

MANUAL OPERATIVO Y DICCIONARIO BASIC

Este volumen contiene información general sobre la instalación del ordenador y algunas convenciones utilizadas en el lenguaje de programación BASIC.

En la segunda parte se presenta, en orden alfabético, el lenguaje BASIC correspondiente a los ORDENADORES LASER 110/210/310/500.

Dado que la mayoría de las instrucciones corresponden al BASIC estándar, este manual es de gran utilidad para cualquier persona que posea un ordenador.

ÍNDICE

GENERALIDADES

- Introducción
- Instalación del ordenador
- Comandos en modo directo
- Teclas de selección, control y edición
- Constantes y variables
- Variables numéricas
- Variables alfanuméricas
- Reglas para la asignación de nombres a variables
- Signos de sintaxis
- Formato de pantalla
- Operadores relacionales
- Operadores lógicos
- Operadores matemáticos (prioridad y paréntesis)
- Numeración binaria y hexadecimal

DICCIONARIO BASIC

INTRODUCCIÓN

Este manual contiene información básica para operar con los ordenadores LASER de la serie 110/210/310/500. Se recomienda leer atentamente esta introducción para comprender mejor las secciones siguientes del texto.

Cada ordenador cuenta con un manual de uso, que en muchos casos no está en español. Es importante hacer referencia al manual original, aunque esté en otro idioma, para consultar tablas y esquemas.

Las informaciones aquí contenidas son válidas para todos los ordenadores LASER, salvo que se indique lo contrario.

Las características que NO son comunes a todos los modelos LASER están identificadas con el modelo que posee o no dicha característica.

Ejemplos:

- **(LASER 500)** indica que solo el LASER 500 cuenta con esa característica.
- **(EXCLUIDO LASER 110/210)** indica que la información no se aplica a los modelos 110 y 210.

En cuanto al diccionario BASIC, los términos están ordenados alfabéticamente para facilitar su búsqueda. Para cada término se indica:

- **Tipo:** Clasificación de la palabra clave (instrucción, comando, función, etc.).
- **Ortografía:** Forma correcta de escribir el comando, incluyendo parámetros en formato genérico.
- **Uso:** Explicación de cómo emplear el comando.
- **Ejemplo:** Muestra un uso práctico del comando.

INSTALACIÓN DEL ORDENADOR

Un **HOME COMPUTER** consta de un conjunto de dispositivos que deben conectarse entre sí y recibir alimentación de la red eléctrica de 220V.

Los principales componentes son:

- Teclado/CPU
- Monitor (o televisor)
- Grabadora de casetes o unidad de disquete
- Impresora o plóter

Los dispositivos básicos son el teclado/CPU y el monitor.

Para instalar el ordenador, siga los diagramas de conexión proporcionados en el manual del equipo, teniendo en cuenta lo siguiente:

- El teclado ya está conectado al ordenador, ya que está integrado en la misma carcasa que la CPU.
- Para conectar la alimentación a 220V, use el adaptador de corriente suministrado.
- La grabadora debe conectarse a la red eléctrica con su propio adaptador y al ordenador mediante un cable especial. Las entradas y salidas de la grabadora suelen estar etiquetadas como **TAPE** o **REC**.
- El ordenador debe conectarse con un cable coaxial a la entrada de antena de un televisor en blanco y negro o color. Dependiendo del tipo de televisor utilizado, ajuste el interruptor en la parte inferior del teclado a **B/W** (blanco y negro) o **C** (color).
- El cable de conexión al televisor debe enchufarse en la salida correcta del ordenador, identificada con la etiqueta **TV**.

- Si se usa un **MONITOR**, conéctelo a la salida identificada como **MONITOR**.
- El televisor debe estar encendido y sintonizado en la señal emitida por el ordenador, buscándola como si fuese un canal de televisión (cuando el ordenador esté encendido y conectado al televisor).

Si tiene dificultades para encontrar o guardar la señal en su televisor, consulte a un técnico especializado.

Cuando todo esté correctamente conectado, en la pantalla aparecerá un cuadro con información sobre el ordenador utilizado, junto con un cursor parpadeante y el mensaje **READY**, indicando que el ordenador está listo para recibir comandos.

COMANDOS EN MODO DIRECTO

Los ordenadores LASER pueden operar en **modo directo**, lo que permite introducir comandos directamente y ejecutarlos inmediatamente al presionar la tecla **ENTER** o **RETURN**.

Ejemplo:

```
PRINT 10
```

Este comando mostrará en la pantalla el número **10**.

Antes de comenzar a operar, es necesario familiarizarse con los distintos comandos.

Si introduce un comando incorrecto y presiona **RETURN**, el ordenador mostrará el mensaje ? **SYNTAX ERROR**.

No se preocupe: no es posible dañar el ordenador simplemente presionando teclas. Si introduce comandos incorrectos, el sistema indicará que no los reconoce, pero no sufrirá ningún daño.

Es importante conocer qué comandos reconoce su ordenador y asegurarse de escribirlos correctamente.

TECLAS DE SELECCIÓN, CONTROL Y EDICIÓN

El teclado de los ordenadores LASER cuenta con varias teclas que pueden realizar una o más funciones.

- **Tecla SHIFT:** Permite activar las mayúsculas (**LASER 500**) o seleccionar símbolos gráficos (**LASER 110/210/310**).
- **Tecla CTRL:** Activa comandos preprogramados para facilitar el uso del lenguaje BASIC.

- **Teclas combinadas:** Tanto **SHIFT** como **CTRL** deben presionarse junto con otra tecla para ejecutar una función específica.

Ejemplo:

CTRL + P

Este comando imprimirá la instrucción **PRINT** en la pantalla.

En el **LASER 500**, la combinación **CTRL + C** equivale a **CTRL + BREAK** en los modelos **LASER 110/210/310**, deteniendo la ejecución de un programa.

MANUAL OPERATIVO Y DICCIONARIO BASIC

Este volumen contiene información general sobre la instalación del ordenador y algunas convenciones utilizadas en el lenguaje de programación BASIC.

En la segunda parte se presenta, en orden alfabético, el lenguaje BASIC correspondiente a los ORDENADORES LASER 110/210/310/500.

Dado que la mayoría de las instrucciones corresponden al BASIC estándar, este manual es de gran utilidad para cualquier persona que posea un ordenador.

TECLAS DE CONTROL Y EDICIÓN

Los ordenadores LASER cuentan con diversas teclas de control y edición. Algunas de las funciones principales son:

- **Teclas de dirección:** Permiten mover el cursor en diferentes direcciones. En los modelos LASER 110/210/310 se combinan con **CTRL** y otras teclas.
- **Teclas especiales (LASER 500):**
 - **INS:** Inserta un carácter.
 - **DEL:** Borra un carácter.
 - **DEL LINE:** Borra desde la posición del cursor hasta el final de la línea.
 - **CLS HOME:** Limpia la pantalla o, si se usa con **SHIFT**, coloca el cursor en la esquina superior izquierda.
- **Teclas de acceso rápido:** Combinaciones como **CTRL + L** permiten insertar caracteres o activar modos gráficos.
- **ESC:** Define secuencias de escape para funciones avanzadas.
- **TAB:** Mueve el cursor a las posiciones de tabulación predefinidas.

CONSTANTES Y VARIABLES

Los ordenadores LASER trabajan con constantes y variables en BASIC, y aplican las reglas estándar del lenguaje para su reconocimiento.

- **Constantes numéricas:** Números enteros o decimales que deben separarse con un punto y no con una coma.
 - Ejemplo: 10, 12.55, 3.141592654
- **Constantes alfanuméricas (cadenas):** Conjunto de caracteres entre comillas.
 - Ejemplo: "BRAVO", "TEL. 031/27.55.88"
- **Constantes hexadecimales (LASER 500):** Se identifican con &H antes del número.

Las variables pueden ser de diferentes tipos y están identificadas por un nombre seguido de un símbolo de tipo:

- **Numéricas enteras (%):** Solo números enteros entre -32768 y +32767.
 - Ejemplo: A%, B1%, COD%
- **Numéricas reales:** Admiten decimales y pueden ser de precisión simple o doble (# para doble precisión en LASER 500).
 - Ejemplo: A, Z1, C1P, A#, B1#
- **Alfanuméricas (\$):** Cadenas de texto.
 - Ejemplo: A\$, PI\$, AB\$

Reglas de nomenclatura

- El nombre debe comenzar con una letra del alfabeto.
- En LASER 110/210/310, los nombres tienen un máximo de 2 caracteres.
 - Ejemplo: PROVAS y PRO\$ se interpretan como PR\$.
- En LASER 500, los nombres pueden tener cualquier longitud.
 - Ejemplo: IMPORTO# e IMPORTA# son distintos.
- Se pueden incluir números, pero el primer carácter debe ser una letra.

Símbolos de sintaxis

- Se pueden usar dos puntos : para separar múltiples instrucciones en una sola línea.
- La coma , y el punto y coma ; tienen funciones específicas en PRINT:
 - ; mantiene la misma línea sin salto de línea.
 - , genera una tabulación.

OPERADORES EN BASIC

Operadores relacionales

Los ordenadores LASER reconocen los siguientes operadores:

- > Mayor que
- < Menor que

- = Igual a
- >= Mayor o igual que
- <= Menor o igual que
- <> Diferente de

En valores numéricos, el criterio es algebraico (-1 es menor que 1). En cadenas, el criterio es ASCII.

Ejemplo:

```
"ALBO" > "ALBERO" ' Devuelve TRUE porque O tiene un código ASCII mayor que E
```

Operadores lógicos

Los siguientes operadores lógicos están disponibles en BASIC:

- NOT (Negación) → A NOT B
- OR (O lógico) → A OR B
- AND (Y lógico) → A AND B
- **Extras en LASER 500:**
 - XOR (O exclusivo)
 - IMP (Implicación)
 - EQV (Equivalencia)

Ejemplo:

```
10 INPUT "Ingrese tres números";A,B,C
20 IF A=B AND B=C THEN PRINT "Los tres son iguales"
30 IF NOT (A=B) OR NOT (B=C) THEN PRINT "Los tres no son iguales"
```

Operadores matemáticos y prioridades

Los símbolos matemáticos utilizados en BASIC son:

- + Suma
- - Resta
- * Multiplicación
- / División
- ^ Potencia

La prioridad de operaciones es:

1. Asignación de signo
2. Potenciación
3. Multiplicación y división
4. Suma y resta

El uso de **paréntesis** modifica la prioridad y agrupa operaciones.

Ejemplo:

```
PRINT 100 / ((5*4) / (2+3)) - 30/2 ' Resultado: 10
```

NUMERACIÓN BINARIA Y HEXADECIMAL

Sistema decimal

Se basa en 10 dígitos (0-9) y la posición de cada número representa una potencia de 10.

Sistema binario

Utiliza solo 0 y 1, donde cada posición representa una potencia de 2.

Ejemplo:

1011 en binario equivale a:
 $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 8 + 0 + 2 + 1 = 11$ decimal

Sistema hexadecimal

Usa base 16 con los dígitos 0-9 y letras A-F (A=10, B=11, ..., F=15).

Ejemplo:

10AF (hex) = $(1 \times 16^3) + (0 \times 16^2) + (A=10) \times 16^1 + (F=15) \times 16^0$
= $4096 + 0 + 160 + 15 = 4271$ decimal

Para diferenciar los números hexadecimales, se usa:

- H al final (3256H)
- \$ al inicio (\$C1FA)

COMANDOS Y FUNCIONES

AUTO

- **Tipo:** Comando de sistema
- **Ortografía:** AUTO n1, n2
- **Uso:** Permite la numeración automática de líneas durante la programación. Para detenerlo, presione CTRL + C.
- **Ejemplo:** AUTO 100,10

BREAK (LASER 500)

- **Uso:** Interrumpe la ejecución de un programa en los modelos:
 - **LASER 110/210/310:** CTRL + BREAK
 - **LASER 500:** CTRL + C

CALL (LASER 500)

- **Tipo:** Función de sistema
- **Ortografía:** CALL nv (I, J, K)
- **Uso:** Salta a una subrutina definida en una dirección de memoria específica.
- **Ejemplo:**
 - 10 MYROUT = &HDOOO
 - 20 CALL MYROUT (I, J, K)

CDBL (LASER 500)

- **Tipo:** Función de conversión
- **Ortografía:** CDBL (x)
- **Uso:** Convierte un número en precisión simple a precisión doble.
- **Ejemplo:**
 - 10 A = 454.67
 - 20 PRINT A; CDBL(A)
 - RUN
 - 454.67 454.670013427344

CHR\$

- **Tipo:** Función de conversión
- **Ortografía:** CHR\$(x)
- **Uso:** Retorna el carácter correspondiente al código ASCII especificado.
- **Ejemplo:**
 - 10 PRINT CHR\$(65)
 - 20 X=66
 - 30 PRINT CHR\$(X)
 - RUN
 - A
 - B

CLEAR

- **Tipo:** Comando de sistema
- **Ortografía:** CLEAR nn[, mm]
- **Uso:** Inicializa variables y reserva memoria.
- **Ejemplo:**

```
CLEAR 1000
```

CLOAD

- **Tipo:** Comando de sistema
- **Ortografía:** CLOAD "nombre programa"
- **Uso:** Carga un programa desde una cinta.
- **Ejemplo:**

```
CLOAD "PROG.3"
```

CLS

- **Tipo:** Instrucción
- **Ortografía:** CLS
- **Uso:** Borra la pantalla y coloca el cursor en la esquina superior izquierda.

COLOR (LASER 500)

- **Tipo:** Función
- **Ortografía:** COLOR I, J, K
- **Uso:** Define los colores de texto, fondo y borde.
- **Ejemplo:**

```
COLOR 0, 4, 6
```

CONT

- **Tipo:** Comando de sistema
- **Ortografía:** CONT
- **Uso:** Continúa la ejecución de un programa interrumpido con BREAK.

COPY

- **Tipo:** Instrucción de impresión
- **Ortografía:** COPY
- **Uso:** Copia el contenido de la pantalla en una impresora.

COS

- **Tipo:** Función trigonométrica
- **Ortografía:** COS (x)
- **Uso:** Retorna el coseno del ángulo en radianes.
- **Ejemplo:**
- 10 A = COS (6.28)
- 20 PRINT A
- RUN
- .999995

CRUN

- **Tipo:** Comando de sistema
- **Ortografía:** CRUN "nombre programa"
- **Uso:** Carga y ejecuta un programa desde una cinta.

CSAVE

- **Tipo:** Comando de sistema
- **Ortografía:** CSAVE "nombre programa"
- **Uso:** Guarda un programa en una cinta.

CSNG (LASER 500)

- **Tipo:** Función de conversión
- **Ortografía:** CSNG (x)
- **Uso:** Convierte un número en precisión simple.
- **Ejemplo:**
 - 10 A# = 975.342145
 - 20 PRINT A#; CSNG(A#)
 - RUN
 - 975.342145 975.342

CVI / CVS / CVD (LASER 500)

- **Tipo:** Funciones de conversión
- **Ortografía:** CVI (x\$), CVS (x\$), CVD (x\$)
- **Uso:** Convierten cadenas a valores numéricos.

DATA

- **Tipo:** Instrucción
- **Ortografía:** DATA [dato1], [dato2], ...
- **Uso:** Almacena datos que pueden ser leídos con READ.
- **Ejemplo:**
 - 10 DATA "PRUEBA", 1, 2, "FIN"
 - 20 READ A\$, B, C, D\$
 - 30 PRINT A\$, B, C, D\$
 - RUN

DEF FN (LASER 500)

- **Tipo:** Instrucción
- **Ortografía:** DEF FN w(x,y)=[expresión]
- **Uso:** Define una función personalizada.
- **Ejemplo:**
 - 10 DEF FN A(N) = N - INT(N)
 - 20 A = 12.12345
 - 30 B = FN A(A)
 - 40 PRINT A; B
 - RUN

DEF INT / SGN / DBL / STR (LASER 500)

- **Tipo:** Instrucciones
- **Ortografía:** DEF tipo x-y, z-t, ...
- **Uso:** Define tipos de variables.
- **Ejemplo:**
 - DEFINT A-E, K, Q-T
 - DEFSTR Z

DEF USR

- **Tipo:** Instrucción
- **Ortografía:** DEF USRn=nn
- **Uso:** Define el inicio de una subrutina en ensamblador definida por el usuario.
- **Ejemplo:**
 - 200 DEF USR0=24000
 - 210 X=USR0(Y Z/2.89)

DELETE

- **Tipo:** Comando de sistema
- **Ortografía:** DELETE [aa] - [bb]
- **Uso:** Borra líneas de un programa BASIC dentro del rango especificado.
- **Ejemplo:**

```
DELETE 100-200
```

DIM

- **Tipo:** Instrucción
- **Ortografía:** DIM x(n1), a\$(n2), b\$(n3,n4), ...
- **Uso:** Declara matrices y vectores de distintos tipos de datos.
- **Ejemplo:**

```
10 DIM A(25), A$(100,100)
```

DRAW (LASER 500)

- **Tipo:** Función gráfica
- **Ortografía:** DRAW x, y
- **Uso:** Dibuja líneas en la pantalla a partir de las últimas coordenadas dadas.
- **Ejemplo:**
 - 10 COLOR 15,0,0:GR1
 - 20 MOVE (79, 15)
 - 30 DRAW (149,100):DRAW(9,100):DRAW(79,15)
 - 40 GOTO 20

END

- **Tipo:** Instrucción
- **Ortografía:** END
- **Uso:** Marca el fin lógico del programa BASIC.
- **Ejemplo:**
 - 10 PRINT "PROGRAMA FINALIZADO"
 - 20 END

ERASE (LASER 500)

- **Tipo:** Instrucción

- **Ortografía:** ERASE x,y,z, ...
- **Uso:** Borra matrices de la memoria y permite redimensionarlas.
- **Ejemplo:**
- 100 DIM A(20), B(20)
- 400 ERASE A,B
- 410 DIM A(12), B(12)

ERR/ERL

- **Tipo:** Variables de sistema
- **Uso:**
 - ERR almacena el código del error producido.
 - ERL guarda el número de línea donde ocurrió el error.
- **Ejemplo:**
- ON ERROR GOTO 100
- 50 PRINT A/0
- 100 PRINT "Error en la línea", ERL, "Código:", ERR

ERROR

- **Tipo:** Instrucción
- **Ortografía:** ERROR x
- **Uso:** Genera errores programados con códigos específicos.
- **Ejemplo:**
- 10 ON ERROR GOTO 100
- 20 ERROR 210
- 100 PRINT "Código de error: ", ERR

EXP

- **Tipo:** Función matemática
- **Ortografía:** EXP(x)
- **Uso:** Retorna e^x .
- **Ejemplo:**

```
10 PRINT EXP(4)
```

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

- **Tipo:** Bucle
- **Ortografía:** FOR x = n1 TO n2 STEP n3 NEXT x
- **Uso:** Ejecuta un bucle controlado por la variable x.
- **Ejemplo:**
- 10 FOR I = 0 TO 20 STEP 5
- 20 PRINT I;
- 30 NEXT I

GOSUB / RETURN

- **Tipo:** Instrucción de salto
- **Ortografía:** GOSUB nn RETURN

- **Uso:** Salta a una subrutina y retorna al punto de origen tras ejecutar RETURN.
- **Ejemplo:**
- 10 PRINT "Inicio"
- 20 GOSUB 100
- 30 PRINT "Fin"
- 40 END
- 100 PRINT "Subrutina"
- 110 RETURN

GOTO

- **Tipo:** Instrucción de salto incondicional
- **Ortografía:** GOTO nn
- **Uso:** Salta a una línea específica del programa BASIC.
- **Ejemplo:**
- 10 PRINT "Inicio"
- 20 GOTO 50
- 50 PRINT "Salto exitoso"

GR (LASER 500)

- **Tipo:** Función gráfica
- **Ortografía:** GRn
- **Uso:** Define el tipo de pantalla gráfica.
- **Ejemplo:**

```
GR1 ' Activa el modo gráfico 160x192 píxeles
```

HEX\$ (LASER 500)

- **Tipo:** Función de conversión
- **Ortografía:** HEX\$(x)
- **Uso:** Convierte un número en su representación hexadecimal.
- **Ejemplo:**

```
10 PRINT HEX$(42)
```

IF ... THEN ... ELSE

- **Tipo:** Estructura condicional
- **Ortografía:** IF condición THEN acción ELSE acción
- **Uso:** Evalúa una condición y ejecuta una acción dependiendo del resultado.
- **Ejemplo:**
- 10 INPUT "Número"; N
- 20 IF N=10 THEN PRINT "Es 10" ELSE PRINT "No es 10"

INKEY\$

- **Tipo:** Función de entrada
- **Uso:** Devuelve la última tecla presionada.
- **Ejemplo:**

- 10 PRINT "Presiona una tecla"
- 20 A\$ = INKEY\$
- 30 PRINT "Tecla presionada: ", A\$

INPUT

- **Tipo:** Instrucción de entrada
- **Ortografía:** INPUT "mensaje"; variable
- **Uso:** Solicita datos desde el teclado y los almacena en una variable.
- **Ejemplo:**

```
10 INPUT "Ingrese un número"; A
```

INPUT#

- **Tipo:** Instrucción de asignación
- **Ortografía:** INPUT# "nombre archivo", [lista de variables]
- **Uso:** Lee datos almacenados en un archivo de cinta en un formato secuencial y los asigna a variables.
- **Ejemplo:**
- 100 INPUT# "PROVA", A, B\$, C, D
- 110 PRINT A, B\$, C, D

(Debe haberse guardado previamente con PRINT#)

INPUT\$

- **Tipo:** Función alfanumérica
- **Ortografía:** INPUT\$(x)
- **Uso:** Permite la entrada de un número determinado de caracteres sin mostrar el cursor.
- **Ejemplo:**
- 100 PRINT "Presione P para continuar, S para detener"
- 110 X\$ = INPUT\$(1)
- 120 IF X\$ = "P" THEN PRINT "Continuando..." : GOTO 500
- 130 IF X\$ = "S" THEN PRINT "Detenido" : STOP ELSE 100

INSTR

- **Tipo:** Función alfanumérica
- **Ortografía:** INSTR(I, x\$, y\$)
- **Uso:** Devuelve la posición de la cadena y\$ dentro de x\$, comenzando desde I.
- **Ejemplo:**
- 10 X\$ = "ABCDEB"
- 20 Y\$ = "B"
- 30 PRINT INSTR(X\$, Y\$) ' Retorna 2
- 40 PRINT INSTR(4, X\$, Y\$) ' Retorna 6

INT

- **Tipo:** Función numérica
- **Ortografía:** INT(x)
- **Uso:** Devuelve la parte entera del número x.
- **Ejemplo:**

```
10 A = INT(3.14) : PRINT A ' Imprime 3
```

JOY (LASER 500)

- **Tipo:** Función de lectura de joystick
- **Ortografía:** JOY(x)
- **Uso:** Devuelve valores relacionados con la posición y los botones del joystick.
- **Ejemplo:**
- 10 CLS
- 20 A = JOY(0) : B = JOY(1) : C = JOY(2)
- 30 PRINT A, B, C
- 40 GOTO 20

KEY (LASER 500)

- **Tipo:** Función de sistema
- **Ortografía:** KEY n, "cadena"
- **Uso:** Asigna una secuencia de comandos a una tecla de función.
- **Ejemplo:**

```
KEY 3, "LIST"
```

LEFT\$

- **Tipo:** Función alfanumérica
- **Ortografía:** LEFT\$(x\$, I)
- **Uso:** Extrae los I primeros caracteres de x\$.
- **Ejemplo:**

```
10 A$ = "ABCDEFGH" : B$ = LEFT$(A$, 4) : PRINT B$ ' Imprime
"ABCD"
```

LEN

- **Tipo:** Función alfanumérica
- **Ortografía:** LEN(x\$)
- **Uso:** Devuelve la cantidad de caracteres en x\$.
- **Ejemplo:**

```
10 A$ = "ABCDEFG" : PRINT LEN(A$) ' Imprime 7
```

LET

- **Tipo:** Instrucción de asignación
- **Ortografía:** LET variable = expresión

- **Uso:** Asigna un valor a una variable.
- **Ejemplo:**
- 10 LET A = 10
- 20 B = A + A
- 30 PRINT A, B

LINE INPUT

- **Tipo:** Instrucción de asignación
- **Ortografía:** LINE INPUT x\$
- **Uso:** Almacena una línea completa de texto en x\$.
- **Ejemplo:**
- 10 LINE INPUT A\$
- 20 PRINT A\$

LIST / LLIST

- **Tipo:** Comando de sistema
- **Ortografía:** LIST [aa] - [bb] / LLIST [aa] - [bb]
- **Uso:** Lista el código BASIC en pantalla (LIST) o en la impresora (LLIST).

LOG

- **Tipo:** Función matemática
- **Ortografía:** LOG (x)
- **Uso:** Devuelve el logaritmo natural de x.
- **Ejemplo:**

```
10 PRINT LOG(100) ' Resultado: 4.605
```

LPOS (LASER 500)

- **Tipo:** Función de impresión
- **Ortografía:** LPOS (x)
- **Uso:** Devuelve la posición de la cabeza de impresión.

LPRINT

- **Tipo:** Instrucción de impresión
- **Ortografía:** LPRINT [expresión]
- **Uso:** Imprime en papel.

MID\$

- **Tipo:** Función alfanumérica
- **Ortografía:** MID\$(x\$, I, J)
- **Uso:** Extrae J caracteres desde la posición I de x\$.
- **Ejemplo:**

```
10 A$ = "ABCDEFGH" : B$ = MID$(A$, 3, 4) : PRINT B$ ' Imprime
"CDEF"
```

MOD

- **Tipo:** Función matemática
- **Ortografía:** $x \text{ MOD } y$
- **Uso:** Devuelve el residuo de x / y .
- **Ejemplo:**

```
10 A = 10 MOD 3 : PRINT A ' Imprime 1
```

MODE (LASER 110/210/310)

- **Tipo:** Comando gráfico
- **Ortografía:** MODE (x)
- **Uso:** Cambia entre modo texto (0) y modo gráfico (1).

MON (LASER 500)

- **Tipo:** Comando de sistema
- **Ortografía:** MON
- **Uso:** Activa el monitor de memoria.

NULL (LASER 500)

- **Tipo:** Función de edición
- **Ortografía:** NULL x
- **Uso:** Establece el número de espacios en blanco entre líneas impresas.
- **Ejemplo:**
- NULL 2
PRINT "Texto con espacio adicional"

OCT\$(LASER 500)

- **Tipo:** Función de conversión
- **Ortografía:** OCT\$(x)
- **Uso:** Convierte un número decimal a su representación octal.
- **Ejemplo:**

```
PRINT OCT$(24) ' Devuelve "30"
```

ON ERROR GOTO

- **Tipo:** Instrucción de control de errores
- **Ortografía:** ON ERROR GOTO xx
- **Uso:** En caso de error, salta a la línea indicada.
- **Ejemplo:**

- 10 ON ERROR GOTO 100
- 20 PRINT 1/0 ' Provoca error
- 100 PRINT "Se ha producido un error"

ON ... GOSUB / ON ... GOTO

- **Tipo:** Instrucción de control de flujo
- **Ortografía:** ON k GOSUB aa, bb, cc / ON k GOTO aa, bb, cc
- **Uso:** Salta a la línea de programa indicada según el valor de k.
- **Ejemplo:**
- 10 INPUT "Ingrese un número (1-3)"; K
- 20 ON K GOTO 100, 200, 300
- 30 PRINT "Número fuera de rango"
- 40 END
- 100 PRINT "Saltó a 100" : END
- 200 PRINT "Saltó a 200" : END
- 300 PRINT "Saltó a 300" : END

OPTION BASE (LASER 500)

- **Tipo:** Comando de sistema
- **Ortografía:** OPTION BASE k
- **Uso:** Establece el índice base de las matrices (0 o 1).
- **Ejemplo:**

```
OPTION BASE 1 ' Establece 1 como el índice mínimo de matrices
```

OUT (LASER 500)

- **Tipo:** Función de sistema
- **Ortografía:** OUT i, j
- **Uso:** Envía un valor a un puerto del sistema.

PEEK

- **Tipo:** Función de sistema
- **Ortografía:** PEEK (x)
- **Uso:** Lee el valor almacenado en la dirección de memoria x.
- **Ejemplo:**

```
PRINT PEEK(30779)
```

POINT

- **Tipo:** Función gráfica
- **Ortografía:** POINT(x, y)
- **Uso:** Devuelve el color del punto en las coordenadas x, y.
- **Ejemplo:**

```
10 IF POINT(40,20)<>0 THEN PRINT "Encendido"
```

POKE

- **Tipo:** Función de sistema
- **Ortografía:** POKE x, y
- **Uso:** Almacena el valor y en la dirección de memoria x.

PRINT USING

- **Tipo:** Instrucción de salida
- **Ortografía:** PRINT USING "formato"; valores
- **Uso:** Formatea la salida de valores.
- **Ejemplo:**

```
PRINT USING "###.##"; 12.345
```

PRINT#

- **Tipo:** Instrucción de salida
- **Ortografía:** PRINT# "nombre archivo", valores
- **Uso:** Guarda valores en un archivo secuencial.

RANDOMIZE

- **Tipo:** Función de sistema
- **Uso:** Reseed para la función RND.

READ

- **Tipo:** Instrucción de asignación
- **Ortografía:** READ variables
- **Uso:** Lee valores de DATA.

REM

- **Tipo:** Instrucción de comentario
- **Ortografía:** REM texto
- **Uso:** Agrega comentarios en el código.

RENUM

- **Tipo:** Comando de sistema
- **Ortografía:** RENUM nn, vn, in
- **Uso:** Renumeración de líneas del programa.

RESTORE

- **Tipo:** Instrucción de sistema
- **Ortografía:** RESTORE
- **Uso:** Restablece la posición de lectura de DATA.

RESUME

- **Tipo:** Instrucción de control de errores
- **Uso:** Continúa la ejecución tras un error.

RETURN

- **Tipo:** Instrucción de control de flujo
- **Ortografía:** RETURN
- **Uso:** Finaliza una subrutina GOSUB.

RIGHT\$

- **Tipo:** Función alfanumérica
- **Ortografía:** RIGHT\$(x\$, i)
- **Uso:** Extrae i caracteres desde la derecha de x\$.

RND

- **Tipo:** Función numérica
- **Ortografía:** RND(k)
- **Uso:** Genera números aleatorios.

RUN

- **Tipo:** Comando de sistema
- **Ortografía:** RUN nn
- **Uso:** Ejecuta un programa desde una línea específica.

SET

- **Tipo:** Función gráfica
- **Ortografía:** SET(x, y)
- **Uso:** Activa un píxel en x, y.

NULL (LASER 500)

- **Tipo:** Función de edición
- **Ortografía:** NULL x
- **Uso:** Establece el número de espacios en blanco entre líneas impresas.
- **Ejemplo:**
- NULL 2
PRINT "Texto con espacio adicional"

OCT\$(LASER 500)

- **Tipo:** Función de conversión
- **Ortografía:** OCT\$(x)
- **Uso:** Convierte un número decimal a su representación octal.

- **Ejemplo:**

```
PRINT OCT$(24) ' Devuelve "30"
```

ON ERROR GOTO

- **Tipo:** Instrucción de control de errores
- **Ortografía:** ON ERROR GOTO xx
- **Uso:** En caso de error, salta a la línea indicada.
- **Ejemplo:**
- 10 ON ERROR GOTO 100
- 20 PRINT 1/0 ' Provoca error
- 100 PRINT "Se ha producido un error"

ON ... GOSUB / ON ... GOTO

- **Tipo:** Instrucción de control de flujo
- **Ortografía:** ON k GOSUB aa, bb, cc / ON k GOTO aa, bb, cc
- **Uso:** Salta a la línea de programa indicada según el valor de k.
- **Ejemplo:**
- 10 INPUT "Ingrese un número (1-3)"; K
- 20 ON K GOTO 100, 200, 300
- 30 PRINT "Número fuera de rango"
- 40 END
- 100 PRINT "Saltó a 100" : END
- 200 PRINT "Saltó a 200" : END
- 300 PRINT "Saltó a 300" : END

OPTION BASE (LASER 500)

- **Tipo:** Comando de sistema
- **Ortografía:** OPTION BASE k
- **Uso:** Establece el índice base de las matrices (0 o 1).
- **Ejemplo:**

```
OPTION BASE 1 ' Establece 1 como el índice mínimo de matrices
```

OUT (LASER 500)

- **Tipo:** Función de sistema
- **Ortografía:** OUT i, j
- **Uso:** Envía un valor a un puerto del sistema.

PEEK

- **Tipo:** Función de sistema
- **Ortografía:** PEEK (x)
- **Uso:** Lee el valor almacenado en la dirección de memoria x.
- **Ejemplo:**

```
PRINT PEEK(30779)
```

POINT

- **Tipo:** Función gráfica
- **Ortografía:** POINT(x, y)
- **Uso:** Devuelve el color del punto en las coordenadas x, y.
- **Ejemplo:**

```
10 IF POINT(40,20)<>0 THEN PRINT "Encendido"
```

POKE

- **Tipo:** Función de sistema
- **Ortografía:** POKE x, y
- **Uso:** Almacena el valor y en la dirección de memoria x.

PRINT USING

- **Tipo:** Instrucción de salida
- **Ortografía:** PRINT USING "formato"; valores
- **Uso:** Formatea la salida de valores.
- **Ejemplo:**

```
PRINT USING "###.##"; 12.345
```

PRINT#

- **Tipo:** Instrucción de salida
- **Ortografía:** PRINT# "nombre archivo", valores
- **Uso:** Guarda valores en un archivo secuencial.

RANDOMIZE

- **Tipo:** Función de sistema
- **Uso:** Reseed para la función RND.

READ

- **Tipo:** Instrucción de asignación
- **Ortografía:** READ variables
- **Uso:** Lee valores de DATA.

REM

- **Tipo:** Instrucción de comentario
- **Ortografía:** REM texto
- **Uso:** Agrega comentarios en el código.

RENUM

- **Tipo:** Comando de sistema
- **Ortografía:** RENUM nn, vn, in
- **Uso:** Renumeración de líneas del programa.

RESTORE

- **Tipo:** Instrucción de sistema
- **Ortografía:** RESTORE
- **Uso:** Restablece la posición de lectura de DATA.

RESUME

- **Tipo:** Instrucción de control de errores
- **Uso:** Continúa la ejecución tras un error.

RETURN

- **Tipo:** Instrucción de control de flujo
- **Ortografía:** RETURN
- **Uso:** Finaliza una subrutina GOSUB.

RIGHT\$

- **Tipo:** Función alfanumérica
- **Ortografía:** RIGHT\$(x\$, i)
- **Uso:** Extrae i caracteres desde la derecha de x\$.

RND

- **Tipo:** Función numérica
- **Ortografía:** RND(k)
- **Uso:** Genera números aleatorios.

RUN

- **Tipo:** Comando de sistema
- **Ortografía:** RUN nn
- **Uso:** Ejecuta un programa desde una línea específica.

SET

- **Tipo:** Función gráfica
- **Ortografía:** SET(x, y)
- **Uso:** Activa un píxel en x, y.