

MicroScript

Sistema de Tratamiento de Textos para el CPC464
Soft 1010

Publicado por AMSOFT, una división de

Amstrad Consumer Electronics plc
Brentwood House
169 Kings Road
Brentwood
Essex
All rights reserved
First edition 1985

La reproducción o traducción de cualquier parte de esta publicación sin el permiso escrito del propietario de los derechos vulnera la ley. Amstrad e Intelligence (Irl) Ltd. se reservan el derecho a enmendar o alterar la especificación sin aviso previo. Aunque se ha efectuado el máximo esfuerzo en verificar que este complejo programa funciona como aquí se describe, no es posible comprobar ningún programa de esa complejidad bajo todas las condiciones posibles. Por lo tanto, el programa y este manual se suministra 'tal como está' sin garantía de ninguna clase, ni expresa ni implícita.

Soft 1010

Editado por **INDESCOMP, S.A.**
Avda. Mediterráneo, 9 - 28007 MADRID (ESPAÑA)

Derechos reservados en lengua española: **INDESCOMP, S.A.**

Traduce y compone: **CONORG, S.A.**

CONTENIDO

1.0	USO DE ESTE MANUAL	1
1.1	A través de este manual, las series de comandos... ...se muestran en negrita	1 1
2.0	¿QUE ES UN PROCESADOR DE TEXTOS?	2
2.1	El texto se introduce a través de un teclado... ...proyectado en la pantalla del terminal de video... ...antes de ser depositado en el disco	2 2 3
2.2	Conviniendo lo que queremos decir por Caracteres... ...y palabras... ...y párrafo	3 3 4
3.0	ARRANQUE DEL SISTEMA	5
3.1	Enciende la fuente de alimentación... ...antes de cargar el disco del sistema... ...para cargar el sistema operativo en la memoria	5 5 5
3.2	Carga MICROSCRIPT en la unidad ductora A; ...para obtener el menú principal	6 7
3.3	Restaurando el ordenador	7
4.0	CREACION DE UN DOCUMENTO MEDIANTE MICROSCRIPT	8
4.1	Elige la opción de Creación en el Menú Principal	8
4.2	Todo Documento debe estar identificado	8
5.0	FORMATO DEL TEXTO EN PANTALLA	10
5.1	El trazado de las líneas está definido por Reglillas... ...a los que reclama para entrar en acción	10 10

5.2	Pueden ponerse en acción hasta nueve reglillas diferentes	11
5.3	Enmienda las reglillas cambiando los símbolos en la línea representativa	11
5.4	Puedes copiar una línea...	12
6.0	CORRECCION DE ERRORES DURANTE LA INTRODUCCION DE TEXTO POR TECLADO	13
6.1	Re-teclea inmediatamente después de observado el error	13
6.2	Otras clases de errores deben ser localizados... ...y situar el cursor mediante las funciones... ...o con las funciones de movimientos de más posiciones	13 13 14
6.3	Habiendo situado el cursor en la posición a corregir efectúa la supresión... ...o la inserción que se requiera... ...antes de re-componer el formato del documento	15 15 16
6.4	Lee y comprueba el documento... ...antes de hacer que lo guarde en disco	16 17
6.5	Devuelve un documento al disco frecuentemente	17
7.0	REVISION DE DOCUMENTOS USANDO MICROSCRIPT	18
7.1	Reclama el texto a la pantalla	18
7.2	Situa el cursor recurriendo a las funciones de movimiento según unidades de texto... ...o a las funciones de control del cursor	18 19
7.3	Alternativamente, puedes hacer que 'busque' y 'halle' una determinada función	19
8.0	EFFECTUANDO ENMIENDAS AL TEXTO QUE HAS GUARDADO	20
8.1	Los cambios pueden hacerse mediante la inserción de nuevo texto	20

8.2	Se puede suprimir texto por caracteres...	20
	...o por palabras...	21
	...o por líneas	21
8.3	Sustituyendo partes de texto por otras a lo largo del documento	21
8.4	Otras funciones de edición adicionales	22
8.5	El texto revisado se archiva en disco...	23
	...guardando el documento en disco	23
9.0	RE-COMPOSICION DEL FORMATO	25
9.1	Extender una línea	25
9.2	Cambio de formato de un párrafo	26
9.3	Recortando y pegando con MICROSCRIPT	26
9.4	Las secciones de texto pueden recortarse de un área...	28
	...y pegado en otra parte	28
10.0	REALCE DE TEXTOS CON MICROSCRIPT	30
10.1	MICROSCRIPT completamente en-guiona...	30
	...o subrayado continuo	30
10.2	Se puede estipular múltiple impacto del cabezal de impresión...	31
	...para imprimir dos líneas de texto dentro de una sola línea horizontal de papel...	31
	...o el texto puede ser bien resaltado...	31
	...o puesto en negrita	32
11.0	ESTIPULANDO EL FORMATO	33
11.1	Se puede centrar un texto automáticamente...	33
11.2	Ayudas mediante topes y sangrados	33
11.3	MICROSCRIPT 'volteará' el texto al llegar a los extremos de la línea	34

12.0	OPERANDO CON MICROSCRIPT EN MODOS DIFERENTES	35
13.0	EL INTERPRETADOR DEL TEXTO ENTRANTE EN TODA MEZCLA	38
13.1	Usa el interpretador...	39
	...para ejecutar controles incrustados en un texto	39
14.0	MEZCLA DE FICHEROS	40
14.1	Preparación del texto a agregar	40
14.2	Indexación en los ficheros agregados...	41
	...de manera que el interpretador pueda cambiar entre secciones del fichero	42
14.3	Anidamiento al congregar ficheros	42
15.0	TRANSFIRIENDO TEXTOS A TRAVES DE MEMORIAS 'COMPENDIADORAS'	44
16.0	CALCULANDO CON MICROSCRIPT	46
16.1	Delimitando los campos del Calculador	46
16.2	Introduce datos numéricos...	46
	...usando expresiones matemáticas...	47
	...teniendo cuidado de no rebasar el campo	48
16.3	Acumula los datos de un campo numérico...	49
	...y calcula el total horizontal...	49
	...y el total vertical	50
16.4	Salida del modo calculador	50
17.0	PREPARACION DE UN DOCUMENTO PARA IMPRESION	51

17.1	Introduce los comandos entre corchetes en línea sin texto normal	51
17.2	Puede imprimir en el modo 'ancho'	52
18.0	IMPRIMIENDO TEXTO CON MICROSCRIPT	54
18.1	Imprimir un documento directamente especificando el nombre	54
18.2	Alternativamente, cambia uno o más de los parámetros de impresión...	55
	...antes de comenzar a imprimir un documento	58
19.0	USA LA GESTION DE FICHEROS PARA ORGANIZAR EL DISCO	59
19.1	Los documentos pueden ser re-denominados...	59
	...o eliminado	59
19.2	Copia de ficheros a otros discos...	60
	...o congrega dos ficheros en uno	60
19.3	Mostrar el catálogo de documentos de un disco	61
20.0	EL PROGRAMA DEL SISTEMA MICROSCRIPT	62
20.1	MICROSCRIPT se suministra como una colección de ficheros en disco	62
21.0	EL FICHERO DE INFORMACION DEL SISTEMA	63
22.0	PROGRAMACION DE APLICACIONES EN MICROSCRIPT	64
22.1	El interpretador actúa sobre instrucciones incrustadas en el texto a agregar...	64
	...que son representadas por equivalentes visibles de los comandos	64
22.2	El MICROSCRIPT puede generar los equivalente automáticamente	66

22.3	Pausas en los programas para aceptar información por teclado	66
22.4	Los programas escritos en MICROSCRIPT pueden ser almacenados...	68
	..como abreviaciones temporales	68
	...o como ficheros en discos	70
22.5	Los ficheros en disco pueden estar indexados para incrementar su flexibilidad	70
22.6	El comando de AGREGAR FICHEROS ayuda a la programabilidad...	71
	...y permite establecer rutas para fallos u omisiones	72
22.7	La facilidad de 'VAYA A ETIQUETA' incrementa el ámbito de los programas	73
22.8	Los valores numéricos obtenidos en los programs pueden ser guardados...	75
	...y recuperados en una aplicación	75
22.9	Se pueden usar modos alternativos para afectar a la programación	76
23.0	PREPARANDO DISCOS PARA SU USO	77
23.1	Discos rígidos	77
23.2	Discos flexibles	77
23.3	Los discos flexibles requieren cuidados	77
24.0	FUNCIONES DISPONIBLES CON MICROSCRIPT	79
25.0	APENDICE I	85
25.1	Código Standard ASCII	85
25.2	Caracteres de función para programas	86
26.0	APENDICE II MENSAJE DE ERROR	87

1.0 Uso de este Manual

MICROSCRIPT es un potente auxiliar en el trabajo administrativo con aplicaciones que varían desde la composición de textos hasta los cálculos matemáticos.

Este manual se ha diseñado teniendo en cuenta todas esas aplicaciones. La primera parte te lleva paso a paso a través del sistema hasta el momento en que tú seas capaz de controlar, todo excepto las rutinas más complicadas.

Los títulos de las secciones a lo largo del manual se han planteado como frases completas normales y se amplían en algunos casos a varias secciones consecutivas, para ayudar a mejorar el flujo lógico de la información. En las primeras pocas ocasiones que se use el sistema debieras leer tanto los títulos como el texto de las secciones. A medida que estés más familiarizado con el sistema, únicamente leerás los títulos; pasando sólo al texto cuando estos titulares continuos no te recuerden lo suficiente el tema tratado en la sección.

1.1 A través de este manual, las series de comandos...

La introducción de comandos involucrará la pulsación de dos clases diferentes de teclas.

Algunas de las cubiertas de las teclas están marcadas con palabras completas o con abreviaturas ([ESC], [CTRL] o [TAB], por ejemplo).

Algunos comandos pueden requerir que sean usadas estas teclas 'dedicadas' junto con las teclas standard alfanuméricas (R, F, ?, *, por ejemplo). En esta situación, no hay ninguna diferencia entre letras mayúsculas y minúsculas.

...se muestran en negrita

Cuando se va a pulsar una tecla como parte de una secuencia de comando, se muestra en negrita:

"Función 30" significa apelar a dicha función, y para ello se ha de pulsar sucesivamente [ESC]R<n> como se indica en el texto.

[ESC]	significa pulsar la tecla marcada	[ESC]
[ENTER]	significa pulsar la tecla marcada	[ENTER]

2.0 ¿Qué es un Procesador de Textos?

La presión ejercida sobre las empresas hoy en día requiere incrementar la eficacia y el rendimiento en la moderna oficina. Los recientes avances en la electrónica han traído a los ordenadores y a los equipos de tratamiento de textos dentro del alcance de oficinas de tamaño medio y pequeño.

El **tratamiento de textos** es la técnica que permite a los ordenadores manejar textos a menudo combinados con datos de diversas maneras. Los textos de todas las clases incluyendo cartas e informes, pueden prepararse rápidamente con un sistema de tratamientos de textos. Un procesador de textos (word processor) es una máquina de escribir con una diferencia: permite registrar el texto introducido a través del teclado y almacenarlo en disco antes de imprimirlo. Se pueden hacer alteraciones al **documento** y eliminar cualquier error que hubiera antes de imprimirlo.

El tratamiento de textos puede ampliarse hasta áreas tales como el envío de cartas circulares personalizadas y otros tipos de preparación automatizada de texto. Usando las posibilidades disponibles en Microscript, serás capaz de producir virtualmente cualquier clase de documentación de una manera que ahorrará tiempo y dinero.

El Microscript ha sido diseñado para ayudar aquellos que usan el tratamiento de textos en la oficina, y está ahora disponible en el microordenador CPC464 AMSTRAD.

2.1 El texto se introduce a través de un teclado...

El teclado es similar al teclado de una máquina de escribir tradicional: en él están colocadas las teclas alfabéticas y numéricas standard.

...proyectado en la pantalla del terminal de video...

En muchas maneras es similar a un aparato de televisión, y el terminal de video (VDU Visual Display Unit) muestra todo lo que se ha introducido a través del teclado.

Un **cuadratín** móvil de luz indica la posición reconocida por el ordenador en cada momento. Es lo que se denomina el **CURSOR** y puede adoptar la forma de un pequeño rectángulo iluminado o del pequeño guión de subrayar. Si se pulsa una tecla, el carácter correspondiente a ella aparecerá en la posición señalada por el cursor en ese momento. El cursor se mueve entonces una posición a la derecha, preparado para mostrar el siguiente carácter que se teclee.

...antes de ser depositado en el disco

El microordenador que efectúa tareas de tratamientos de textos, está montado como una unidad de **Consola** autónoma.

Junto con la consola (teclado y monitor) están las **ductoras de disco** (drives): las unidades que permite que la información sea **depositada** permanente y magnéticamente en el disco, y posteriormente **recuperada** para ser aprovechada.

2.2 Conviniendo lo que queremos decir por Caracteres...

Un **carácter** es la unidad básica y fundamental al hablar de textos. Es igual que **símbolo** y es por tanto cualquier letra, cifra, signo de puntuación, e incluso el **espacio en blanco** producido al pulsar la barra espaciadora del teclado; con todos ellos se pretende que aparezcan en pantalla como si fuera una **copia** de lo que se obtiene a través de la impresora.

...y palabras...

Los grupos de caracteres adosados unos a otros y tomados como una unidad, que van seguidos de un espacio en blanco, son los que se conoce en el tratamiento de textos como una **palabra** (también se denomina literal, cadena, sarta,...). Por tanto no sólo lo que se llama palabra en gramática es lo que entra dentro de esta categoría, sino simplemente retahílas de caracteres tal como \$5000 serán considerados como palabras dentro del tratamiento de textos.

... y párrafo

Cualquier serie consecutivas de palabras que formen un bloque de texto y que vaya seguido de **dos pulsaciones** de 'retorno de carro' la tecla **[ENTER]** es considerado un **párrafo**. Un párrafo puede por tanto ser tan corto como para contener un único carácter, o ampliarse tanto que sea exactamente la totalidad de un **documento**.

Eso implica que si empleas la pulsación de **[ENTER]** para terminar una línea, antes de alcanzar el margen derecho del documento -como haces cuando tecleas por ejemplo una dirección- no estas creando párrafos separados.

No es hasta que esa dirección esté completa, y necesites una **pulsación extra** de **[ENTER]**, para forzar una línea en blanco, cuando el sistema lo reconoce como final de un párrafo.

3.0 Arranque del Sistema

3.1 Enciende la fuente de alimentación...

Habiendo comprobado que el cable de red está enchufado correctamente, pon el conmutador de alimentación del monitor en la posición ON, y luego el conmutador del teclado del CPC464. Mientras que esté alimentado con energía, el piloto indicador rojo estará iluminado.

...antes de cargar el disco del sistema...

Incluso con la alimentación al sistema conectada, el ordenador es simplemente una colección de unidades independientes. Necesita ser programada con instrucciones que le hagan **funcionar** como un equipo de tratamiento de textos.

El programa con el **sistema operativo** debe cargarse desde el DISCO DEL SISTEMA cada vez que se enciende el ordenador. Una vez que el ordenador esté preparado la pantalla mostrará el mensaje típico de bienvenida o saludo.

Inserta el disco de sistema en la ranura abierta de la Unidad Ductora de Disco A. Inserta suavemente el disco en la ductora hasta que se asiente en su lugar. Teclea entonces:

[CPM[ENTER]

...para cargar el sistema operativo en la memoria

Cuando el disco está correctamente en su sitio, el programa será leído del disco y automáticamente traído a la memoria del ordenador en dos etapas. El piloto situado en la puertecilla de una unidad ductora de disco estará encendido mientras está cargando el programa.

Una vez que el sistema operativo haya quedado **implantado** en la memoria, la pantalla muestra un mensaje en la forma:

CP/M 2.2 Amstrad Consumer Electronics plc

A>

En que puedes apreciar la divisa típica del CPM, constituida por la pareja de símbolos **A>**.

3.2 Carga MICROSCRIPT en la unidad ductora A:

Cuando recibes tu copia del programa **MICROSCRIPT**, deberás haber sacado dos copias de él, que te servirán de **respaldo** si algo falla; y también habrás copiado los ficheros de los programas de trabajo sobre todos los discos que estés usando para tratamiento de textos cuando sólo dispones de una única ductora de disco. Recuerda activar la lengüeta de protección ante escritura en el disco maestro original antes de comenzar a copiar!. El disco maestro original no debiera ser usado de nuevo, a no ser que las otras copias de trabajo estén dañadas o se hayan borrado. Una de las dos copias que has sacado es tu copia de seguridad y la otra la copia de **trabajo**.

Consulta el manual del operador para el ordenador y su sistema operativo para mayores detalles de como sacar una copia de un disco.

Inserta la **copia de trabajo** del **MICROSCRIPT** exactamente de la misma manera que lo has hecho con la del disco del sistema, asegurándote que lo haces en la ductora de disco **A:** (como se identifica en CP/M). Como tendrás en pantalla seguramente la divisa típica del CP/M, que es un "aviso" ("prompt") que te indica que está **pronto** (preparado, presto) para recibir tus comandos:

A>

teclea **SCRIPT** bien sea en mayúsculas o en minúsculas y concluye pulsando **[ENTER]**.

Los documentos almacenados mediante MICROSCRIPT deben también serlo en el disco que contiene este programa de tratamiento de textos. Los discos sin el programa se usan como discos de información exclusivamente, (discos 'pasivos') y pudieras emplearlos siempre que dispusieras de un sistema con dos unidades lectoras de disco. Los discos de usuarios nuevos deben ser preparados **conforme** a las reglas del sistema para que pueda tratarse la información grabada en ellos. El proceso, se conoce como **inicialización, estreno, conformado del disco** ("formateo"), y se describe en las instrucciones del sistema operativo para tu unidad de disco AMSTRAD DDI-1.

...para obtener el menú principal

Siempre que está funcionando MICROSCRIPT, se muestran en pantalla una serie de opciones. Es el llamado MENU PRINCIPAL, que te permite las mostradas debajo:

C	Crear nuevo documento	Sección 4.0
E	Editar-revisar documento	Sección 7.0
R	Reformar un documento existente	Sección 9.0
S	Búsqueda y Sustitución	Sección 8.3
P	Imprimir un documento	Sección 18.0
F	gestión de Fichero	Sección 19.0
?	índice de documentos en disco	
*	salida al sistema	

Cada una de estas posibles actividades se describe en la sección indicada a la derecha.

Con el disco MICROSCRIPT alojado en la lectora A: elige tu opción.

3.3 Restaurando el ordenador

Cuando se pone en funcionamiento la máquina, es esencial que todos los circuitos internos y todos los valores de los parámetros sean **restaurados** a un estado predeterminado (normalmente las condiciones iniciales). Eso permite que el programa que controla el funcionamiento interno comience con las condiciones adecuadas para los parámetros.

Esta acción se llama **restauero**, o restauración del ordenador (en inglés RESET: Re-poner, Re-fijar, Re-establecer... un valor o condición).

4.0 Creación de un documento mediante MICROSCRIPT

4.1 Elige la opción de Creación en el Menú Principal

Si ya ha sido cargado en memoria el programa MICROSCRIPT, inserta en la dadora B: un disco de usuario ya inicializado. Cuando aparece el menú principal, pulsa la tecla C y con esta opción podrás confeccionar un documento.

El espacio de almacenamiento queda repartido automáticamente por el ordenador y organizado en 'recintos' claramente delimitados y denominados con un **nombre de referencia** que se conocen como **ficheros**.

4.2 Todo Documento debe estar identificado

Cada **nombrefichero** consta de tres partes:

Un identificador de la dadora de disco en el que está alojado el disco que contiene al fichero (en la forma A: o B:).

Si se omite este identificador el Microscript supondrá que el documento a crear va a quedar registrado en la dadora **corriente** (que es la vigente o ligada al sistema en ese momento).

Un nombre elegido por el usuario para ese fichero, y que puede contener ocho caracteres como máximo (incluyendo los espacios separadores) -CARTA001, por ejemplo.

Un 'apellido' o ampliación del nombre, que consta de tres caracteres como máximo y se usa para indicar la **clase** de fichero a la que pertenece (DOCumento, por ejemplo). Esta ampliación, también llamada 'nombre de familia' suele requerirse en algunos casos para hacer que el nombrefichero quede completo.

La parte del nombrefichero elegida por el usuario no puede incluir ninguno de los caracteres siguientes:

, < > ; : = ? * [] { }

El signo de puntuación (.) se usa -y sólo puede usarse para ello- para separar la parte del nombre de la parte del 'apellido' del nombrefichero. Si se incluye un punto en la parte del nombre, MICROSCRIPT acepta los tres primeros caracteres siguientes a dicho punto como la parte del apellido, o sea como clase, asignada a ese fichero.

Con la opción de Creación elegida en el menú principal, aparece en pantalla el mensaje indicando que se ha seleccionado la opción de Crear un nuevo documento, y a continuación solicita que se le diga el documento en cuestión.

Se debe entonces asignar un nombrefichero completo para dicho documento, tecléando y concluyendo con [ENTER]. Muchas instalaciones de MICROSCRIPT han sido configuradas para usar el disco del sistema en la ductora A: y el disco del usuario en la ductora B:. Para crear ficheros en el disco de usuarios debes por tanto dar el identificador del fichero en la siguiente forma:

B:<nomefich>.<fam>

B:CARTA001.DOC como la primera carta de la clase documento

B:FC/1068.VTA como factura 1068 de los documentos de venta

B:23JUN.REP como el reporte enviado de fecha 23 de junio

Observa que los ficheros debieran identificarse con nombres tan significativos como para que la clase y contenido de los mismos fuera inmediatamente patentes al operador. No es muy buena idea construir una serie de ficheros con nombres tales como Pepe1, Pepe2, Pepe3, ...Pepe57, ya que al intentar buscar cuál de ellos contiene la información vital que nos interesa puede resultar una labor ímproba.

Las reglas para los nombrefichero harían que los siguientes no fueran válidos:

B:AC:1068.INV Contiene un dospuntos (: en el nombre)

B:23.06.84.REP Contiene varios puntos dentro del identificador

Después de que hayas asignado y tecléado un nombrefichero válido, la pantalla se quedará automáticamente en 'claro' y puedes comenzar a teclear la información que deseas ver contenida en ese fichero.

5.0 Formato del texto en pantalla

5.1 El trazado de las líneas está definido por Reglillas...

Al igual que en una máquina de escribir tipográfica convencional el MICROSCRIPT dispone de **topes de tabulación** para el control del 'carro' de la máquina, (en este caso es la posición del cursor). Este sistema te permite estipular los márgenes izquierdo y derecho del documento, colocar los topes de tabulación, y otras posibilidades que afectan al aspecto de un documento.

MICROSCRIPT usa un sistema de Reglillas para este control. Dichas **reglillas** son en esencia una línea de puntos que marcan las columnas a lo ancho de la pantalla. En ella se colocan letras clave para definir ciertos puntos de control, como en el siguiente ejemplo:

W.....T.....T.....T.....R

...a las que se reclama para entrar en acción

El MICROSCRIPT tiene reservados en memoria diez **depósitos** para conservar estas reglillas, permitiendo que en cada momento haya hasta nueve reglillas diferentes -identificadas del 1 a 9 depositadas en ellos en cualquier momento. El depósito décimo -que está numerado como 0- contiene siempre la reglilla que se está usando en un documento (la **corriente o vigente**).

Cuando está funcionando el MICROSCRIPT los depósitos para las reglillas quedan automáticamente cargados con las reglillas **preceptivas** en el sistema. Además la reglilla 1 se carga en el depósito 0 cuando se pone en marcha el sistema. Estas reglillas estipuladas para omisiones pueden cambiarse, pero el procedimiento para hacerlo **no** cae dentro del ámbito de un manual standard de usuario MICROSCRIPT.

No hay límite a las veces en que pueden reclamarse las reglillas para dar forma a un documento, durante la creación o revisión del mismo. Cuando quiera que se elige una nueva reglilla, automáticamente se trasfiere desde el depósito pertinente una copia de ella hasta el depósito numerado como 0, y entra en vigor a partir de ese momento.

5.2 Pueden ponerse en acción hasta nueve reglillas diferentes

En cualquier momento de la creación o edición de documentos puede reclamarse que entre en acción una reglilla cualquiera de las definidas, haciendo que sea **recuperada** del depósito en que se encuentra, a través de la **función 30**, como en el siguiente ejemplo.

Para reclamar la reglilla 6, ejerce la función 30 **([ESC]R)**,
y luego pulsa 6
Para reclamar la reglilla 4, cita la función 30 **([ESC]R)**,
y luego pulsa 4

Las reglillas pueden quedar incrustadas en el texto sin ningún problema, ya que son desechadas al imprimir un fichero.

Para ocultar las reglillas y que no sean exhibidas en pantalla, puedes apelar a la función 12 de suprimir línea **([CTRL]Y)**.

5.3 Enmienda las reglillas cambiando los símbolos en la línea representativa

Puede cambiarse una reglilla en pantalla y luego dejarla depositada en MICROSCRIPT para posteriormente ser recuperada y usada en el texto.

Usa uno o más de los siguientes caracteres de control para cambiar lo estipulado en una reglilla:

Posición en la reglilla con NINGUNA **[CTRL]** FUNCION...(signo).

Topo de TABULACION **T**

FIJACION DEL MARGEN DERECHO(Right). .. **R**

FIJACION DEL MARGEN IZQUIERDO para una sección de texto con margen derecho 'rasgado' **W**

FIJACION DE MARGEN IZQUIERDO para una sección de texto con margen derecho alineado ("justificado") **J**

PRIMER CARACTER en un párrafo, cuando ha de estar 'sangrado' (indentado) hasta una columna diferente a la del margen izquierdo. **I**

CENTRO de una línea de texto. El texto queda centrado cuando al escribirlo se cita la función 27 de centrar texto **([ESC]C)**. **C**

Posición de dígito en el CALCULO numérico
(la descripción completa del calculador en la sección 16) #

Efectúa los cambios necesarios en tu reglilla. Para quitar un caracter de control de una reglilla basta teclear **encima** de dicho caracter el signo punto.

Mueve el cursor a la línea arriba de la reglilla y luego bájalo pasando por encima de dicha reglilla para hacer que se **active**, o pase a ser la reglilla **corriente**.

Para hacer que **vierta** a uno de los depósitos la reglilla expuesta en pantalla recurre a la función 29 **([ESC]Sn)**, siendo n el numero de depósito en memoria que la identifique.

Una vez que una reglilla ya está guardada en el depósito, puedes hacer que la **uses** para establecer el formato de las líneas, si citas la función 31 **([ESC]Un)**.

5.4 Puedes copiar una línea...

Si primeramente **"despegas"** la línea que vas a copiar en otro sitio, colocando para ello el cursor al comienzo de la línea y apelando a la función 55 de COGER UNA LINEA DE TEXTO **([COPY])**, puedes luego **'pegar'** esa línea de texto en la sección del documento en que las requieras. Para esta operación 'pegado' sitúa el cursor en la posición dentro del documento donde quieres que quede la copia de esa línea, y luego cita la función 56 de DEJAR UNA LINEA DE TEXTO **([ESC][COPY])**.

La línea de texto que ha sido recogida o despegada, puede dejarse o pegarse en tantos sitios como sea necesaria. Dicha línea de texto queda guardada en una parte única de la memoria ("la despensa de línea") hasta que tú hagas que se escriba encima de ella otra que desees.

6.0 Corrección de errores durante la Introducción de texto por teclado

6.1 Re-teclea inmediatamente después de observado el error

Hay posibilidades (mayores y menores) de que cometas equivocaciones al teclear y te des cuenta de la equivocación inmediatamente después de haberla cometido. Puedes citar la función 2 **(DEL)]** para borrar ('delectar') el último carácter recién tecleado. Observa que esta operación se efectúa en dos pasos:

- El cursor retrocede ('arredra') una posición y se sitúa sobre el carácter que se desea eliminar;
- El carácter situado ahora en la nueva posición señalada por el cursor es eliminado;

Es equivalente a la famosa tecla **correctora**. Una vez eliminado el error puedes continuar tecleando normalmente.

6.2 Otras clases de errores deben ser localizados...

Cierta cantidad de errores cometidos al teclear no serán detectados hasta que posteriormente se revise el documento, y subsecuentemente se **edite** (que es la palabra informática usada para volver a proyectar en pantalla un documento ya guardado en el disco y efectuar alteraciones, correcciones, etc.). Dado que esos cambios solamente pueden efectuarse cuando se ha colocado el cursor en la posición correspondiente al carácter, palabra, línea, etc., donde ha de comenzarse esta corrección, deben usarse las funciones que ejercen el control sobre el movimiento del cursor para desplazarlo hasta el sitio deseado.

...y situar el cursor mediante las funciones...

El cursor puede moverse hacia la izquierda ('arredra'), hacia la derecha ('avance'), hacia arriba ('suba') y hacia abajo ('baje'). Las funciones que controlan estos movimientos del cursor son las numeradas en el MICROSCRIPT como funciones 3-6, y corresponden a las teclas rotuladas con las **flechas** pertinentes.

AVANCE (mover el cursor una posición a la izquierda Flecha izquierda

ARREDRE (retrocede) mover el cursor una posición a la derecha Flecha derecha

SUBA (mover el cursor una línea hacia arriba a partir de la posición corriente y siguiendo la misma columna) Flecha arriba

Manteniendo pulsada esta tecla el tiempo suficiente puedes hacer que el texto en pantalla se 'deslice' hacia arriba' hasta 23 líneas. Si suena un pitido al usar esta función de correr el texto, indicará que o se ha alcanzado la parte superior del documento, o que la parte de la memoria donde está contenido el texto está llena.

BAJE (mover el cursor una línea hacia abajo a partir de la posición corriente y siguiendo la misma columna de texto). Flecha abajo

Manteniendo esta tecla pulsada haremos primero que el cursor llegue al fondo de la pantalla, y luego continúe corriendo el texto que aparece en pantalla hacia arriba.

Las teclas de control del cursor son **autorepetitivas**. Si se mantienen pulsadas durante un momento, su efecto se repite automáticamente. Usa sin embargo esta facilidad con precaución.

Para situar el cursor al final de un documento, apela a la función 57, de bajar hasta el final del fichero, y que se consigue con **([ESC]B)** .

Para hacer que el cursor suba hasta el principio del documento que se está tratando, cita la función 53 **([ESC]T)** DE SUBIR HASTA EL INICIO DEL FICHERO.

...o con las funciones de movimientos de más posiciones

Para mover el cursor más rápidamente por todo el texto, pueden ejercerse las funciones de movimiento alrededor del texto, con las que se desplaza teniendo en cuenta las unidades de textos: palabras, topes, líneas, etc. Usando por ejemplo la función #14 **([CTRL]→)**, harás que el cursor avance de un salto hasta el extremo de la línea actual donde está situado. Son las siguientes funciones:

TABULE (mover el cursor avanzando hacia la derecha hasta que alcanza el siguiente tope de tabulación estipulado en la **reglilla vigente**) [TAB]

AVANCE PALABRA (mover el cursor hasta el primer caracter de la siguiente palabra a la derecha dentro de la misma línea) [CTRL]T

RETROCEDA-ARREDRE-UNA PALABRA (mover el cursor hasta el primer caracter de la palabra previa a la izquierda a lo largo de esa línea) [CTRL]R

AVANCE línea (mover el cursor hasta el extremo derecho de la línea) [CTRL]→

RETROCEDA-ARREDRE-LINEA (mover el cursor hasta el extremo izquierdo de la línea) [CTRL]←

RETROCEDA-ARREDRE-PLANA (subir el cursor hasta la línea superior de la pantalla) [ESC]↑

AVANCE PLANA (mover el cursor hasta la línea inferior de la pantalla) [ESC]↓

SUBA AL INICIO DE PLANA, que es mover el cursor hasta la esquina superior izquierda de la pantalla [CTRL]↑

6.3 Habiendo situado el cursor en la posición a corregir efectúa la supresión...

Usa la función apropiada para eliminar el error (véase sección 8.2)

...o la inserción que se requiera...

Suprime los errores antes de pasar a **insertar** texto adicional.

Como un ejemplo, corrige la frase:

'Devuelve por favor este impreso adjunto'
 'Devuelve por favor el impreso adjunto'

Efectúa estas modificaciones como una serie de pasos:

Suprime la palabra 'este' recurriendo a la función 2 **([DEL])**, por ejemplo.

Apela a la función 40 **([ESC]I)** para **entrar** en el modo de insercción, y teclea lo que debas intercalar.

Apela a la función 42 **([ESC][ESC]I)** para **salir** del modo de insercción.

...antes de re-componer el formato del documento

Durante el curso de tus supresiones e inserciones de texto, es probable que hayan aparecido en el texto espacios en blanco que no son convenientes. Utiliza las funciones concernientes al cursor para situarlo en el punto apropiado del texto y luego apela a la función 9 de SACAR TEXTO **([CLR]P)** para quitar los espacios o caracteres. Alternativamente puedes insertar espacios extra apelando a la función 8 de METER TEXTO **([CTRL]N)**.

Si el texto se hace demasiado largo con respecto a la longitud de línea especificada en la reglilla, automáticamente se 'voltarán' las palabras pasándola a la siguiente línea. Las palabras son consideradas indivisibles por el MICROSCRIPT.

6.4 Lee y Comprueba el documento...

Para comprobar el contenido de los documentos antes de guardarlos, puedes volver al comienzo del mismo empleando la función 53 **([ESC]T)**.

Habiendo alcanzado el inicio del documento, repasa el texto usando las funciones de movimiento del cursor, haciendo pausas para efectuar cualquier alteración o enmienda en los errores que detectes.

...antes de hacer que lo guarde en disco

Cuando no hay más correcciones y deseas dejar archivado el documento apela a la función 52 **([ESC]E)** de SALIR (EXIT) con el DOCUMENTO GUARDADO.

Esta función se utiliza normalmente para dejar el modo de edición y regresar a menú principal. Durante la ejecución de esta función todo el texto que se ha estado editando queda **guardado** en el disco, y al mismo tiempo se produce automáticamente un fichero de **respaldo** (back-up) en el mismo disco que corresponde al fichero que se ha editado tal y como estaba antes de la última sesión de edición.

6.5 Devuelve un documento al disco frecuentemente

Aunque MICROSCRIPT es un producto demostrado y comprobado, siempre es conveniente guardar en disco el texto que se está escribiendo o revisando, a intervalos regulares de tiempo. Se recomienda que el documento corriente sea guardado cada dos páginas, apelando para ello a la función 52 **([ESC]E)** de SALIR Y GUARDAR DOCUMENTO.

Cuando se ha editado un documento, la versión previa del mismo es automáticamente retenida en el disco bajo el mismo nombre del fichero, pero especificado de clase **.BAK** (back-up = respaldo) para indicar que es una **copia de seguridad**. Esta versión del documento quedará sobrescrita cada vez que la copia principal del documento es traída a memoria para **edición** y una vez con texto modificado vuelva a escribirse en el disco.

Nota: Para salir del modo de creación, o del modo de edición, sin que el texto corriente quede guardado en disco recurre a la función 50 -pulsando **[ESC]Q** para que ABANDONE EDICION.

7.0 Revisión de documentos usando MICROSCRIPT

La revisión de documentos **-la edición-** cubre todas las operaciones involucradas cuando se está examinando y alterando un documento que ya estaba archivado en el disco.

Habiendo creado un documento y habiéndolo archivado ya en el disco, puedes desear volver a comprobar el texto antes de imprimirlo. En este momento puedes efectuar inserciones, supresiones y otras enmiendas, apelando a las mismas funciones que usastes durante la introducción del texto primitivo.

7.1 Reclama el texto a la pantalla

Con el menú principal mostrado en la pantalla, elige la función de **edición-revisión**, pulsando la letra E. En pantalla aparecerá un mensaje indicando que se ha elegido dicha actividad, y se te pedirá que especifiques el **nombre fichero** del documento que deseas sea traído del disco a la pantalla.

Por ejemplo, si especificas:

A:JBK.DOC corresponderá a un documento del disco alojado en la ductora A

B:020384.CTA será de un fichero en la ductora B

7.2 Situa el cursor recurriendo a las funciones de movimiento según unidades de texto...

Dado que los cambios únicamente pueden efectuarse con el cursor en la posición correcta, puedes usar una de las diversas técnicas disponibles para mover el cursor hasta el punto requerido dentro del documento.

El uso de estas funciones de movimiento se explicó en la sección 6.

...o a las funciones de control del cursor

Las funciones de control del cursor te permiten hacer el corrimiento del documento, bien sea verticalmente o bien de izquierda a derecha por todo lo ancho y largo de la pantalla. Con el cursor a comienzo del documento, puedes repasar el texto haciendo que se desplace verticalmente y haciendo una pausa cuando detectes errores y desees corregirlos.

El uso de estas funciones de control del cursor se explicó en la sección 6.

7.3 Alternativamente, puedes hacer que 'busque' y 'halle' una determinada función

Utilizando la función de BUSQUEDA de una determinada palabra, puedes localizar la primera aparición de dicha palabra en el texto.

La función de BUSCAR TEXTO que es la función 45 ([ESC]F), puede ser ejercida bien sea para buscar cierto trozo de texto dentro del fichero corriente, o bien para hacer que pase a través del fichero hasta un punto determinado.

Para localizar un trozo de texto usando la función 45 de **búsqueda de texto**, ha de teclearse la palabra o frase que se desea **hallar**, con el cursor situado al comienzo de una línea en blanco, y luego apelar a la función 45 ([ESC]F).

El MICROSCRIPT escruta el fichero en busca de la primera aparición en el texto de un trozo que concuerde con lo tecleado, y que sea la primera que aparece después de donde se tecleó el texto. Observa que la línea del cursor donde se escribió el texto a buscar es **eliminada** antes de pasar a efectuar la operación de búsqueda.

Cuando MICROSCRIPT ha encontrado el texto a buscar, emite un pitido y sitúa el cursor en la posición correspondiente al texto encontrado.

Si el texto tecleado no aparece en el texto corriente antes del final del fichero, aparece un mensaje en la parte superior de la pantalla indicándolo así, y el proceso de edición vuelve a comenzar a partir del inicio del documento.

8.0 Efectuando enmiendas al texto que has guardado

8.1 Los cambios pueden hacerse mediante la inserción de nuevo texto

Cuando hayas situado el cursor en la posición donde quieres intercalar el primer carácter, apela a la función 40 **([ESC]I)** de ENTRAR EN EL MODO DE INSERCIÓN.

El texto que añadas se acepta desplazando todo el texto situado a la derecha del cursor en la línea corriente, para hacer sitio a los nuevos caracteres que estás incluyendo.

Como un ejemplo de este proceso de inserción, la frase 'el borrador final de' a de **insertarse** en la siguiente frase, inmediatamente delante de la palabra 'agenda'...

Adjuntamos para su información la agenda de la reunión y apreciaríamos sus comentarios.

El procedimiento a seguir:

Situar el cursor sobre la PRIMERA letra "a" de la palabra agenda.

Apelar a la función 40 de entrar en el modo inserción **([ESC]I)**.

Teclear la frase que se desea insertar.

El efecto conseguido sería:

Adjuntamos para su información el borrador final de la agenda para la reunión y apreciaríamos sus comentarios.

Apelar a la función 41 **([ESC][ESC]I)** de SALIR DEL MODO DE INSERCIÓN.

8.2 Se puede suprimir texto por caracteres...

La función de DELECTAR CARACTERES que es la función 2 **([DEL])**, se utiliza normalmente para corregir los errores detectados antes de seguir, o para efectuar eliminaciones carácter a carácter durante una revisión posterior. (Véase sección 6.1 para más detalles).

...o por palabras...

Esta función se usa para eliminar el texto desde la posición corriente del cursor hasta el siguiente espacio en blanco que aparezca, permitiendo así que sea suprimida parte de una palabra o toda ella.

La función 'DELECTAR' PALABRA, que es la función 37 **([CTRL]V)**, opera en la forma siguiente:

Sitúa el cursor en la primera letra de la palabra que quieres eliminar.

Apela a la función 37 de suprimir palabra, pulsando **([CTRL]V)**.

Si el cursor no está situado 'dentro' de una palabra, emitirá el pitido de aviso.

...o por líneas

Una manera alternativa de suprimir partes del texto es apelar a las funciones de BORRAR HASTA FINAL DE LINEA y de BORRAR DESDE PRINCIPIO DE LINEA, con las que se elimina todo el texto situado en esa línea a la derecha o a la izquierda del cursor, respectivamente.

Para suprimir texto a la derecha del cursor, apela a la función 35 **([ESC]D)**.

Para suprimir texto a la izquierda del cursor, recurre a la función 36 **([ESC]I)**.

8.3 Sustituyendo partes de texto por otras a lo largo del documento

Si sólo quieres localizar - y situar el cursor- en un determinado punto del texto pero no sustituir ese texto por otro, puedes apelar a la función 45 de hallar texto, que hemos mencionado en la sección 7.3.

La opción BUSQUEDA - SUSTITUCION disponible en el menú principal permite que una cierta **serie de caracteres** consecutivos sea sustituida por otra. El número de caracteres en ambas series no tienen por que ser el mismo.

La opción de BUSQUEDA y SUSTITUCION es accesible en el menú principal pulsando la tecla S.

El sistema pide entonces un nombrefichero válido sobre el que va a operar en esta modalidad de búsqueda y sustitución . Teclea primero el nombrefichero y luego la serie de caracteres que deseas buscar dentro de ese fichero **-el argumento de búsqueda-**.

El MICROSCRIPT dispone de tres modalidades dentro de esta actividad de búsqueda:

1. Búsqueda y recuento de las veces que aparece
2. Búsqueda y supresión
3. Búsqueda y sustitución

La opción 1 escrutará todo el fichero anotando cada aparición en el mismo de la serie de caracteres especificada como argumento, y dará como resultado de la operación el número de veces en que esto ocurre.

La opción 2 eliminará del fichero la serie de caracteres especificada, tantas veces como aparezca.

La opción 3 hallará todos los sitios donde aparece esa serie de caracteres y reemplazará en ellos la serie de caracteres especificada para sustitución. Al elegir esta opción, el sistema pide que se le especifique la serie de caracteres a usar como sustitutiva **-el argumento sustitutivo-**.

8.4 Otras funciones de edición adicionales

Las funciones que serán provechosas pero que todavía no habíamos comentado son:

La función 13 de ABRIR LINEA en blanco **[[ESC]O)**

Eso permite intercalar texto adicional en el espacio abierto, o simplemente mejorar la apariencia del texto.

La función 18 de TRANSFORMAR A MAYUSCULAS ([ESC]>)
Convierte a mayúsculas todos los caracteres situados a la derecha del cursor y hasta el extremo de la línea, si el cursor está situado precisamente en un espacio en blanco; en los demás casos sólo los caracteres que aparecen antes del siguiente espacio en blanco.

La función 19 de CONVERTIR A MINUSCULAS ([ESC]<)
Convierte a minúsculas todos los caracteres situados a la derecha del cursor y hasta el extremo de la línea, si el cursor está situado precisamente en un espacio en blanco; en los demás casos sólo los caracteres que aparecen antes del siguiente espacio en blanco.

La función 32 de SUBGUIONADO de la línea corriente ([ESC]-)

La función 33 de SUBRAYADO de la línea corriente ([ESC]_)

Observa que al **subrayar** haces que en la impresión el documento no se ocupe una línea horizontal adicional, como sucede al emplear los **guiones**.

8.5 El texto revisado se archiva en disco...

Al reclamar un documento usando la función de EDICION se transfiere en realidad una copia del documento existente en el disco, hasta la memoria de pantalla. Hasta que hayas concluido la edición y hayas vuelto a guardar la copia revisada, el documento original que quedó en el disco permanece sin cambios.

...guardando el documento en disco

Para conservar en disco la versión revisada de un documento, apela a la función 52 ([ESC]E), de salir de la edición guardando el documento.

Alternativamente, puedes **abandonar** la edición de un documento, sin reflejar en el original que hay en el disco ninguno de los cambios que hayas hecho durante la sesión de edición. Para hacer eso, recurre a la función 50 ([ESC]Q).

Siempre que se haya reclamado a pantalla un documento para su revisión, queda conservada en el disco una copia de ese documento, en forma de fichero con el mismo nombre que tenía, pero con la clase o nombre de familia .BAK (bak-up = respaldo) en lugar de la que tuviera definida. Las subsecuentes ediciones del mismo fichero irán sobrescribiendo este fichero de respaldo; quedando automáticamente suprimidas las versiones más anteriores.

Ejemplo:

Se crea un documento titulado CARTA1.DOC. Después de haber sido revisado por primera vez, la versión primitiva y sin modificar se convierte en CARTA1.BAK.

En la siguiente ocasión en que se actualize CARTA1.DOC, se volverá a escribir CARTA1.DOC sobre la vieja versión de respaldo CARTA1.BAK.

Si requieres copias adicionales de las diferentes "versiones" durante la edición, se puede sacar una copia del documento sobre el que se está trabajando bajo un nuevo nombrefichero (usando las operaciones de manejo de ficheros descritas en la sección 19 de este manual).

9.0 Re-composición del formato

Esta opción, R, del menú principal del MICROSCRIPT debiera elegirse siempre que desees llevar a cabo cualquiera de las siguientes actividades:

- Cálculos en el documento
- Extender las líneas de texto (función 44)
- Re-composición del formato de párrafos enteros (función 28)
- Recorte y pegado de secciones de texto (funciones 20-26 inclusive)

La opción de Re-composición del formato- es por tanto más amplia que lo que su nombre indica, aunque en principio se usará más a menudo para ordenar y dejar más pulcro un documento, e introducir datos numéricos que requieren cálculos, inmediatamente antes de imprimir el documento.

El cálculo con MICROSCRIPT se explica con detalle en la sección 16.

Cuando eliges en el menú principal esta opción, se te pedirá que especifiques el nombre fichero con que el documento está registrado en el disco.

9.1 Extender una línea

La función de extender una línea permite que el texto sea **alineado** con respecto al margen derecho hasta el final del párrafo corriente, sin efectuar una verdadera operación de cambio de formato. Apela a la función 44 ([ESC]X), cuando desees aprovechar esta posibilidad.

Como ejemplo, supongamos que desees sacar el siguiente párrafo de texto, con las terminaciones de líneas en el lado derecho **justificadas**. Al introducir el texto, su apariencia será similar a:

- ★ **La tecnología de las oficinas ha evolucionado en gran manera desde las primitivas máquinas de escribir eléctricas, durante la década de los 20 y de los 30. Quizá el mayor avance fue hecho inmediatamente después de la Segunda Guerra Mundial, en Inglesaamerica por IBM.**

Sin embargo después de recurrir a la función 44 **([ESC]X)** el mismo párrafo aparecería **alineado**, como:

- ★ **La tecnología de las oficinas ha evolucionado en gran manera desde las primitivas máquinas de escribir eléctricas, durante la década de los 20 y de los 30. Quizá el mayor avance fue hecho ineditamente después de la Segunda Guerra Mundial, en Inglesaamerica por IBM.**

Esta función 'justificará por la derecha' un párrafo de cualquier tamaño, desde una línea hasta un fichero completo del disco; pero debe recurrirse a ella sucesivamente para cada **párrafo**. (Observa que el alineamiento por la derecha del margen durante la operación de re-composición del formato (función 28) sólo puede usarse en un párrafo con menos de 23 líneas, y que sobrepase a la línea siguiente.

Si durante la ejecución de esta función encontrara una línea que sólo contiene una palabra, se emitiría el pitido de aviso y MICROSCRIPT saltaría a la línea siguiente.

9.2 Cambio de formato de un párrafo

Situa el cursor en cualquier parte del párrafo cuyo formato deseas cambiar, y luego apela a la función 28 **([ESC]J)** de RE-COMPONER PARRAFO. El MICROSCRIPT inmediatamente reformará el párrafo de acuerdo con el nuevo formato de márgenes y topes de tabulación contenido en la reglilla activa en ese momento.

Antes de apelar a esta función, puede que desees usar otra reglilla. Eso ya se ha explicado en la sección 5.

9.3 Recortando y pegando con MICROSCRIPT

Una de las ventajas del tratamiento de texto es que se pueden mover secciones de texto de un lado a otro dentro de un documento, o entre dos ficheros diferentes, sin la necesidad de volver a teclear nada.

El término **RECORTE** guarda relación con el marcaje de un bloque de texto en la pantalla y la transferencia de una copia de ese bloque de texto a un depósito especial en la memoria del **MICROSCRIPT**. Se suele mencionar como memoria de **montaje**.

En esta etapa de 'recortado' del trozo a mover, se dispone de tres opciones:

- Si el bloque recortado va a quedar relleno de espacios en **BLANCO**, el texto se borrará de la pantalla; así se permite insertar otro texto alternativo en ese punto.
- Como alternativa, puede hacerse que el texto recortado continúe quedando en esa posición. Esa función se denomina a veces **copiar bloque** ya que deja ('Left') el texto original sin alteración.
- Finalmente el texto recortado puede ser quitado ('Removed'), operación que se conoce como **mover bloque** dado que el recortado queda suprimido cuando es transferido al otro sitio del texto. El texto subsiguiente se corre para rellenar el espacio que ocupaba el recortado. Observa que cualquier texto que permanezca en las líneas de las que se ha quitado un bloque quedará sobrescrito cuando el texto debajo del bloque se corra hacía arriba de la pantalla.

El **PEGADO** es la operación opuesta a la de recortado, y guarda relación con depositar una copia de texto conservado en la memoria de montaje en otro punto dentro del mismo documento, o en otro punto dentro de otro fichero. Al igual que con las modalidades de recortado, el **MICROSCRIPT** ofrece tres modalidades para el pegado:

- La de **SOLAPANDO (OVERLAY)** en que se monta el texto en el punto requerido escribiéndose encima del texto allí existente. Es por tanto una operación 'destructiva', ya que cualquier texto que allí hubiera queda sobrescrito y por ende perdido.
- La modalidad más segura es la de **INSERTANDO**, en la que se abre una nueva línea en blanco para cada una del texto que se está pegando, de manera que el texto existente no se pierde.
- La de **CODEANDO (ELBOW = codo)**, habitual cuando se desean mover columnas de texto, ya que todo el texto existente se corre hacía la derecha. Pero observa que si se sobrepasara el margen derecho, los caracteres sobrantes se pierden.

9.4 Las secciones de texto pueden recortarse de un área...

Para aprovechar la facilidad del MICROSCRIPT de recorte y pegado, elige la opción R en el menú principal para Re-componer el formato del párrafo.

Para que RECORTE un área de texto, el texto debe señalarse **marcando** la esquina superior izquierda del mismo y la esquina inferior izquierda. Situa el cursor en la esquina superior izquierda de la sección requerida y **marcálo** apelando a la función 20 **((ESC])O**.

Luego mueve el cursor hasta la esquina inferior derecha de la sección requerida, y marca ese final del bloque apelando a una de las tres modalidades de recortado:

- Función 21 recortado poniendo el bloque marcando a blancos **((ESC])B)**
- Función 22 recortado dejando el bloque **((ESC])L)**
- Función 23 recortado quitando el bloque y corriendo el texto subsiguiente **((ESC])R)**

Observa que en ese momento el bloque así marcado queda depositado en la llamada memoria de montaje que el sistema tiene en activo. Observa también que esta posibilidad de recortado y pegado de bloques está restringida a una **plana** o texto presente en pantalla.

..y pegado en otra parte

La operación de pegado recupera de la memoria de montaje una copia del texto allí conservado y lo monta o pega en el lugar señalado por la posición corriente del cursor.

Para hacer esta operación de pegado (y la puedes llamar de 'empastado: = pasting'), situa el cursor en el punto donde deseas montar el texto que recortaste, y luego usa una de las tres modalidades de pegado:

- Función 24 pegado solapando ([ESC]*O)
- Función 25 pegado insertando ([ESC]*I)
- Función 26 pegado 'codeando' ([ESC]*E)

El MICROSCRIPT te permite que uses tantas veces como desees el contenido de la memoria de montaje, que solamente se cambia al efectuar una nueva operación de recortado, o cuando se sale de la corriente sesión con MICROSCRIPT.

10.0 Realce de textos con MICROSCRIPT

El MICROSCRIPT dispone de una cierta variedad de funciones que pueden usarse para adornar un determinado texto **realzando** partes de él.

10.1 MICROSCRIPT completamente en-guiona...

Cuando se requiere realzar mediante **guiones** una determinada línea, automáticamente se inserta una línea en blanco por debajo de la señalada por el cursor, y se colocan los guiones desde el principio de texto hasta el final del texto presente en la línea corriente.

Para efectuar esta operación apela a la función 32, **([ESC]-)**

Al final de la operación el cursor queda situado en el extremo de la línea de guiones.

Esto es enguionado tal y como aparece al imprimir

...o subrayado continuo

Se aplica el mismo principio que para el caso anterior, excepto que al imprimir los caracteres **subrayados** no ocupa ninguna línea horizontal adicional, y así el texto produce un verdadero efecto de subrayado.

Para efectuar esta operación apela a la función 33, **([ESC]_)**

Automáticamente se coloca un único puntoycoma en el extremo derecho de la pantalla y los caracteres **raya** se colocan una línea por debajo de la línea de texto. Estos caracteres **raya** pueden revisarse cuando sólo van a subrayar palabras específicas.

Así es como aparece el subrayado en la pantalla del monitor:

Esto es subrayado tal y como aparece al imprimir

IMPORTANTE: esta es una de las poquísimas ocasiones en que MICROSCRIPT no refleja en pantalla una exacta representación de lo que aparecerá al imprimir el texto.

10.2 Se puede estipular múltiple impacto del cabezal de impresión...

Hay tres técnicas que permiten al cabezal de la impresora golpear más de una vez en la misma posición correspondiente a un único carácter:

Sobre golpeo - en que se imprime dos líneas de textos sucesivas sobre la misma línea de papel.

Resaltado - en que cada carácter se imprime dos veces exactamente en la misma posición.

- En que un carácter golpea también dos veces, pero con el cabezal levemente desplazado una pequeñísima cantidad entre los dos impactos;

Aunque los comandos para imprimir encima, resaltar, y colocar en negrita sí quedan señalados en pantalla, obviamente el efecto sólo se ve en el momento de imprimir el documento.

...para imprimir dos líneas de texto dentro de una sola línea horizontal de papel...

Un puntoycoma en la última columna del texto (habitualmente columna 79) impide que la impresora efectue un **avance de línea** después de imprimir la primera de ella, con lo que la siguiente se efectúa a la misma altura de línea.

```
0000000000      ;  
-----
```

Así aparecería en pantalla, mientras que a la hora de imprimir el resultado sería el siguiente:

```
0000000000
```

...o el texto puede ser bien resaltado...

El MICROSCRIPT reconoce dos puntoycoma en las últimas dos columnas de una línea, como una instrucción para el cabezal de impresión golpee dos veces cada carácter para que el texto quede **destacado**.

Primeramente, debes usar la función 33 de subrayar (**[ESC]_**) y luego añadir un puntoycoma extra inmediatamente a la izquierda del que ya hay en dicha línea. Observa que el carácter raya no se imprimirá.

```
Capitulo 1: Los or extos           ;;
```

Con los dos puntoycoma seria:

```
Capitulo 1: Los or extos
```

...o puesto en negrita

El procedimiento para imprimir con caracteres en negrita es similar al de doble impacto, con un tercer puntoycoma situado en la antepenúltima columna de la línea.

Como ejemplo de esta técnica.

```
Capitulo 1: Los or extos           ;;;
```

Estos tres puntoycoma producen el siguiente efecto de negrita:

```
Capitulo 1: Los or extos
```


11.0 Estipulando el formato

Un equipo de tratamiento de textos produce la máxima calidad de presentación con un mínimo esfuerzo de teclado. El centrado automático de títulos, el alineamiento por la derecha y la tabulación son algunas de las características de las que están dotados con este propósito.

11.1 Se puede centrar un texto automáticamente...

Cualquier línea de texto puede **centrarse** entre los márgenes izquierdo y derecho en vigor en cada momento, apelando a la función 27 de centrar línea **([ESC]C)**.

Esta función hará que vuelva a escribirse el contenido de la línea señalada por el cursor, centrada alrededor del símbolo **C** que debe figurar en la reglilla corriente.

11.2 Ayudas mediante topes y sangrados

Los topes para la estipulación de topes de tabulación se esbozó en la sección 5. Pueden emplearse para conseguir diversas clases de indentación (sangrado, escotado, decalado) para el texto, como en los siguientes ejemplos:

(a)J.....I.....R.....

Con esta reglilla en vigor podemos mostrar el efecto de usar un código de indentado de párrafo (I) con lo que haremos que la primera línea de texto de cada párrafo empiece precisamente a partir de esa columna. El operador tiene que usar la tecla de TAB o las de flechas para mover el cursor hasta la posición indicada por la letra I.

(b)I.....J.....R.....

Es lo contrario del ejemplo anterior, y mantiene el comienzo de la primera línea de cada párrafo a la izquierda de donde comienzan las líneas siguientes del párrafo.

Si el símbolo I no se usara en la reglilla en ninguno de los dos ejemplos, se produciría un problema cuando se intentara re-componer el formato del párrafo.

(c)J.....T.....T.....T.....R.....

Hay situaciones en que el texto y los datos tienen que colocarse pulcramente en columnas: estipulando topes de tabulación ayuda en esas situaciones, como en el ejemplo:

```
Aluminio 16.09 gramos
Magnesio 17.98 gramos
Tungsteno 25.97 gramos
```

11.3 MICROSCRIPT 'volteará' el texto al llegar a los extremos de la línea

Cuando se alcanza el final de una línea de texto, el MICROSCRIPT automáticamente **volteará** -pasará a la línea siguiente- evitando que la palabra final sea dividida.

La siguiente serie de ejemplos muestra como MICROSCRIPT trata la composición de texto en los extremos de las líneas:

.....J.....R.....

Las palabras y cifras se

.....J.....R.....

Las palabras y cifras se acept

.....J.....R.....

Las palabras y cifras se
aceptan sobre una sola línea
hasta que no hay más espacio
disponible. Las palabras no
se rompen; son indivisibles.

12.0 Operando con MICROSCRIPT en modos diferentes

El MICROSCRIPT permite elegir entre siete diferentes MODOS de operación.

Para operar MICROSCRIPT en otro modo, debe estar en principio efectuando la actividad Edición, con el sistema siendo controlado bien a través del teclado o bien a través de una secuencia previamente programada para controlar un trabajo (esta programación se describe en la sección 22.

Estos modos de operación se han diseñado para proporcionar la máxima flexibilidad: el sistema puede **conmutar** -cambiar- entre modos siempre que se requiera.

El MICROSCRIPT cancela automáticamente todos los modos **alternativos** al regresar al menú principal, a no ser que haya comandos específicos que estén en vigor y que impidan el regreso.

Para **entrar** en un modo alternativo apela a la función 48 de ENTRAR EN MODO 'n', conseguida en el teclado pulsando **([ESC]M<n>)**

0 TEXTO EN VERTICAL

Este modo cambia la introducción normal de texto que pasa ahora a ser con el cursor moviéndose verticalmente hacia abajo, de manera que cada carácter tecleado aparece en la siguiente línea e inmediatamente debajo del anteriormente citado.

1 AUTO TAB

Al pulsar **[ENTER]** en el modo 1 se logra que el cursor "indente" hasta el margen izquierdo de la siguiente línea, en lugar de hacerlo como es habitual hasta el borde izquierdo de la pantalla. Esta facilidad es muy aprovechable cuando han de prepararse grandes cantidades de texto con **sangrados** (indentados, decalados, escotados).

2 RETENCION DEL FORMATO DE LINEA DURANTE LA MEZCLA DE DOCUMENTOS

Cuando se mezcla (merge = congregar, combinar) al texto corriente con el que se está trabajando, el procedente de otro fichero en disco, el texto entrante adopta automáticamente el formato de línea en cuanto a márgenes y tabuladores que esté en vigor en el documento corriente en el punto donde se efectua la mezcla.

El modo seis predomina sobre esta condición y conserva intacto el formato de línea del texto entrante en la mezcla.

3 'FRUSTRACION' DE TODOS LOS EQUIVALENTES VISIBLES

Este modo impide que los 'equivalentes visibles' de los caracteres de control sean interpretados como tales al incorporarse a la pantalla. Esta facilidad para la programación permite que los equivalente visibles sean almacenados en **memorias de abreviación** e insertados en un documento cuando se requiera, sin que las acciones que representan sean llevadas a cabo en ese momento.

Como un ejemplo, ~ sería normalmente interpretado como la acción de pulsar [ENTER], cuando fuera mezclado al documento con el que se está trabajando en memoria. Si el sistema se mantiene en el modo 3, el símbolo visivo equivalente a ese carácter de control sería incluido en esa notación empleada.

4 TECLEADO INVISIBLE

El modo 4 esta pensado como una ayuda al diseñador de sistemas. Todas las operaciones de escritura sobre la pantalla quedan anuladas. Eso permite que un operador programe el sistema para tratar funciones tales como la supresión de líneas y el borrado de pantalla en la 'retaguardia' (background). Los programas escritos en MICROSCRIPT tienen así una apariencia más profesional para el usuario final.

Un ejemplo de como debiera usarse este modo sería el de una rutina para preparar facturas. El cursor queda oculto mientras sea manipulado el texto y los datos por toda la pantalla.

5 MODO DE INSERCIÓN

El MICROSCRIPT tiene prescrito que para omisiones emplee la técnica de que cualquier carácter tecleado sea introducido y replicado en la pantalla en la posición señalada por el cursor en ese momento, sobre-escribiendo el carácter que hubiera en esa posición.

El modo 5 permite un método alternativo para la introducción de caracteres, en que el nuevo texto que se introduce automáticamente fuerza al texto existente en pantalla a desplazarse a la derecha. En este modo de inserción, los caracteres reflejados en la pantalla sólo pueden eliminarse usando algunas de las funciones de supresión de texto ('delección').

6 RETENCIÓN DEL FORMATO DE ESPACIO DURANTE LA MEZCLA DE DOCUMENTOS

Cuando se mezcla al texto sobre el que se está trabajando el procedente de otro documento, automáticamente adopta el formato correspondiente al documento corriente en el punto donde se efectúa la mezcla.

El modo 6 revoca esta condición prescrita y conserva intacto el formato del texto entrante.

7 APRENDIZAJE DE PROGRAMA

Este modo es de lo más valioso para un usuario de MICROSCRIPT que desarrolle programas de aplicación. El sistema automáticamente convertirá las 'secuencias de acciones' introducidas a través del teclado, a su 'notación simbólica equivalente' para incluirlas dentro del documento sobre el que está trabajando.

Con el sistema en modo 7, al pulsar la tecla **[ENTER]** se expondría en pantalla el símbolo **!** que es el equivalente visivo de esa acción que se ha mandado, y por tanto en este modo no se produciría el retorno del cursor a la siguiente línea de pantalla.

La utilización de este modo 7 para APRENDIZAJE DE PROGRAMAS se comenta detalladamente en la sección 22 de este manual.

Para salir de cualquiera de los modos alternativos, apela a la función 49, de SALIR DEL MODO 'n', tecleando para ello **([ESC][ESC]M<n>)**.

13.0 El interpretador del texto entrante en toda mezcla

Usando las posibilidades descritas hasta este momento, se puede preparar y revisar cualquier documento con gran aprovechamiento.

Hay sin embargo ocasión, en que un documento va a ser **confeccionado** a partir de otros documentos que ya están archivados en disco. El MICROSCRIPT incluye dos métodos mediante los cuales el texto ya guardado en un disco puede ser **congregado**, o 'mezclado', con el documento que en ese momento se está editando-revisando:

- Recuperando el texto que entra en la mezcla de un fichero depositado en disco Sección 14.0
- Recuperando el texto que entra en la mezcla de un **memoria de abreviación** Sección 15.0

En ambos casos antes de congregar ambos textos, el texto que entra en dicha mezcla pasa a través de un sistema de control denominado **INTERPRETADOR** que asegura que el texto traído hasta el sistema para ser congregarado con el que ya hay en él, no contiene los espacios innecesarios que pudiera haber para conservar el formato pertinente.

Si MICROSCRIPT se está usando como un procesador de texto, eso sería la medida en que se usa dicho interpretador.

Pero sin embargo, este interpretador ejerce una segunda función -y mucho más importante: la comprensión y ejecución de ciertos **símbolos de control** que obligan a llevar a cabo las **acciones** que indican. Esta capacidad que tiene MICROSCRIPT se transforma de un buen procesador de textos en un potente sistema capaz de efectuar operaciones pre-definidas complejas.

De esta manera, se puede usar MICROSCRIPT para programar la ejecución de muchas tareas incluyendo la de manipulación de datos, de manera que se puedan construir sistemas completos que ejecuten trabajos incluyendo facturar y recabar datos.

13.1 Usa el interpretador...

El interpretador puede usarse para mejorar el agregamiento de texto al documento sobre el que se está trabajando, del procedente de un fichero en disco, tal como un párrafo standard; mediante la inserción en el documento existente y al final de uno de sus párrafos de dos marcas de 'retorno de carro' forzando así una línea en blanco antes del comienzo del siguiente párrafo.

Para estas operaciones, se emplean **caracteres de control** (que efectúan acciones) y que pueden insertarse en cualquier parte del texto y ejercerán la misma función que si en ese punto se hubieran tecleado los comandos que representan.

...para ejecutar controles incrustados en un texto

El interpretador tiene predefinido los caracteres de control que pueden usarse en las operaciones de 'congregamiento' de textos.

El EQUIVALENTE VISIBLE de cada carácter de control consiste en un símbolo marcador o (**marca**), de los que raramente se usan en otras áreas del tratamiento de texto. El carácter que va inmediatamente detrás de esa **marca** representa la acción que se requiere.

En un sistema que se use en los países angloparlantes, el símbolo (^) constituye la elección ideal para esa **marca**.

14.0 Mezcla de ficheros

El MICROSCRIPT permite que al documento que se está editando se le **agregue** cualquier fichero existente en el disco. Esta operación recibe el nombre de mezcla o congregamiento de ficheros (merge).

Para congregare ficheros apela a la función 39 **([ESC]G)** de agregar ficheros.

Para conseguir que el sistema ejecute esta acción, ha de teclearse primeramente -y al comienzo de una línea libre- el nombrefichero que se quiere agregar, y luego teclear los caracteres de control **([ESC]G)**. Automáticamente las especificación del fichero que se va a agregar se elimina de la línea donde está, y el sistema procede a buscar en el disco el fichero especificado y congregarlo con el fichero sobre el que está trabajando.

Si el fichero especificado como fichero agregado no existe, el sistema emite un pitido y el cursor regresa al comienzo de la línea.

Todos los ficheros **agregados** usando este método son 'tramitados' por el interpretador (Sección 13). El interpretador efectua la lectura de cada carácter del texto que entra desde el disco, y si encuentra cualquier símbolo de control lo ejecuta en ese momento.

Los ficheros agregados son leídos desde su comienzo hasta su final usando MICROSCRIPT. Alternativamente, el fichero en disco puede estar **indexado** de manera que dentro de un simple fichero en disco están incluidas secciones separadas, cada una de las cuales puede ser tomada individualmente y agregada al texto que está editando el sistema.

14.1 Preparación del texto a agregar

La forma más simple de un fichero **agregable** es la de un párrafo standard pero puede prepararse una colección completa de párrafos standard para conseguir confeccionar rápidamente un texto.

Una secuencia de símbolos de control, tales como la correspondiente a 'Use reglilla 5' puede incluirse al comienzo del párrafo agregable de manera que al producirse el congregamiento con el que está editando el sistema, se emplee una reglilla concreta para agregar ese párrafo. El párrafo pudiera además terminarse con el símbolo de control correspondiente 're-composición de formato' para completar la operación de congregamiento de textos de acuerdo con un formato predeterminado.

Durante el congregamiento de textos, el símbolo ! incrustado en el texto que se agrega, se usa para hacer que el sistema se detenga y pida se le introduzca algo por teclado. Cuando el interpretador de MICROSCRIPT encuentra este carácter en el texto que se está agregando, el proceso de congregamiento se detiene temporalmente, para que pueda teclearse textos o datos directamente por teclado. El interpretador de MICROSCRIPT incluirá esta información tecleada dentro del documento que va a agregar. Para continuar la operación de congregamiento de textos, pulsa [ENTER]. Para abandonarla en cualquier momento, pulsa [ESC].

14.2 Indexación en los ficheros agregados...

El interpretador de MICROSCRIPT ofrece un método sencillo de **indexación** que permite conservar en un sólo fichero de disco uno o más párrafos o secciones de manera que sean **accesibles** de manera individual.

Cada una de estas secciones separadas tienen que tener una **clave numérica** que la identifique y diferencie de la siguiente, y ha de estar terminada por los dos símbolos))).

Cuando se especifica el nombrefichero de un fichero en disco, esta clave numérica puede añadirse como **calificativo** de dicho nombrefichero. El interpretador de MICROSCRIPT que efectúa la actividad de agregar el fichero, escrutará todo el fichero especificado hasta que encuentre esa clave numérica y agregará todo el texto que encuentre a partir de ella y hasta que detecte la marca final de sección -los dos))). Esta marca de final de sección puede aparecer en una nueva línea o puede directamente colocarse después del último carácter que pertenezca en esa sección.

El procedimiento para agregar al fichero existente en memoria, la sección indexada de otro fichero en disco, es similar al procedimiento normal para agregar ficheros completos, excepto que al especificar el fichero que se desea agregar el nombre fichero pertinente debe ir seguido de una **coma** y luego de la clave numérica correspondiente a la sección que se desea.

<nomefich.fam>, <claveordinal> y luego ESC G (función 39)

...de manera que el interpretador pueda cambiar entre secciones del fichero

Puede programar una operación de congregamiento de dos ficheros en uno, de forma que automáticamente durante la operación se pase a otra sección diferente del fichero en disco que se va a agregar. Esta facilidad pudiera aprovecharse para estructurar un sistema standard de correspondencia basado en ficheros y párrafos en disco. Se deduce que las cartas standard pudieran programarse de manera que saltaran a una terminación común tal como 'sinceramente suyo'.

La técnica empleada para efectuar estos 'saltos' de un lado a otro, se basa en el principio de emplear una **clave de indexación** dentro del fichero que se va a agregar al existente en memoria. Se describe con más detalle en la sección 22 de programación.

El interpretador de MICROSCRIPT escruta el fichero guardado en disco en busca de cualquier clave de indexación a partir del principio mismo del fichero. Por lo tanto la instrucción de 'salto' puede usarse para provocar tanto un salto hacia adelante como un salto hacia atrás dentro del fichero que se va a agregar.

14.3 Anidamiento al congregar ficheros

Puedes producir operaciones de congregamiento de ficheros 'en cascada', mediante la inclusión del comando de congregamiento dentro de un fichero que va a ser agregado al existente en memoria. Cuando el interpretador encuentra la función 39 de AGREGAR FICHERO, dentro de un fichero que a su vez ya se está agregando al existente en memoria, el sistema pasará a traer del disco y depositar en memoria dentro del fichero existente en ella el nuevo fichero especificado, antes de proseguir con la actividad original de congregamiento de ficheros.

Dado que el MICROSCRIPT permite hasta 'siete niveles de anidamiento' en la operación de congregamiento de ficheros, se puede preparar una completa 'jerarquía' de ficheros agregables para una aplicación.

15.0 Transfiriendo textos a través de memorias 'compendiadoras'

El concepto de memorias temporales para **retener** textos frecuentemente usados, o transferibles, se presentó en la sección 9 (operaciones de recortado y pegado, y memorias de montaje). Se dispone de diez memorias adicionales para archivar en ellas **abreviaciones, o 'dichos'** que pueden corresponder a textos, a comandos, o a una mezcla de ambos.

La capacidad de almacenamiento de estas memorias compendiadoras (habitualmente también llamadas '**dichos**' por ser las frases más usadas) está limitada a una línea de pantalla menos dos posiciones; pero puedes depositar en ellas texto adicional si en una de dichas abreviaciones incluye comandos que ponga en acción otras memorias.

Cada una de estas memorias de abreviación, tiene su contenido **prescrito** cuando se genera el sistema por primera vez. Sin embargo se pueden hacer cambios en cualquiera de ellas, o en todas, efectivos durante cada sesión de edición.

Las abreviaciones prescritas para el sistema se emplean para frases comúnmente utilizadas o para caracteres de control, y serán las mismas cada vez que el sistema se pone en funcionamiento.

Para reclamar a pantalla una abreviación prescrita, apela a la:

Función 47 de RECABAR ABREVIACION'n' **[[ESC]@n)**

Para verter a una de esas memorias una frase de texto o un comando, apela a la :

Función 46 de **RETENER** ABREVIACION'n' **[[ESC]#n)**

Si una de esas memorias está vacía, quedará depositado en ella el contenido de la línea del cursor. Sin embargo cuando ya hay información en una memoria concreta, esa información será mostrada en pantalla junto con el número identificativo de dicha memoria.

En ese momento tienes que indicar lo que deseas que se haga con el texto **retenido** en la memoria especificada. Puedes pulsar uno de estos dos caracteres:

- C Para cancelar (i.e. poner en 'claro') la información temporal, tanto la de pantalla como la retenida en la memoria.
- L Para cancelar sólo la información temporal escrita en la pantalla y dejar ('Leave') sin cambio la contenida en la memoria en cuestión.

16.0 Calculando con MICROSCRIPT

Una de las más importantes características del MICROSCRIPT es que dispone de un calculador de cinco funciones completamente programable que te permite acumular el total de las columnas vertical y horizontalmente, retener datos hasta en diez memorias numéricas y el alineamiento completo de datos según el 'punto' fraccionario.

16.1 Delimitando los campos del Calculador

Se usan también las reglillas para especificar la posición exacta de los campos calculables. Los caracteres '.' y '#' se sitúan en la reglilla para definir cada posición que interviene en el cálculo. La reglilla típica para cálculo podría ser:

```
.....#####.....#####.....#####.....  
.....#####.##.....#####.##.....#####.##.....  
.....#####.##.##.....#####.##.##.....#####.##.##..
```

Puedes emplear cualquier número de posiciones en una reglilla, restringida por la longitud de la reglilla. Cualquier campo puede tener hasta 13 posiciones numéricas sin ninguna restricción sobre la posición del punto fraccionario.

16.2 Introduce datos numéricos...

Para entrar en el modo calculador, elige en el menú principal la opción de RE-COMPOSICIÓN DE FORMATO. Asegurate que la reglilla activa contiene los símbolos marcadores apropiados para campos aritméticos. Observa que mientras estás en el modo de cálculo, todos los controles de cursor y las operaciones generales de control no son válidas.

Usa la tecla [TAB] para avanzar el cursor hasta el primer campo aritmético: el cursor se situará en la columna a la izquierda de dicho campo, y aparecerán tres ceros dentro de ese campo:

```
.....###.##..###.##..###.##....  
                                  0.00
```

Teclea las cifras requeridas usando el **punto fraccionario (.)** para pasar desde la parte entera de un número hasta las posiciones fraccionarias (si has definido un lugar para ello):

El valor 34.09, por ejemplo, aparecería como:

```
.....####.....  
34.09
```

En este momento, puedes elegir entre dejar el campo aritmético corriente o efectuar un cálculo con el valor que has introducido en él:

- Pulsa **[TAB]** para mover el cursor situándolo inmediatamente a la derecha del campo, que es un área no numérica.
- Alternativamente, usa uno de los cinco operadores para efectuar un cálculo:

Más	+
Menos	-
Multipliación	*
División	/
Porcentaje	%

Inmediatamente después de haber introducido uno de esos operadores, el campo queda en 'claro' esperando el siguiente valor que intervendrá en el cálculo.

...usando expresiones matemáticas...

La serie de operaciones es similar a la empleada en los calculadores científicos, ya que introduces la expresión de la misma manera en que la pronuncias. Por ejemplo:

9*33-12= es una expresión válida

Esta expresión sería introducida, y aparecería en pantalla, como:

```
.....####.....  
9
```

```
.....####.....  
* (no es visivo en pantalla)
```

.....####.....
33

.....####.....
297 (Producto 9x27)

.....####.....
- (no es visivo en pantalla)

.....####.....
12

.....####.....
= (no aparece en pantalla)

.....####.....
285

La expresión válida para calcular el 12% de 45, por ejemplo, sería:

$$45 * 12\% =$$

Pulsando una tecla numérica haces que se emita el pitido de aviso y que se introduzca un cero como valor del campo mostrado en pantalla.

...teniendo cuidado de no rebasar el campo

Si intentas colocar un valor que desborde la capacidad del campo numérico mostrado, el calculador automáticamente detectará el rebase, indicando por el pitido de aviso y mostrado en pantalla rellenando con ! el campo.

Intentando por ejemplo introducir el valor 61234 en el campo aritmético usando en el último ejemplo, rebasaríamos las cuatro posiciones asignadas al campo, y obtendríamos:

.....####.....
!!!!

16.3 Acumula los datos de un campo numérico...

Además de las cinco operaciones aritméticas admitidas por MICROSCRIPT hay seis rutinas aritméticas adicionales. Las primeras cuatro guardan relación con el contenido de un campo específico, y tienen por claves:

- S n Almacena un valor en la memoria numérica n (0 @ 9)
- R n Recupera el valor depositado en la memoria numérica n (0 @ 9)
- X Cambia el signo del valor introducido en el campo
- C Anula el valor del campo situado inmediatamente a la derecha del cursor.

Observa que estas memorias -registros- numéricos son de almacenamiento temporal y perderán todo su contenido cuando el sistema salga del modo de recomposición de formato.

...y calcula el total horizontal...

Cuando se ha estipulado más de un campo a lo ancho de la página que hay en pantalla, el total de todos los valores de los campos numéricos de una sola línea horizontal, puede ser calculado.

Para ello, sitúa el cursor inmediatamente a la izquierda del campo en que quieres colocar el total horizontal, y pulsa **H**. En el caso del ejemplo:

```
.....#####.....#####.....#####.....  
          123.45      234.56
```

Y luego al pulsar **H**:

```
.....#####.....#####.....#####.....  
          123.45      234.56      358.01
```

Observa que si las rutinas de cálculo encuentran en un campo un carácter que no es de índole numérica, el cálculo del total horizontal no puede proseguir más allá del margen izquierdo. En el siguiente ejemplo, la letra E que aparece en el segundo de los campos, impediría que el valor asignado al primero de los campos interviniera en el cálculo del total:

```
.....#####.##.....#####.##.....#####.##.....#####.##.....
      1000.00          E          1000.00          1000.00
```

...y el total vertical

Cuando se ha introducido una serie de valores en unos campos aritméticos que forman una columna vertical, el total acumulado en ellos puede calcularse situando el cursor inmediatamente a la izquierda del campo donde quieres colocar dicho total, y pulsando la tecla **V**.

Las rutinas aritméticas calcularán el total correspondiente a todos los valores colocados en la misma columna vertical. Es posible por tanto calcular subtotales rellenando los campos de una línea de pantalla con guiones. MICROSCRIPT sólo calculará el total hasta esa línea, no más allá.

16.4 Salida del modo calculador

Para dejar este modo de cálculo, debes pasar al siguiente tope de tabulación, pulsando la tecla **[TAB]**. El cursor se desplaza una posición a la derecha del campo mostrado en ese momento.

Si el campo final mostrado contiene un valor distinto de cero, entonces el resultado queda meramente como texto en el documento. En los demás casos, el campo mostrado queda relleno de espacios en blanco.

Si deseas abandonar el modo calculador durante una condición de rebase de la capacidad de un campo numérico, entonces el campo en cuestión queda con espacios en blanco al igual que si existiera una condición de cero.

NB: Pulsar **[TAB]** también te permite desplazarte a través de las columnas, dentro del modo calculador.

17.0 Preparación de un documento para impresión

Las rutinas a seguir para imprimir un documento con MICROSCRIPT se describen en la sección 28 de este manual.

Cuando se está creando o revisando un documento, es posible incrustar en él una serie de comandos de impresión. Estos comandos -caracterizados por ciertas claves- son examinados e interpretados por el MICROSCRIPT sólo en el momento en que el documento se envía desde la consola hasta la impresora.

Hay tres clases de comandos que pueden estipularse de esta manera:

- Secuencias especiales para realce de texto Sección 10.0
- Comandos entre corchetes angulados Sección 17.1
- Series de símbolos de control pre-definidos (especificados en el fichero de información del sistema (SIF))

17.1 Introduce los comandos entre corchetes en líneas sin texto normal

Los comandos entre corchetes angulados constan de un único carácter de control encerrado entre corchetes.

Los comandos entre corchetes angulados se sitúan en el extremo izquierdo de la pantalla. Una vez que el sistema ha identificado el comando, cualquier texto que hubiera en la misma línea no será tenido en cuenta.

El MICROSCRIPT reconoce los siguiente comandos:

- <F> Para terminar la impresión de la hoja corriente y forzar un avance de página en la impresora. Los números de página se imprimirán si así corresponde, y todos los otros comandos y controles reconocerán este salto de página.
- <R> Para indicar una línea de comentario. Eso indica al MICROSCRIPT que cualquier texto que hubiera en la línea corriente de impresión deberá no ser tenido en cuenta: no hay ningún otro comando implicado.

- <S> Para indicar que la siguiente línea es un **subtítulo** y no debe imprimirse si quedan para el pie de página menos líneas de las especificadas como mínimo para subtítulos (véase sección 18.2).
- <T> Para indicar que la línea corriente es la línea del **título** para cada página que se va a imprimir. Este título puede cambiarse en cualquier momento usando este comando.
- <I> Para identificar el texto en la línea corriente como especificación de un fichero que ha de **incluirse** al imprimir. La impresión del fichero corriente queda suspendida hasta que se haya enviado a imprimir el fichero que se manda incluir; luego se reanuda la impresión del fichero primitivo. Un fichero que va a incluirse queda identificado de la misma manera que el fichero que se agregará durante la actividad de congregamiento de ficheros. Es decir, por ejemplo:

<I>B:OTROFICH,#A

- <W> Para marcar el comienzo del modo de imprimir **ancho**. (Véase sección 17.2).
- <X> Para marcar el final del modo de imprimir **ancho**. (Véase sección 17.2)

17.2 Puede imprimir en el modo 'ancho'

Con el fin de imprimir documentos e informes que requieren una **anchura** de línea mayor de la que puede mostrarse en pantalla, el MICROSCRIPT trabajará con la impresora en el modo ancho, en el que se permite que la longitud horizontal de la línea de texto tenga hasta 255 caracteres, para un máximo de 100 líneas a imprimir.

Las secciones se construyen y añaden al final de la sección previa, confeccionando así internamente la línea a imprimir. Se deduce por tanto que si el último carácter de una sección está en la columna 58, por ejemplo, el primer carácter de la siguiente sección se imprimirá en la columna 59, a no ser que se haya colocado como mínimo una pulsación de [ENTER] entre las secciones.

Marca el comienzo de la sección usando el comando **<W>**. Esa línea no debiera contener ningún texto -sólo este comando- y un marcador **->**. Eso corresponde al final de las líneas de texto en esa sección.

Marca el final de la sección con **<X>**. El MICROSCRIPT regresará al modo normal de impresión.

Las secciones deben contener el mismo número de líneas incluso aunque eso involucre la inserción de líneas en blanco.

18.0 Imprimiendo texto con MICROSCRIPT

La impresión de texto desde el MICROSCRIPT es una opción en el menú principal distinta que la de creación y la de edición o revisión. La rutina para imprimir es accesible mediante el menú principal, pulsando **P**.

Al elegir la opción **P** (de printer) en el menú principal, aparecerá en pantalla después de una breve pausa el siguiente mensaje:

ENTER para fijar los parámetros de impresión o ESPACIADOR para avanzar hoja

Nombre del Documento, m?- para listar catálogo o ESC para salir

Elige la ? para que te muestre el catálogo de documentos en disco. Especifica el nombrefichero, incluyendo cuando corresponda la ductora (A: o B:), cuando aparece el mensaje en que se te solicita el nombre del documento.

18.1 Imprimir un documento directamente especificando el nombre

En la mayoría de las aplicaciones del MICROSCRIPT, se puede imprimir un documento directamente desde el disco, sin efectuar ningún cambio en los parámetros de formato prescritos para omisiones o sin cambiar ninguno de los comandos que puede haber incrustados en el texto para producir realces al imprimir.

Si el documento requerido está en la ductora B, recuerda especificar B: por delante del nombre del documento. Para imprimir el documento 'TEST1' en ese momento guardado en el disco alojado en la ductora B: por ejemplo debieras teclear como nombrefichero:

B:TEST1

El MICROSCRIPT buscará el documento en el disco y lo preparará para imprimir. Durante la actividad de impresión, el MICROSCRIPT mostrará en pantalla la página corriente y los números de línea, junto con el texto completo del fichero que está imprimiendo.

18.2 Alternativamente, cambia uno o más de los parámetros de impresión...

Cuando el programa MICROSCRIPT queda implantado en memoria desde el disco, contiene un cierto número de parámetros -con valores prefijados- para usar a la hora de imprimir, y que determinan características tales como el número de copias a sacar y la posición de los números de página en cada hoja que se imprime.

Al pulsar [ENTER] cuando aparece el mensaje inicial de pantalla, se consigue mostrar un menú con los parámetros de impresión. Comprueba la lista y pulsa [ENTER] para aceptar los 18 con los valores que tienen en ese momento.

- | | |
|--------------------------------------|--|
| 1 Tamaño de página | Define el tamaño longitudinal del papel según las líneas a imprimir. El valor dado para este parámetro es el usado para calcular los números de página. Valor mínimo es de 10. |
| 2 Línea para final de texto | Define el número de línea donde acabará la impresión del texto. El papel avanzará hasta el comienzo de la siguiente página. El valor mínimo es de 10. |
| 3 Columna de comienzo a la izquierda | Usado para desplazar el comienzo de la impresión contada desde el borde izquierdo. Permite que se efectue la impresión de acuerdo con los alimentadores de hojas. |
| 4 Pausa al inicio de página | Permite que si es preciso el sistema efectúe una pausa en la impresión al comienzo de cada nueva página. |
| 5 Mínimo de líneas para subtítulos | Impide que se imprima un subtítulo a no ser que quede todavía en esa hoja el número de líneas que se especifique para este parámetro. |
| 6 Doble interlineado | Se utiliza para activar o desactivar la posibilidad en algunas impresoras de usar una separación entre líneas doble de lo habitual. |

- 7 Página inicial Define la primera página que se ha de tomar para imprimir el fichero.
- 8 Página final Define la última página del fichero que se va a imprimir.
- 9 Forma adoptada para números de páginas Define como van a aparecer en el texto los números de páginas. El símbolo de 'intercalo' o exponenciación (\uparrow) representa el número de página, de manera que reseñando \uparrow indicaría que el número apareciera en la forma -123 \uparrow -, por ejemplo.
- 10 Posición de los números de página Contiene la descripción de en que posición debe aparecer al imprimir los números de página, por ejemplo, A o A2 da como resultado una numeración de página alternada.
- 11 Número de línea donde colocar los números de página Define el número correspondiente a la línea donde se van a imprimir los números de página. Un valor de 0 cancela la posibilidad de la numeración de páginas.
- 12 Posición de los títulos Controla la posición de los títulos de referencia para las páginas. Los válidos son L (izquierda), C (centrado) y R (derecha).
- 13 Número de línea para títulos Define el número correspondiente a la línea donde aparecen los títulos de las páginas. Un valor de 0 cancela la posibilidad de sacar titulares en cada página.
- 14 Longitud de línea Define la anchura en columnas del papel que se está usando.

- | | |
|---------------------------------------|---|
| 15 Número de copias | Habitualmente puesto al valor 1 pero puede alterarse para sacar las que se deseen. |
| 16 Líneas para comienzo de texto | Define el número de la línea donde comenzará la impresión de cada página. |
| 17 Número de la primera página | Permite estipular un valor a partir del cual comenzará la numeración de páginas de las que constituyen el fichero. Estipulando un valor de cero se dejan las páginas sin numerar. |
| 18 Opción de usuario para cada página | Permite al usuario enviar uno de los controles preestablecido para la impresora antes del comienzo de cada página. |

Para efectuar un cambio a cualquiera de los parámetros anteriores, que sea activo durante la sesión de impresión de documentos:

Teclea el número correspondiente al parámetro que se desea cambiar
Pulsa **[ENTER]**
Teclea ahora el nuevo valor de ese parámetro
Pulsa **[ENTER]**

Cuando hayas hecho todos los cambios pulsa **[ENTER]** una vez más para hacer que vuelvan a aparecer en pantalla los mensajes originales para imprimir:

[ENTER] para fijar los parámetros de impresión o **ESPACIADOR** para avanzar hoja

Nombre del Documento, m?- para listar catálogo o **ESC** para salir

...antes de comenzar a imprimir un documento

Una vez que se hayan estipulado valores para todos los parámetros, pulsa **[ENTER]** para que el sistema pase a imprimir el fichero.

Teclea el nombrefichero cuando se te requiera y concluye pulsando **[ENTER]**. El MICROSCRIPT buscará en el disco el fichero especificado y luego hará una pausa indicándote que puede comenzar a imprimir.

Durante la actividad de impresión, el MICROSCRIPT va mostrando en pantalla la página corriente y los números de líneas, conjuntamente con los detalles completos del fichero que está imprimiendo.

La actividad de impresión puede detenerse en cualquier momento pulsando **[ESC]**. El sistema regresará entonces al menú principal.

19.0 Usa la gestión de ficheros para organizar el disco

El MICROSCRIPT contiene un sistema de **gestión de ficheros** para todo el movimiento y trasiego de documentos archivados en los discos. Elige en el menú principal la opción de gestión de ficheros, pulsando la letra **F**.

La pantalla indicará que se ha elegido dicha opción, y luego aparecerá el siguiente submenú de posibles actividades:

1. Cambiar el nombre de un fichero
2. Suprimir un fichero del disco
3. Copiar de un fichero a otro
4. Congregar dos ficheros en uno
5. Mostrar catálogo de ficheros en disco

19.1 Los documentos pueden ser re-denominados...

Al elegir la opción 1 se puede cambiar el título o nombre fichero de un documento en el disco. La pantalla solicita cual es el documento al que desea cambiarse el nombre. Teclea el nombre fichero y pulsa **[ENTER]**.

...o eliminado

Elige en submenú la opción 2 para suprimir un fichero del disco. En pantalla se te solicita el nombre fichero correspondiente al documento que deseas eliminar. Introduce el nombre del fichero y pulsa **[ENTER]**.

Una vez especificado el nombre fichero, el sistema pide que le confirmes que deseas borrarlo. Pulsa **[ENTER]** y harás que abandone la operación de borrado y regrese al submenú de opciones, sin que el fichero especificado para eliminación sea borrado del disco.

Si pulsas el * seguido de **[ENTER]**, el sistema procederá a eliminar el fichero del disco. Observa que una vez el fichero ha sido eliminado, no hay manera de recuperarlo a no ser que dispongas de una copia del disco, como respaldo de las operaciones.

19.2 Copia de ficheros a otros discos...

Esta opción de submenú está pensada para copiar desde un disco hasta otro ficheros de texto y de datos sencillos. NO copiara ficheros de programas ni tampoco sacará copias de seguridad de discos completos.

Quita el disco del MICROSCRIPT de la ductora A y sustitúyelo con el disco que contiene el fichero que va a ser copiado.

Inserta un disco ya preparado en la ductora B: y luego elige la opción 3 de copiar.

La pantalla te pide primero el nombre del fichero desde el que vas a obtener la copia. Teclea dicho nombrefichero y pulsa **[ENTER]**. A continuación en pantalla se te pedirá que des el nombrefichero donde quieres que quede reflejada la copia que va a sacar.

Teclea el nombre del fichero correspondiente a la copia y pulsa **[ENTER]**. A menudo vas a usar el mismo nombre que el que tenía el documento original, pero recuerda que has de usar como prefijo el identificador de la ductora B: que es donde va a ser copiado el fichero cuyo original está en la otra ductora.

Una vez se haya realizado la operación de copiar, volverá a reaparecer el menú.

...o congrega dos ficheros en uno

Al elegir en el menú de gestión de ficheros la opción 4, puedes combinar dos ficheros para producir un nuevo fichero.

Con la opción 4 elegida quita el disco del MICROSCRIPT de la ductora A e inserta el disco con el fichero al que va a ser agregado el otro. En pantalla aparecerá el mensaje que te pide el nombrefichero correspondiente, que será considerado como fichero número 1.

Teclea dicho nombre de fichero y pulsa **[ENTER]**. Luego se te pedirá el nombrefichero correspondiente al segundo de los que van a entrar en la operación, el que va a ser agregado al otro, que es el considerado número 2.

Teclea dicho nombre y pulsa **[ENTER]**.

Finalmente, el sistema te solicita el nombre fichero con el que quieres denominar al que ha resultado de esta operación de congregación, y que está identificado como fichero -2. Teclea el nombre que desees para el fichero combinado obtenido y pulsa **[ENTER]**.

19.3 Mostrar el catálogo de documentos de un disco

Puedes comprobar cuales son los documentos guardados en un disco, si eliges esta opción en el submenú de gestión de ficheros. Es la opción 5, y al elegirla, la pantalla solicita que le identifiques la ductora (A: o B:) cuyo catálogo desees examinar.

Teclea el identificativo de la ductora en cuestión y pulsa **[ENTER]**. La pantalla te sugiere que:

- le des únicamente el nombre de **familia**, o clase, de los documentos que desees ver si están en ese disco.
- puedes también pulsar **[ENTER]** para que te muestre todos los registrados en el catálogo del disco.

Otra manera alternativa de hacer que el sistema te muestre en pantalla la lista de todos los documentos guardados en un disco dado, es elegir la opción apropiada (?) en el menú principal.

20.0 El programa del sistema MICROSCRIPT

20.1 MICROSCRIPT se suministra como una colección de ficheros en disco

El disco del programa entregado como standard por la fábrica contiene una serie de ficheros

FUNCION	FICHERO
Editor maestro	SCRIPT .COM
Impresión	SCRIPT-P .COM
Búsqueda y Sustitución	SCRIPT-S .COM
Gestión de ficheros	SCRIPT-F .COM
Ficheros standard de comandos	SCRIPT .SIF
Re-componer formato	SCRIPT-R .COM

También hay una copia de seguridad que actúa como respaldo del fichero de información del sistema **SCRIPT.SIF** (y lleva como identificador de clase los mismos tres caracteres **.SIF**).

21.0 El fichero de información del sistema

El fichero de información del sistema (System Information File: SIF) es un fichero grabado en disco que contiene los textos, mensajes, y parámetros de configuración del programa MICROSCRIPT. En cada ocasión en que entra en funcionamiento el MICROSCRIPT, el programa consulta el fichero SIF y extrae de la lista de información que el SIF contiene datos y valores tales como los mensajes que han de ser proyectados en pantalla y las series de letras clave empleadas para los comandos.

22.0 Programación de aplicaciones en MICROSCRIPT

Las secciones 1 a 21 de esta guía de usuario, han considerado el papel de MICROSCRIPT como un sistema de tratamiento de textos. Realmente, la colección de programas MICROSCRIPT constituye un surtido completo de facilidades para desarrollar **programas de aplicación** industriales.

22.1 El interpretador actúa sobre instrucciones incrustadas en el texto a agregar...

La 'llave' de las capacidades de programación de MICROSCRIPT es un segmento de programa conocido con el nombre de INTERPRETADOR y que interviene en toda operación de congregamiento de ficheros. Esa característica se esbozó brevemente en la sección 3, cuando se explicaron las actividades para agregar al fichero en memoria párrafos de ficheros en discos y textos y comandos depositados en las memorias de abreviación.

En el contexto de la programación de aplicaciones, el interpretador ante el comando de agregar un fichero, examina las series de instrucciones y claves incrustadas en él y las convierte en **acciones** sobre la memoria del ordenador o sobre la pantalla. El interpretador responde a esas instrucciones clave de la misma manera que si hubieran sido introducidas directamente por teclado. Por ejemplo, el comando para suprimir la línea corriente señalada por el cursor, incrustada en el texto que se va a agregar, sería tratada igualmente que si se hubieran teclado los dos caracteres ([CTRL]Y) que corresponden a la función 12. Se deduce por tanto, que para cada una de las acciones standard del MICROSCRIPT, hay un **comando** correspondiente para ser usado al programar.

...que son representadas por equivalentes visibles de los comandos

Las funciones que pueden ejercerse con el MICROSCRIPT, tales como las de suprimir o la de subrayar, son ejecutadas normalmente en el mismo momento en que son introducidas a través del teclado. Se requiere por tanto un método, para introducir esos comandos como instrucciones -como comandos 'estáticos' cuya acción se llevará a cabo posteriormente-, de manera que no sean obedecidas hasta que el programa del que forman parte haya pasado a través del interpretador al realizar el congregamiento de dos ficheros: el existente en memoria y el que se agrega desde el disco o desde las memorias especiales de abreviación.

Ese método es lo que aquí se denomina **-equivalentes visibles-**, que es de hecho una anotación taquigráfica que puede incluirse dentro de un texto y que representa un programa cuyas acciones serán posteriormente ejecutadas, al agregar ese texto que ya hay existente en memoria.

Los símbolos equivalentes visibles para algunas de las acciones más frecuentemente ejecutadas, son:

~D para la **Función 36** de borrar desde comienzo de línea

~B para la **Función 19** de convertir a minúsculas

~/# para la **Función 7** de avanzar el cursor hasta el siguiente tope de tabulación

~P 3 por ejemplo para la **Función 48** de conmutar al modo alternativo 3

Observa que todos los equivalentes van **precedidos** por un símbolo especial -en esta serie de ejemplos ha sido la tilde (~). Eso indica al interpretador -durante una operación de congregamiento de ficheros- que el siguiente carácter que aparece inmediatamente detrás (y el número que le sigue cuando el comando lo requiere como parte de su sintáxis). La de ser interpretado como un **comando para la acción**, y no como un simple texto. Por tanto el interpretador lo ejecutará.

Además se pueden interpretar (e inmediatamente ejecutar) series de comandos sucesivas: cada comando **estando** precedido por el símbolo especial que caracteriza a los equivalente visibles. La aplicación puede requerir que se avance el cursor hasta el segundo tope de tabulación, por ejemplo, y luego se suprima todo el texto que haya en la línea corriente a la izquierda del cursor. La secuencia apropiada de comandos sería la representada por los símbolos equivalentes ~~/~D. La tilde es el símbolo **prescrito** para designar a los equivalentes visibles, porque raramente es usado para cualquier otro propósito en Inglés.

Hay sin embargo una posibilidad de **frustrar** el efecto de cualquier equivalente visible sencillo, haciendo posible así que incluso el caracter especial ~ pueda ser usado en el texto. El comando representado por ~/ hará que la siguiente serie de símbolos que aparezcan no sean interpretadas como un comando, sino como un simple texto: así por ejemplo ~/~D haría que los dos caracteres ~D pasaran a través del interpretador como si fueran meros caracteres de texto, y no de control.

Observa que la supresión del efecto de los caracteres visibles sólo se aplica al carácter que va inmediatamente detrás de la expresión que **frustra** (derrota) la interpretación. Hay situaciones -explicadas con detalle en la sección 22.3- donde una serie larga de equivalentes visibles no ha de ser interpretada hasta una etapa posterior. En ese caso, el modo alternativo numerado 3 suspenderá el efecto de los equivalentes hasta que dicho modo haya sido cancelado.

22.2 El MICROSCRIPT puede generar los equivalentes automáticamente

La naturaleza de un equivalente visible es tal que a menudo no lleva ninguna relación aparente con la acción que representa. No es inmediatamente obvio por ejemplo, que ~! sea el equivalente visible del comando **[ENTER]**, o que a la instrucción para bajar el cursor una línea le corresponda como equivalente visible la pareja de caracteres ~&.

Un usuario que desarrollara un programa de aplicación con MICROSCRIPT, encontraría que le llevaba mucho tiempo si tuviera que consultar los equivalentes en una tabla y luego teclearlos en el programa. El modo alternativo numerado 7 del MICROSCRIPT, de aprendizaje del programa, le alivia esta tarea haciendo que con cada pulsación del comando automáticamente aparezcan en pantalla los equivalentes visibles.

Con el modo 7 activado (función 48, i.e: **[ESC]M7**) al pulsar la tecla **[ENTER]** se generan en pantalla los símbolos ~!, mientras que al mover el cursor una línea hacia abajo hace que aparezcan los símbolos ~&, sin acción adicional por parte del usuario.

Para salir del modo 7, basta apelar a la función 49 con ese número (**ESC ESC M7**).

22.3 Pausas en los programas para aceptar información por teclado

Se puede incrustar un comando en el texto que está siendo tramitado por el interpretador para que cierto sistema efectúe una pausa y pida que se le introduzca un dato por teclado. El equivalente visible de este comando es la !. La introducción de datos sólo es posible con el cursor situado en el extremo izquierdo de las líneas.

Cuando el sistema se encuentra con este comando, el control del programa vuelve al operador. Cualquier texto o dato literal puede ser introducido en ese momento; y como siempre la introducción se concluye pulsando [ENTER]. El programa pasa entonces a interpretar y a ejecutar el siguiente comando.

Alternativamente, se puede pulsar la tecla [ESC] para hacer que el programa abandone esa acción y el sistema regrese al menú principal del MICROSCRIPT.

Ejemplo:

Se requiere un breve programa que solicite el nombre de un documento del MICROSCRIPT, y luego almacene la respuesta en la memoria temporal para abreviaciones numerada como 6.

Las acciones requeridas son:

Sacar un aviso por pantalla

'Por favor teclea nombrefichero'

Colocar el cursor al comienzo de la línea

Función 15

Pausa para que el usuario introduzca el dato

!

Borrar desde el cursor hasta el extremo de esa línea

Función 35

Depositar el nombrefichero dado como abreviación 6 sustituyendo el contenido existente en ella

Función 46 seguida de 6 y R cuando se solicite.

Los equivalentes visibles de estos comandos tal y como estarían registrados en el programa serían:

Por favor teclea nombrefichero ~/!~C~N6

22.4 Los programas escritos en MICROSCRIPT pueden ser almacenados...

Hay dos maneras de depositar un programa de aplicaciones para ser ejecutado por MICROSCRIPT de manera que pueda ser reclamado cuando se requiera utilizarlo. El método a elegir dependerá de la naturaleza de aplicación y del tamaño del programa en cuestión.

...como abreviaciones temporales

Puede ser apropiado alterar el contenido de una o más de las memorias compendiadas de abreviaciones durante una sesión corriente de creación/edición/revisión de un fichero. El procedimiento para retener un programa de esta manera es el mismo que para depositar cualquier serie de caracteres en una de las memorias temporales de abreviación:

<series de claves><función 46><n>

Donde **n** representa el número de la memoria de abreviación empleada. Recuerda que el límite para el tamaño del programa que puede retenerse temporalmente en una de estas memorias de abreviación (**de dichos**) es el de la longitud de línea menos dos columnas, lo que habitualmente es 78 símbolos.

Siempre que se reclama el contenido de una de las memorias de abreviación, cualquier equivalente visible depositado en ella sería **traducido** al pasar a través del interpretador. Eso crearía problemas si la memoria que se está usando retuviera series repetitivas de comandos preparadas para ser combinadas en un programa mayor que todavía se está desarrollando. El equivalente visivo para el comando de elegir modo alternativo 3 (función 48 seguida de 3) puede incluirse en esas ocasiones al comienzo de una memoria de abreviación (y quitarse al final) para permitir que la serie de equivalentes visivos pase a través del interpretador sin ser traducida, y por tanto, sin ser convertida a la serie de acciones correspondientes en ese momento

Ejemplo:

Se está confeccionando un programa, que tiene que recurrir varias veces a la misma serie de comandos. Carga esa serie -mostrada a continuación- en la memoria de abreviaciones numerada 4, de manera que pueda usarse en cualquier momento mientras se está preparando el programa (la serie no va a ser ejecutada inmediatamente):

Mover el cursor a la esquina izquierda de la pantalla

Usar reglilla numerada 6

Activar reglilla número 6

Avanzar hasta segundo tope de tabulación

Usando el modo 7 del MICROSCRIPT para el 'aprendizaje de programa', el programa a introducir sería:

[ESC]M seguido de 7	Entrar en el modo alternativo 7 (aprendiendo)
[ESC]M seguido de 3	Frustra todos los equivalentes visibles
[CTRL] ↑ ascendente	Subir a la línea superior de pantalla
[ESC]U seguido de 6	Usar reglilla 6 para formato
Función 1	Activar la reglilla 6
[TAB] dos veces	Avanzar hasta el segundo tope de tabulación
[ESC][ESC]M seguido de 3	Restaurar efecto de equivalentes visibles
[ESC]#4	Depositar todos los caracteres a la izquierda del cursor como abreviación número 4.
[ESC][ESC]M seguido de 7	Salir del modo alternativo 7.

En pantalla, esa misma serie vendría representada como:

~P3~F~?6~!~!~Q3~N4

Observa que pasar al modo 7 y regresar al modo normal de introducción de texto, son **transparentes** al programa y al usuario, y no aparece ningún equivalente en pantalla para denominar estos pasos.

...o como ficheros en discos

Los programas retenidos en las memorias de abreviaciones están restringidos en tamaño (incluso enlazando las diez memorias obtenemos sólo una capacidad total de 750 símbolos).

No hay ninguna restricción sin embargo, cuando los programas se guardan como ficheros en disco (restringidos únicamente por el espacio disponible en el disco). Cualquier equivalente visible contenido en un fichero del programa será traducido y ejecutado -puesto en acción- cuando ese fichero sea agregado al documento que ya hay en pantalla al pasar a través del interpretador durante una operación de congregamiento de ficheros (a no ser desde luego que se haya 'frustrado' el efecto de un equivalente).

Los ficheros en disco con MICROSCRIPT pueden ser 'anidados', o puestos en cascada, de manera que un comando de congregamiento de fichero incrustado en un fichero puede poner en acción el congregamiento de otro fichero mencionado, y así sucesivamente, hasta un máximo de 7 niveles de anidamiento.

22.5 Los ficheros en disco pueden estar indexados para incrementar su flexibilidad

Si no se da ninguna instrucción en contrario, todo el texto del fichero especificado en el comando será traído del disco y agregado al fichero existente en memoria. El siguiente comando, por ejemplo, agregaría por completo el fichero **INPUT.DOC** al que se estuviera tratando en el sistema:

INPUT.DOC seguido de **[ESC]G** (función de agregar fichero)

Se deduce pues, que diez bloques de instrucciones guardadas en disco exigirían diez ficheros separados en disco (diez documentos), cada una con un nombre unívoco para identificarlas. Pero si se recuerda, ya se mostró en la sección 14, que partes concretas del fichero mencionado en el comando pueden agregarse individualmente al documento corriente en memoria, aprovecharemos esa facilidad.

Esta actividad involucra la INDEXACION interna del fichero en disco de manera que segmentos de un solo fichero puedan ser extraídos del mismo cuando se requieran, y agregados a través del interpretador al documento existente en memoria. Los programas de aplicaciones en MICROSCRIPT suelen aprovecharse de esa misma facilidad.

Los ficheros pueden desglosarse en unidades de cualquier longitud, usando un identificador adecuado (y el término habitual para eso es etiqueta o 'label'). Esa etiqueta, y los símbolos)) que marcan el final de cada una de esas unidades de texto, han de estar colocados en una línea separada.

El documento **INPUT.DOC** por ejemplo, puede contener tres de tales unidades de programa, etiquetadas como INICIO, PRUEBA y FINAL. Si se desea reclamar sólo la segunda de ellas, debieramos dar el comando:

INPUT.DOC, PRUEBA seguido de [ESC]G (función de agregar ficheros)

Observa que el comando para pasar la sección PRUEBA a través del interpretador viene especificada fuera del fichero **INPUT.DOC** que contiene dicha sección -en este caso, la instrucción fue proporcionada por las teclas **[ESC]G** que aparecen en este documento corriente 'de trabajo' que está abierto en el sistema en ese momento.

22.6 El comando de AGREGAR FICHEROS ayuda a la programabilidad...

Es posible la programación extremadamente avanzada con MICROSCRIPT si se construyen menús de opciones como 'vanguardia' de una aplicación. La función de AGREGAR FICHERO **[ESC]G** se usa para **encadenar** desde el menú con otros programas concretos tal y como se muestra en la sección 21.4.

Por ejemplo, una serie de rutinas de contabilidad se pueden empalmar al menú principal del MICROSCRIPT.

...y permite establecer rutas para fallos u omisiones

La función de agregar un párrafo se aprovecha considerablemente en menús tal como el mostrado más abajo. Su ámbito sin embargo puede incrementarse todavía más si dicho comando se matiza con una o más 'rutas' preparadas para cuando el primero, y los programas subsiguientes, no pueden ser localizados por alguna razón.

Considera la instrucción al MICROSCRIPT:

INVCE~G DAYBK~G ACMEN~G

Los nombres de los programas mencionados en la instrucción se toman consecutivamente de izquierda a derecha. Si **INVCE** no está presente en el disco, el sistema tiene prescrito pasar inmediatamente al siguiente nombre de programa que sea válido -en este caso **DAYBK**. Y si este programa tampoco se pudiera localizar, el sistema continúa su ruta pasando al tercero de los nombres mencioandos - **ACMEN**.

Mientras que la disponibilidad de primeras y segundas 'prescripciones' puede tener únicamente importancia teórica en sistemas basados en una estructura simple de menús, adquieren una importancia mucho mayor cuando la respuesta del usuario es adoptada por el sistema como parte de un nombre de programa.

Un sistema puede estar controlado a través de un menú de pongamos tres programas:

- A Contabilidad**
- W Tratamiento de textos**
- S Hoja de cálculo**
- X Salida del sistema**

¿Qué actividad deseas?

Dentro del sistema, los programas pueden estar identificados por ejemplo como **ACCTA**, **ACCTW**, **ACCTS** y **ACCTX** respectivamente. La función de reclamar un párrafo puede ponerse en acción aprovechando la respuesta dada al mensaje sacado en pantalla:

ACCT (Pausa para introducir datos)~G

Tecleando A, W, S, o X cuando el sistema efectúa la pausa, se completaría un nombre de programa válido y se conseguiría que actuara la función de reclamar párrafos, y se trajera del disco a través del interpretador el programa en cuestión para ser agregado al existente en memoria.

Pero habría problemas si el operador cometiera una equivocación al teclear, y pulsara un carácter que no estuviera en la lista de respuestas aceptables. Si por ejemplo pulsara la letra D, el sistema 'vería' el comando **ACCTD~G**, que no es el nombre de ninguno de los programas que figuran en la instrucción. El sistema por tanto fallaría en ese momento.

Si se incorpora al comando un valor prescrito para omisiones o fallos, se evitaría que sucediera éso:

ACCT (Pausa para la introducción de datos)~G ACCTX~G

En el caso en que se produjera una respuesta no válida, el sistema seguiría su ruta pasando por tanto al programa prescrito **ACCTX**, que en este caso sería una rutina para hacer el cierre del sistema de una manera elegante, o para hacer que siguiera su ruta hacia otra -y esperemos que esta vez válida- respuesta.

22.7 La facilidad de 'VAYA A ETIQUETA' incrementa el ámbito de los programas

Teniendo muchas similitudes con la función de agregar fichero, la función de saltar a una etiqueta (**GOTO LABEL**) es una de las facilidades que permite al **MICROSCRIPT** avanzar a través de un programa de aplicaciones en busca de una determinada **etiqueta**. Habiendo llegado a ese punto, el sistema actuaría (ejecutaría) todo el programa a partir de ese punto, o hasta que volviera a encontrar otra instrucción de salto posteriormente.

Observa que este comando de salto hasta etiqueta es un comando **interno** ya que accesa parte del fichero en que está incrustado el propio comando. La función de **AGREGAR FICHERO** por contraste, reclama secciones de otros ficheros.

La función de SALTAR A ETIQUETA hace posible el diseño de sistemas con **rutas** lógicas que se extienden por todo el programa. El camino preciso tomado por la lógica del sistema depende del resultado obtenido en una etapa anterior del programa.

El saltar a etiqueta pudiera usarse por ejemplo para manejar un submenú dentro de una aplicación. La respuesta a un mensaje mostrado en pantalla completa una etiqueta y hace que el sistema desvie su ruta hasta el punto requerido.

En el siguiente ejemplo, se está preparando un pequeño sistema industrial. Las rutinas para introducción de datos están controladas a través de un menú dentro del fichero del programa:

- 1) Ventas domésticas
- 2) Ventas exportación
- 3) Facturas de compras

Teclea opción y pulsa **[ENTER]**

La serie de comandos adecuada para manejar esta tarea podría ser:

!	Pausa para introducción por teclado
Función 46 seguida de 1	Retener la respuesta como abreviación 1
ESC{	Suprimir cualquier carácter situado a la derecha de la respuesta.
OPCION	La etiqueta en cuestión
[ESC]#n seguido de 1	Recoger el contenido de la abreviación 1.
@	Saltar a la etiqueta definida anteriormente.
INVALIDO	Etiqueta para fallos
@	Saltar a etiqueta anterior

Eso sería mostrado como la serie de equivalentes visibles siguientes:

!~N1~OPCION.~01 @ INVALIDO @

Observa que las etiquetas que corresponden a las secciones del programa que pueden identificarse a través del menú serían:

OPCION.1 OPCION.2 OPCION.3 INVALIDO

Cuando el sistema efectúa la pausa y se tecléa por ejemplo el 6- el sistema ensayaría una ruta a través de una etiqueta que no existe (en este ejemplo **OPCION.6**). Como sería incapaz de localizar esta etiqueta el MICROSCRIPT tomaría la primera etiqueta prescrita para fallos que encuentre (en este caso **INVALIDO**) y ejecutaría todas las rutinas contenidas desde esa etiqueta hasta la siguiente pareja de símbolos delimitadores, que son los dos paréntesis de cierre.

22.8 Los valores numéricos obtenidos en los programas pueden ser guardados...

Hay bastante probabilidad de que un valor numérico generado en un cálculo sea requerido en cualquier otra parte dentro de la misma aplicación. La posibilidad de almacenar datos numéricos (comentada en sección 16.3 sobre el calculador del MICROSCRIPT) puede usarse también dentro de los programas de aplicaciones.

Al alcanzar el punto en el programa donde han de depositarse datos obtenidos de un campo numérico, tecléa S seguido de un número del 0 al 9 (porque hay diez memorias numéricas separadas).

...y recuperados en una aplicación

El cursor debiera estar preparado para aceptar un dato numérico a introducir en un campo de cálculo, cuando va a ser recuperado el contenido de una de las diez memorias numéricas disponibles. La siguiente secuencia en el programa debiera ser entonces **Rn** siendo n el número identificativo de la memoria que retiene el dato.

22.9 Se pueden usar modos alternativos para afectar a la programación

De los ocho modos alternativos admitidos por MICROSCRIPT, cuatro de ellos tienen una importancia particular en la programación de aplicaciones.

MODO 3 Frustra todos los equivalentes visibles.

Se comentó en la sección 22.1.

MODO 4 Modo Invisible.

Este modo de operación se diseñó específicamente para la escritura de programas de aplicación. Mientras está en vigor este modo quedan suspendidas todas las operaciones de escrituras sobre la pantalla. El programa puede entonces efectuar tareas tales como la supresión de líneas y el limpiado de pantalla 'en la retaguardia' (background). El resultado es un programa con una presentación más profesional, y sin que los movimientos rápidos del cursor en la pantalla estorben al manejo de textos y datos.

MODO 6 Retención del formato al agregar un fichero.

Cuando al texto existente en memoria se agrega texto de otro fichero, o de una memoria de abreviación, está prescrito que se efectúe de acuerdo con el formato y márgenes aplicables en el momento de efectuar la operación de congegamiento de los dos ficheros.

El modo 6 revoca esta condición prescrita, y permite que el texto que se agrega retenga el formato que previamente tuviera asignado. Un texto indentado por ejemplo, perdería esas indentaciones al ser agregado a otro, a no ser que se haya elegido este modo.

MODO 7 Aprendizaje de programa.

Se comentó en la sección 22.2.

23.0 Preparando discos para su uso

Los discos informáticos son de dos clases: flexibles y rígidos.

23.1 Discos rígidos

Los discos rígidos son el método más efectivo de almacenamiento de información disponible normalmente en el ambiente de oficinas: con ellos se puede implementar un sistema de gran capacidad en un mínimo espacio y con un tiempo muy rápido de respuesta. Además, no requiere ninguno de tus cuidados.

Todo el mantenimiento debe ser realizado por personal con cualificación técnica.

Las capacidades de almacenamiento de un disco rígido hacen innecesario que se manipulen directamente los soportes magnéticos de almacenamiento.

23.2 Discos flexibles

Los discos flexibles ('Floppy') están ampliamente utilizados en microordenadores. La información se graba en un disco plano de material magnético que gira dentro de una funda. Los tamaños disponibles son 3"; 3,5"; 5,25" y 8".

Cada disco tiene una abertura central y una ventana a través de la cual actúan los cabezales de lectura y grabación durante el arrastre.

El AMSTRAD CPC464 usa discos de 3".

23.3 Los discos flexibles requieren cuidados

Si la máquina se apaga accidentalmente antes de quitar el disco de la ductora no dejes por tiempo indefinido el disco dentro.

Nunca acerques los dedos u otros objetos a la ventana alargada que hay en cada cara del disco.

Siempre inserta el disco en la unidad ductora con cuidado y cierra la portezuela con suavidad.

Nunca quites un disco en la mitad de un trabajo.

Mantén los discos alejados del café, té, los cigarrillos y el polvo.

Usa un rotulador de punta de fieltro cuando etiquetes los discos para evitar dañar su superficie.

Conserva los discos en posición vertical, o semivertical, en un cofre de seguridad, a prueba de fuegos, y estanco a la humedad.

SIEMPRE saca copias de seguridad de los discos importantes y guárdalas por separado. Recuerda que no es el valor del DISCO lo que te estás jugando, si no el valor de toda la información que has almacenado en él.

24.0 Funciones disponibles con MICROSCRIPT

Esta sección del manual contiene una descripción completa de los efectos de cada una de las funciones:

['aDENTRO']

Esta función opera de manera similar al **retorno de carro** de una máquina de escribir tradicional. No afecta al texto en la pantalla, de manera que puede usarse desde cualquier posición dentro de una línea. Esta función no es válida cuando el MICROSCRIPT está en el modo de cálculo.

#2 'DELECTE'

El cursor retrocede un espacio hasta situarse sobre el caracter a suprimir, y luego elimina ese caracter.

#3 CURSOR IZQUIERDA

El cursor retrocede una posición a la izquierda.

#4 CURSOR DERECHA

El cursor avanza una posición a la derecha.

#5 CURSOR SUBE

Mueve el cursor una línea hacia arriba manteniéndose en la columna corriente.

#6 CURSOR BAJA

Mueve el cursor una línea hacia abajo manteniendo la columna corriente

#7 TABULACION

Mueve el cursor hasta el siguiente tope de tabulación.

#8 METE TEXTO

Inserta espacio en blanco en cualquier lugar a lo largo de la línea corriente.

#9 SACA TEXTO

Corre hacia la izquierda el texto situado a la derecha sacando los espacios en blanco no requeridos.

#10 AVANZA PALABRA	Mueve el cursor hasta el primer caracter de la siguiente palabra situada a la derecha.
#11 RETROCEDE PALABRA	Mueve el cursor hasta el último caracter de la palabra previa (situada a la izquierda).
#12 SUPRIME LINEA	Elimina la línea del cursor corriendo el texto acordemente.
#13 ABRE NUEVA LINEA	Inserta líneas en blanco para permitir añadir nuevo texto.
#14 AVANZA A FINAL DE LINEA	Coloca el cursor en la posición a la derecha del último carácter de la línea corriente.
#15 RETROCEDE LINEA	Mueve el cursor hasta el extremo izquierdo de la línea corriente.
#16 ARRIBA DE PANTALLA	Mueve el cursor hasta la línea superior de la pantalla.
#17 ABAJO DE PANTALLA	Mueve el cursor hasta la línea inferior de la pantalla.
#18 CONVIERTE A MAYUSCULAS	Convierte a mayúsculas todos los caracteres situados en la derecha del cursor.
#19 CONVIERTE A MINUSCULAS	Convierte a minúsculas todos los caracteres situados en la izquierda del cursor.
#20 RECORTE Y PEGADO MARCA DE COMIENZO	Señala la esquina superior izquierda de la sección que va a montarse durante la operación de recortado y pegado.

- #21 MARCA FINAL CON BLANQUEADO** Señala la esquina inferior derecha del área que va a montarse. El texto es en ese momento copiado en la memoria 'temporal de montaje' y el área primitiva queda rellena de espacios en blanco.
- #22 MARCAJE FINAL Y DEJANDO** Similar a la función 21 pero el área original permanece en pantalla una vez que se ha copiado el bloque en la memoria temporal.
- #23 MARCAJE FINAL Y QUITADO** Marca la esquina inferior derecha del área de texto a montar. Pero una vez que dicho área ha sido copiada de la memoria temporal, el texto es suprimido.
- #24 PEGADO SOLAPANDO** Coloca el texto retenido en la memoria temporal sobre un documento, pero sobreescribiendo encima del texto existente.
- #25 PEGADO INSERTANDO** Inserta en un documento el texto retenido en la memoria temporal.
- #26 PEGADO CODEANDO** Inserta en el documento el texto retenido en la memoria temporal, desplazando el texto existente de manera que no se escribe encima.
- #27 CENTRAR LINEA** Cambia el formato de todo el texto colocándolo centrado bajo el control de la reglilla en vigor en ese momento.
- #28 CAMBIAR FORMATO** Cambia el formato de todo el texto bajo el control de la reglilla corriente.

#29 DEPOSITAR REGLILLA	Comprueba la validez de una reglilla, y luego la vierte sobre la memoria de reglilla escribiendo encima de la allí existente.
#30 RECUPERA REGLILLA	Recaba la reglilla depositada en una memoria concreta.
#31 USAR REGLILLA	Indica a MICROSCRIPT que emplee la reglilla depositada en una memoria específica.
#32 ENGUIONADO	Coloca una línea de guiones por debajo de la línea del cursor.
#33 SUBRAYAR	Subraya todo el texto situado en la línea del cursor.
#34 AVANCE DE PAGINA	Coloca el cursor en la esquina superior izquierda de la pantalla de la siguiente página.
#35 BORRA HASTA FINAL LINEA	Elimina el texto situado a la derecha del cursor en la línea corriente.
#36 BORRA DESDE COMIENZO DE LINEA	Elimina el texto a la izquierda del cursor en la línea corriente.
#37 SUPRIME PALABRA	Elimina parte o la totalidad de una palabra.
#38 SUBE AL INICIO DEL FICHERO	Envía al cursor a la posición BASE, que es la esquina superior izquierda de la pantalla.
#39 AGREGAR FICHERO	Congrega el documento que se está editando y un fichero procedente del disco.
#40 ENTRA INSERCION	Pasa al modo inserción.
#41 CODEAR TEXTO	Desplaza hasta la línea siguiente el texto a la derecha del cursor.

#42 SALIR INSERCION	Deja el modo de inserción.
#43 IR AL FINAL TEXTO	Desplaza el cursor hasta el final de la pantalla.
#44 EXTENDER LINEA	Alinear por la derecha -justificado el texto a partir de la línea del cursor y hasta el final del párrafo.
#45 ENCONTRAR TEXTO	Busca en un fichero una determinada palabra o trozo de texto.
#46 DEPOSITA TEXTO EN UNA MEMORIA ABREVIACION CONCRETA	
#47 RECABA ABREVIACION	Recupera el texto depositado en una memoria de abreviación.
#48 ENTRA ALTERNATIVO	Pasa a funcionar en uno de los modos alternativos.
#49 SALE ALTERNATIVO	Deja de funcionar en el modo alternativo concreto.
#50 ABANDONA EDICION	Deja el documento que estaba editando sin guardarlo.
#51 RECLAMA DOCUMENTO	Trae a pantalla un documento específico.
#52 GUARDE DOCUMENTO	Abandona la edición guardando el documento.
#53 SUBA HASTA INICIO FICHERO	Coloca el cursor al principio del documento.
#54 RECLAMA FICHERO INFORMACION SISTEMA	Trae a pantalla el fichero SIF.

#55 COPIA LINEA

Deposita una línea de texto en una memoria de copiado para ser incluida en otra parte del documento.

#56 DEJE LINEA

Coloca la línea de texto copiada mediante la función 55 en un documento y en la posición corriente del cursor.

**#57 BAJE HATA FINAL
DOCUMENTO**

Coloca el cursor al final del documento corriente.

25.0 Apéndice I

25.1 Código Standard ASCII

Car.	Cód.	Car.	Cód.	Car.	Cód.	Car.	Cód.
^@	0	Space	32	@	64	\	96
^A	1	!	33	A	65	a	97
^B	2	"	34	B	66	b	98
^C	3	#	35	C	67	c	99
^D	4	\$	36	D	68	d	100
^E	5	%	37	E	69	e	101
^F	6	&	38	F	70	f	102
^G	7	'	39	G	71	g	103
^H	8	(40	H	72	h	104
^I	9)	41	I	73	i	105
^J	10	*	42	J	74	j	106
^K	11	+	43	K	75	k	107
^L	12	^	44	L	76	l	108
^M	13	-	45	M	77	m	109
^N	14	.	46	N	78	n	110
^O	15	/	47	O	79	o	111
^P	16	0	48	P	80	p	112
^Q	17	1	49	Q	81	q	123
^R	18	2	50	R	82	r	114
^S	19	3	51	S	83	s	115
^T	20	4	52	T	84	t	116
^U	21	5	53	U	85	u	117
^V	22	6	54	V	86	v	118
^W	23	7	55	W	87	w	119
^X	24	8	56	X	88	x	120
^Y	25	9	57	Y	89	y	121
^Z	26	:	58	Z	90	z	122
^[27	;	59	[91	{	123
^\	28	<	60	\	92		124
^]	29	=	61]	93	}	125
^^	30	>	62	^	94	~	126
^-	31	?	63	-	95	Del	127

25.2 Caracteres de función para programas

Función	Código	Función	Código
1 Retorno	!	31 Use reglilla n	?
2 Supresión	"	32 Enguionado	@
3 Izquierda	#	33 Subrayado	A
4 Derecha	\$	34 Avance de página	B
5 Arriba	%	35 Borrado a derecha	C
6 Abajo	&	36 Borrado a izquierda	D
7 Tabulación	'	37 Borrado palabra	E
8 Mete	(38 Línea superior	F
9 Saca)	39 Agregar fichero	G
10 Avance palabra	*	40 Entrar inserción	H
11 Retrocede palabra	+	41 Voltear línea	I
12 Suprime línea	^	42 Salir inserción	J
13 Abra línea	-	43 Línea inferior	K
14 Extremo derecha	.	44 Extender línea	L
15 Extremo izquierda	/	45 Buscar texto	M
16 Línea superior	0	46 Depositar abreviación	N
17 Línea inferior	1	47 Recuperar abreviación	O
18 Mayúsculas	2	48 Alternativo puesto	P
19 Minúsculas	3	49 Alternativo quitado	Q
20 Inicio recorte	4	50 Abandone sin quitar	R
21 Recorte blanqueado	5	51 Reclame fichero	S
22 Recorte dejado	6	52 Guarde documento	T
23 Recorte quitado	7	53 Inicio documento	U
24 Pegado solapando	8	54 Reclamar SIF	V
25 Pegado insertando	9	55 Pie de documento	W
26 Pegado codeando	:	56 Pantalla previa	X
27 Centrado línea	;	57 Salto a etiqueta	Y
28 Reforma párrafo	<	58 Pare congregamiento	Z
29 Deposite reglilla n	=	59 Pausa para entrada	[
30 Recupere reglilla n	>	60 Frustre equivalente	\

26.0 Apéndice II Mensaje de Error

Hay tres clases de mensajes de error:

Errores de usuarios El MICROSCRIPT te informa que has cometido un error.

'Pifias' Pueden surgir en condiciones en que se descubra un fallo en la programación. Los errores de esta clase serán expresados en la forma:

Error 64/960

Esta clase de mensaje va habitualmente seguida de una línea de explicación en 'Lengua Inglesa' (o casi). Cuando ocurre uno de estos errores, automáticamente provocará que se deje el MICROSCRIPT y se regrese al sistema operativo. Debieras anotar lo siguiente:

La fecha de entrega de la versión del disco.

El texto exacto del mensaje de error aparecido.

La operación que se estaba efectuando en ese momento.

Debes informar estos hechos por teléfono a tu distribuidor.

Errores externos Se pueden producir errores debido a un espacio insuficiente en el disco, o en directorio del disco. Estos errores serán detectados por el sistema operativo y una vez remediada la causa del error, el MICROSCRIPT puede funcionar de nuevo.

INDICE

Abandono edición sin guardar documento	22
Abandonar edición	22
Agregamiento de documentos	8
Ampliación de documentos	8
Ancho, imprimiendo en el modo	52
Anidamiento al congregar ficheros	42
Anulando un campo aritmético	50
Blanqueo de texto después recortado	21
Borrado de un documento	59
Búsqueda de un documento	18
Cálculos con MICROSCRIPT	51
Cambiando abreviaciones al tecleo	66
Cambiando parámetros de impresión	55
Cambiando título de un documento	59
Cambio del signo aritmético	49
Campos aritméticos, definiendo	69
Campos en el modo aritmético	49
Carácter, definición	3
Caracteres de control en reglillas	11
Caracteres de función al programar	86
Catálogo de documentos	61
Centrado de textos, carácter de control	11
Centrar texto	33
Código ASCII	85
Colores-cambiando distribución	66
Comandos entre corchetes al imprimir	32
Congregamiento de documentos	61
Congregamiento de ficheros en disco	61
Congregamiento de ficheros en pantalla	40
Congregamiento de ficheros indexados	41
Congregando ficheros en disco	61
Congregando ficheros en pantalla	40
Conmutando entre secciones de un fichero	41
Conservando resultados numéricos de los programas	74
Controlando el cursor	13
Control del margen	11
Controles de dirección del cursor	13
Conversión de equivalentes visibles	37
Conversión mayúsculas/minúsculas	72
Copiado de ficheros entre disco	60
Copiado de ficheros entre discos	60

Copiado de líneas	12
Copias de documentos entre discos	60
Copias de seguridad de los discos	78
Corrección de errores durante el teclado	13
Creación de documentos	8
Creación de un documento	8
Cuidado de disco	70
Cursor	2
Datos en registros numéricos	49
Depositando abreviaciones	44
Depositando datos en registros numéricos	49
Depósitos para abreviaciones	44
Deslazamiento de texto durante el pegado	28
Disco del sistema	5
Disco del sistema	5
Disco del usuario	6
Edición de documentos	18
Edición de un documento existente	18
Ejecución de códigos y comandos	39
Eliminación de un documento	59
Enguionado, automático	37
Enguionado, función	21,30
Entrega del MICROSCRIPT por fábrica	62
Equivalente de los comandos MICROSCRIPT	64
Equivalente visibles, frustración del efecto	36
Escutando un documento	18
Extensión de líneas	25
Familia, parte del nombrefichero	8
Ficheros, catálogo	61
Formato durante el agregamiento de texto	37
Frustración de equivalentes visibles	36
Función de búsqueda de texto	18
Funciones -breves descripciones	79
Funciones de movimiento de cursor	13
Funciones MICROSCRIPT - listado	79
Gestión de disco	59
Gestión de ficheros	59
Guardando aplicaciones de MICROSCRIPT	67
Guardando documentos en disco	16

Identificador de disco	8
Identificador de ductora	8
Impresión comandos de	51
Impresión de un documento	54
Impresión en modo ancho	52
Impresión en negrita	28
Impresión en negrita	32
Impresión parámetros de	55
Imprimiendo texto con MICROSCRIPT	54
Indentación de párrafos	34
Indexación de ficheros en disco	70
Inicio del documento, función	13
Inserción de líneas en blanco	21
Inserción de texto	15
Inserción, modo	37
Insertando texto mediante pegado	28
Intercambio global de texto	20
Interpretación de comandos durante la mezcla	38
Interpretador del texto a agregar	38
Introduciendo datos para el calculador	11
Introduciendo números para el cálculo	46
Invisibilidad de palabras	34
Justificación de líneas de texto	25
Líneas en blanco, inserción	36
Líneas, inserción de	21
Listando los documentos de un disco	61
Localizando errores en un documento	13
Manejo de datos	49
Manipulación de datos	49
Manual, organizacional del	1
Matemáticos, cambios de signo	49
Matemáticos, definición de campos	46
Mayúsculas, conversión	21
Mayúsculas/minúsculas	21
Memorias de abreviación	44
Memorias de abreviación	44
Mensajes de error	87
Menú principal	7
Minúsculas, conversión	21
Modo aprendizaje de programa	37
Modo auto-inserción	36
Modo auto-justificación	36

Modo auto-tabulación	35
Modo de auto-enguionado	49
Modo invisible	36
Modo retaguardia para programas	16,22
Modos alternativos	35
Modos alternativos en programación	75
Movimiento del cursor	13
Múltiple impacto al imprimir	31
Nombre de un documento	8
Numéricos, definiendo campos	46
Numéricos, registros	49
Palabra, definición de	3
Parámetros para imprimir	54
Párrafo reforma de	26
Párrafos, definición de	3
Pausas en programas	66
Pausas en programas para introducir datos	66
Pegado de texto en un documento	28
Preparación de discos para su uso	77
Principal, menú	7
Programa, caracteres de función para	86
Programación de aplicaciones	35
Programación de aplicaciones	64
Programas de aplicaciones guardados en disco	62
Programas MICROSCRIPT en disco	62
Programas retenidos como aplicaciones	67
Programas retenidos en disco	69
Quitando el texto después de recortado	28
Quitando una reglilla de la pantalla	11
Reclamando abreviaciones	44
Reclamando datos de los registros numéricos	49
Reclamando párrafos de ficheros en disco	46
Recomponer formato en pantalla	10
Recomposición de documentos en la impresión	55
Recortado y pegado	25
Recuperando datos almacenados en programas	75
Re-denominando a un documento	51
Reemplazando memorias de abreviación	44
Reglas para los nombres de los documentos	8
Reglilla corriente (en vigor)	10

Reglillas	10
Reglillas prescritas, cambiando parámetros	66
Reglillas, validación	11
Respaldo de documentos	35
Restauración del ordenador	7
Restaurando el ordenador	7
Reteniendo el formato durante el congegamiento	37
Reteniendo programas en abreviaciones	67
Rutas de programas hasta etiquetas prescritas	71
Rutinas de congegamiento, retención del formato	37
Rutinas para guardar documentos	16,22
Saltar a etiqueta, función al programar	73
Secuencia de operaciones aritméticas	46
Secuencias de control predefinidas	51
Secuencias de operaciones aritméticas	47
Selección de párrafos de un fichero en disco	70
Signos aritméticos	47
Símbolos para el calculador	46
Sistema operativo	5
Sobre impacto al imprimir	31
Solapando el texto en el pegado	28
Standard, códigos ASCII	85
Subrayado, función	21,30
Suministro del MICROSCRIPT en disco	62
Supresión de un carácter	13
Suprimiendo el efecto de los equivalentes	65
Tabulación, carácter de control	11
Tabulación de datos por reglilla	34
Texto a doble impacto	31
Texto en negrita	32
Texto, formato en pantalla	10
Texto, inserción de	15,20
Texto resaltado	31
Tipografiado vertical	35
Totalización de filas y columnas de valores	49
Totalización horizontal de columnas	49
Totalización vertical de columnas	50

Usuario, disco de	6
Usuario para documentos, nombre de	8
Validación de una reglilla	11
Vertical, tipografía	35
Visibles, equivalentes	6
Volteando texto en los extremos de línea	