puede estar en cualquier parte, e incluso ser cambiado en cualquier momento. Si movemos el perro variando la fila y columna podemos hacer:

Veamos el programa anterior:

```

20 y < RETURN >
30 y < RETURN >

5 USEC = 3
55 ECO 25
60 DIBUJE PERRO EN FIL #6 COL #C
67 USE C = C + 1
68 FIN

```

Hasta aquí sólo cambiamos la forma de mover el perro, el resultado es igual que antes. Se dieron cuenta que el perro sigue negro y la casa azul. Si sacan la 10, verán que el FONDO no sigue amarillo.

Conclusión: El color del fondo debemos inicializarlo cada vez, sino lo pone azul, pero una vez que 'coloreamos' una figura, guarda ese color.

Ahora agreguemos

```

56 SI C = 13 HAGA
57 COLOR PERRO ROJO
59 FIN

```

¿Qué pasó? El perro empezó en negro y a mitad del recorrido cambió a rojo; si volvemos a dar ARRANQUE el perro empezará en rojo, ya que fue el último color con que lo pintamos, entonces agreguemos:

```

20 COLOR PERRO NEGRO

```

y ahora veremos que cada vez que damos ARRANQUE, el perro comienza en negro y cambia a rojo.

Si quisiéramos hacer esto mismo en modo texto tenemos también dos instrucciones. Una es la ya conocida:

```
COLOR FONDO color
```

y la otra

```
COLOR LETRAS color
```

Con esto cambiamos el color de todo lo que no sea fondo, es decir de las letras.

Veamos el siguiente ejemplo:

```
10 BORRE
20 ECO 10
40 ESCRIBA "A B C D E F G H I J K"
50 FIN
60 COLOR FONDO AMARILLO
70 COLOR LETRAS AZUL
80 ECO 30
90 FIN
100 COLOR LETRAS NEGRO
110 ECO 30
120 FIN
130 COLOR LETRAS ROJO
140 ECO 30
150 FIN
160 COLOR LETRAS BLANCO
170 COLOR FONDO AZUL
```

SONIDO:

Volvamos al primer programa en el que el perro lo movíamos variando C, es decir:

```
5 USE C = 3
10 COLOR FONDO AMARILLO
20 COLOR PERRO NEGRO
40 GRAFICOS
50 DIBUJE CASA EN FIL #6 COL #28
55 ECO 25
56 SI C = 13 HAGA
57 COLOR PERRO ROJO
59 FIN
60 DIBUJE PERRO EN FIL #6 COL #C
```

```

67 USE C = C +1
68 FIN
70 ECO 30
80 FIN
90 FIN-GRAFICOS

```

y agreguemos

```

58 SONIDO MI

```

¿Qué escuchamos? Cuando el perro cambia de color se escucha un MI, ahora queremos que se escuche una pequeña melodía, hacemos:

```

58 SONIDO MI MI SOL FA SOL

```

y escuchamos una pequeña melodía cuando cambia de color, entonces para crear sonidos bastará con poner instrucciones del tipo:

```

SONIDO nota [nota nota ... ..]

```

Esto quiere decir que es obligatorio poner SONIDO y al menos una nota musical, y si queremos podemos agregar hasta 255 notas en la misma instrucción. Podríamos poner:

```

63 SONIDO FA

```

entonces cada vez que el perro se mueve una posición, se escuchará un FA y con:

```

85 SONIDO MI MI MI FA FA SOL LA LA FA SOL SOL RE
    MI

```

escucharemos todas estas notas cuando el perro llegue hasta la casa.

Pruebe y diviértase creando dibujos y melodías con su computadora y el LPC.

CAPITULO 13

SUBROUTINAS

Al llegar a este capítulo es muy posible que Ud. ya haya hecho solo algún programa; si no lo hizo, no importa, ya profundizaremos en eso un poco más adelante, en todo caso si lo hizo puede ser que en alguno haya notado que tuvo que repetir instrucciones más de una vez. Veamos un ejemplo del capítulo anterior:

```
10 BORRE
20 ECO 10
40 ESCRIBA "A B C D E F G H I J"
50 FIN
60 COLOR FONDO AMARILLO
70 COLOR LETRAS AZUL
80 ECO 30
90 FIN
100 COLOR LETRAS NEGRO
110 ECO 30
120 FIN
130 COLOR LETRAS ROJO
140 ECO 30
150 FIN
160 COLOR LETRAS BLANCO
170 COLOR FONDO AZUL
```

Acá
ECO 30
FIN

Lo repetimos tres veces en el programa. Si pudiéramos escribirlo una sola vez y cuando lo necesitamos vamos a esa parte del programa y terminada esa tarea volvemos, podríamos ahorrar instrucciones y espacios de memoria. Veamos cómo puede quedar esto.

```
10 BORRE
20 ECO 10
40 ESCRIBA "A B C D E F G H I J"
50 FIN
60 COLOR FONDO AMARILLO
70 COLOR LETRAS AZUL
80 EJECUTE ESPERA
90 COLOR LETRAS NEGRO
100 EJECUTE ESPERA
110 COLOR LETRAS ROJO
120 EJECUTE ESPERA
130 COLOR LETRAS BLANCO
```

```

140 COLOR FONDO AZUL
150 PARE
160 RUTINA ESPERA
170 ECO 30
180 FIN
190 VUELVA

```

Aquí es evidente que nuestro programa fue más largo. Vamos a explicar que hicimos y a dar un ejemplo en el que una rutina es conveniente.

¿Qué es una rutina? Es una parte del programa que escribimos al final, tiene un nombre para reconocerlo y una instrucción que nos dice donde termina la rutina. En el caso del ejemplo la rutina que tiene el nombre ESPERA comienza en la línea 160 y la instrucción que indica donde termina es VUELVA en la 190.

¿Cómo la llamamos o invocamos? Muy sencillo, con la instrucción EJECUTE nombre-rutina.

Lo que hace esta instrucción es buscar en el programa una instrucción RUTINA nombre-rutina y empezar a ejecutar desde esa línea, cuando encuentra la instrucción VUELVA, vuelve al punto donde fue llamada y continúa con la instrucción siguiente. En nuestro ejemplo, cuando llega a la línea 80 va hasta la línea 160, ejecuta 170 y 180 y cuando se encuentra con 190 VUELVA, vuelve a continuación de la línea 80 y continúa ejecutando la 90.

Veamos las tres instrucciones que nos permiten ejecutar rutinas.

- EJECUTE nombre-rutina

Busca una rutina cuyo nombre es nombre-rutina y empieza a ejecutar desde allí.

- RUTINA nombre-rutina

Es la instrucción que encabeza una rutina y que puede ser invocada desde cualquier punto del programa, incluso desde adentro de la misma RUTINA.

- VUELVA es la instrucción que marca el fin de la rutina y hace que el programa vuelva a continuación de la instrucción EJECUTE desde donde salió o fue invocada la subrutina.

Decíamos que en el caso del ejemplo teníamos menos líneas sin rutina. Que con rutina, pero pensemos en el siguiente caso:

Supongamos que en el programa sin rutina agregamos las siguientes líneas:

```

85 USE C = C+1
87 ESCRIBA C
89 ESCRIBA C*2
115 USE C = C+1

```

```
117 ESCRIBA C
119 ESCRIBA C*2
135 USE C = C+1
137 ESCRIBA C
139 ESCRIBA C*2
```

Aumentamos nueve líneas, es decir, tenemos siete líneas más que en el programa con rutinas; si quisiéramos que los dos programas hicieran lo mismo, en el último bastaría hacer:

```
175 USE C = C+1
177 ESCRIBA C
179 ESCRIBA C*2
```

Sólo agregamos tres líneas, es decir que ahora tiene cuatro líneas menos que el programa sin rutinas. Entonces podemos llegar a la conclusión que las rutinas son útiles siempre que acortemos la cantidad de líneas de programa, como en este último ejemplo. Hay otro caso en que también es muy útil.

Veamos el siguiente caso:

```
10 USE A = 0
20 EJECUTE RUT1
30 ESCRIBA A
40 ESCRIBA "LINEA 40"
50 PARE
60 RUTINA RUT1
70 USE A = A+1
80 SI A < 3 HAGA
90 EJECUTE RUT1
100 FIN
110 EJECUTE RUT2
120 VUELVA
130 RUTINA RUT2
140 ESCRIBA "RUTINA 2"
150 VUELVA
```

¿Qué hay acá de nuevo?

RUT1 la llamamos desde 20, es decir del cuerpo principal del programa, que termina en 50 PARE (cuando se usan rutinas, es necesario usar esta instrucción para separar lo que es un programa principal de lo que son rutinas), y la llamamos en 90, que es dentro del mismo bloque de RUT1, y a RUT2 la llamamos desde RUT1. Es decir, podemos llamar a cualquier rutina desde el programa principal, desde otra rutina y desde sí misma (esto

último se lo conoce como recursividad). Notamos además que para salir del "ciclo recursivo" fijamos una condición, ya que si no existiera esta condición la RUTINA RUT1, se estaría "autollamando" siempre, sin volver nunca al programa principal. Por supuesto que llegaría un momento en que cancelaría por no tener más memoria, para guardar el número de línea de la instrucción a la que debe regresar.

Vamos a ver un último ejemplo en el que el mismo programa nos muestra por dónde pasa.

```
10 ESCRIBA "LINEA 10"  
20 EJECUTE RUT  
30 ESCRIBA "LINEA 30"  
40 PARE  
50 RUTINA RUT  
60 ESCRIBA "LINEA 60"  
70 ESCRIBA "LINEA 70"  
80 VUELVA
```

Como última reflexión diremos que una rutina nos ayuda a acortar un programa, pero sabiéndola usar. Además se puede convertir en una herramienta muy poderosa, sobre todo si sabemos usar la recursividad, o si la usamos para crear estructuras lógicas y hacer los programas más fáciles de entender y de alterar en el caso que sea necesario. No se preocupe si ahora no queda claro, con el tiempo y los programas ya lo irá entendiendo solo.

CAPITULO 14

HACIENDO FUNCIONAR SUS PROGRAMAS

Cuando uno sube a un coche para manejar por primera vez y le explican como funciona cada cosa por separado, en general es muy simple de entender; ahora, cuando debemos coordinar todo para que este se mueva, pare, doble, retroceda y además podamos salir de un punto y llegar adonde nosotros queramos, ya el problema no es tan simple. En programación sucede algo similar: nosotros ya hemos visto como funciona cada instrucción, pero ahora es necesario saber armarlas y Coordinarlas para que la computadora haga lo que nosotros queremos, es decir que llegue a entregar los resultados correctos a partir de datos que nosotros proporcionamos, ya sea con sentencias USE o con sentencias INGRESE.

En general tenemos tres tipos de errores en un programa:

- 1) De escritura o sintaxis
- 2) De ejecución
- 3) De lógica

El primero es muy fácil de solucionar, ya que cuando ingresamos una instrucción la máquina nos responde MAL ESCRITO; si ésta no es correcta, bastará con fijarse que es lo que escribimos mal para solucionarlo.

El segundo caso si bien es fácil ya es un poco más complicado, ya que puede ser que el error no esté exactamente en la última instrucción ejecutada sino en otra parte del programa, y este "se arrastró" hasta producir el error. Veamos este ejemplo para ilustrar mejor lo último.

```
5 BORRE
10 USE A = 3
20 ECO 38
30 DIBUJE EN FIL #3 COL #A
40 USE A = A+1
50 FIN
```

Cuando A supere 36 aparecerá un mensaje que dice

```
MAL VALOR EN 30
```

El error se produjo en la línea 30, pero su causa no está allí sino en el ECO 38 que hace que A supere ese valor, entonces la línea que debemos corregir es 20 poniendo por ejemplo 20 ECO 30.

El tercer tipo de error ya es más difícil, ya que depende de como hayamos estructurado nuestro programa; es común al principio, que si bien el programa funciona y termina bien,

no hace lo que nosotros queríamos, y es porque nosotros no le hemos indicado correctamente a la computadora lo que debe hacer. Para esto último es que existe una herramienta llamada DIAGRAMACION y que nos sirve para visualizar mejor que es lo que queremos y para corregir errores de lógica.

Existen tres tipos de estructuras o bloques:

1) Secuencia de instrucciones

Cada bloque representa una instrucción o un conjunto de instrucciones.

(Puede ser algo general)

Ejemplo: yendo de lo general a lo particular

SUMAR 2 NUMEROS



INGRESAR 1er. NRO.
INGRESAR 2do. NRO.
REALIZAR SUMA
DAR RESULTADO

Problema general

Desarrollo del
problema general

Esto mismo en un programa LPC sería:

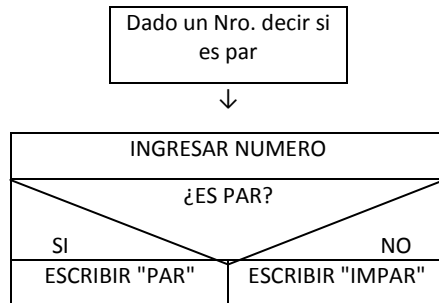
10	INGRESE DATO A
20	INGRESE DATO B
30	USE C = A+B
40	ESCRIBA C

2) Bloqueo condicional

CONDICION	
SI	NO
Secuencia 1	Secuencia 2

Se fija Una condición; por verdadero. sigue por la secuencia 1, y por falso ejecuta la secuencia 2.

Ejemplo

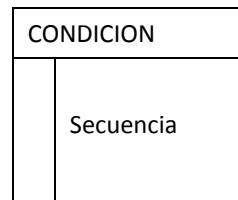


En LPC

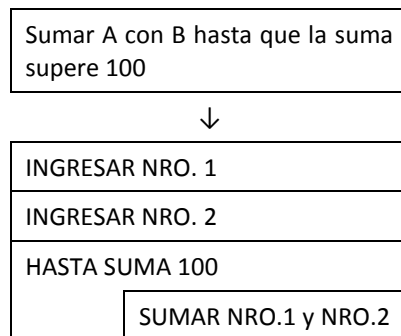
10	INGRESE DATO A		
20	ENTERO $V = A/2$		
	$A = V * 2$		
30	SI		NO
40	ESCRIBIR "PAR"	50	OTRO
		60	ESCRIBA "IMPAR"
70	FIN		

3) Bloque de ciclo condicional

Si se cumple la condición ejecuta la secuencia.
Una vez finalizada vuelve a chequear la condición y si se cumple la ejecuta y así, hasta que se deje de cumplir.



Ejemplo:



En LPC

10	INGRESE DATO A						
20	INGRESE DATO B						
30	USE $A = A + B$						
40	ECO HASTA $A > 100$						
	<table> <tr> <td>50</td><td>ESCRIBA A</td></tr> <tr> <td>60</td><td>USE $A = A + B$</td></tr> <tr> <td>70</td><td>FIN</td></tr> </table>	50	ESCRIBA A	60	USE $A = A + B$	70	FIN
50	ESCRIBA A						
60	USE $A = A + B$						
70	FIN						

Note que el tercer bloque del segundo diagrama fue a su vez desarrollado en cuatro instrucciones.

Con estos tres elementos Ud. debería ser capaz de diagramar sus programas LPC. Intente diagramar algunos de los ejemplos dados en capítulos anteriores.

En general si el programa no es muy simple conviene primero diagramar; primero el problema general, luego ir desglosando en bloques hasta llegar a nivel de detalle, es decir de instrucciones LPC. Esto reducirá bastante la posibilidad de que existan errores de lógica. Si después de diagramar y pasar a LPC surgen problemas es recomendable tomar lápiz y papel e ir escribiendo instrucción por instrucción, como si uno fuera la computadora, lo que va pasando; sobre todo anotando los valores que van asumiendo las variables.

Con estos elementos Ud. estaría en condiciones de generar sus programas y corregirlos.

Suerte y a divertirse !!!

CAPITULO 15

ALMACENAMIENTO EN CINTA MAGNETICA

Seguramente ya habrá descubierto lo limitado que resulta el computador por el hecho de que cada vez que lo apaga pierde los programas ingresados por Ud., pero esto también es fácil de solucionar. Bastará con que Ud. guarde sus programas LPC en un cassette.

Lo primero que hay que hacer es tener conectado el grabador. La ficha de cinco patitas (también conocida como DIN) va en su computadora, la ficha blanca del otro extremo (mini-jack) va en la salida externa del grabador y la roja en la conexión para micrófono. Una vez conectado, deberá tener en memoria el programa que Ud. desea grabar (si no lo tiene escriba cualquier programa como prueba).

El programa deberá tener un nombre que será el que debe usar cuando lo lea para volver a ponerlo en memoria. Por ejemplo lo llamaremos PRUEBA.

Entonces escriba

```
] GRABE CAS: PRUEBA
```

CAS: Es para indicar que el mecanismo de grabación es un cassette. Si fuera un Quick Disk sería QD0: y si fuera una disketera sería A: o B: según de qué disketera se trate.

Luego ponga el grabador en posición de grabar y por último oprima RETURN. Verá que el cursor █ desaparece y luego de un tiempo vuelve a aparecer; cuando esto suceda su programa LPC ya estará a salvo en un cassette.

Ahora veremos cómo podemos traerlo de nuevo a la memoria.

Ejecute el comando LIMPIE, así nos aseguramos que no hay nada; para estar seguro que no quedó nada ponga LISTE (o F1) y verá que aparece:

```
] LISTE
** LPC **
```

█

Ahora rebobine el cassette. Escriba

```
] LEA CAS: PRUEBA
```

Ponga el cassette en Play, es decir como para escuchar música y luego pulse RETURN. Nuevamente el cursor va a desaparecer y después de un tiempo volverá a aparecer.

Ejecute el comando LISTE y verá como su programa vuelve a estar en la memoria, y al dar ARRANQUE lo ejecuta tal cual como cuando Ud. lo escribió por primera vez.

En un cassette podemos guardar todos los programas que nosotros queramos, limitados sólo por el espacio del cassette. Luego Ud. podrá traerlos a memoria con sólo invocar el comando LEA y el nombre, además de indicar el mecanismo en el que se lee.

Resumiendo:

Para grabar el formato es:

GRABE nombre mecanismo: nombre programa
 nombre mecanismo es alguno de los siguientes:

CAS: cassette

QDO: Quick Disk

A o B: Diskette (según el caso)

nombre programa es cualquier nombre que Ud. desee y que luego deberá usar exactamente cuando lo lea.

Para leer es:

LEA nombre-mecanismo: nombre programa

Ud. puede hacer la siguiente prueba:

Haga un programa y ejecute

GRABE CAS: PGMA1

luego haga un segundo programa y ejecute

GRABE CAS: PGMA2

Por supuesto que deberá estar en el cassette sin tapar al primer programa.

Y por último haga un tercer programa y ejecute

GRABE CAS: PGMA3

Luego ejecute LIMPIE, rebobine el cassette y ejecute

LEA CAS: PGMA2

Verá luego de listarlo que el computador leerá ese programa, sin tomar en cuenta lo que hay en el resto del cassette, luego haga lo mismo con PGMA1 y PGMA3.

CAPITULO 16

USANDO LA IMPRESORA

Si Ud. tiene una impresora es posible que haya pensado ¿por qué no puedo poner lo mismo que tengo en pantalla, en el papel de la impresora? Efectivamente, puede.

Para esto tenemos una instrucción y un comando.

Escriba un programa, puede ser alguno de los ejemplos de capítulos anteriores. Luego escriba

] LISTE

Esto ya no es novedoso para nosotros. Ya sabemos que nos va a aparecer por pantalla, es decir en el televisor, el listado del programa; supongamos que queremos tenerlo escrito, para corregirlo o simplemente para tenerlo. Una solución es tomar lápiz y papel y copiar línea por línea. Esto mismo lo puede hacer la computadora.

Teniendo la impresora conectada y encendida (y por supuesto con papel) escriba:

] IMPRIMA y oprima RETURN

¿Qué pasó? El programa fue escrito en la impresora.

Bueno, con esto ya nos hemos dado cuenta que el comando IMPRIMA nos saca el programa de memoria y lo escribe por medio de la impresora. También nos puede interesar que alguno de los datos de un programa queden en un papel para luego corroborar resultados y ver por ejemplo si el programa funciona correctamente.

Veamos el siguiente programa:

```
10 INGRESE TEXTO "CUAL ES SU NOMBRE" @NOMBRE
20 INGRESE DATO "CUAL ES SU EDAD" EDAD
30 INGRESE DATO "QUE ANIO ES" ANIO
40 ASIGNE @NOMBRE = "NOMBRE: " + @NOMBRE
50 IMPRIMIR @NOMBRE
60 IMPRIMIR "UD. NACIO EN"
70 IMPRIMIR ANIO - EDAD
```

y ARRANQUE

```

¿CUAL SU NOMBRE?
PEDRO
¿CUAL ES SU EDAD?
20
¿QUE ANIO ES?
1984

```

Luego verá que en la pantalla aparece

```

** FIN **

```

¿Qué pasó con los resultados? Verá que en la impresora está escrito:

```

NOMBRE: PEDRO
UD. NACIO EN
1964

```

Seguramente ya descubrió que con la instrucción IMPRIMIR se puede hacer lo mismo que con ESCRIBA pero en vez de salir en la pantalla aparece en la impresora. Pruebe con los ejemplos anteriores reemplazando los ESCRIBA por IMPRIMIR.

Ejemplo:

```

10 INGRESE DATO A
20 SI A > 100 HAGA
30 ESCRIBA "A ES MAYOR QUE 100"
40 OTRO
50 IMPRIMIR "A ES MENOR O IGUAL A 100" >
60 FIN
IMPRIMA
ARRANQUE

```

Pruebe con números mayores a 100 o menores y verá que pasa.

APENDICE :

INSTRUCCIÓN	EXPLICACION	EJEMPLO
1) USE <Var1>=<Exp1>	Asigna el resultado de evaluar <Exp 1> a la variable <Var 1 >;<Exp 1 > puede ser un número, una variable numérica o una operación entre variables y/o números.	USE A=B+5
2) INGRESE DATO ["Texto"] <Var>	Asigna el valor numérico ingresado desde el teclado a la variable <Var>. Texto es opcional y es un texto que aparece y en pantalla antes de pedir el ingreso del dato.	INGRESE DATO "Ingrese un Nro" NUMERO
3) INGRESE TEXTO ["Texto"] @<Var alf. >	Asigna el valor ingresado desde el teclado a la variable @<Var alf.>	INGRESE TEXTO "NOMBRE:" @NOMBRE
4) ASIGNE @<Var1>= @<var 2>/"Txt 1" [+ @<var3 >/"Txt 2")]	Asigna el valor de @<Var2> o <texto1> a @<Var1 >, en caso de usar + y una variable o constante alfanumérica, concatena estas a @<Var2> o <texto1> antes de asignar.	ASIGNE "NOMBRE:" + @NOMBRE
5) SI <cond> [Y/O cond] HAGA instrucción-1 . . instrucción -n [OTRO Instrucción -n +1 . . Instrucción -n +m] FIN	Evalúa <condición> y obtiene un valor de verdad, si este es verdadero ejecuta Instrucción -1 hasta Instrucción -n. Si es falso y se usó la opción OTRO ejecuta las instrucciones de n+1 a n+m, si no se usó OTRO continúa con la instrucción que sigue a FIN. En todos los casos continúa con esta instrucción.	10 SI A>= 5 HAGA 20 ESCRIBA "A >= 5 " 30 OTRO 40 ESCRIBA "A MENOR QUE 5" 50 FIN
6) ECO <exp1> instrucción-1 . . instrucción -n FIN	Evalúa <EXP> que puede ser un número, una variable o una operación entre variables y/o números, y ejecuta la cantidad de veces de la evaluación las instrucciones de 1 a n.	10 ECO * 2 20 ESCRIBA "HOLA" 30 FIN

7) ECO HASTA <Condición> Instrucción – 1 . . Instrucción – n FIN	Evalúa <condición> y si esta no se cumple ejecuta las instrucciones de 1 a n, continúa este ciclo HASTA que la <condición > se cumpla.	10 USE A = 0 20 ECO HASTA A = 5 30 ESCRIBA A 40 USE A = A+1 50 FIN
8) COLOR FONDO/figura <color>	Pinta del color dado en <Color> ya sea <figura> o Fondo según sea la opción <figura> es una formación creada con la instrucción FORME.	10 COLOR FONDO AZUL
9) FORME <figura>	Permite crear una formación con el nombre de <figura>. Al ejecutarse aparecerá grilla de 8 x 8 donde se podrá pintar cada punto o no para formar el dibujo. Tecla 1: Pinta 0: No pinta BS: Retrocede	10 FORME CASA
10) DIBUJE [<exp 1> Derecha/<exp2> Abajo] <figura >/PIP/TOC EN FIL #<exp3 > COL #<exp4>	Dibuja la formación <figura> creada con la instrucción FORME en FILA dada por la evaluación de <exp3> y columna <exp4> existiendo 24 filas y 36 columnas. Si se usa la opción Derecha o Abajo moverá la figura en esa dirección <exp1> o <exp2> veces. PIP y TOC son figuras predefinidas y son un cuadrado del color del fondo y uno negro respectivamente.	DIBUJE CASA EN FIL #A COL #5
11) ESCRIBA "<texto> "/<var>/@<var >	Escribe en pantalla ya sea <texto> , el valor de <var> o el de @<var> según sea la opción.	ESCRIBA "LPC"
12) BORRE	Borre la pantalla.	5 BORRE 10 ESCRIBA "LPC"
13) PARE	Detiene en ese punto la ejecución del programa.	10 SI A=0 HAGA 20 PARE 30 OTRO 40 USE C = C/A 50 FIN
14) SONIDO <nota musical> [<nota > ...[]]	Permite obtener alguno de los sonidos de la escala musical DO RE MI FA SOL LA SI.	10 SONIDO DO MI FA
15) GRAFICOS	Permite indicar que todo lo que viene a continuación se tratará en la pantalla de alta resolución. Los dibujos serán de acuerdo al FORME correspondiente. Todo DIBUJE que no esté en la parte GRAFICOS hará que se ponga en pantalla lo que se defina en la misma instrucción.	10 GRAFICOS

LPC - 55

16) FIN-GRAFICOS	Hará finalizar la pantalla de alta resolución. Nota: Cualquier instrucción ESCRIBA o INGRESE será ignorada en Alta Resolución.	100 FIN-GRAFICOS
17) Entero <var> = <exp>	Asigna a la variable <var> la parte entera que resulte de evaluar la expresión <exp> (funciona como el USE).	5 USE A = 5 10 ENTERO B = A/3 (en B queda 1).
18) EJECUTE < Nombre rutina >	Permite ejecutar una subrutina que deberá ser definida en la parte de abajo del programa, después de la instrucción PARE. La subrutina deberá estar en un bloque del tipo RUTINA < nombre subrutina > INSTRUCCION 1 . . INSTRUCCION N VUELVA Una vez encontrada la instrucción EJECUTE se comenzará a ejecutar a partir de la INSTRUCCION 1 hasta que encuentre la Instrucción VUELVA en cuyo caso seguirá ejecutando la instrucción siguiente a EJECUTE.	5 USE A = 0 10 INGRESE DATO NUM 20 EJECUTE CUENTA 30 PARE 40 RUTINA CUENTA 50 ESCRIBA A 60 USE A = A+1 70 SI A < 10 HAGA 80 EJECUTE CUENTA 100 VUELVA
19) RUTINA < Nombre subrutina >	Marca el principio de una rutina. Es una instrucción de marca, no hace nada.	
20) VUELVA	Vuelve el control a la instrucción siguiente de la última instrucción EJECUTE. Marca el final de una subrutina.	

COMANDOS

COMANDO	EXPLICACION
ARRANQUE	Se usa para ejecutar el programa que está en ese momento en memoria.
AUTO [<línea inicial>] [<incremento >]	Numera automáticamente las líneas de programa a partir del número <línea inicial> aumentando en el valor de <incremento>. Los valores asumidos de <línea inicial > e <incremento> son 10.
AYUDA	Muestra en pantalla los comandos e instrucciones del LPC.
CALCULE <n1 > <OP> <n2>	Realiza la operación matemática <OP> con el primer operando <n1> y el segundo <n2> poniendo el resultado en pantalla.
FIN-SECUENCIA	Deshabilita el modo, seguimiento o SECUENCIA.
GRABE <mecanismo>: <nombre programa>	Graba el programa almacenado en memoria con rótulo <nombre programa> en el periférico <mecanismo>.
LEA <mecanismo>: <nombre de programa>	Lee el programa <nombre del programa> desde el periférico <mecanismo>.
LIMPIE	Borra la pantalla y limpia la memoria, es decir elimina programa y variables almacenadas.
LISTE	Sirve para listar en pantalla un programa previamente almacenado en memoria.
SECUENCIA	Habilita seguimiento de programa, poniendo durante la ejecución los números de línea por donde pasa el programa, en pantalla.
Tecla [ESC]	Detiene automáticamente el listado o la ejecución de un programa. Estos se continúan oprimiendo cualquier tecla.
Tecla [?]	Interrumpe la ejecución de un programa o un listado, apareciendo la palabra PAUSA.

INDICE TEMATICO

INTRODUCCION	3
CAPITULO 1.....	4
El intérprete L.P.C.	
Cómo se carga L.P.C.	
CAPITULO 2.....	6
Cómo pedirle a la computadora lo que debe hacer	
Uso de ← → INS DEL ENTER	
CAPITULO 3.....	9
Uso de la computadora como una calculadora	
Sentencia: CALCULE	
Operaciones: + - * / ^	
Expresiones y notación científica	
CAPITULO 4.....	10
Programando su computadora	
Sentencias: ESCRIBA, ARRANQUE, LISTE, BORRE, LIMPIE, ESC (tecla)	
Programas	
Edición de programas empleando ↑ ↓	
CAPITULO 5.....	15
Variables numéricas	
Sentencia: USE	
CAPITULO 6.....	17
Variables alfanuméricas (Textos)	
Sentencias: ASIGNE	
CAPITULO 7.....	19
Más programación de la computadora	
Sentencias: INGRESE DATO, INGRESE TEXTO, LIMPIE, ESCRIBA, ENTERO	
CAPITULO 8.....	23
SI	
Sentencias: SI, PARE	
Operaciones: =, <, >, <=, >=, <>, Y, O	
CAPITULO 9.....	27
Ciclos	
Sentencias: ECO, ECO HASTA, ?	

CAPITULO 10	31
El juego de caracteres	
Sentencia: DIBUJE	
CAPITULO 11	33
Gráficos	
Sentencias: GRAFICOS, FIN-GRAFICOS, FORME, DIBUJE	
CAPITULO 12	37
Colores y sonido	
Sentencias: COLOR, SONIDO	
CAPITULO 13	41
Subrutinas	
Sentencias: EJECUTE, RUTINA, VUELVA	
CAPITULO 14	45
Haciendo funcionar sus programas	
Diagramación, corrección de errores	
CAPITULO 15	49
Almacenamiento en cinta magnética	
Sentencias: GRABE, LEA	
CAPITULO 16	51
Usando la impresora	
Sentencias: IMPRIMA, IMPRIMIR	
APENDICES	53
Sentencia L.P.C.: COMANDOS e INSTRUCCIONES	
Códigos de error	
Ejemplos	



LPC es un intérprete que busca comunicar al hombre con la computadora a través de un lenguaje sencillo de sintaxis simple y directa, a través de una 'conversación' y no de complicados códigos que en general suelen olvidarse fácilmente, en el lenguaje cotidiano y sobre todo en castellano, ya que uno de los problemas para el que toma el primer contacto con la computadora, además del hecho en sí, es tener que traducir o intentar entender qué significa cada instrucción. Otra característica es la de estar dotado de estructuras lógicas con el propósito de que la persona que lo usa, además de poder comunicarse con la máquina aprenda a resolver los problemas en forma ordenada, yendo de lo general a lo particular.

El uso del intérprete responde también a una forma de 'conversación natural' permitiendo ingresar programas, visualizarlos a través de una terminal de video o una impresora, eliminar los mismos de la memoria del computador, grabarlos en algún medio periférico o leerlos de los mismos, además de poder realizar cálculos matemáticos en forma directa y obtener en la pantalla los comandos e instrucciones del intérprete LPC.

Es por estas razones que LPC es un lenguaje educativo e introductorio que no necesita conocimiento previos.

© SISTEMAS LOGICAL
MSX: Marca Registrada de MICROSOFT Corp.
Telemática S.A. - 1986 Todos los derechos reservados

Producido en Argentina por Telemática S.A.
