

CAPÍTULO 7

PeCés: MS-DOS/Windows.

Aplicándonos el cuento

Un caso práctico: Luis Ricardo PeCé

Distribución a librerías
Mira Editores
Concepción Arenal, 22
50005 Zaragoza
Tel: 976 354165 Fax: 976 351043
e-mail: lcentral@ctu.es

El precio incluye envío
o contrarrembolso dentro
de España.

Distribución a particulares
Luis Membrado Giner
Andador Anayet, 4, 6º C
50015 Zaragoza
e-mail: lmg00009@inicio.es lmg00009@teletime.es

La cosa normalmente consiste en comenzar diciendo algo así como: “MS-DOS¹ es el sistema operativo del PeCé. Lo componen las órdenes que se describen a continuación y admiten los parámetros que en ellas se detallan. Son, en orden alfabético...”. Y lo que sigue a continuación es, en efecto, una descripción en orden alfabético de las órdenes que componen MS-DOS, con sus parámetros y para qué sirve cada cosa.

Para mí, meterme por ese camino resulta muy tentador. Es sencillo conseguir que el ordenador produzca de forma semiautomática más de la mitad del texto necesario para llenar las 30 o 40 páginas siguientes, y el resultado final suele permitir cubrir el expediente con nota a poco que se ponga un poquito de imaginación y las explicaciones sean claras.

No es eso lo que vamos a hacer, por desgracia para el que esto escribe y tal vez para el que lo lee. Al parecer, siento una cierta atracción inconsciente por las camisas de once varas y los abismos procelosos. ¿Porqué no?.

En primer y principal lugar porque lo que dijéramos sería en gran parte una repetición (quizá mejor explicada) de la información que el ordenador pone a nuestro alcance en todo momento. Como ya dijimos, una interfase de usuario puede hacerse bastante acogedora con sólo incorporar una serie de mecanismos (manuales en línea, parámetros de ayuda) que nos eviten la necesidad de memorizar con excesivo detalle. Y, felizmente, el sistema operativo de un PeCé cuenta ya con tan maravillosos inventos², por lo que con él tenemos de entrada un cúmulo de información de consulta accesible en todo momento que vendría a ser la que incluiríamos de haber seguido por el camino insinuado en el primer párrafo. Además, ya hay otros 200 libros dedicados a tal tarea, con títulos análogos a “Aprenda MS-DOS y Windows en 3 minutos”, “MS-DOS fácil, muy fácil,... y Windows más”, o “Pero como, ¿aún no es Vd. un experto en MS-DOS y Windows?”. Respetemos el esfuerzo realizado por sus autores, y no invadamos un terreno ya ampliamente cubierto por otros.

Y en segundo y no menos importante lugar, porque hay una buena cantidad de cosas además de órdenes y parámetros que son anteriores a ellos, que están en la línea de exposición que nosotros estamos siguiendo, y que conviene comprender antes de meternos a decirle “DIR”

- 1 La abreviatura corresponde, como ya deberíamos saber, a “Microsoft Disk Operating System”. Veremos a no mucho tardar que hay más fabricantes de DOS para PeCés. Muchas veces se emplea “MS-DOS” como genérico para referirse a cualquier DOS, sea o no el de Microsoft. Se intentará precisar un poco en lo sucesivo, aunque no debería Vd. hacerse muchas ilusiones.
- 2 No hace tanto, no crea. A los chicos del MS-DOS les costó mucho decidirse a montarlos. Pero al fin lo hicieron, o sea que alegrémonos y a usarlos cuando sea menester.

© Luis Membrado Giner. Todos los derechos reservados.
Se autoriza la copia sin modificación de los ficheros originales en formato Pdf. Si desea una copia impresa, por favor, compré un ejemplar en lugar de imprimirlo Vd. mismo. Le saldrá más económico y el resultado será de mejor calidad.

Título: Léeme, ya (Readme, Ist) Manual mínimo crítico para PeCés
Autor: Luis Membrado Giner **Editor:** John Pigeon Publisher
ISBN: 84-605-7033-9 **Dep. Legal:** Z-3314-97
Formato: 17x24 cm, 771 páginas **PVP:** 5.000 Ptas (30'05 euros)

CAPÍTULO 7. PeCés: MS-DOS/Windows.

o “CHKDSK” a Luis Ricardo PeCé (al que en adelante, y en aras de la brevedad, denominaremos LR-PeCé). O, si lo prefiere en plan telegráfico, porque el autor cree más conveniente enseñarle a acceder con provecho a las ayudas en línea del PeCé que contarle lo que va a encontrar en ellas.

El material que se presenta a continuación no es fácil de situar dentro de las clasificaciones habituales. Va de lo “pre-básico” y, por evidente, no tratado, a lo que se suele reservar para textos avanzados. Creo que tan horrible mezcolanza puede resultar digerible, como casi siempre, en tanto en cuanto no se profundice en detalles técnicos. En consecuencia, pasaremos sobre ellos con todo el sigilo permitido por nuestra innata pedantería. Aún así, intentaremos dar una visión general de las posibilidades ofrecidas en la actualidad por el sistema operativo del PeCé (MS-DOS) y su extensión GUI más popular (Windows). Describiremos, faltaría más, las dos o tres órdenes más importantes y explicaremos porqué lo son. Intentaremos que se entienda como son posibles cosas aparentemente maravillosas y explicaremos cuales son los trucos en que se basan. Trataremos ante todo de quitarle magia al PeCé y procuraremos que se vean las pocas ideas que hay debajo de su funcionamiento. Se recuerda que solucionar los problemas que el manejo cotidiano del ordenador nos va a proporcionar con prodigalidad es sólo posible si tenemos clara su forma de funcionar.

¿Y cómo vamos a hacerlo?. Espero que en este momento todo LAO esté gritando a coro con el resto de sus congéneres, cual si de un programa infantil se tratara, algo así como: “¡Viendo el sistema de ficheros y las interfases de usuario en los PeCés!”. ¡Pues claro que sí!

Tuercas 1: Discos

Los discos de LR-PeCé

Estamos frescos y descansados. Conviene que nos quitemos de encima lo antes posible aquellos detalles técnicos de obligado conocimiento que posteriormente necesitaremos y tal vez no sepamos donde meter. Veamos en primer lugar como se almacenan los datos en un disco, sea del tipo que sea, en un PeCé funcionando con el sistema operativo MS-DOS (o Windows), que es lo más normal³. Comencemos por el sistema de ficheros y pasemos a describir algunas de sus tuercas y tornillos.

La cuestión es que mientras en un disco de vinilo los datos se almacenan sobre su superficie como un surco en espiral, en un disco de ordenador los datos se guardan en cachos bien definidos de la misma a los que se denomina “sectores” (en inglés “sectors”)⁴.

Un sector no es más que un trozo de una de las bandas circulares sobre las que se puede colocar con comodidad la cabeza de lectura-escritura del disco. Puesto que éste se halla en rotación, y las cabezas avanzan o retroceden a pasos, no en continuo, debería estar claro

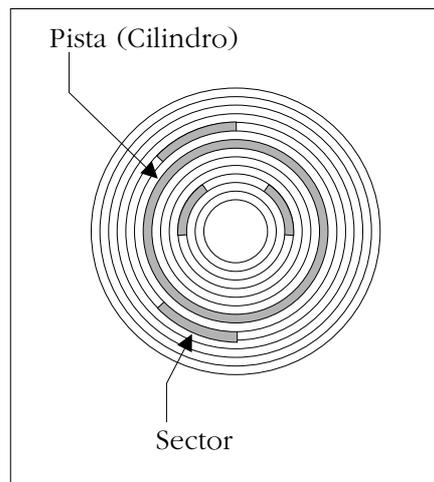
3 Subrayo esto porque es necesario que nos mentalizemos de que todo lo que sigue se refiere a este sistema operativo, que no es el único existente. Con otro sistema operativo, el mismo PeCé tal vez hiciera las cosas de forma muy diferente.

4 En un sector de un disco de PeCé suelen almacenarse 512 bytes. No es que tenga mayor importancia, pero quizá le apetecía saberlo.

Tuercas 1: Discos

porqué hablamos de bandas de grabación. A las que por cierto se denomina “pistas” (en inglés “tracks”). Y como en el caso de un disco duro tenemos varios platos y varias cabezas⁵, una pista en ellos adquiere una cierta profundidad, por lo que se habla de “cilindros” (en inglés “cylinders”).

Parece que procede poner una figura para que se vea un poco la cosa. En la que aparece a continuación se han sombreado una pista (sería un cilindro si habláramos de un disco duro) y varios sectores distribuidos al azar.



Bien. Hemos dicho que en un PeCé, los disquetes de 3'5" (pulgadas) y alta densidad son capaces de albergar 1'44 Megabytes. Si Vd. es más bien chinche, tal vez se haya fijado en que en las cajas de algunos de ellos aparecen notas similares a: “Capacidad sin formatear 2'0 Megabytes. Formateados para PeCé 1'44 Megabytes”. ¿Qué quiere decir eso?. El asunto consiste en que de lo que se trata es de almacenar ficheros dentro del antedicho disquete; que los ficheros eran datos almacenados con un nombre en un periférico; y que de igual modo que almacenamos los datos habrá que guardar el nombre bajo el que los dejamos. Y eso también ocupa sitio.

¿Cómo se grabará un fichero?. Supongamos que tenemos uno de tamaño moderado, cosa de 2 Kilobytes. Y supongamos asimismo que queremos grabarlo con el nombre de “pequeño.txt”, un nombre por ahora tan válido como otro cualquiera. Supongamos finalmente que le hemos dado la orden a LR-PeCé para que lo haga.

5 Al principio de los tiempos, los discos duros tenían de verdad varios “platos” metálicos montados uno encima de otro y equipados con sus correspondientes cabezas de lectura-escritura. Se hablaba de discos de 4 platos y 8 cabezas, por ejemplo. Ya no es tan frecuente que los discos que se montan en un PeCé tengan varios platos y el doble de cabezas, si bien se considera a nivel lógico (virtual, si nos gusta más la palabra) que sigue siendo así. Espere pues ser preguntado en alguna ocasión por cuantos platos o cabezas tiene su disco duro.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Se plantean en este momento interesantes cuestiones. A saber: ¿cómo sabe LR-PeCé los sectores del disco que están vacíos y son utilizables para el nuevo fichero?. Ya que espero que convenga conmigo en que no interesa que se borre ninguno de los ficheros ya existentes a base de que uno de los sectores en que están contenidos sus datos se vea sobrescrito al ir a grabar “pequeño.txt”. ¿Una vez localizados los sectores libres, con qué criterio podemos seleccionar los más convenientes para nuestros propósitos?. Y en cuanto al nombre ¿lo grabamos junto a los datos o en una zona aparte?. Si necesitamos más de un sector ¿cogemos varios seguidos o uno de aquí y otro de allá?. Y así podríamos seguir un buen rato.

Ninguna de las preguntas anteriores es irrelevante. De hecho todas pueden contestarse de diversas maneras, y elegir una u otra respuesta a la hora de diseñar un sistema de ficheros hace que éste adquiera unas propiedades determinadas. Que sea más seguro o menos, más rápido o más lento, más o menos tolerante a los errores, que aproveche mejor o peor el espacio de que se dispone en el disco.

En un PeCé la cosa se resuelve básicamente del siguiente modo: el sistema operativo se reserva dos zonas del disco para sí, una en la que guarda los nombres de los ficheros que se graban, y otra en la que lleva un índice de qué sectores corresponden a cada fichero (o están libres, es decir que no pertenecen a ninguno). Además, no es necesario que los sectores que corresponden a un mismo fichero estén juntos de verdad sobre la superficie del disco. O sea, que nuestro “pequeño.txt” podría consistir en 2 Kilobytes de datos almacenados en los cuatro sectores sueltos sombreados en la última figura, algunas entradas en la tabla de índices para dejar bien claro que esos sectores y sólo ellos le pertenecen, y su nombre “pequeño.txt” grabado en la zona reservada a la lista de nombres de ficheros (el directorio). No nos hace falta saber más.

Volviendo al tema de la capacidad, formatear un soporte consiste en grabar en su superficie una estructura de almacenamiento de datos acorde a los principios definidos por el sistema operativo con que lo vayamos a emplear. Como los que acabamos de nombrar, por ejemplo. Esto significa que el sistema operativo se va a reservar algunas de las zonas del disquete (o disco duro) para sus cosas, y el resto es lo que nos deja usar a nosotros. O sea, que tras preparar el disquete de 3’5" HD para trabajar en MS-DOS (formatearlo), sólo nos quedan 1’44 libres de verdad en lugar de 2’0 Megabytes.

Por si no lo ha notado, llevamos un rato hablando del sistema de ficheros de un PeCé trabajando en MS-DOS, en concreto de algunas de sus particularidades técnicas. Y no vendría mal que dijéramos como se llama tan importante componente del sistema operativo. Pues bien, por desgracia no tiene un nombre definido. Es innombrable, al parecer, o como mucho se le llama “sistema de ficheros F.A.T.”, lo que no puede ser más inapropiado. Por supuesto, no se traduce.

Y es inapropiado porque lo de F.A.T. no es sino el nombre de uno de los elementos ya citados que lo integran. A saber, la “File Allocation Table”. Traducido la “tabla de ubicación de ficheros”, es decir, la tabla de índices que se encarga de llevar el registro de qué sectores ocupa cada uno y cuales quedan libres para su uso a la hora de almacenar cosas nuevas. Al no grabar los ficheros en sectores contiguos, la antedicha tabla es fundamental para orientarse en el mar de sectores que componen un disco⁶.

6 Como se ha dicho, esta forma de trabajar tiene, como todo, sus ventajas y sus inconvenientes.

Tuercas 2: Teclado

Por ahora, vale. Por supuesto eso no es todo, pero no se asuste. Es casi todo. Quédese con la copla de que no hace falta que, como si de una casete se tratara, Vd. se dedique como un poseso a apuntar a mano en la etiqueta del disquete los nombres de los 100 ficheros que ha ido grabando⁷ en él. El sistema operativo construye y guarda esa lista por Vd., y va a poder mostrársela siempre que lo desee. Pronto veremos como pedírselo.

Tuercas 2: Teclado

Las teclas de LR-PeCé

Estamos frescos y descansados, aunque quizá algo menos que antes. Sigue conviniendo que nos quitemos de encima lo antes posible aquellos detalles tal vez algo prolijos y quizá demasiado técnicos que, a pesar de ello, son de obligado conocimiento y tal vez no sepamos luego donde meter. Ocupémonos en segundo lugar nuestro instrumento de mandarle cosas a LR-PeCé. Pasemos a la interfase de usuario y procedamos a describir algunas de sus tuercas y tornillos.

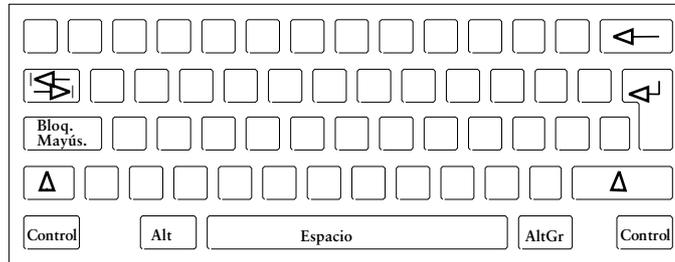
Puesto que MS-DOS es un sistema operativo de los que se comunican a través de una línea de órdenes, su uso se basará ante todo en el empleo del teclado. Mal vamos a poder manejar a LR-PeCé si no lo conocemos un poco. No nos vendría mal por lo tanto ver qué teclas tenemos y para qué sirven las más importantes. En el capítulo anterior ya mostramos un teclado tipo AT, el más normal hoy en día en un PeCé, con los distintos bloques señalados. Pasemos a verlos de uno en uno.

Y comencemos por el más familiar, el que más se parece a lo que de toda vida hemos visto en las máquinas de escribir. Es decir, por el bloque alfanumérico. Sin especificar las teclas alfabéticas, que deberían resultar identificables por sí mismas, su aspecto en un teclado castellano⁸ es como sigue:

Entre las primeras se encuentra que se aprovecha el espacio bastante bien, ya que todo sector libre es susceptible de ser utilizado a la hora de grabar un fichero nuevo. Y entre los segundos, un riesgo bastante cierto de que cualquier daño en la FAT dé lugar a pérdidas de datos y una cierta lentitud de operación, ya que al leer o escribir puede ser necesario desplazar repetidamente la cabeza del lector, lo que cuesta un tiempo precioso. No es en absoluto el único esquema posible, y sistemas operativos hay que insisten en que los ficheros se graben en sectores contiguos y un fichero tras otro, sin dejar espacios vacíos. No insistiremos más. Tan sólo diremos que MS-DOS NO PROHIBE que los sectores de un fichero se encuentren juntos. Simplemente, cuando no puede grabarlos de esta forma, los distribuye por el disco lo mejor que puede.

- 7 Dificilmente podrán los recién llegados a los ordenadores hacerse idea de como era la microinformática cuando SÍ QUE SE TRATABA DE UNA CASETE Y SÍ QUE HABÍA QUE HACERLO. El autor lo sufrió en sus carnes y garantiza que la adquisición de una unidad de disquetes para su New Brain supuso toda una liberación.
- 8 El teclado debe adaptarse a los usos y costumbres de cada país e idioma. No es tan simple. ¿Sabía Vd. por ejemplo que los teclados franceses no siguen la distribución normal de letras?. En lugar de empezar con QWERTY, usan una disposición AZERTY. Además, las cifras (en la fila superior) se obtienen pulsando a la vez la tecla de alzamiento de mayúsculas. Como cada idioma tiene predilección por algunos caracteres más bien raros, determinadas teclas alfabéticas cambian con frecuencia de un país a otro. Un teclado español incluye la “ñ”, el acento, la diéresis y hasta la cedilla.

CAPÍTULO 7. PeCés: MS-DOS/Windows.



Antes de que se me subleve la tropa, debo advertir que aunque las rotulaciones están en castellano o emplean símbolos gráficos, que es lo más común (no he hecho sino fusilar lo que veo en los teclados que uso todos los días), indicaremos al tratarlas sus denominaciones en inglés. No se trata de exhibir estúpidamente mi dudosa erudición, ni de incordiar lo más posible al lector. En la práctica, no está de más conocerlas, se corresponden mejor con la tabla de códigos ASCII que generan al pulsarlas, y no son en absoluto complicadas.

Pasemos a estudiar un poco el dibujo, y comencemos por la tecla gorda de la derecha, nuestra vieja amiga, cuyo fin ya explicamos⁹. Es la que cierra las líneas 2 y 3, y luce un símbolo gráfico más o menos equivalente a la sigla CR, abreviatura de “Carriage Return” o “Retorno de carro”, por los motivos que ya deberíamos conocer. No olvidemos que, en esencia, el bloque alfanumérico es una adaptación muy directa del teclado de una máquina de escribir eléctrica.

Siguiendo con las adopciones directas desde el teclado de máquina de escribir, es necesario citar la tecla de retroceso (o “BackSpace”, es la gorda de la derecha de la fila 1), las teclas de alzamiento de mayúsculas (“Caps”, abreviatura de “Capitals”, que significa mayúsculas en inglés y que también se suele nombrar “Shift”, traducible por “Alzamiento”, y que son las de los extremos izquierdo y derecho de la línea 3 con un triángulito hacia arriba), la tecla de tabulación (“Tab”, la de la izquierda de la fila 2, con dos flechas apuntando en sentidos opuestos), el bloqueo de mayúsculas (en inglés “Caps Lock” y para variar rotulado en castellano, o sea que búsquelo a ver si lo encuentra) y la barra espaciadora (“Space”, en la figura llamada “Espacio” y en la práctica totalmente en blanco). Son necesarias algunas aclaraciones.

En primer lugar, que mientras en una máquina de escribir la barra espaciadora se usa para desplazar la posición en la que escribimos, en un teclado de ordenador genera un carácter como cualquier tecla alfabética. Genera un espacio en blanco, tan válido como una coma, una “q”, o una “ñ”. No se emplea para desplazar nada. Para movernos tenemos otras teclas que ya veremos (las del cursor). La barra espaciadora suele servir asimismo para marcar opciones en esos paneles de diálogo típicos de los GUIs. Conviene saberlo por si no tenemos el ratón a mano, y porque los tales paneles aparecen en bastantes más sitios que en esos GUIs tan modernos.

9 Por si alguien no se acuerda, en una interfase de línea de órdenes es la tecla que emplearemos para decirle a L.R.PeCé que hemos terminado de escribir una orden, que tenga a bien leerla, y que, si sus conocimientos se lo permiten, proceda a llevarla a cabo de inmediato.

Tuercas 2: Teclado

La tecla de retroceso es la primera tecla de corrección (“edición”, en jerga inglesa) que encontramos. Se encarga de borrar el carácter a la izquierda de la posición del cursor. Y al hilo de su descripción podemos pasar a ocuparnos de algunas consideraciones de ergonomía que dejamos pendientes al hablar de los teclados en el capítulo anterior. Según el teclado que Vd. adquiera, esta tecla puede ser más gorda de lo normal, como se ha representado en la figura. Es conveniente que sea así, pues es una tecla que se usa con mucha frecuencia ya que solemos meter la pata bastante al escribir y una de las grandes ventajas de los ordenadores es la facilidad con que pueden corregirse los errores. También la tecla de retorno de carro se usa mucho, y por tanto conviene asimismo que sea en efecto una tecla gorda, como la de la figura.

La tecla de tabulación se emplea para insertar tabuladores en un documento. Y si Vd. se molesta en mirar el código ASCII, verá que, de nuevo, al igual que ocurría con el espacio en blanco, el tabulador es otro carácter más, al igual que la coma, la “q”, o la “ñ”. No es un desplazamiento del carro¹⁰. Esta tecla tiene frecuentemente otro uso. Cuando tenemos varias opciones (paneles de diálogo de nuevo) o varias zonas de trabajo en un programa, se encarga de pasarnos de una o otra. Podemos ir hacia delante pulsando “Tab”, o hacia atrás pulsándola junto con la tecla de alzamiento “Shift+Tab”. Y por favor, lea la nota al pie¹¹

Las teclas de alzamiento de mayúsculas (“Shift”, que es el nombre más habitual, o “Caps”) se emplean como en una máquina de escribir normal, es decir, para obtener una letra mayúscula si se pulsa en combinación con una letra alfabética, o para sacar el carácter de la parte de arriba con aquellas teclas que tienen dos símbolos grabados. El bloqueo de mayúsculas fija el teclado en este modo, y una luz en algún sitio del mismo se encarga de recordárnoslo. Por motivos de ergonomía es mejor que dicho testigo se encuentre en la misma tecla de bloqueo, aunque es frecuente que se halle en la esquina superior derecha del teclado, junto a otros dos que luego veremos. En un teclado de ordenador, este bloqueo es algo particular, ya que sólo actúa sobre las teclas alfabéticas, no sobre los números y los signos de puntuación, y además se desactiva tan sólo cuando se pulsa de nuevo¹², no cuando le

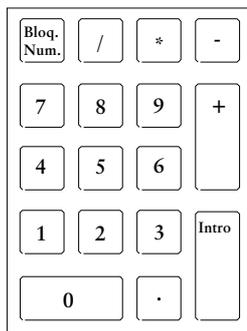
-
- 10 Si Vd. emplea en algún momento un procesador de textos, la pulsación de la tecla de tabulación le desplazará el cursor a la siguiente posición del tabulador. Es así porque dicho programa se encarga de simular en lo posible una máquina de escribir. Pero conviene que no olvide que lo que en realidad ha hecho es insertar un carácter particular en una posición del texto que está creando.
 - 11 Para un novato, esto de pulsar dos teclas es más difícil de lo que parece. Cuando estaba dando clases de introducción a la informática observé frecuentemente, para mi sorpresa, que muchos recién llegados creen que pulsar “Caps+Tab”, o en general cualquier tecla de alzamiento junto a otra, consiste bien en pulsarlas sucesivamente (primero una y luego otra) o bien en pulsarlas simultáneamente (tal como si estuviéramos poniendo un buen par de banderillas). Nada más lejos de la realidad. Se pulsa en primer lugar la primera tecla que se indica, y, sin soltarla, asegurándonos de que la tenemos perfectamente asida por el cuello, se pulsa luego la otra. Perpetrado el estrangulamiento, pueden soltarse ambas. El procedimiento no es distinto al que se emplea para generar una “A” mayúscula, por ejemplo, en una máquina de escribir. Y es por eso por lo que no entiendo porqué mucha gente se monta semejantes líos.
 - 12 Este tipo de funcionamiento, en el que una misma tecla pone y quita algo, puede denominarse “de conmutador” y es muy típico en informática y de uso universal para los distintos bloqueos que pueden activarse en un teclado.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

damos a la tecla de alzamiento simple. Es así ya que en bastantes ocasiones nos interesa poder acceder con rapidez y simultáneamente a mayúsculas, números, y signos de puntuación más frecuentes, que son justo los que se encuentran en la parte inferior de las teclas. Cuidado, pues, los buenos mecanógrafos.

Y tan sólo nos quedan aquellas teclas de alzamiento raras que vimos eran típicas de los teclados de ordenador. Tenemos por supuesto las teclas de control (“Ctrl”), encargadas de generar los correspondientes caracteres de este tipo cuando se pulsen junto con una tecla alfanumérica. Hay dos, una a cada lado, igual que hay dos teclas de mayúsculas. Y luego hay otras dos un poco más particulares, las rotuladas “Alt” y “AltGr”, a ambos lados de la barra espaciadora. No merece la pena intentar explicar tales abreviaturas. “AltGr” da acceso al carácter que aparece en la cara delantera de aquellas teclas que lucen orgullosamente tres símbolos en lugar de uno, y baste decir que, aunque parezcan raros, son de uso muy frecuente en la comunicación con el Pecé. Y la tecla “Alt” permite generar en un PeCé cualquier código ASCII que deseemos, aunque no se encuentre representado explícitamente en el teclado. Para obtener por ejemplo el carácter “~”, correspondiente al ASCII 126, basta mantener pulsada la tecla Alt mientras se tecléa en el bloque numérico 126¹³. Y permite además con bastante frecuencia activar los menús en entornos GUI o similares (Windows), si es que no se puede o no se desea recurrir para ello al ratón.

Pasemos al bloque numérico. Su aspecto real es ligeramente más complicado que el que aquí se ve. En la mayor parte de las teclas que corresponden a cifras aparecen otras inscripciones.



Recuerde el LAO lo que se dijo en el capítulo anterior sobre las aberraciones cometidas por IBM en el diseño de su primer teclado para PeCé y las lamentables consecuencias de ese error primigenio sobre el teclado del AT. Lo que se muestra es lo que debería ser y lo que haremos que sea. Dentro de poco diremos como. Ignoraremos el resto.

Por lo demás, sólo tiene dos cosas que pueden chocar un poquito. Una tecla “CR”, idéntica a la del teclado alfabético y con la que se puede usar indistintamente, pero que por algún motivo indecifrible suele ahora lucir el nombre “Intro”, lo que nos obliga además a aclarar que la tecla “CR” recibe también este nombre y en ocasiones hasta el de “Enter”, en inglés por si fuera poco. Y justo en el extremo opuesto una tecla de bloqueo, necesaria para evitar las aberraciones ya relatadas. Esta última nos permitirá restaurar la cordura si, en algún momento y por alguna razón desconocida, el bloque numérico deja de funcionar como tal. Un testigo al efecto, de los dos que nos quedaban, se encarga de chivarse de que el bloque numérico está activado, algo que debe llenarnos de satisfacción.

Y ahora, ocupémonos del bloque de teclas de función, cuyo aspecto es cual sigue:

13 Estas teclas son típicas de TODOS los teclados de ordenador. En el Mac, por ejemplo, también existen. Recibían, eso sí, nombres menos agresivos. “Tecla Manzana”, “Tecla Comando”... y se rotularon con caprichosos dibujitos. No entraremos a comentar las enormes ventajas de ergonomía y funcionalidad que tan audaz revolución en el diseño de los teclados de ordenador proporcionó a los usuarios del Mac pues se comentan solas.

Tuercas 2: Teclado



Estrictamente hablando, las teclas de función son las 12 que lucen un nombre indefinido de F-algo, y no deberían desempeñar ninguna acción concreta. Cada programa podrá usarlas como desee para sus propios fines, y lo que hagan en cada caso debería ser explicado mediante algún mecanismo fácil de descifrar, tal como el interfase Hewlett-Packard del que ya hablamos. No siempre es así, como a estas alturas ya resultará obvio, pero no puede decirse mucho más de ellas.

Hemos incluido algunas más, de todos modos, por cuanto se encuentran en la misma fila, y es de estas de las que más hay que hablar. La primera empezando por la izquierda, rotulada “Esc” y universalmente conocida como “Escape”, es la más importante con diferencia. Su objetivo es más o menos el opuesto a “CR”. Si estamos en algún lugar en el que no deseemos seguir, si queremos salir de un menú, o anular una opción y volver a la anterior, esta tecla se encarga de hacerlo. La podemos leer como “Déjalo estar y vuelve al sitio anterior”. Si se encuentra en apuros, púlsela. Mírela con atención y memorice su posición en el teclado porque es de las que se usan con mayor frecuencia.

El bloque de tres de la derecha tiene asimismo una función bien definida en cada caso. La primera, “Impr. Pant.” no hace sino enviar a la impresora lo que veamos en la pantalla en un momento determinado. Por ello se denomina así, simplemente “Imprimir Pantalla” abreviado. Funciona muy bien si estamos en modo texto y tiene conectada una impresora, pero yo no lo intentaría en modo gráfico, al menos por ahora, ni depositaría grandes esperanzas en este mecanismo a la hora de imprimir nada. Es un recurso interesante de conocer si queremos en algún momento una copia en papel de la pantalla que el ordenador nos muestra en MS-DOS. Fue de este modo como nuestro desdichado pardillo del capítulo 5 consiguió por error obtener una copia impresa.

Olvídese de que existe la segunda, la rotulada “Bloq. Despl.”, abreviatura de “Bloqueo de Desplazamiento”, pues su fin no es sino convertir el bloque numérico en el antiguo bloque de edición de los teclados anteriores al AT, lo que suele proporcionar serios quebraderos de cabeza. El último testigo que quedaba nos informará oportunamente de tan triste situación. Mírela tan sólo para asegurarse de que está desconectada esta forma de trabajar, pues por desgracia y aduciendo motivos de compatibilidad no ha sido enviada todavía a las profundidades de los infiernos exteriores.

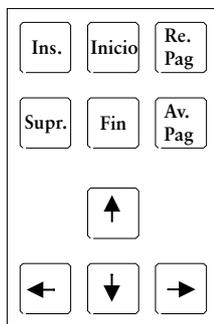
La tercera, rotulada “Pausa”, permite paralizar la pantalla del PeCé, por supuesto en modo texto y siempre que contemos con unos reflejos felinos. Aunque eso sí, siempre es un poco demasiado tarde cuando la pulsamos y nunca suele producir el efecto deseado. Invariablemente nos encontramos con que aquello que queríamos leer ya se ha perdido. La pantalla recobra vida en cuanto pulsamos cualquier otra tecla. No es muy útil, no. Hay otros medios mucho más adecuados para visualizar correctamente un texto largo y luego nos ocuparemos de ellos. En su cara delantera luce el Rótulo “Inter”, de “Interrumpir”. Si la pulsamos junto con la tecla de Control (“Ctrl+Inter”), algunos programas, no todos, se terminan. En resumen, este pequeño grupo de teclas no tiene mayor interés para un usuario no experto. Incluso los viejos lobos de mar las emplean sólo de vez en cuando.

Terminaremos con las teclas de función en general ocupándonos de una que realmente debería lucir su propio nombre, y orgullosamente por cierto. La tecla F1, la más cercana a

CAPÍTULO 7. PeCés: MS-DOS/Windows.

“Escape”, tiene desde tiempo inmemorial un cometido único en todo programa de PeCé. Nos lleva a las pantallas de ayuda, y por eso mismo es fundamental conocerla. Si Vd. está usando un programa y no sabe qué hacer, pulse F1. Es probable que ante sus ojos se desplieguen a continuación una serie de explicaciones más o menos detalladas, los ya famosos textos de ayuda. Al igual que decíamos con la tecla “Escape”, si se encuentra en apuros, púlsela. ¿Que por qué no se llama “Ayuda” en lugar de F1?. La tradición. Se cuenta con que todo usuario de PeCés lo sabe desde que nace, lo que no siempre es cierto. No se olvide de ella¹⁴.

Y para acabar con los bloques del teclado, presentemos a aquel que más vamos a usar tras el alfabético y que se va a convertir en nuestro mejor amigo, pues nos va a permitir corregir y movernos: el bloque de edición. No me diga que no tiene un aspecto acogedor.



Las cuatro teclas de abajo, las que están formando una “T” invertida y tienen unas flechitas, son las archifamosas teclas de cursor. Si es posible, es decir si hay sitio en la dirección elegida, las horizontales desplazan el cursor una posición a derecha o izquierda. Lo desplazan, digo, no borran nada, o no deberían hacerlo. Y si por ventura queremos desplazar el cursor en horizontal más deprisa, digamos al principio de la palabra anterior o de la siguiente, suele ser posible lograrlo pulsando las teclas de cursor horizontales junto a la tecla “Control”. No está de más saberlo, se lo aseguro.

Las verticales son algo más complicadas. Con según qué programas, en concreto con aquellos que se comuniquen con nosotros en ese momento mediante un editor de pantalla completa, no hacen sino desplazar el cursor arriba o abajo una línea. Con aquellos que empleen una línea de órdenes mejorada pueden tener a bien recuperar la última orden que hayamos dado. Además, ya vimos que las teclas de cursor, todas, estaban activamente implicadas en el mecanismo de selección de opciones de los menús modernos.

Pasemos al bloque de seis que está sobre las teclas de cursor. De izquierda a derecha encontramos primero “Ins.” y “Supr.”. La primera es la tecla “Insertar” y se encarga de conmutar el modo de escritura. Si no está activada y escribo sobre un texto ya existente, simplemente lo nuevo sustituye a lo ya escrito y no deja ni rastro. Se habla de “sobrescribir”. Mucho mejor que el “Typex”. Y si la pulso y escribo a continuación, el texto que creo se va haciendo hueco y desplazando el ya existente a su derecha, conservándolo. Un invento maravilloso. Desgraciadamente no hay un testigo, aunque algunos programas tienen la delicadeza de indicar cuando se va a insertar o a sobrescribir mediante un rótulo en pantalla

14 En realidad F1 puede, o no, llevarnos a un texto de ayuda. Y éste puede ser, o no, sensible al contexto. Y tal vez sea inteligible. O no. No olvide en ningún momento ese “casi” implícito desde hace muchos capítulos. Al final, cada programa hace las cosas como desea su programador y, en ocasiones, no se respetan ni las más sagradas convenciones. Y la de F1 es de las más intocables. Lo que no ha impedido que programas de gran éxito la ignoraran. Por ejemplo WordPerfect, que en su momento usaba F3 y no F1 para dar acceso a las ayudas. Más adelante volveremos sobre como WordPerfect pudo sobrevivir a semejante desatino, hace tiempo felizmente superado, pero por ahora no olvide lo anterior.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

en su totalidad; si estamos señalando un fichero, pulsar cualquiera de ellas hará que se borre; y si es un elemento de un dibujo, pues también. No es nada mala la idea y funciona francamente bien. Conviene que se acostumbre a asignar acciones más generales que concretas a esas teclas tan interesantes que acabamos de ver. “F1” es la ayuda en casi cualquier parte, “ESCAPE” sale de casi cualquier lugar, “SUPRIMIR” y “RETROCESO” borran casi cualquier cosa y “RETORNO DE CARRO” dice que sí a todo lo que se le pregunte. ¿Entendido?.

Y porque también se me olvidaba y para que vea como son las cosas, sepa asimismo que algunas teclas no hacen sino reproducir el comportamiento de combinaciones con la tecla de control que aparecieron mucho antes. Así, “Ctrl+C” (Control y C) es equivalente a “Inter.”, “Ctrl+P” es similar en cierta forma a “Impr. Pant.” y “Ctrl+S” a veces es como “Pausa”. Como dijimos, estos recursos siguen existiendo, aunque ocultos tras un velo de silencio.

¿Contento, ahora que ya sabe para que sirven las teclas?. Desde luego, puede sentirse legítimamente satisfecho, pero enfríe un poco su entusiasmo. La más cruda realidad consiste en que el teclado no es para el ordenador más que un montón de botones sin significado, y cada programa puede emplearlo de la forma que desee. Conviene no olvidarlo aunque no sea frecuente que las distintas teclas hagan otra cosa que lo expuesto. Se suele intentar no despistar al usuario, que puede experimentar cierto pasmo si al pulsar la “A”, por ejemplo, se le borra la pantalla. Sin embargo hay que estar preparado para que ciertos programas usen algunas teclas de forma en principio sorprendente. Por ejemplo, la tecla “Alt”, por sí sola y como ya vimos, activará en Windows la barra de menú de la ventana en que nos encontremos. Algunos juegos pueden leer la pulsación de las teclas de “Control” para activar el disparo de un arma figurada. Y así sucesivamente. Las teclas de alzamiento (todas ellas) suelen verse envueltas asiduamente en este tipo de manejos, y no es extraño tener que pulsar “Ctrl+Shift+otra tecla” para acceder a alguna sutil variación en el funcionamiento de un programa. Preparémonos pues para que nuestro teclado “haga cosas raras” de vez en cuando.

¿Y el ratón?. No existe en MS-DOS, que emplea un interfase de línea de órdenes. Nos ocuparemos del ratón cuando hablemos de Windows¹⁷.

que es más sencillo con el ratón. En ambos casos, la zona marcada se indica de algún modo (visualizándola en video inverso, por ejemplo). Existen un buen montón de alternativas para facilitar y mejorar tan frecuente operación y que permiten entre otras cosas seleccionar bloques discontinuos o especialmente grandes. Con la práctica los irá aprendiendo y, si desea más pistas, en las ayudas de los programas suelen indicarse este tipo de cosas. Una razón más para que las mire.

17 Los programas MS-DOS son perfectamente capaces de usar el ratón, de desplegar una interfase gráfica propia, y de cualquier otra cosa. Lo que quiero decir es que el ratón no es lo que MS-DOS requiere para su funcionamiento básico. O si lo quiere de otra forma, Windows, como ya vimos y veremos de nuevo, no es sino una extensión de MS-DOS, una capa de software que añade el ratón y otras muchas cosas a lo que MS-DOS sabe hacer solito.

Tuercas 3: Nombres

Como llama LR-PeCé a sus cosas

Estamos frescos y descansados, aunque a estas alturas probablemente tenemos nuestras dudas de que sea cierto. Sigue conviniendo que nos quitemos de encima lo antes posible aquellos detalles tal vez algo prolijos y quizá demasiado técnicos que... bla, bla, bla... . Tras ocuparnos del teclado, es decir de asuntos relativos a la interfase de usuario en un PeCé, tiempo es de que volvamos de nuevo la mirada al sistema de ficheros. Sigamos con él y procedamos una vez más a describir algunas de sus tuercas y tornillos.

Un fichero son datos guardados en un medio de almacenamiento con un nombre, ya lo vimos y lo volvemos a repetir a ver si se le queda de una vez. Para poder seguir con comodidad, conviene aclarar qué nombres podemos dar a los ficheros en MS-DOS. Cuales son las reglas a observar para nombrarlos.

En realidad es muy simple. El nombre de un fichero en MS-DOS debe seguir una antigua convención informática y ajustarse al formato "8.3". Lo que no quiere decir nada más que puede tener hasta 8 caracteres y, si así se desea, añadir una extensión de hasta 3 más, eso sí, separada de lo anterior por un punto. Nombres válidos serán por lo tanto el ya referido PEQUEÑO.TXT y otros muchos que podemos inventar sobre la marcha, como FICHERO1.WPD, PAISAJE.GPH, TEXTO21.DOC, CARTA5, FACTURAS, NADADAT, y NADA.DAT, que no es igual que el anterior. ¿Que quiere más?. Dejemos que el ordenador en que estoy trabajando nos muestre un poco sus tripas y aquí van un buen montón:

1050.CPI	4201.CPI	4208.CPI	5202.CPI	ANSI.SYS
APPEND.EXE	ASSIGN.COM	ATTRIB.EXE	BACKUP.COM	CHKDSK.EXE
CHOICE.COM	COMMAND.ICO	COMP.COM	COUNTRY.SYS	CREATE.COM
CURSOR.EXE	DBLBUF.SYS	DCONVERT.COM	DCONVRT2.EXE	DEBUG.EXE
DEFAULT.FB	DEFAULT.INI	DEFAULT.ND7	DELPURGE.EXE	DELWATCH.EXE
DEVSWAP.COM	DISKCOMP.COM	DISKCOPY.COM	DISKMAP.EXE	DISKOPT.EXE
DISPLAY.SYS	DOSBOOK.EXE	DOSKEY.EXE	DPMI.EXE	DPMS.EXE
DRIVER.SYS	E.EXE	EDIT.COM	EGA.CPI	EMM386.EXE
EMMXMA.SYS	EXE2BIN.EXE	FASTOPEN.COM	FBWSCHED.DAT	FBWX.EXE
FBWXDRVR.DLL	FBWXHELP.HLP	FBX.EXE	FBXFIL.HLP	FBXHELP.HLP
FC.COM	FDISK.COM	FILELINK.EXE	FIND.EXE	FORMAT.COM
GRAFTABL.COM	GRAPHICS.COM	HELP.BAT	HIMEM.SYS	HWTESTX.FBM
JOIN.EXE	KEYB.COM	LABEL.COM	LEAME.TXT	LOCK.EXE
MEM.EXE	MEMMAX.COM	MODE.COM	MORE.COM	MOVE.EXE
NETWARS.EXE	NLSFUNC.EXE	NWCACHE.EXE	NWCACHE.OV1	NWCACHE.OV2
NWCDEX.EXE	NWDOS.INI	NWDOS.ND7	PASSWORD.EXE	PNUNPACK.EXE
POROMISI.FB	PREVIEW.EXE	PRINT.COM	PRINTER.SYS	README.TXT
RECOVER.COM	RENDIR.EXE	REPLACE.EXE	STORE.COM	RPLDOC.TXT
SCONVERT.EXE	SCREATE.SYS	SCREXEC.EXE	SCRIPT.EXE	SDEFRAG.COM
SDEFRAG2.EXE	SDRES.EXE	SDSCAN.EXE	SECURITY.OVL	SERNO.EXE
SETUP.EXE	SETUP.HLP	SETUP.INI	SETUP.ND7	SETUP2.EXE
SETVER.EXE	SHARE.EXE	SORT.EXE	STACFM.DLL	STACFM.HLP
STACHIGH.SYS	STACKER.COM	STACKER.EXE	STACLOAD.BIN	SUBST.EXE
SYS.COM	TASKMGR.BAK	TASKMGR.EXE	TASKMGR.INI	TASKMGR.ND7
TOUCH.EXE	TREE.COM	UI.CDB	UNDELETE.EXE	UNFORMAT.COM
UNINSTAL.EXE	UNSTACK.COM	UNSTACK2.EXE	UT.CDB	VDISK.SYS
VIPX.386	VIR-INFO.VRB	WINSWAP2.EXE	WSDRES.EXE	WSDSCAN.EXE
WSDSCAN.HLP	XCOPY.EXE	XDEL.EXE	XDIR.EXE	

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Una inspección cuidadosa de lo anterior permite extraer interesantes conclusiones. En primer lugar, que el 8.3 da para un montón de nombres diferentes, por lo que es difícil que, con un poquito de cuidado, se repitan¹⁸. En segundo lugar, que normalmente sólo se emplean letras y cifras para componerlos. No hace falta mucho más. Si acaso desea hacer separaciones, puede añadir a la lista el guión y el subrayado. DAT_0001.TXT o DAT-0001.TXT son nombres asimismo perfectamente válidos. No separe mediante espacios en blanco. No se complique la vida y créame, con esto tiene más que suficiente. Letras, cifras, guión o subrayado, y a funcionar. Por cierto al hablar de letras, da igual mayúsculas que minúsculas. No se distinguen en el sistema de ficheros FAT¹⁹.

La convención 8.3 lleva implícitas algunas otras cosas. Por ejemplo que la extensión de tres letras tras el punto de separación va a ser empleada para indicar el tipo de datos que contiene el fichero en cuestión. Como recordará, éste no era sino uno de los mecanismos que permitían darle pistas al ordenador añadiendo algo de significado a las cosas que maneja, volvemos a repetir de nuevo a ver si se le queda de una vez. En realidad no es sólo el ordenador el que necesita pistas. Si Vd. deja una carta a su amante esposa almacenada bajo la extravagante denominación XXR-T21, pues bueno, pues es válida y el ordenador no se va a quejar. Pero esté seguro de que un mes después ni Vd. va a saber, cuando vea semejante nombre, qué diablos es lo que esconde. Y tampoco va a ser sencillo echarle un vistazo, pues si por ejemplo lo ha creado con el programa A y lo intenta mirar con el programa B, tal vez no se deje. Es por tanto de obligado cumplimiento en todo usuario de un ordenador la siguiente regla: ponga a sus ficheros nombres fáciles de leer, que permitan deducir su contenido y el formato con que se han almacenado.

Observando la regla anterior, elegiríamos por ejemplo “esposa”, como nombre para la antedicha carta. Para saber el tipo de datos añadiríamos una extensión. Si estuviera en ASCII, un formato como vimos universal y legible con multitud de programas, podríamos añadir “.txt” o “.asc”. Si la hubiéramos creado mediante WordPerfect 5.2, que emplea un formato de datos más particular, podríamos añadir una extensión “.wp5”. O “.wri” si en lugar de WordPerfect 5.2 fuera Write, el procesadorcillo de textos de Windows, el instrumento empleado para escribirla. La cosa quedaría al final como ESPOSA.WP5, por ejemplo.

Lo de los formatos en que se almacenan las cosas es tan importante (y crea tantos problemas), que algunos programas acostumbran a añadir las extensiones ellos solitos cuando se graban las cosas. Por distintos motivos no acaba de ser tan buena idea como parece, y no es nada raro que sea el usuario final, es decir Vd., el que deba, o al menos pueda si así lo quiere,

18 ¿Cuántos, pregunta?. No vamos a hacer un cálculo exacto. Tirando por lo bajo tenemos 30 caracteres diferentes (hay muchos más), en grupos de 11 y con posibilidad de repetirlos. Lo que hace la bonita cifra de 30^{11} , es decir $1'7715 \times 10^{16}$ o 17.715 seguido de 12 ceros, es decir 17.715 billones españoles, billones de verdad, no como los anglosajones que son de mil millones. ¿Le parecen bastantes como estimación a la baja?.

19 Una nota algo estúpida, pero que tal vez le venga bien. Por supuesto el cero y la “o” mayúscula, o el uno y la “l” minúscula, por citar sólo dos antiguas convenciones mecanográficas, sí que se distinguen. Cuidado porque, en ocasiones, su apariencia al verlos escritos en pantalla puede inducir a confusión y llevar a persistentes rascamientos de cabeza. La “l” minúscula no existe en un nombre de fichero, pero algunos programas pueden listar los nombres en minúsculas. En Windows, por ejemplo, es lo habitual.

Tuercas 3: Nombres

especificar la extensión deseada. Como sobre lo de los formatos ya dijimos algo, y vamos a ir diciendo muchísimo más, lo dejaremos estar así por ahora, pero váyase haciendo a la idea de poner extensiones a sus archivos.

En un sistema de ficheros era necesario dar nombre a algunas cosas más. En concreto, las unidades de almacenamiento debían tener el suyo para poder referirnos a ellas y decirle al ordenador que mire en el disquete o en el disco duro, o en el CD-ROM o en el magnetoóptico regrabable que acabamos de comprar. Y en el caso de que tengamos la suerte de estar en un sistema de ficheros jerárquico, como por ventura ocurre con el PeCé y el FAT, habrá que ver qué nombre les damos a los directorios. Por ponerlo de otra forma, vamos a ver qué tenemos que saber sobre el nombre completo de los ficheros.

Lo de las unidades se resuelve siguiendo un método ligeramente chocante a primera vista. Se denominan mediante letras únicas seguidas de “:”. Las dos primeras, por si no se acuerda la “A” y la “B”, se reservan por MS-DOS para las unidades de disquetes. La “C” se reserva para el primer disco duro del sistema. Y de ahí en adelante el sistema operativo va designando los demás periféricos de almacenamiento con las siguientes. Un PeCé típico hoy día, como el que describimos en el anterior capítulo, tendrá una unidad “A:” que será un lector de disquetes de 3’5”, y una unidad “C:”, que será el disco duro.

Complicuemos un poco la cosa. Si Vd. monta, por algún ignoto motivo, un lector de disquetes adicional de 5’25” y otro disco duro, tendrá entonces además una unidad “B:” (el lector de disquetes de 5’25 pulgadas) y un disco duro “D:” (el segundo que ha montado). Tales adiciones son sencillas ya que, como recordará, las tarjetas controladoras de disquetes admiten más de un lector, y el controlador IDE soporta hasta dos discos duros²⁰. Algo más difícil es añadir un CD-ROM, que pasaría a ser la unidad “E:” de este ejemplo. MS-DOS va asignando más o menos automáticamente las correspondientes letras²¹.

20 En los tiempos que corren, con programas cada vez más gordos y discos duros cada vez de mayor capacidad y más baratos, ésta es una posibilidad de ampliación del ordenador francamente interesante. O puede ser que Vd. quiera comprar un ordenador nuevo, no tenga destino para el que va a sustituir, y quiera aprovecharlo en lo posible. Es perfectamente viable desmontar el disco duro viejo e instalarlo en el nuevo ordenador, junto al que equipaba de origen.

21 Salga de aquí ahora mismo si no quiere liarse. En realidad, la asignación de letras a unidades por parte de MS-DOS es sólo parcialmente automática y bastante más compleja que lo que se acaba de decir. Algunas reservas, sólo para abrir boca, son las siguientes:

1.- Si no se le dice otra cosa, MS-DOS suele estar preparado para admitir unidades hasta la “E:”. Si se quieren más hay que entrar a configurarlo, fácilmente eso sí.

2.- Las letras se refieren en realidad a unidades lógicas, no a dispositivos físicos reales. Es posible por ejemplo hacer que un único disco lo suficientemente grande aparezca para el sistema operativo como dos unidades diferentes. Podríamos tener así una unidad “C:” y una “D:” que no serían sino dos trozos del mismo disco duro. O una serie de unidades “C:”, “D:” y “E:”, donde la primera sería un disco duro completo y las “D:” y “E:” dos trozos de un segundo disco. De hecho, en MS-DOS fue preciso recurrir a este tipo de operaciones durante cierto tiempo, pues no dejaba manejar directamente discos de más de 32 Megabytes, por lo que un disco mayor debía ser necesariamente troceado si se quería usar en su totalidad. El proceso se conoce como “establecer las particiones del disco duro”, y felizmente ya no es tan necesario como antes, pues MS-DOS ha superado hace tiempo dicha limitación.

3.-Es posible asignar directamente a determinados periféricos de almacenamiento la letra que los denomine. Un CD-ROM, por ejemplo, lo admite, y podemos tener un ordenador con

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Respecto a los directorios, basta con decir tres cosas. A saber, que emplean el símbolo “\” como separador, que sus nombres siguen asimismo la convención 8.3 antes referida aunque al no haber necesidad de especificar el tipo de datos que contienen²² no suele usarse la extensión, y que son fundamentales si se quiere usar un ordenador en MS-DOS. O sea que mírelos con buenos ojos. Vamos a hacerles un poquito de propaganda. Puede que a primera vista le intimiden un poco, pero recuerde que están ahí para facilitarle la vida, pues sin ellos sería imposible orientarse en una unidad con miles de ficheros. Al permitir su agrupación, en buena medida según sus deseos, hacen sencillas operaciones de mantenimiento que son básicas para el uso productivo del ordenador. Haga un esfuerzillo por entenderlos y familiarizarse con ellos. Serán unos de sus mejores amigos, no lo dude.

Uno de sus primeros efectos beneficiosos es que si a Vd. le parecen poco explícitos los nombres 8.3 y desea tener más capacidad expresiva a la hora de nombrar ficheros, tenga en cuenta que en realidad su nombre completo incluye la unidad y el directorio en que se encuentre, hasta un límite de 67 caracteres útiles en total. Retomando y convirtiendo uno de los ejemplos vistos en capítulos anteriores, nuestro antiguo “hdisk1_Contabilidad_Año1993_Febrero_Cuentas” pasaría a ser en un PeCé con MS-DOS algo así como “C:\CONTABIL\1993\FEBRERO\CUENTAS.WKS”, que resulta bastante expresivo. Pondremos unas gotas de suspense al tema y dejaremos en paz por ahora la extensión que le hemos puesto. Más adelante tendrá sentido.

El único inconveniente que tiene lo de los directorios es encontrar el maldito “\”, tan necesario para manejarlos. En un PeCé con teclado en castellano, se encuentra en la cara delantera de la tecla superior izquierda del bloque alfanumérico, la que tiene los símbolos “o” y “a”, por lo que para obtenerlo hay que pulsar “AltGr” y la antedicha tecla. Un poco complicado, lo reconozco, y excesivamente gimnástico. En los teclados en inglés USA tiene una tecla para él solo, con lo que a ellos no les resulta tan difícil. Es la que en los teclados españoles luce los símbolos “<” y “>”²³.

unidades “A:”, “C:” y “R:”, donde esta última designaría el CD-ROM.

4.- Algunos periféricos de almacenamiento no se incorporan al sistema de ficheros FAT y no reciben, en consecuencia, denominación alguna, con lo que no son accesibles sino desde programas especiales. Las unidades de cinta suelen ser de este tipo.

5.- Determinados procesos muy frecuentes en la actualidad, como la compresión de discos duros que veremos posteriormente, alteran la asignación de unidades.

22 Los directorios no contienen datos. Contienen otros directorios o ficheros, y son estos últimos los que contienen los datos, por lo que es con ellos con los que es recomendable usar la extensión. Nada impide de todos modos, y MS-DOS no se va a quejar lo más mínimo, que se usen con los directorios. Tal vez Vd. les encuentre alguna utilidad

23 ¿Porqué le digo esto?. Pues para que vea que no mentía con lo de que el teclado no es más que un conjunto de botones que cada programa maneja a su antojo. Es igual de fácil hacer que cuando se pulse la tecla “A” en la pantalla aparezca la tal “A”, que hacer que aparezca una “S”. El teclado español del que tan orgulloso está, lo es sólo porque un programa del sistema operativo hace que funcione como tal. El mismo programa puede hacer que trabaje como si fuera de USA. Y esto es lo que hace si aparecen algunos problemillas que luego veremos. No está de más tener localizados entonces donde se encuentran escondidos los caracteres necesarios para especificar los nombres completos de los ficheros. Aquí van. Como se ha dicho, la barra invertida “\” está en la tecla con “<” y “>”, los dos puntos, en la “ñ” al pulsarla junto al alzamiento de mayúsculas, el guión y el subrayado sustituyen en la misma tecla al apóstrofe y el cierre de interrogación, y

Un poco de historia de MS-DOS

Terminemos por ahora con el sistema de ficheros realizando un pequeño ejercicio que tal vez nos permita dejar las cosas un pelín más claras. La “C” para MS-DOS no es más que una letra. “C:” en cambio indica la primera unidad de disco duro del ordenador. “C:\” a su vez es el directorio principal de dicha unidad, el primero de todos, a partir del cual se montan los demás, y que universalmente recibe el nombre de “directorio raíz”. “C\” es un error de escritura sin sentido. “C:\MISDATOS\” es un directorio con ese nombre, en tanto “C:\MISDATOS” es un fichero situado en el directorio raíz con el mismo nombre que el directorio anterior, situación perfectamente posible y nada ilegal. Si añadimos un fichero de verdad al directorio anterior, podríamos tener un “C:\MISDATOS\IRPF93.WKS” por ejemplo. A su vez “A:\MISDATOS\IRPF93.WKS” sería una copia de lo anterior, con directorios y todo, en un disquete.

Una buena parte del empleo en la práctica de un ordenador consiste en moverse por su sistema de ficheros. Entrar en un directorio, salir de él, buscar un fichero en una unidad determinada, inspeccionar un directorio. Para hacerlo es preciso que los nombres de ficheros, directorios, y unidades, nos sean absolutamente familiares y que les tengamos el aprecio que merecen. Que los queramos de verdad y no que los soportemos como un dolor de muelas irremediable. Aplíquese el cuento si de verdad quiere trabajar codo a codo con cualquier tipo de Luis Ricardo.

Y terminemos por fin las sucesivas tuercas con una nota al pie, destinada a aquellos con un ansia irrefrenable de complicarse la vida²⁴.

Un poco de historia de MS-DOS

De como LR-PeCé se dotó de un DOS

Tal vez Vd., querido LAO, haya oído en alguna ocasión quejas sobre las limitaciones y peculiaridades del sistema operativo del PeCé. Suelen consistir en letanías más o menos amargas que muestran su pasmo ante una serie de cosas aparentemente injustificables. La gente se pregunta, y no se contesta, cosas tales como: ¿porqué hay tantos tipos de memoria MS-DOS diferentes?, ¿cómo es posible que teniendo muchos Megabytes de RAM y diciéndome el ordenador que algunos de ellos están libres, no pueda ejecutar según qué programa porque al parecer la memoria disponible no llega a, digamos, 590 Kilobytes?, ¿porqué MS-DOS no es multitarea, o multiusuario?, ¿porqué no se les pueden poner nombres más

la barra normal “/”, también de interés, ocupa por su parte el lugar del guión.

- 24 La norma 8.3 es más compleja que la muy simplificada versión que hemos expuesto en el texto principal y que es la que, en la práctica, suele usarse de forma rutinaria. Pero casos hay en los que no está de más conocerla exactamente. Hela aquí, ligeramente refundida a partir de una descripción bastante adecuada encontrada recientemente (“Windows 95”, Adrian King, pg. 245, ISBN 84-481-1700-X, Editorial McGraw-Hill, 1.995):
- El nombre puede contener hasta 8 caracteres con una extensión opcional de 3 caracteres.
 - Los caracteres válidos pueden ser letras, dígitos, espacios, cualquier carácter con un valor mayor que ASCII 127 y cualquiera de los siguientes: \$, %, comillas simples de apertura y cierre, apóstrofo, guión, subrayado, arroba (@), tilde (~), acento grave, !, (,), {,}, #, &.
 - El nombre completo, incluidos unidad y directorios, puede tener hasta 67 caracteres útiles.
 - No se distinguen mayúsculas y minúsculas. Se almacenan como mayúsculas.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

largos a los ficheros?, ¿porqué las unidades de almacenamiento se denominan mediante letras únicas?. Y así podríamos continuar largas horas.

La situación no es en absoluto distinta a la ya vista para el hardware del PeCé, asimismo proclive a la formulación de cuestiones de similar cariz. En aquel caso, la Historia aportaba algo de luz sobre un presente difícil de comprender de otro modo. MS-DOS parece también mucho más coherente, y por ello se entiende mejor, si se conoce su árbol genealógico. La Historia suele ser bastante agradecida. Volvamos a ella antes de meternos realmente en harina.

Si así lo desea, échele un vistazo al apartado de los orígenes del PeCé para refrescar su memoria sobre la situación en la microinformática allá hacia finales del 70 y principios del 80. Los micros de 8 bits reinaban, y con ellos un sistema operativo, estándar por excelencia para el i8080 y compatibles como el Z80. Llamado CP/M, desarrollado inicialmente por Gary Kildall, y distribuido por la compañía que luego fundó, Digital Research, no era cómodo para el usuario, no era muy bonito, pero funcionaba lo suficiente y era casi universal. En consecuencia, poseía una razonable biblioteca de programas, era asimismo razonablemente sólido, conocido, y útil. No era ni multitarea ni multiusuario, no porque entonces no existieran esas cosas, sino porque se diseñó para CPUs muy poco potentes y ordenadores con poca memoria y poca capacidad de almacenamiento, hechos para ser usados por una sola persona y que debían ser cuanto más baratos mejor. En resumen, CP/M era un sistema operativo rústico, ante todo sencillo, práctico y barato, sobre el que, simplemente, se podía ir construyendo una biblioteca de programas lo más grande posible para dotar de utilidad a la mayoría de micros de 8 bits con pretensiones de seriedad.

La aparición de los microprocesadores de 16 bits significó, como ya se expuso, que el mercado microinformático volvía a ser terreno de nadie. Y también significó que había que dotar a los nuevos ordenadores con un sistema operativo a la altura de los tiempos, ya que no sólo las nuevas CPUs eran incompatibles con las viejas y no podían utilizar sus programas, sino que ofrecían una mayor potencia que convenía aprovechar. ¿Porqué, pregunta?.

Los micros de 8 bits no andaban sobrados de músculo. Cualquier mejora era esperada como agua de mayo, sobre todo si se trataba de duplicar (al menos) la potencia. Por poner sólo un ejemplo, la cantidad de memoria con que se podía trabajar en ellos sin trucos adicionales era de 64 Kilobytes, lo que desde luego no daba para mucho. Y CP/M, en justa consecuencia, se limitaba a administrar dicha cantidad de memoria. Las CPUs i8086 (o i8088), ya de 16 bits, permitían manipular directamente todo un Megabyte completo. Hacía falta un sistema operativo para los ordenadores basados en ellas que diera acceso a ese nuevo espacio, y que permitiera a los programas que se ejecutaran sobre él beneficiarse asimismo de tan inconcebibles holguras. Había que sentar las bases para el desarrollo de programas mucho más potentes, que hicieran a los ordenadores de 16 bits más útiles y cómodos que los de 8 bits, y en consecuencia fueran capaces de desplazarlos del mercado aún vendiéndose más caros.

Convenía hacerlo lo antes posible²⁵ y seguía sin tener mucho sentido meterse en berenjenales de multitareas o multiusuarios principalmente porque aún faltaba potencia. Además, para eso ya estaban los ordenadores más gordos de las grandes marcas, que no era cuestión de

25 Cuando escribo este tipo de frases no puedo evitar pensar que también va siendo hora de que

Un poco de historia de MS-DOS

dejar de vender. Por lo tanto, lo más propio era adaptar CP/M. Nació así CP/M-86, es decir CP/M para el i8086 y compatibles, y la antigua versión de 8 bits pasó a llamarse CP/M-80, es decir CP/M para el i8080 y compatibles. Al mantener la misma estructura general y aspecto que su progenitor era inmediatamente utilizable por todo usuario anterior de CP/M, y bastaba con una modesta adaptación de los programas desarrollados inicialmente para las CPUs de 8 bits para que pudieran trabajar en los nuevos equipos. Pero no fue CP/M-86, la adaptación oficial de CP/M realizada por la casa madre, Digital Research, el sistema operativo estándar de los PeCés.

Hacia 1.979, más o menos a la vez que se desarrollaba CP/M-86, un programador particular llamado Tim Patterson procedió también a hacer más o menos lo mismo, tal vez con mayor prisa por ser para uso propio, obteniendo un resultado similar al que llamó 86-DOS. Como sabía que Digital Research estaba en ello por su cuenta, y casualmente su más cualificada competencia acababa a la sazón de inaugurar su nueva sede en las proximidades de su residencia (por si es precisa la aclaración, Microsoft, de un tal Bill Gates, en Redmond, Washington, EEUU), se dio una vuelta por allí para ofrecerles su invento. Y la propuesta no despertó inicialmente mayor interés entre las huestes de Bill Gates.

Pero un año más tarde, IBM eligió a Microsoft para realizar el software básico de su PC, al parecer tras no llegar a un acuerdo satisfactorio con Digital Research. En lugar de ponerse a trabajar desde cero, Microsoft decidió sabiamente comenzar adquiriendo el producto de Tim Patterson, para continuar su desarrollo hasta llegar a algo lo suficientemente completo como para ser tan bueno o mejor que el original CP/M-86. En 1.981 salió el PeCé, equipado con MS-DOS 1.0. Aunque deberíamos decir que montaba PC-DOS 1.0, ya que éste era el nombre que adoptaba en las máquinas de IBM. Para el resto de la plebe, los fabricantes de clónicos, el nombre era MS-DOS.

Está claro que con tales orígenes, y especialmente si se quería facilitar la adaptación del software ya existente para CP/M-80 (una colección ya más que respetable en aquellos entonces), las diferencias entre CP/M-86 y MS-DOS 1.0 debían ser más bien escasas. La estructura básica era la misma. Un usuario de CP/M seguía estando en casa en MS-DOS. Y para darle un toque de distinción, algunos puntos particularmente horribles de CP/M se pulieron. Por ejemplo, CP/M tenía una notable y justificada mala fama de sistema arisco e incómodo de uso. Si había un error, uno se encontraba ante cosas parecidas a “SYS ERROR #16. Terminated.”.

Uno miraba los manuales, tras revolver papeles durante cosa de dos horas y sacarlos de entre los periódicos atrasados de hace dos años donde no se sabía como habían ido a parar, y acababa enterándose, tras vagar más bien sin rumbo por sus páginas durante un período de tiempo adicional, que tal cosa no quería decir sino que no se había colocado el disquete adecuado en el momento de encender el ordenador. MS-DOS, enfrentado a idéntica situación, presentaba un mucho más educado, aunque eso sí también en inglés, “Disco de arranque no encontrado. Inserte el disco correcto y pulse una tecla para continuar”. Desde luego, era un cambio.

termine de escribir este texto, que asimismo convenía que hubiera estado disponible cuanto antes.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Ya dijimos que, en la práctica, CP/M-86 y MS-DOS resultaron prácticamente intercambiables, y muchos ordenadores admitían cualquiera de ellos. Las ventajas iniciales de MS-DOS, y especialmente el “boom” arrollador del PeCé, hicieron que éste último acabara siendo aún más estándar que lo que CP/M-80 había llegado a serlo nunca. Con el tiempo, CP/M desapareció en cualquiera de sus versiones, aunque todavía duró unos años²⁶.

¿Qué más debemos saber sobre MS-DOS 1.0?. Por supuesto que, como ocurre con cualquier programa con éxito, no fue sino una primera versión a la que en rápida sucesión siguieron otras cada vez más capaces en diversos aspectos. Por ejemplo, MS-DOS 1.0 sólo trabajaba con disquetes de 160 KB y 5'25", pero la 1.25 ya admitía disquetes de 360 KB y el mismo tamaño. También que más o menos el mismo producto se denominaba PC-DOS cuando era IBM la que lo vendía con sus PC. Que Microsoft no podía vender directamente al usuario, sino que debía limitarse a ser un proveedor a terceros. Un fabricante de clónicos compraba MS-DOS a Microsoft, lo adaptaba más o menos a su equipo si era necesario, y lo vendía por su parte al usuario final. No resulta extraño, por lo tanto, que incluso la misma versión base de MS-DOS resultara ligeramente distinta cuando se compraba a Olivetti o a Compaq, por poner un ejemplo²⁷. Que un montón de cosas se tomaron directamente de CP/M tal cual. Por ejemplo la norma 8.3 para los nombres de ficheros. Y, finalmente, que se hizo una distribución de la memoria disponible que sigue fundamentalmente vigente en nuestros días y que es obligatorio conocer. Del Megabyte que podía manejar el i8086, se reservaron los primeros 640 Kilobytes para los programas que fueran a ejecutarse, y el resto se dejó para memoria de video, ROM, y demás. Ningún programa podía usar más de 640 KB de vez. Lo que parecía una barbaridad, ya que los programas disponibles, normalmente adaptaciones más o menos directas de sus versiones CP/M, estaban diseñados para trabajar con 64 Kilobytes. Además de que los PeCés de entonces no solían montar más allá de 128 o 256 Kilobytes de RAM. Fue una decisión lógica, desde luego, y escasamente censurable, pero que con el tiempo ha provocado no pocos inconvenientes.

Resumiendo, no es arriesgado decir que el software base del PeCé, su sistema operativo, empezó heredando la mayor parte de las características del sistema operativo hasta entonces estándar para los equipos de 8 bits, si bien camuflado bajo otro nombre. Lo que constituye toda una declaración de principios. Se quería un sistema informático que primara ante todo la conservación del software existente. La palabra clave era “compatibilidad”, y a todos los niveles. Que nadie tuviera que tirar nada, ni de software ni de hardware, si podía evitarse. Tal enfoque permitió un primer beneficio inmediato: los programadores y usuarios ya formados sobre CP/M no tuvieron dificultad en pasarse al PeCé, los programas importantes se adaptaron rápidamente, y en cuatro días el PeCé comenzó a tener una biblioteca de programas más que respetable y en continua expansión.

26 Recuerde que ya con el PeCé impuesto como estándar, Amstrad seguía vendiendo con enorme éxito en España y en Europa sus gamas CPC y PCW, que trabajaban sobre CP/M-80.

27 Debe destacarse que las diferencias entre adaptaciones particulares de una misma versión de MS-DOS no han causado problemas importantes de incompatibilidad. No era tan extraño poder montar el DOS distribuido con sus equipos por una marca determinada en los de otra distinta. Las diferencias solían consistir en la presencia de más o menos programas de utilidad o en que estuvieran ligeramente mejor terminados en un caso que en otro.

Un poco de historia de MS-DOS

Retomemos la evolución de MS-DOS y hagamos un rápido resumen de sus sucesivas versiones y lo que fueron añadiendo a lo que inicialmente era poco más que un esqueleto de lo que hoy en día es²⁸. El PeCé XT aparece en Marzo de 1.983, dotado por fin de un disco duro, y MS-DOS celebra tan feliz natalicio pasando a la versión 2.0, dotada ya de un sistema de ficheros jerárquico. Al parecer, la insistencia de IBM lleva a que se adopte la barra invertida (\) como elemento separador de directorios en los nombres completos de los ficheros. En 1.984 se pasa a la versión 2.1, en la que se dedica una mayor atención a la internacionalización, acogiendo distintos tipos de indicadores monetarios, fechas, y separadores de decimales (ver nota)²⁹. MS-DOS 2.11 se traduce a 60 idiomas. En agosto, IBM lanza el PC-AT, y en noviembre aparece MS-DOS 3.1, capaz de trabajar en red si se le añaden los módulos necesarios. Esta versión se convertirá en la referencia durante largo tiempo. En 1.986 aparece MS-DOS 3.2, que incorpora el soporte para disquetes de 3'5" con 720 Kilobytes de capacidad, cada vez más usados en los ordenadores portátiles.

El 2 de abril de 1.987 se anuncian los PS/2 y aparece MS-DOS 3.3 para poder trabajar con disquetes de 3'5" y 1'44 Megabytes. Permite seleccionar distintos conjuntos (mapas) de caracteres y hasta 4 discos duros siempre que ninguno de ellos supere los 32 Megabytes, lo que era una antigua limitación impuesta a su tamaño. Para entonces muchas adaptaciones de distintos fabricantes (recuérdese que MS-DOS se vendía a través de terceros) intentaban por su cuenta solventar algunos enojosos problemas de MS-DOS. Tales como la escasa capacidad de memoria y el límite en el tamaño de los discos duros. Lo de la memoria consistía en que ya para entonces no era extraño que muchos programas desearan trabajar con un par o más de Megabytes de RAM, lo que era posible ya que hacía tiempo que los PeCés montaban CPUs de segunda generación (el i80286) que permitían trabajar con bastante más que el único Megabyte del i8086. En concreto llegaban a los 16 Megabytes de RAM. Pero MS-DOS seguía en sus trece con los 640 KB para el usuario y el Megabyte de memoria total, por lo que la CPU y la RAM adicional que se le pusieran al ordenador no valían para nada.

La cosa se resolvió con un truco más bien inmundo. Una zona de 64 Kilobytes de las reservadas para MS-DOS quedaba digamos libre, y era posible hacer desfilar por ella, de 64 KB en 64 KB, los Megabytes necesarios de memoria. Se desarrolló una norma para poner de acuerdo a la gente y apareció así la "memoria expandida", o EMS ("Expanded Memory Specification"), también conocida como "norma LIM", abreviatura a su vez de "Lotus-Intel-Microsoft", las compañías que la propusieron. La cosa era lenta y poco elegante pero

28 Bastantes de estos datos históricos sobre MS-DOS se han sacado de uno de los muchos resúmenes que han aparecido en las revistas sobre informática. En concreto del "Science et Vie Micro" de Abril de 1.991, pg. 78.

29 La cuestión es que la gente en España suele necesitar poner en el ordenador algo así como "254.000 Pts" cuando quiere decir "Doscientas cincuenta y cuatro mil pesetas", y los PeCés hasta el momento preferían claramente poner "\$ 254,000", que no queda tan bien. Los anglosajones separan los miles con una coma y los decimales con un punto, en tanto en España y otros países se hace justo al contrario. Esta estupidez da lugar a un buen montón de estúpidos errores en el uso de aquellos programas destinados a manejar números, y conviene tenerlo en cuenta. Por desgracia, incluso en la actualidad, sigue siendo imposible poner la coma arriba, como a mí me gusta, para sacar números tan bonitos como "123.000'834". Y no hablemos por supuesto de poner los unos y los doses que deberían indicar millones y billones. También las fechas se dan en distinto orden y con distintos formatos según el país.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

mantenía la compatibilidad y no había nada mejor, ya que el i80286 tampoco es que permitiera demasiadas alegrías. Según IBM y Microsoft, el problema sólo merecía ser resuelto con dignidad dentro de un sistema operativo nuevo y más avanzado, entonces en desarrollo, y que se asociaba a los PS/2. El invento respondía al nombre de OS/2 (“Operating System 2”, es decir “Sistema Operativo 2”), y se suponía que debía ir sustituyendo paulatinamente a MS-DOS.

En julio de 1.988 aparece MS-DOS 4.0. Supera por fin el límite de los 32 MB para los discos duros, que pasan a poder ser de hasta 2 Gigabytes (2.048 Megabytes), e incluye una interfase de usuario semigráfica opcional, lo que nosotros hemos llamado “interfase WMP”, que soporta el uso del ratón. No se acepta muy bien, sin embargo, ya que deja menos memoria libre en la zona de 640 KB, ya para entonces casi totalmente saturada, es más gordo que las versiones 3.3 entonces en uso y, por si no bastara, tiene algunos problemas con los discos duros que habían saltado por su cuenta el fatídico límite de los 32 MB³⁰ con versiones avanzadas de MS-DOS 3. En 1.990 Microsoft se desquita lanzando Windows 3.0, un GUI para PeCés que es finalmente aceptado por todo el mundo de forma fulminante. Para ello, deja de lado OS/2, al que sólo IBM sigue prestando atención en lo sucesivo. Por aquel entonces hace algún tiempo ya que los PS/2 están en vías de extinción.

En 1.992 aparece MS-DOS 5.0. Los PeCés tienen en esas fechas CPUs i80386 en distintas versiones. Las menos potentes (el i80386SX) admiten hasta 16 Megabytes de memoria y están capacitados para someterla a múltiples malabarismos a la hora de direccionarla. Lo que permite descongestionar en alguna medida la zona de 640 KB, ya a rebosar, y hace aparecer otro tipo de memoria de obligado conocimiento: la memoria extendida. No, no es lo mismo “expandida” que “extendida”. Por supuesto, se producen multitud de confusiones entre denominaciones tan similares³¹. Además de esto, se incorporan los parámetros de ayuda a las distintas órdenes que componen MS-DOS, se añaden órdenes para deshacer borrados de ficheros y hasta para recuperar los datos de un disco formateado, se incluye un interfase de línea de órdenes mejorado como el que en su momento describimos, los disquetes de 3’5" pueden llegar hasta los 2’88 Megabytes, hay un conmutador de tareas (pueden lanzarse varios programas y pasar de uno a otro sobre la marcha aunque sólo trabaja aquel en que nos encontramos), y hasta un editor de pantalla completa para modificar cuando haga falta ficheros de tipo texto de tamaño moderado. Ya era hora.

30 Por ejemplo, el intento de instalar MS-DOS 4.0 sobre un ordenador con disco duro de 40 Megabytes equipado con la correspondiente versión “avanzada” de MS-DOS 3.3, llevó en mi caso a la inmediata desaparición de todo su contenido, lo que no me hizo ninguna gracia.

31 Digamos que con el i80386, el PeCé fue finalmente capaz de gestionar, si así se deseaba, toda su memoria como un conjunto homogéneo de bytes numerados desde el uno al 16 millones, por lo menos. El esquema de direccionamiento más sencillo que nosotros siempre hemos usado. Hacía más de diez años que las CPUs de la competencia permitían tal maravilla. De todos modos, la compatibilidad a ultranza defendida en el diseño de los PeCés seguía y sigue complicando enormemente la situación. La memoria extendida sería en principio definible como este tipo de memoria lineal. El i80386 y todos los chips que le han seguido son capaces de llevar a cabo, y de hecho así lo hacen al trabajar en MS-DOS, intrincadísimas manipulaciones de la memoria que, una vez más, ignoraremos.

Un poco de historia de MS-DOS

Hacia junio de 1.994 se lanza la versión 6.0, rápidamente sustituida por la 6.22³², que acaba de redondear la cosa con el soporte de discos comprimidos, un manual en línea, un mejor aprovechamiento de la fatídica zona de los primeros 640 KB, y una colección más completa de utilidades de mantenimiento (compactadores de disco, antivirus, optimizadores de memoria...). Y aquí estamos en este momento, más o menos, pues Windows 95 añadió una nueva versión semiclandestina que suele conocerse como MS-DOS 7.0.

A modo de resumen, digamos que MS-DOS comenzó a ser plenamente aceptable a partir de la versión 5.0, en 1.992. Hasta ese momento, cosas tan elementales como corregir una orden errónea eran cuando menos poco cómodas, siendo muy indulgentes con nuestros calificativos. No es extraño que los PeCés tuvieran fama de incómodos. Su IU era una línea de órdenes mal implementada, como la que narramos al principio del capítulo 5, antes de que nuestro programador le dedicara sus esfuerzos. Pero, repitamos, no es esa la situación actual, y ya no hay porqué temerla como antaño. Todos nuestros maravillosos inventos (parámetros de ayuda, posibilidades amplias de corrección y recuperación de órdenes, manuales en línea y demás caterva) están ya disponibles, y conviene saberlo. Y aprovecharlo. Incluso puede decirse que, ahora mismo, MS-DOS es un sistema operativo más que razonablemente tolerante ante los inevitables errores de uso que más pronto o más tarde todos cometemos. No debe tenerle ningún miedo. MS-DOS es ya un buen amigo potencial, con todo y que siga requiriendo el uso del teclado para mandarle cosas.

¿Qué pasó con Digital Research mientras tanto, se pregunta?. Aparte de agotar el filón de CPM/80 mientras duró, intento al principio seguir vendiendo su CPM/86, en distintas versiones también cada vez más avanzadas. Pero a partir de determinado momento, quedó claro que lo único que la gente estaba dispuesta a comprar era MS-DOS, o algo que, aun llamándose de otra forma, fuera lo mismo. Es decir, un clónico de MS-DOS, a ser posible mejor que el original. Digital Research sacó tal producto, lo llamó DR-DOS (abreviatura de Digital Research Disk Operating System) y, como buen segundón, proporcionó a mejor precio un mejor producto. Cuando no se podían usar discos duros de más de 32 Megabytes con MS-DOS, DR-DOS ya había superado dicho límite. Montó una línea de órdenes mejorada cuando todavía no se incluía en MS-DOS. DR-DOS 5.0, contemporáneo de MS-DOS 4.0, ya tenía parámetros de ayuda para las distintas órdenes. DR-DOS 6.0 contaba con manual en línea y compresor de disco cuando MS-DOS 5.0 incorporó finalmente la línea de órdenes mejorada y los parámetros de ayuda. Novell DOS 7.0, la versión siguiente, es un auténtico multitarea, cosa que no puede decirse de MS-DOS 6.22. DR-DOS ha venido siendo el sistema operativo de aquellos usuarios avanzados de PeCés que querían ese “algo más” que MS-DOS reservaba para la siguiente versión³³.

32 La razón fueron los problemas aparecidos con el sistema de compresión de discos. Microsoft cobró la actualización a los desdichados que adquirieron la versión 6.0, aunque el precio fue muy reducido. Esto provocó las iras de bastantes usuarios.

33 DR-DOS incluyó, ya en la versión 6.0, extras tales como un programa de intercambio de ficheros entre ordenadores a través de la puerta serie o paralelo, o la posibilidad de proteger mediante claves el acceso no autorizado a los ficheros del ordenador. Novell DOS 7 llega hasta a incorporar todo lo necesario para gestionar un sistema de red de tamaño modesto. Hay que precisar que MS-DOS 6.22 es también bastante completa, y que Microsoft siempre a incluido un Basic (un lenguaje de programación, que ya veremos qué es).

CAPÍTULO 7. PeCés: MS-DOS/Windows.

El cambio de nombre se debe a que Digital Research ya no existe como tal. Novell, otra editora de software, la compró allá por 1.993. En el capítulo siguiente daremos una pequeña lista de las empresas del ramo y volveremos sobre el tema. Hasta entonces. Gary Kildall, el fundador de Digital Research e inventor de CP/M, falleció hace no mucho. Vaya para él un merecido recuerdo, como corresponde al que de alguna forma dio sentido a los micros como ordenadores serios.

No ha sido DR-DOS el único clónico de MS-DOS. También IBM ha vendido por su cuenta su versión particular, el PC-DOS, normalmente más avanzado que aquel. Una de las ventajas de DR-DOS y PC-DOS es que son auténticos sistemas operativos completos tal cual salen de sus correspondientes cajas³⁴. Microsoft no puede vender directamente al usuario final un auténtico DOS para PeCé. Debe, por sus acuerdos con IBM, suministrarlo tan sólo a fabricantes de hardware, y dejar que sean estos los que traten con el cliente. A partir de MS-DOS 5.0, Microsoft soslayó astutamente el problema vendiendo “actualizaciones de MS-DOS”, que sólo sirven si se instalan sobre una copia de MS-DOS anterior ya funcional y adquirida a un tercero. Por contra, tanto IBM como Novell no hacen sino vender clónicos, por excelsos que sean, de lo que es el auténtico producto de referencia, MS-DOS³⁵.

Y tras tanto rollo, podríamos preguntarnos: ¿qué hace, en realidad, MS-DOS o cualquiera de sus clónicos?. La exposición anterior nos ha permitido ir dando una lista. Como hemos podido ver, poco más que permitir el uso en un PeCé de distintos tipos de disquetes, discos duros, y periféricos en general (teclados, CD-ROMs, pantallas de texto y gráficas, etc.), gestionar la memoria de diversas maneras, y aunque no lo hemos visto vamos adelantándolo, especificar los mecanismos de ejecución de programas, proporcionar utilidades básicas de gestión y mantenimiento de ficheros, y casi nada más. No es difícil aprender tan poca cosa. Vamos a ello.

Empezando por el principio

LR-PeCé, ¡en marcha!

Bien, pongámonos en marcha con MS-DOS y, por favor, lea la nota³⁶. Hace ya mucho rato, en el capítulo 4 para ser exactos, nos ocupamos a nivel general de lo que era un “boot” y

34 Otras ventajas han sido menos evidentes pero no menos reales. Baste un ejemplo. Cuando apareció DR-DOS 6.0, Windows andaba por la versión 3.0. Microsoft lanzó casi inmediatamente después Windows 3.1, mucho mejor que la 3.0. Casualmente, resultó no ser compatible con DR-DOS 6.0, entonces en fuerte competencia con MS-DOS 5.0. También inmediatamente, Digital Research modificó DR-DOS 6.0 para que pudiera ejecutar Windows 3.1 y proporcionó dicha actualización a sus clientes sin coste alguno. Una cierta diferencia si se compara con lo acontecido con los problemas del compresor de disco de MS-DOS 6.0. Compresores de disco que llevaban ya bastante tiempo funcionando en DR-DOS sin ningún problema y que algunos fabricantes, por ejemplo Hewlett-Packard, incluían en sus versiones de MS-DOS 5.0.

35 El asunto de los legalismos en la distribución de MS-DOS y Windows no es baladí, pero no aparece con excesiva frecuencia en las publicaciones. Aunque en su momento leí algún artículo bastante detallado, lamento no poder dar su referencia bibliográfica concreta o precisar más cosas tales como hasta cuando Microsoft tiene prohibido vender MS-DOS al usuario final, o en que momento dejó o dejará IBM de tener derechos sobre Windows.

36 A partir de aquí, iremos incluyendo ejemplos reales. Debo decir que soy usuario habitual de DR-DOS, como habrá intuido al leer alguno de los comentarios anteriores y puede constatar

Empezando por el principio

un “reset”. Arrancar y reinicializar un ordenador son operaciones inevitables y altamente provechosas a nivel educativo si se analizan con un mínimo detalle. Vamos a ver qué pasa con ellos en un PeCé, y como hay que empezar por el principio y lo primero es enchufar el ordenador, pues empezamos por el “boot”. Que además es prácticamente idéntico al “reset”, por lo que mataremos dos pájaros de un tiro con suma comodidad.

Para poder hacer algo, lo primero es darle al interruptor general del PeCé. Tan simple acción desencadena un torrente de acontecimientos. El “boot” ha comenzado y el ordenador se pone inmediatamente a hacer cosas. Si el monitor está también en marcha, situación muy conveniente, esto se traduce en la aparición de distintos mensajes en la pantalla. Significa que se está ejecutando algún programa. ¿Cual y de donde ha salido?.

No debería a estas alturas ser una sorpresa el decir que el programa que tan alegremente a comenzado a poner en práctica LR-PeCé se halla en la ROM. La poca que tiene un PeCé contiene lo que se denomina el BIOS (“Basic Input -Output System”, sin mayor significado). Se le conoce también en la actualidad como “firmware”, traducible tal vez como “software fijo” o, si lo prefiere, “software residente”. Se encarga, como tampoco debería sorprender, de identificar los distintos componentes que puedan estar instalados y comprobar su estado de funcionamiento. ¿Hay RAM?. Sí. ¿Cuanta?. 8 Megabytes. ¿Están todos bien?. Vamos a verlo. Y así con teclado, pantalla, disquetes, discos duros, y algún que otro chisme más. Los mensajes que de inmediato van apareciendo en la pantalla tienen como objeto mantenernos informados del progreso del programa y hacernos saber, Dios no lo quiera, si aparece algún problema³⁷. LR-PeCé es bastante locuaz y conviene permanecer atentos a lo que nos vaya diciendo. Una vez está seguro de que todo está bien y, en consecuencia, de que al otro lado hay un PeCé en condiciones de trabajar, pasa el control a otro programita. De nuevo podemos preguntar ¿cual y dónde se encuentra?.

Tras haberse realizado la comprobación del sistema, pueden entrar en funcionamiento los discos de distintos tipos con que cuente nuestro equipo. MS-DOS es un sistema operativo basado en discos, y la mayor parte de él estará allí. El programa de que hablamos, en justa consecuencia, se encontrará en uno de ellos. Para ver un poco la lógica del invento, hay que hacer notar que en este momento LR-PeCé sabe que tiene unos discos determinados pero

leyendo la lista de hardware y software empleado para redactar este texto. Dichos ejemplos corresponderán por lo tanto, estrictamente hablando, a dicho clónico, y no son absolutamente idénticos a sus equivalentes MS-DOS. En bastantes ocasiones especificaremos dichas diferencias, pero no siempre. En cualquier caso, carecen de importancia. Se trata ante todo, recuerde, de entender el funcionamiento general del sistema operativo del PeCé, cualquiera que sea la versión.

37 ¿Le encantan a Vd. las abreviaturas informáticas?. ¿Aún no se ha hartado?. Aparte de avisarle de que ese momento llegará inevitablemente, especialmente a partir de que se dé cuenta de que muchas veces no quieren decir nada y son simplemente una forma más de hacer el fantasma, y que también en muchas ocasiones parece que primero se ha elegido la abreviatura para que resulte sonora y llamativa y después se ha buscado como encajar algo que tenga un mínimo de sentido en tan aparente acrónimo, procedamos una vez más a saciar en lo posible sus bajos instintos. Al programa que en el momento de arrancar el ordenador se encarga de comprobar el funcionamiento de los distintos componentes se le conoce también en general como “POST”, abreviatura de “Power On Self Test”, traducido “Auto comprobación en el encendido”. Pues nada, nada, a disfrutarlo mientras dure.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

desconoce como están organizados. Es, eso sí, capaz de colocar las correspondientes cabezas de lectura al principio del disco que sea. Es el único sitio al que puede acceder y donde debe estar lo que busca: el MBR (“Master Boot Record”, más o menos, traducible como “Sector de Arranque”). Normalmente, LR-PeCé comienza mirando en el primer lector de disquetes, la unidad “A:”. Si la unidad “A:” está vacía, mira a continuación en el primer disco duro, la unidad “C:” y, en un PeCé moderno, es donde suele encontrarlo. Si por casualidad hemos arrancado con un disquete en “A:” pueden ocurrir dos cosas. A saber, que el disquete tenga un MBR o que no lo tenga. Si no lo tiene, el proceso no puede seguir, y el ordenador se para y muestra un mensaje más o menos similar a: “No se encuentra el disco de arranque. Pulse una tecla para continuar”. LR-PeCé nos da una oportunidad para corregir el error. En este momento, podemos hacer otras dos cosas. Bien insertamos otro disquete que sí tenga el dichoso MBR, o bien quitamos el disquete que no lo tiene y dejamos que, tras constatar que “A:” está vacía, pase a mirar en “C:”.

Localizado el MBR en una u otra unidad, LR-PeCé sabe que el resto del sistema operativo se encontrará en el mismo sitio. Vamos a suponer que estamos arrancando desde el disco “C:”³⁸. El MBR no es sino una mínima serie de instrucciones, pero podríamos decir que se encarga de activar el sistema de ficheros. Algo absolutamente fundamental, como sabemos. Tras haberlo digerido, LR-PeCé es ya capaz de localizar ficheros de verdad. Y eso es lo que busca a continuación. ¿Cuales y en qué directorio estarán, pregunta ahora?. Es usted insaciable, querido LAO. Apliquemos una vez más la lógica y preguntémosnos: ¿cual es el único directorio que podemos estar seguros de que existe en cualquier circunstancia?. La respuesta es obvia: el directorio raíz. Fuera de él, nada está garantizado. Los ficheros que busca ahora LR-PeCé estarán pues en “C:\”. Sus nombres tradicionales son “IBMBIO.COM” e “IBMDOS.COM”, y así se siguen llamando si por ventura trabajamos en DR-DOS o PC-DOS. Si estamos siendo fieles a Microsoft y su MS-DOS, se llaman “IO.SYS” y “MSDOS.SYS”

En uno u otro caso, tras digerir su contenido (cosa que hace en cuanto los localiza), LR-PeCé es ya capaz de controlar con propiedad casi todos sus componentes principales, pues podemos decir que contienen los drivers necesarios para ello (los programas piloto ¿recuerda?). Valen su peso en oro si queremos que cosas tales como teclados, pantallas, y unidades de almacenamiento sigan en marcha, por lo que LR-PeCé los deja amorosamente cargados en su RAM. Y está en disposición de pasar a dejar que le mandemos cosas.

Pero, desgraciadamente, aún no conoce un lenguaje mediante el cual comunicarse con el género humano. Para aprenderlo, debe buscar y cargar en RAM un fichero adicional, llamado siempre “C:\COMMAND.COM”. Tras hacerlo, ya tenemos una serie de palabrejas con un montón de parámetros a nuestra disposición. Básicamente, el arranque acaba aquí. Alegre, contento y satisfecho, LR-PeCé nos informa de que está listo para trabajar sacando en pantalla su famoso “prompt”, la indicación de que podemos darle órdenes, con su correspondiente cursor parpadeante que tan ansiosamente esperábamos. También él ha sobrevivido desde los tiempos de CP/M y normalmente es algo así como:

```
C:\>_
```

38 En caso de que estuviéramos arrancando desde el “A:”, la única diferencia sería que todo lo que sigue tendría lugar en dicha unidad, nada más.

Empezando por el principio

No muy acogedor, pero suficiente. Y ya está. Tras haber arrancado el ordenador, todo un nuevo mundo se abre ante nosotros. ¿Que qué le mandamos a LR-PeCé que haga?. Usted sabrá. ¿Que no sabe qué hacer ahora, dice?. ¿Que está un poco desorientado, dice?. Acaba Vd. de topar con el síndrome de la hoja en blanco. Para superarlo, hay que saber algo más y enseguida lo explicaremos, tras recapitular un poquito.

Y empecemos a recapitular llamando la atención sobre el hecho de que las cosas son como son simplemente porque LR-PeCé tiene poquísima ROM. Casi todo hay que cargarlo desde disco, y eso obliga a ir paso a paso. No se puede cargar un fichero si antes no disponemos de un sistema de ficheros activo, no se puede buscar éste si antes no sabemos con qué discos cuenta el ordenador, y así sucesivamente. Eso justifica las distintas etapas.

Y continuemos recapitulando a base de hacer notar que, para que sea posible el “boot”, hace falta que pongamos al alcance de LR-PeCé un disco duro o disquete bastante especial. Uno en el que sea capaz de encontrar el MBR y los ficheros IBMBIO.COM, IBMDOS.COM y COMMAND.COM (o equivalentes). Un disco así, capaz de poner el PeCé en marcha proporcionándole todo lo necesario, se denomina en justa consecuencia “disco de arranque”. Como vimos, la mera estructura de un disquete tipo FAT se lleva una parte no despreciable de su capacidad. Añadir además todo lo anterior hace que perdamos aún más espacio. Por ejemplo, en Novell DOS 7, IBMBIO.COM tiene un tamaño de unos 28 KB, IBMDOS.COM de más o menos 29 KB, y COMMAND.COM por su parte necesita casi 54 KB más. Nuestros disquetes de 2 MB en bruto se quedan en 1'34. Además, en realidad el arranque no ha terminado (no me diga que no lo sospechaba) y es necesario añadir un buen montón de cosas todavía, que no está de más poner en el mismo sitio. O sea que nos podemos olvidar de que nos quede algo de sitio para nosotros en el tal disquete. Por lo tanto no tiene sentido que todos los disquetes sean de arranque, y esto obliga a distinguir dos tipos: los de arranque (también llamados “de sistema”) y los de datos. Estos últimos contendrán simplemente la estructura de almacenamiento y nos dejarán los famosos 1'44 MB a nuestra total disposición para lo que queramos. Pero no serán capaces de poner en marcha el ordenador.

Ante este panorama no es raro que se prefiera arrancar desde el disco duro³⁹. Como tiene mucha mayor capacidad, no tendrá problema en albergar todo lo anterior y bastantes cosas más, dejándonos todavía sitio para nosotros y nuestros datos. Como además es mucho más rápido que un disquete, el arranque nos hará perder menos tiempo. En general, estas mismas consideraciones pueden hacerse para cualquier otro programa que queramos ejecutar (muchos ni siquiera caben en un disquete⁴⁰), y es por eso por lo que los programas a emplear en un ordenador se “instalan” en el disco duro, junto a los datos con que vayamos a emplearlos.

Podemos decir que, al ser el disco duro una especie de memoria adicional (recuerde el esquema general de un ordenador), simplemente añadimos en ella los programas necesarios para que LR-PeCé llegue a ser un hombre de provecho. Dijimos también que el disco duro

39 Lo de “arrancar desde” es la forma habitual de indicar que es en un disco determinado donde se encuentra el sistema de arranque.

40 Muchos ni siquiera caben en 10 disquetes, y bastantes tampoco en 15. Por supuesto de 1'44 MB.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

va a ser el lugar de trabajo habitual, en tanto el disquete queda como un soporte de almacenamiento secundario, proporcionando ante todo la capacidad de intercambiar datos con el prójimo. Ahora mismo, sólo en contadas ocasiones tiene sentido siquiera trabajar directamente sobre los datos de un disquete⁴¹.

Pero eso no quiere decir que un disquete de arranque no tenga interés. ¿Qué pasa si, por error, avería, o por lo que sea, nuestro MBR, IBMBIO.COM, IBMDOS.COM y COMMAND.COM que tan adecuadamente tenemos en el disco duro se borran o modifican, de modo que el arranque no pueda continuar?. Pues que tenemos un ordenador incapaz de funcionar. La única manera de salir de tal atolladero es disponer de un disquete de arranque que permita reinstalar de nuevo el sistema operativo en el disco duro. Es ABSOLUTAMENTE NECESARIO tener uno de tales disquetes. Si Vd. usa PC-DOS o DR-DOS no es problema. Ya dijimos que eran distribuciones completas, capaces de poner por sí solas en marcha un PeCé. Basta con que conserve a buen recaudo los disquetes originales. Si Vd. emplea una de las modernas versiones de MS-DOS, vendidas como actualizaciones, más vale que lo primero que haga con ellas sea generar un disquete de arranque. El propio programa de instalación en disco duro le propone realizar tal operación y Vd. debería contestar que sí a tan razonable propuesta.

De todos modos, más que un disco de arranque simple, lo que nos interesa es tener un “disquete de emergencia”, capaz en caso de apuro no sólo de poner el ordenador en funcionamiento sino también de proporcionarnos todas las herramientas básicas para arreglar el contenido de un disco duro que se ha desviado del recto camino. Si nos ponemos estrictos, en realidad nos conviene tener un par de copias del tal disquete, sólo por si acaso. A lo largo de las siguientes páginas iremos montando uno y explicando el porqué de cada una de las cosas que incluyamos. Veremos que es un proceso muy sencillo y, una vez entendido qué hace falta para un realizar un “boot” en un PeCé, nada misterioso.

Y terminemos la recapitulación justificando porqué, en caso de disponer de interruptores separados para los distintos componentes, la unidad central es lo último a conectar. Una parte fundamental del proceso de arranque es la identificación y comprobación del estado funcional de las distintas partes del ordenador. Si tengo un CD-ROM externo y está apagado, no podrá responder cuando MS-DOS le pregunte si está allí. LR-PeCé no es capaz de distinguir si está averiado o simplemente apagado. Lo único que ve es que no funciona, y en consecuencia no lo incluye entre los componentes utilizables. Como máximo el ordenador arrancará pero, aunque yo enchufe el CD-ROM después, no podré trabajar con él. Para MS-DOS, no estaba en el momento de comenzar a trabajar y no va a estar hasta que no sea capaz de identificarlo la próxima vez que haga un “boot”⁴².

41 Sólo si no tenemos sitio en el disco duro para hacer una copia de los datos del disquete. En caso contrario se copian a un lugar conocido del disco duro, se trabaja, se vuelven a pasar las nuevas versiones al disquete, y, si así se desea, se borra la copia del disco duro. Aun con toda esta complicación adicional, merece la pena.

42 No es aplicable a todos los sistemas operativos. Casos hay en que pueden ponerse y quitarse cosas sobre la marcha y el sistema operativo es capaz de reconocerlas sin necesidad de volver a arrancar. Es algo bastante conveniente. La última versión de Windows, conocida en su etapa de desarrollo como “Chicago” y lanzada finalmente en Septiembre de 1.995 bajo el nombre de Windows’95, debería permitir, en algún grado, este tipo de manipulaciones. Otra razón para enchufar la unidad central en último lugar era, en equipos antiguos, el intentar protegerla de

Empezando por el principio

Puestos a practicar lo que sabemos, lo cierto es que por ahora no tenemos mucho repertorio donde elegir. Nuestras posibilidades se reducen a rearrancar el ordenador algunos cientos de veces. Y hasta para esto es mejor ver como se hace un “reset”, por no estar todo el rato encendiendo y apagando⁴³. Es algo que, ciertamente, va tener que realizar con cierta frecuencia. Con tanta frecuencia que en un PeCé es posible hacerlo hasta de tres formas diferentes, como dijimos en un capítulo anterior⁴⁴. Veámoslas por orden de menor a mayor contundencia.

Si las cosas han ido bastante mal y el ordenador se ha vuelto loco pero conservamos el uso, siquiera parcial, del teclado, es posible “resetear” un PeCé pulsando simultáneamente las teclas “Control+Alt+Suprimir”. Hacen falta las dos manos para que no sea fácil pulsar semejante combinación accidentalmente. Recuerde que un “reset” no era algo deseable ya que se perdían todos los datos que no hubiéramos puesto a buen recaudo en el correspondiente fichero. Primera posibilidad.

Si el control del teclado ha pasado a mejor vida, tal cosa no funciona, y pulsar la antedicha combinación no produce ningún efecto. Ha llegado el momento de probar la segunda forma de “resetear”. Uno de los botones de la parte delantera de la unidad central estaba rotulado, justamente, “reset”. Pulsarlo envía, a través del correspondiente cable, una señal eléctrica a la CPU que fuerza el “reset”.

Y si lo anterior tampoco funciona, cosa que en alguna ocasión ocurre, ha llegado el momento de adoptar medidas drásticas. Se apaga el ordenador mediante el interruptor principal, se espera algunos segundos (veinte o treinta)⁴⁵, y se vuelve a encender de nuevo. Tercera posibilidad⁴⁶. Si aún así nuestro PeCé no arranca, es que tenemos verdaderos problemas.

posibles voltajes transitorios inducidos al conectar el resto de componentes. No tiene sentido en la actualidad. La impresora y el monitor, por supuesto, sí que pueden enchufarse en MS-DOS en cualquier momento sin mayor precaución.

- 43 No se asuste, pero cuando más sufren los circuitos electrónicos no es cuando están en marcha varios días, sino cuando se encienden o apagan. Se debe a los cambios de temperatura a que se ven sometidos. A pesar de todo, es posible que se le caiga el dedo antes de que se funda la CPU por haber encendido el ordenador demasiadas veces.
- 44 Algunos ordenadores, como por ejemplo el Apple Macintosh, han pretendido en algún momento hacer creer que jamás se iban a colgar, por lo que no facilitaban llevar a cabo tan necesaria operación. No por ello dejaban de colgarse, de manera que la única consecuencia práctica de tal enfoque consistía en que el “reset” era más difícil de hacer. El PeCé es en este aspecto mucho más realista.
- 45 Algunas zonas de la RAM de un PeCé no son normales y, aún sin alimentación eléctrica, les cuesta un tiempo borrar su contenido. Es por esto que hay que esperar un poco, para asegurarse de que absolutamente toda la RAM va a estar completamente vacía.
- 46 Los portátiles son un caso un tanto particular. Aquellos que almacenan sus datos en RAM no se apagan nunca del todo, como ya se dijo. En estos casos, el manual indica claramente como proceder para “resetear” el chisme con autoridad. Un motivo más para darle un vistazo más pronto que tarde.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Los dos primeros tipos de “reset” no requieren que se apague el ordenador. Se habla de “reset en caliente” y, como el primero es menos contundente que el segundo, se distingue en ocasiones entre ellos designándolos respectivamente “reset blando” y “reset duro”. El tercero no es más que realizar un nuevo arranque, desde el principio, hace falta apagar el equipo, y por ello se denomina “reset en frío”⁴⁷.

Pues nada, a practicar. Cuando se canse, lo que ocurrirá pronto ya que hay que reconocer que resulta un espectáculo un tanto monótono, pase al siguiente apartado, en el que veremos qué tenemos hasta ahora en un PeCé con MS-DOS en condiciones de marcha.

Todo son ficheros, más o menos

LR-PeCé se intenta unificar

El ordenador adora los ejercicios de ocultamiento de información y transformismo, ya vimos. Una de las obligaciones del sistema operativo consiste en poner a disposición del usuario sus recursos más importantes, y en tal tarea se encuentra un campo abonado para la aplicación de dichos principios. Ver cómo quedan los tales recursos, hay que verlo, pues más pronto que tarde vamos a querer usarlos. Podemos en consecuencia, como muchas otras veces, aprovechar la ocasión para matar varios pájaros de un tiro y contemplar además un buen ejemplo de dos de los rasgos más importantes de la personalidad del ordenador en acción⁴⁸.

Establezcamos en primer lugar claramente de qué se trata. Un disco, duro o blando, era un mar de sectores, definido inicialmente por características tales como el número de cabezas, sectores por pista, y demás etcéteras. Pero lo que a mí me interesa al final es trabajar con mis datos y me importa un bledo (al menos en principio) en qué sector de qué cara del disco están en realidad, siempre que pueda referirme a ellos con un nombre, y haya un programa que se encargue, a base sólo de ese nombre, de buscarlos en los sectores, pistas y cabezas correspondientes⁴⁹. Para poder hacerlo, se diseña un programa (el sistema operativo⁵⁰) que

47 En cualquier ordenador, aunque no disponga de botón de “reset” o de combinación de teclas al uso, es posible realizar éste último.

48 Ya que es un buen ejemplo, y ambos principios son más importantes de lo que parece, aprovecharemos también para profundizar un poco. Como criterios de diseño del software, tanto el ocultamiento de información como la simulación de comportamientos pueden ser sumamente provechosos y una auténtica bendición para el usuario final. Pero el emplearlos en exceso puede acabar llevando a situaciones más complejas que la que inicialmente se quería simplificar y hacer el ordenador todavía menos utilizable de lo que ya era. Y desgraciadamente no es tan raro que se abuse de ellos, pues para diseñar un buen programa no basta con ocultar TODOS los detalles ni con darle CUALQUIER apariencia final. Debe elegirse con buen criterio qué mostrar y qué no y qué aspecto conviene más a nivel global. Puesto que lo anterior es un buen ejemplo, intentemos quedarnos a partir de él con las ideas principales.

49 Aprovecho para hacerle notar, de todos modos, que en ocasiones hace falta conocer este tipo de detalles, por lo que una información tan cuidadosamente oculta debe sacarse de nuevo a la luz. Cuando de ordenadores se trata, “oculto” está bien pero “inaccesible” no tanto. Si hay problemas en el disco y se deteriora la tabla de ficheros, puede que sí que me interese conocer con total exactitud los sectores que integran mi fichero, porque al menos así será posible intentar recuperar aunque sea parte de su contenido. En cualquier situación no estrictamente normal suele ser preciso destapar esos detalles que tan engorrosos nos resultan si los tenemos siempre delante.

Todo son ficheros, más o menos

primero oculta esos enojosos detalles y luego hace aparecer el tal disco como una unidad, sea la "A:", la "C:" o cualquier otra, donde yo podré colocar mis datos con su nombre correspondiente sin preocuparme de nada más.

Se oculta una información innecesaria, haciendo que lo mismo sea tener un disco duro marca Seagate de 1049 cilindros, 63 sectores por pista y 16 cabezas, que uno Conner de 524 cilindros, 63 sectores por pista y 32 cabezas⁵¹; y luego se le da un aspecto sencillo y útil que nos permita acceder a él, presentando lo anterior como, pongamos, "un disco C: de 540 Megabytes", que son justo los datos que le hacen falta a un programa normal. O a nosotros, sin ir más lejos.

Lo que MS-DOS deberá hacer será ni más ni menos que montar los distintos elementos que para él van a componer nuestro ordenador y, en el proceso, darles una apariencia lo más útil y sencilla posible.

Y ya que estamos con asuntos relacionados con arquitectura en general, digamos también que, a la hora de diseñar un sistema cualquiera, no está de más intentar simplificarlo cuanto más mejor, haciendo que aquellas cosas que deben integrarse en él y que muy probablemente son en realidad notablemente diferentes entre sí, se conviertan en algo lo más parecido posible. De este modo se consigue tratarlas a todas con los mismos procedimientos, con lo que nos ahorramos un montón de trabajo en un montón de sitios. Puestos a otorgar apariencias graciosamente, podríamos tratar de seguir tan sabio principio. Siempre que no vayamos absolutamente en contra de su naturaleza será posible presentarlas como más nos apetezca. Y tal vez aún entonces, aunque eso sí, con mayor esfuerzo.

No es extraño pues que, continuando en esa línea, se intente unificar lo más posible el aspecto que ante el sistema operativo deben adoptar los distintos elementos a manejar, y tratemos de hacer que todos ellos se presenten vestidos con un mismo ropaje. Que se intenten ocultar aquellos detalles, nimios sin duda, que diferencian a una impresora de un monitor, de modo y manera que para el sistema operativo ambos sean más parecidos que diferentes y se puedan manejar del mismo modo.

Para lograrlo, el camino a seguir es bastante evidente. Si nos acordamos del esquema general de funcionamiento de un ordenador, sabremos que todo lo que éste hace es pasar datos de un sitio a otro, más o menos. Tal vez los cambie algo en el camino, pero eso va a ser todo. Puesto que nuestros datos se encontrarán normalmente almacenados en ficheros, la cosa debería ser equivalente a un intercambio de ficheros. Tal vez cambien algo en el camino,

Muchos sistemas operativos solventan el expediente haciendo que los detalles a ocultar, más que desaparecer para siempre, se queden simplemente en las habitaciones de servicio menos concurridas del edificio, de forma que no se les vea pero estén disponibles si hacen falta. Podemos decir que MS-DOS es de ese estilo.

- 50 El sistema de ficheros, con mayor precisión. El sistema operativo es un conjunto de programas, o mejor dicho, un conjunto de conjuntos de programas.
- 51 Seagate y Conner son dos marcas de fabricantes de discos duros. Si quiere más, hay van otras dos: Quantum, Maxtor. Y por supuesto hay más todavía. Las cifras anteriores, junto al tamaño de sector que empleen, definen por completo un disco.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

pero de nuevo eso va a ser todo. Nada hay de raro en lo anterior ya que, en sentido general, un fichero es un conjunto de datos con la entidad propia suficiente como para manipularse como un todo único. Es el “dato” con que trabajaremos normalmente.

Pongamos algún ejemplo. Si yo estoy almacenando aquello que tecleo en un momento determinado en un fichero del disco duro, puedo representar el proceso como un flujo de datos que salen del teclado y van a parar al fichero que sea. Si lo que hago es visualizar un texto de ayuda, no estoy haciendo sino pasar un fichero desde donde quiera que se encuentre a la pantalla. Y así sucesivamente. Puedo decir que en el primer caso estoy copiando un fichero desde el teclado al disco duro, y en el segundo, desde éste último a la pantalla. Pantalla y teclado pueden verse como unos ficheros más, algo especiales quizá, pero sobre los que más o menos vamos a poder realizar las mismas operaciones básicas que con los demás.

MS-DOS se apunta a este enfoque⁵², y los distintos recursos a gestionar se incorporan al sistema operativo como tales “ficheros”⁵³. Simplemente recursos con un nombre, un tanto especial eso sí, susceptibles de usarse en lugar de cualquier otro fichero la mayor parte de las ocasiones. Si volvemos a mirar la página del capítulo 4 en la que le echábamos una miradita al contenido de la memoria del ordenador tras haberlo arrancado, veremos algunos de tales inventos. Sólo por evitarle ir hasta allí, lo que siempre es un engorro, pongamos un pequeño resumen con lo que nos interesa ahora. A saber:

<u>Dirección</u>	<u>Usuario</u>	<u>Tamaño</u>	<u>Tipo</u>
70:050B	PRN		Controlador disp. incorporado
70:051D	LPT1		Controlador disp. incorporado
70:052F	LPT2		Controlador disp. incorporado
70:0541	LPT3		Controlador disp. incorporado
70:0553	AUX		Controlador disp. incorporado
70:0565	COM1		Controlador disp. incorporado
70:0577	COM2		Controlador disp. incorporado
70:0589	COM3		Controlador disp. incorporado
70:059B	COM4		Controlador disp. incorporado
70:0602	CLOCK\$		Controlador disp. incorporado
70:0645	CON		Controlador disp. incorporado
70:0671	A:-C:		Controlador disp. incorporado
123:0048	NUL		Controlador disp. incorporado

52 Se apunta, pero tampoco es que lo haga con demasiado convencimiento. Otros sistemas operativos son mucho más drásticos al respecto. El del New Brain, tan citado en este libro, era uno de ellos.

53 Esta es una nota para gente más bien escrupulosa, esos a los que les gusta la exactitud por encima de todo. Probablemente sería más correcto decir que los recursos se incorporan como “unidades”, del mismo modo que lo hacían los discos, en lugar de hablar de “ficheros”. El hecho es que dicha precisión puede más confundir que ayudar, ya que muchas veces pueden ocupar el lugar de un fichero de verdad en una orden y, al contrario de lo que ocurre con las unidades normales, no pueden crearse ficheros en su interior, ya que no son periféricos de almacenamiento. Como lo de la precisión no es algo que nos deba preocupar en exceso, y con esta nota debería valer, seguiremos con nuestra línea. Que como se verá, permite continuar la exposición con bastante coherencia.

Todo son ficheros, más o menos

No necesitamos fijarnos nada más que en un par de cosas. Que lo que aquí aparece son recursos que MS-DOS es capaz de controlar, que están cargados en la RAM del chisme para poder usarlos cuando haga falta, y que MS-DOS les asigna unos nombres un tanto curiosos. PRN es la abreviatura de “Printer”, impresora en inglés. LPT es la abreviatura del inglés “Line Printer” o impresora de líneas, o si lo preferimos, impresora a secas una vez más. Hay tres de ellas. COM a su vez quiere decir “Communications”, o sea puerta de comunicación, por otro nombre puerta serie. En principio, hasta 4 de ellas, aunque ya vimos que los PeCé no suelen contar con más allá de dos⁵⁴. A: y C: deberían resultarnos familiares.

El nombre correcto de tales cosas, de lo que nosotros hemos llamado “ficheros un tanto especiales”, es el de “controladores de dispositivo incorporados”. Lo que sólo quiere decir que lo que estamos viendo son los “drivers”, los controladores⁵⁵, ya cargados en la RAM, del conjunto de chismes que MS-DOS necesita para trabajar a nivel básico y por lo tanto incluye irremediablemente.

Hay algunos más, pero sólo prestaremos atención a dos un tantico peculiares: CON y NUL. CON es la consola, el conjunto de pantalla y teclado que tantas veces hemos dicho era la base de nuestra comunicación con el ordenador. NUL es una especie de desagüe. Suele preferirse añadir “:” a los nombres de los recursos. Esto permite identificarlos como “ficheros especiales”. Tendremos pues LPT1:, LPT2:, CON:, NUL:, y así sucesivamente⁵⁶.

Nuestros “ficheros especiales”, a los que en lo sucesivo y en aras de la brevedad vamos a llamar “dispositivos” a secas, son por otra parte un ejemplo bastante contundente de un concepto que no hemos introducido todavía y que resulta ser de más que notable interés en la práctica. Y por si fuera poco, tiene cierto aire de material avanzado, con lo que vamos a

54 ¿Por qué cuatro?. Bueno, puesto que tenemos que cargar un controlador, el mismo para todas, igual da ocho que ochenta. No perdemos mucho dejando que haya hasta cuatro. Los Pecé han tenido siempre vocación de ordenadores serios y, en este contexto, las comunicaciones con ordenadores distantes no son algo que se pueda ignorar. Al montar 4 puertas serie, MS-DOS está ya preparado de entrada para añadir una tarjeta modem que permita ponernos en contacto con el ordenador principal de nuestra compañía a través de la línea telefónica. La conexión al modem, en caso de existir y si no le decimos lo contrario, suele aparecer en MS-DOS como COM3.

55 Los “drivers” eran los “pilotos” de un determinado componente de hardware, como en su momento dijimos. También dijimos que íbamos a preferir esta denominación a la también aceptable de “controladores”, la que aquí aparece, para evitar confusiones con los otros controladores, los que no eran sino ordenadores más o menos invisibles en la actualidad incorporados a multitud de aparatos. Y también porque lo de “driver” se emplea como tal, o sea que conviene acostumbrarse. No se líe.

56 No ha sido así siempre, pero por suerte una vez más, ya lo es. MS-DOS era, como buen sistema operativo arcaico con interfase de usuario basado en línea de ordenes, un tanto niquitoso con la sintaxis. Vamos, que si había que poner dos puntos o no, había o no que ponerlos, o no funcionaba. Pero con el tiempo, el intérprete de órdenes, nuestro COMMAND.COM, se ha ido civilizando, y ahora mismo muchas veces da igual poner unas cosas u otras. Cuando MS-DOS tiene alguna duda nos pregunta, se le contesta para aclararle las cosas (lo que requiere simplemente pulsar una u otra tecla de las que nos indique), y en paz.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

darnos el gustazo de explicar, y quizá aprender, cosas que la gente normal que no tiene tanta suerte a la hora de elegir su primer libro no ve hasta que no lleva algunos años en el negocio.

Si reflexionamos un pelín, caeremos en la cuenta de que nuestros dispositivos presentan importantes diferencias entre sí. Tan grandes que deberemos incorporarlas de una u otra forma si queremos que la cosa funcione. Cojamos por ejemplo a “NUL:”, nuestro desagüe, que, no sé por qué, me resulta particularmente simpático. Mirándolo con atención a los ojos, constataremos que si bien los datos pueden acabar allí, es decir que NUL: puede ser un destino perfecto para una operación de intercambio de datos, especialmente si no tienen mayor interés, es difícilmente admisible que sirva como origen de algo. Al desagüe pueden ir cosas, pero no pueden salir de él. “CON:”, en cambio, la consola, el conjunto de teclado y pantalla, admitirá actuar como origen y como destino. Podremos crear datos con CON: mediante el teclado, o mandarlos allí para que se visualicen en la pantalla. Si aplicamos la jerga, diremos que CON: será un dispositivo de lectura-escritura, en tanto que NUL: será de sólo escritura. Para que el sistema operativo no se líe y se ponga a esperar eternamente a que un dato salga mágicamente del desagüe, deberemos añadir algo que permita distinguir estas cosas. Ya que si no lo hacemos, puede en efecto ponerse a esperar eternamente cuando así se lo mandemos en un momento de debilidad mental, no nos dejará hacer nada mientras tanto, y en consecuencia se colgará. Perderemos el control del ordenador⁵⁷.

Se solventa el problema asociando a los controladores unos “atributos” que definen su comportamiento en este tipo de operaciones. NUL: será un dispositivo de “sólo escritura”, en tanto que CON: lo será de “lectura-escritura”. La cosa no acaba ahí, ya que podemos a su vez mirar a los ficheros como un potencial origen o destino de datos, es decir como si fueran dispositivos. La escritura supone la posibilidad de modificar el contenido de un fichero, pudiendo llegar hasta a borrarlo. El ejemplo más claro de fichero que debe leerse siempre que sea necesario pero es mejor no modificar o borrar es el de aquel contenga un programa clave para el funcionamiento del ordenador, tal como IBMBIO.COM. No estaría de más impedir dichas operaciones sobre él. Para abrir el manejo de ficheros a tales sutilezas, todo lo que hay que hacer es ampliar el ámbito de aplicación del concepto de “atributo” desde nuestros “ficheros especiales” a cualquier fichero de a pie. Bastará con declarar “sólo de lectura” a IBMBIO.COM. Ventajas de la unificación⁵⁸.

57 Estoy dramatizando un poco, por aquello de que hay que motivar al lector. La realidad es menos traumática en este caso. Con las versiones de DOS con que suelo trabajar, el intento de sacar datos de NUL: y dejarlos en un fichero produce como único resultado que se cree un fichero vacío. O sea, en resumen, que no se pueden sacar datos del desagüe. Pero es forzoso precisar que no se cuelga. Lo que por otra parte me permite hacerle notar que la manera en que el sistema operativo reaccione en situaciones extremas puede, sin dejar de ser lógica, añadir una notable robustez al funcionamiento del ordenador. Esta “tolerancia a fallos” es una bendición de los cielos.

58 De nuevo una nota para amantes de la precisión. MS-DOS no permite en condiciones normales (las redes son otra historia) definir ficheros como “de sólo escritura”. NUL: y sus colegas son, de verdad, casos bastante especiales de los que se encarga el sistema operativo por su cuenta, sin dejarnos meter baza. No existen como ficheros reales. Pero prefiero exponer las ideas resaltando el concepto general. Hay que unificar, para aprendiendo poco llegar a saber lo suficiente.

Todo son ficheros, más o menos

Un fichero en MS-DOS, por tanto, será un conjunto de datos almacenados con un nombre y una serie de atributos que regulen el acceso a ellos. Los atributos accesibles al usuario son los de “sólo lectura”, “sistema”, “oculto”, “archivo” y “directorio”. En inglés, ya que como de costumbre no viene mal saber la traducción, serán respectivamente “read only”, “system”, “hidden”, “archive” y “directory”, y se abrevian mediante su inicial, con lo que se transformarán respectivamente en “r”, “s”, “h”, “a” y “d”, que será como los veamos normalmente. Ya que hablamos de cosas que se guardan con un archivo, aprovechemos para decir que además del nombre, los atributos, y los datos y cuantos hay en caso de haberlos, MS-DOS le saca partido a tener un reloj a su disposición grabando junto a cada fichero la fecha y hora en que se creó, cosa que viene muchas veces de perlas. Conviene saberlo, y también conviene tomarse la molestia de comprobar de vez en cuando que el reloj del ordenador está en hora, no sea que cuando lo necesitemos resulte que no lo podemos usar por haber hecho creer a LR-PeCé que eran las 3 de la tarde del 15 de junio de 1.976 cuando eran las 8 de la mañana del 15 de octubre de 1.995. Es fácil hacerlo, luego diremos como.

Como para poderles sacar partido a los atributos hay que conocerlos un poco, vamos a explicarlos. En primer lugar decir que son atributos de “todo” o “nada”, y que en consecuencia o están o no, no hay términos medios⁵⁹. Un fichero será “sólo lectura” o no, o tendrá el atributo de “archivo” o no. En segundo lugar decir que son independientes entre sí, o sea que el hecho de que uno esté puesto o quitado no tiene ninguna consecuencia sobre los demás. Y en cuanto a significados, el más fácil de entender es “r”, el de “sólo lectura”. Si un fichero lo tiene puesto será posible leerlo, por ejemplo para copiarlo a otro sitio, pero no se podrá modificar, ni borrar, ya que cualquier operación de escritura sobre él está terminantemente prohibida mientras persista la situación⁶⁰. El atributo “a”, el de “archivo”, es bastante discreto, lo que es una pena ya que resulta de los más útiles. Más que “archivo”, deberíamos traducirlo por “pendiente de archivar”. Siempre que se modifica un fichero, y sin que nadie le diga nada, de manera totalmente automática, MS-DOS pone el atributo “a”. Con lo que todo archivo que haya sido modificado alguna vez, o varias, lo tendrá colocado. Hasta que no lo quitemos, ya veremos cómo, este atributo nos permitirá fácilmente localizar todos los archivos que en cualquier momento, con cualquier programa, hayan sufrido un cambio de su contenido. Los atributos “s”, “sistema”, y “h”, “oculto” son parecidos en tanto en cuanto sirven respectivamente para indicar qué ficheros son de interés para el sistema operativo, por lo que es mejor dejarlos en paz; o cuales no se desea que aparezcan fácilmente ante los ojos del usuario, aunque estén allí⁶¹, por parecidos motivos.

59 Ya deberíamos ir traduciendo según que cosas a interioridades del chisme. Esto nos permite deducir que a nivel interno, cada atributo vendrá representado por único bit. Si está a 1, el atributo existe. Si está a 0, no. Un bit nos permitía almacenar sólo dos posibilidades únicas.

60 No hay que confundir los atributos con la protección de escritura de los disquetes. Si se acuerda, moviendo una pestaña del disquete de 3'5", hacemos imposible la modificación de cualquiera de los ficheros que se encuentren en él. Los atributos se aplican a cada uno de los ficheros de un soporte, disco duro, disquete o lo que sea, y podemos perfectamente tener un disquete sobre el que se pueda escribir, con alguno de sus ficheros protegidos mediante el atributo de “sólo lectura”. Estos ficheros, y sólo ellos, no se dejarán manipular. Los demás sí. La pestaña es una protección global de todo el disquete y tiene sus usos. Los atributos afectan a ficheros individuales y también tienen los suyos.

61 Si Vd. sabía el suficiente MS-DOS como para conocer la orden “DIR”, tal vez se haya sentido tentado cuando explicamos el proceso de arranque de mirar a ver si estaban los ficheros citados,

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Un fichero oculto no puede verse, manipularse, ni borrarse, con facilidad⁶². En cuanto al atributo “d”, de directorio, sirve para que MS-DOS sepa qué entrada de la tabla de ficheros define un directorio y cual no. Como un directorio no contiene datos, su entrada en la lista de nombres será un nombre más, sin datos, pero con el atributo de directorio colocado. No tiene mucho interés, pero explica que los nombres de directorios se ajusten también a la norma 8.3. Acordémonos en lo sucesivo de que un fichero incluye nombre, fecha, hora, los cinco atributos citados, y los datos y cuantos hay, en caso de haberlos, y vale por el momento.

Recapitulemos, ya que es importante. MS-DOS se encarga de poner a nuestra disposición los recursos del ordenador. Con el fin de unificar su manejo y ocultar detalles innecesarios, los hace aparecer ante el usuario como si fueran ficheros. Y los diferencia del resto de ficheros normales tan sólo mediante sus nombres, predeterminados por el propio sistema operativo, y dotándolos de atributos acorde con su naturaleza. A partir de aquí, es posible manipularlos como a cualquier otro fichero. Para hacer casi cualquier cosa en MS-DOS bastará por lo tanto con disponer de órdenes dedicadas a manejar ficheros de distintas formas. Y es ni más ni menos que eso lo que la mayor parte de las órdenes de MS-DOS van a hacer.

Y con esto, estamos ya en disposición de comenzar a aprender, en rápida sucesión, las palabras mágicas que obligarán a LR-PeCé a que haga algo que nos pueda resultar de interés.

Abracadabra

Palabras mágicas para LR.PeCé

Dominar un ordenador, y utilizar sus recursos en nuestro propio beneficio, tiene un cierto halo de magia negra. Aunque, como se verá, en realidad de eso nada, me resulta imposible renunciar a tan literaria posibilidad como hilo conductor del que servirme en este trascendental momento que sin duda cambiará el rumbo de su vida, querido LAO. Vamos a comenzar con vigor nuestro particular curso para brujos y hechiceros que deseen dominar a LR-PeCé, pues para poder comenzar a practicar lo que ya sabemos y continuar aprendiendo cosas nuevas, es necesario finalmente introducir una orden MS-DOS de las de verdad, una de esas palabras mágicas que el ordenador en MS-DOS entiende. Y será sólo la primera, porque vendrán más en rápida sucesión.

Todo buen hechicero debe controlar el desarrollo de su magia, y eso requiere saber qué está ocurriendo. Empezamos sentados ante un PeCé con el “boot” realizado. Nuestro particular genio ha salido de su lámpara, bajo la influencia del potente conjuro que desencadenamos al pulsar el botón de arranque, y, tras acabar de cargar COMMAND.COM, nos ha mostrado

tales como IBMBIO.COM y IBMDOS.COM o equivalentes. Y puede que ande ligeramente mosca por no haberlos encontrado. Ahora ya sabe el suficiente MS-DOS como para figurarse que están, pero ocultos. De hecho, ambos son ficheros ocultos y de sistema.

62 Un usuario inexperto es, en principio, bastante peligroso para sí mismo. El atributo de “oculto” permite quitar de su vista aquellos ficheros con los que no debería enredar por su propio bien. Como lo de la sobreprotección tiene también sus problemas, existen múltiples posibilidades en MS-DOS, si bien ligeramente retorcidas, para ver lo que no debería ser visto. El usuario las aprende con la experiencia, y para entonces ya no es tan problemático que enrede con esas cosas ya que él sabrá lo que hace, que ya es mayorcito. En resumidas cuentas, de esta forma cumplimos con aquello de dejar los detalles que no deben ser vistos ocultos pero no inaccesibles.

el “prompt”. Leámoslo como si nos dijera “Ordena y obedezco, ¡Oh, amo!”. Protegidos de su perversa influencia en el interior del pentáculo que hemos trazado a nuestro alrededor, nos aprestamos a comunicarle nuestros deseos. Al contrario que con los genios de todos los días, en este caso no son sólo tres, sino todos los que queramos. LR-PeCé los hará realidad en cuanto tecleemos algo que él sea capaz de entender y pulsemos CR. Lo cual, en la prosaica realidad tecnológica que deberá ser nuestra guía para enfrentarnos a las fuerzas primigenias que tratamos de controlar en nuestro provecho, no querrá decir nada más que COMMAND.COM, el programa con el que estamos trabajando, será capaz primero de interpretar y a continuación de ejecutar la línea de órdenes que le pasamos al pulsar CR.

¿Que orden merece ser la primera que aprendamos?. ¿Cual es la palabra mágica más importante?. Este tipo de preguntas no suele admitir una respuesta precisa, pero quizá deberíamos empezar por algo que refleje lo más posible el funcionamiento del ordenador. Y no nos vamos a poner a explicar cosas raras, por lo que, al igual que todas las que sigan, convendría que fuera de uso muy frecuente. Sea. Ya que estamos definiendo el funcionamiento de un ordenador ante todo como un intercambio de datos, de ficheros en resumidas cuentas, la primera orden que aprendamos hará esto básicamente. Hela aquí: “COPY”. ¿Que está en inglés?. Las oscuras artes de la hechicería siempre han requerido el uso de ignotos lenguajes. Confórmese con que para domeñar a LR-PeCé no haga falta aprender sánscrito. Además, “COPY” se parece asombrosamente a “COPIAR”, que es lo que dicha orden hace. “¡Pues vaya, tanto rollo para eso!”, tal vez esté pensando. No se acelere, querido LAO. Nuestra primera palabra del vocabulario en MS-DOS sirve para decirle a LR-PeCé que tenga a bien copiar algo, fundamentalmente un fichero. O cualquier cosa equivalente. Tal como un flujo de datos desde o hacia uno de nuestros “dispositivos”. Puesto que en buena medida el ordenador no hace básicamente otra cosa, veremos que da para mucho más de lo que pueda parecer a primera vista.

Ocupémonos en primer lugar de reflexionar sobre cual será la forma exacta que deberá adoptar para surtir los efectos deseados. Ya se sabe que los genios, espíritus, y cualquiera de las distintas clases de demonios habitualmente objeto de manipulación mágica son siempre muy quisquillosos en materia de sintaxis. Si hay que poner una coma, hay que ponerla, o las consecuencias pueden ser terribles⁶³. Si hay que copiar, deberemos decirle qué queremos copiar (el origen de la operación), donde dejarlo (su destino), y tal vez sea necesario indicarle además el modo en que queremos hacerlo, en caso de que existan varios. Con esto deberíamos evitar cualquier tipo de sorpresa desagradable. El aspecto de la pantalla en MS-DOS, con “prompt”, cursor, y la orden en forma esquemática, será algo así como:

```
C:\>COPY ESTO AQUELLO /PARAMETROS_
```

Explicuemos algunos detalles. Con lo anterior, LR-PeCé copiará “esto” a “aquello” según le indiquen los parámetros que añadamos⁶⁴. Destaquemos en primer lugar que los espacios

63 Hay muchos chistes que se ocupan del tema. Baste con uno. Se cuenta que un desdichado, incapaz de articular correctamente una de sus manos por tenerla bloqueada hacia abajo, consiguió los famosos tres deseos. Pidió en primer lugar ser inmensamente rico. Luego, ordenó que sus dos manos fueran iguales. Y el genio le bloqueó las dos hacia abajo. Pasmado, dijo: “No hombre, no. Al revés”. Y se las bloqueó hacia arriba. Tras lo cual, volvió a su lámpara.

64 Dijimos que CP/M no tenía fama de sistema operativo agradable de usar. Para que se vea con detalle qué queríamos decir, bastará con que precisemos que el equivalente del COPY anterior

CAPÍTULO 7. PeCés: MS-DOS/Windows.

en blanco separarán las distintas partes de la orden y que no pasa nada si ponemos más de uno, pero que MS-DOS no va a aceptar que no estén. O sea que nada de poner COPYES-TOAQUELLO. Y que da igual que escriba en minúsculas o en mayúsculas. Tampoco MS-DOS se va a quejar, ya que no las distingue a este nivel. En segundo lugar que los parámetros se indican en la orden anteponiéndoles una barra normal, o en ocasiones un guión. En jerga, diremos que la “/” actúa como un separador que indica que lo que sigue son los parámetros (por favor, lea la nota⁶⁵). Y en tercero y por supuesto que “esto” y “aquello” serán cualquier nombre de fichero de verdad o de dispositivo aceptable para MS-DOS. Lo que significa que deberán ser nombres válidos de fichero, completos o no, o dispositivos que MS-DOS reconozca. No vamos a poder copiar a CDX1:, ya que no es un dispositivo que tenga sentido para MS-DOS. No vamos a poder copiar ESPOSA.TXT a ESPOSA_C.TXT si antes ESPOSA.TXT no existe. O mejor dicho, y ésta es una puntualización importante, aunque exista, no vamos a poder copiarlo si MS-DOS no es capaz de encontrarlo.

¿Y cómo saber si MS-DOS lo va a encontrar?. Ha llegado el momento de hablar de las ubicaciones de los ficheros y de su importancia a la hora de manejar el ordenador. Nosotros tenemos problemas para encontrar las cosas cuando hay un montón enorme de ellas. LR-PeCé también. Sería posible hacer que LR-PeCé mirara en todos los ficheros de todos los discos que tenga cuando le diéramos cualquier orden. Pero aunque el ordenador sea rápido, buscar un fichero concreto entre 14.265⁶⁶ lleva su tiempo, y hacerlo así conduce inexorablemente a enlentecer la máquina más allá de lo soportable. En esta situación, los directorios entran al rescate, y es otro de los impagables servicios que nos prestan.

Ya vimos, comenzamos diciendo una vez más, que el fin primordial de los sistemas de ficheros jerárquicos era permitir agrupar los ficheros existentes en conjuntos más pequeños, en resumidas cuentas organizar el informe montón de información que podamos tener en una unidad de almacenamiento lo suficientemente capaz. Entre otras cosas, esto nos permite limitar las búsquedas de ficheros a grupos concretos. Por seguir con el símil del callejero, diremos que le vamos a proporcionar a LR-PeCé, por diversos medios, una lista de direc-

era en CP/M ni más ni menos que algo así como PIP DESTINO=ORIGEN[PARAMETROS]. No hay error. Origen y destino van al revés y se llama PIP, no COPY. Para que C:\ESPOSA.TXT se copiara a A:\ESPOSA.CPY la cosa podía ser PIP A:\ESPOSA.CPY=C:\ESPOSA.TXT[E]. Aunque se tengan preferencias, no debe dejarse que nos cieguen, y hay que arrearle a todo el mundo cuando y por lo que se lo merezca. CP/M, desde luego, era muy suyo. Si bien no exento de cierto informático sentido del humor, cuando se le cogía el chiste. Por ejemplo, la orden que servía para depurar programas se llamaba DDT. Aunque en teoría era la abreviatura de “Dynamic Debugging Tool”, no era probablemente más que uno de tantos casos de siglas metidas con calzador. Por si no coge el chiste, se lo explico. Un error en un programa se denomina “bug”, traducible por “insecto” de forma general. Nada mejor que el nombre de un famoso insecticida para el programa que debía ayudarnos a eliminarlos. Por si la curiosidad le corroe, aprovecho para precisar que PIP era a su vez la abreviatura de “Peripheral Interchange Program”.

65 Tanto la “/” como el “-” suelen servir para delimitar los parámetros de cualquier orden en MS-DOS. Algunas versiones de MS-DOS son más quisquillosas de lo habitual respecto a cual puede usarse en concreto. Nosotros emplearemos la barra normal, pero tal vez Vd. deba emplear el guión.

66 Y no me estoy inventando la cifra, que corresponde a un caso real no particularmente raro y contando sólo los ficheros en discos duros.

ciones donde debe buscar las cosas que tenga que buscar, y de este modo haremos que en lugar de tener que hurgar entre los 14.265 ficheros de antes, mire únicamente entre 200 o 300, lo que ya puede hacer con eficacia. Le diremos que cuando tenga que buscar algo mire en Zaragoza, no en Madrid, Bilbao, Valencia y Zaragoza, y no en toda, sino tan sólo en los barrios de Delicias y Las Fuentes.

Por el momento nos vamos a conformar con decir donde va a mirar LR-PeCé en primer lugar. Y la cosa es absolutamente banal. ¿Dónde busca Vd. las cosas en primer lugar?. Pues ni más ni menos que en el sitio donde está. Si no están allí, ya se irá a otra parte. De nuevo, el ordenador no es nada original. En un sistema de ficheros jerárquico, nosotros estamos en todo momento situados en algún directorio, el que se denomina directorio actual, y es allí donde se van a buscar en primera instancia los ficheros necesarios para cualquier cosa. ¿Cómo sabemos cual es el directorio en que nos hallamos en un momento determinado?. LR-PeCé suele mostrárnoslo a través del prompt. Si miramos de nuevo la orden resumida que poníamos anteriormente, veremos que comienza con “C:\”, el prompt, y termina con “_”, con lo que convinimos en su momento que indicaríamos la ubicación del cursor. Leamos el prompt. “C:\” nos dice que estamos en ese momento en el disco C:, y dentro de él en el directorio raíz. La “_” simplemente nos indica que podemos escribir lo que deseemos, que LR-PeCé está listo para escucharnos. Si, mediante las órdenes que más adelante veremos, nos desplazáramos por el sistema de directorios hasta colocarnos por ejemplo en “C:\CUENTAS\1995\AGOSTO\”, el prompt se visualizaría entonces como “C:\CUENTAS\1995\AGOSTO_”.

Bueno, empecemos. Como estamos en el directorio raíz, vamos a copiar un fichero que seguro, seguro, que está ahí. Se llama CONFIG.SYS. Hagamos una copia. Para distinguirla, le cambiaremos el nombre, ya que por supuesto no es posible copiar un fichero a otro que se llame igual y esté en el mismo sitio. Recuerde que no puede haber dos ficheros con el mismo nombre completo. A la copia la llamaremos CONFIG.CPY, para indicar que lo es. ¿Por qué esa extensión y no otra?. Por el momento las extensiones las elige Vd. a su gusto, querido LAO, y quieren decir lo que Vd. decida, o sea que sírvase Vd. mismo. La orden para perpetrar la anterior majadería sería ni más ni menos que:

```
C:\>COPY CONFIG.SYS CONFIG.CPY_
```

Aunque no muy sobrado de entendederas, nuestro genio de la lámpara particular, COMMAND.COM, será capaz de interpretarlo sin error. Buscará CONFIG.SYS en el directorio actual, el directorio raíz del disco C: tal como nos indica el “prompt”, lo encontrará por razones que posteriormente expondremos, y copiará su contenido a un nuevo fichero que llamará CONFIG.CPY. LR-PeCé no improvisa, y si le hemos dado ese nombre como destino, ese nombre le pondrá. Al final tendremos el fichero original y un fichero nuevecito y reluciente, nuestro CONFIG.CPY, que es con el que vamos a enredar en lo sucesivo. Hemos completado nuestra primera operación de proceso de datos con el ordenador. Hemos copiado un fichero. Maravilloso. Impresionante. Trascendental. ¿A que sí?.

Ya que estamos, podríamos preguntarnos algo así como lo siguiente: “Hemos dicho que LR-PeCé va a buscar en el directorio actual para encontrar el origen de una operación de copia, pero ¿dónde va a dejar el fichero destino?”. La respuesta es de nuevo banal. Si busca en el sitio donde está, va a dejar las cosas allí mismo. Mientras no le digamos lo contrario, todas las operaciones tendrán su origen y su destino en el directorio actual. Y es por ello por lo que es sumamente importante que sepamos siempre en qué directorio estamos trabajando.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

“Si no le decimos lo contrario” quiere decir, para todo LAO que lo sea y que además aspire a ser un brujo competente⁶⁷, que sí que podremos decírselo. ¿Cómo, pregunta?. Bueno, convendría que tuviéramos más de un directorio para poder explicarlo. Por lo tanto vamos a presentar la segunda orden MS-DOS. La palabra mágica encargada de crear un directorio casi de la nada es “MD”, abreviatura de “Make Directory” en inglés, lo que traducido al castellano quiere decir ni más ni menos que “Crear un directorio”, como debe ser. Si Vd. invoca a LR-PeCé con la cabalística fórmula, que presentamos de nuevo en forma esquemática:

```
C:\>MD NOMBRE-DIRECTORIO_
```

por ejemplo diciéndole:

```
C:\>MD EJEMPLO_
```

LR-PeCé obedecerá ciegamente y creará el directorio “C:\EJEMPLO\” a partir de la casi nada, es decir, una vez más a partir del directorio en que se encuentre. En este caso el directorio raíz, con lo que “EJEMPLO” será un subdirectorio suyo.

Lo de crear cosas a partir de la casi nada tiene su atractivo, y no es extraño que, cual prestidigitador sacando conejos de la chistera, nos apetezca crear más directorios, y más, y más... . No hay nada que oponer, sino es que para hacerlo con propiedad, una vez más, es preciso aprender otra orden. Hemos creado “C:\EJEMPLO\” pero seguimos situados en “C:\”. ¿Cómo vamos desde la Plaza de España a nuestra nueva calle?. La palabra mágica correspondiente es “CD”, abreviatura de “Change Directory” en inglés, y de nuevo, traducido al castellano quiere decir ni más ni menos que “cambia de directorio”, como debe ser. Si Vd. invoca a LR-PeCé con la fórmula, no menos cabalística, y que una vez más presentamos de forma esquemática:

```
C:\>CD NOMBRE-DIRECTORIO_
```

por ejemplo diciéndole:

```
C:\>CD EJEMPLO_
```

LR-PeCé obedecerá tan ciegamente como antes y, si encuentra el directorio en cuestión a partir de aquel en que se halle en ese momento, se situará raudo y veloz en él, y le informará de que así lo ha hecho cambiando el “prompt” en consecuencia. A partir de aquí podemos seguir creando directorios cual posesos⁶⁸:

67 Para ser un buen brujo, mago, o hechicero, son necesarios muy diversos dones, entre ellos el de observación y una cierta suspicacia. Ya decía Javier Krahe en una de sus letras (“Ciencias ocultas”) que “...siendo sólo fea, siendo sólo arpía, nunca serás bruja de categoría”.

68 Lo siento, no me da la imaginación para más ni me apetece escribir demasiado. Tal vez a Vd. le pasen por la cabeza nombres más ocurrentes. Por otra parte, muchas veces hace falta crear directorios de usar y tirar. No es mala idea llamarlos cosas como X, Y, o Z, que no son proclives a la confusión y con las que apenas hay que teclear. Al hilo de esto, váyase preparando para que hablemos un poco más adelante de la organización del disco duro, un asunto cuya importancia es difícil de exagerar.

```
C:\EJEMPLO\>MD X_  
C:\EJEMPLO\>MD Y_  
C:\EJEMPLO\>MD Z_
```

Y ya vale con tres más, no hay que pasarse. Vamos a movernos por ellos, que también sabemos:

```
C:\EJEMPLO\>CD X_
```

y se nos ha vuelto a acabar la cuerda, porque una vez que hayamos entrado en C:\EJEMPLO\X\ no es fácil que sepamos salir solitos. ¿Qué hacer?. Tal vez a Vd. se le ocurra que “CD C:\EJEMPLO” debería llevarnos atrás. Y tiene razón. Y a partir de aquí podríamos ir entrando y saliendo de uno y otro:

```
C:\EJEMPLO\X\>CD C:\EJEMPLO_  
C:\EJEMPLO\>CD Y_  
C:\EJEMPLO\Y\>CD C:\EJEMPLO_  
C:\EJEMPLO\>CD Z_  
C:\EJEMPLO\Z\>CD C:\EJEMPLO_  
C:\EJEMPLO\>CD C:\_  
C:\>CD EJEMPLO_
```

Tras semejante borrachera, volveríamos a estar en el punto de partida, es decir en C:\EJEMPLO\. Tal vez, si mira con atención las órdenes anteriores, le parezca que hay cierta sutil diferencia entre “CD C:\EJEMPLO” y “CD X”, por poner algo. En efecto, en la primera indicamos explícitamente el nombre completo del directorio, en tanto en la segunda damos sólo su nombre, sin más, siendo LR-PeCé el que debe completarlo a base de anteponerle la ubicación actual. Puede también que nos parezca que hay demasiado “EJEMPLO” para tan poco movimiento. Su sagacidad, una vez más, debe verse recompensada diciéndole que de nuevo tiene razón. ¡Da gusto trabajar con gente tan inteligente!. El hecho es que lo de ir hacia atrás, al directorio anterior, o si lo desea, inmediatamente superior, conocido habitualmente por “directorio padre” es una operación tan frecuente que admite una forma abreviada que es la que de verdad se emplea casi siempre. Entre otros motivos porque de este modo, como vamos a ver, no es necesario saber el nombre del directorio de partida, cosa de la que no siempre nos acordamos. ¿Que podríamos mirar el “prompt”? Sí, desde luego, pero ni eso va a hacer falta.

La cuestión es que hasta ahora hemos estado trabajando a ciegas. El contenido del directorio actual es algo que el ordenador puede mostrarnos a nada que se lo pidamos adecuadamente, y no hay que malgastar esfuerzos en intentar acordarse de qué tenemos en cada directorio, cosa por otra parte imposible. Si “COPY” era una orden muy apropiada como paradigma del funcionamiento del ordenador, la más popular sin duda es la que presentamos a continuación: “DIR”. Abreviatura del inglés “Directory”, o de la traducción castellana “directorio”, sin ir más lejos, su cometido es el de obligar a LR-PeCé a mostrarnos el contenido de un directorio. Esquemáticamente es algo así como:

```
C:\>DIR NOMBRE-DIRECTORIO /PARAMETROS
```

pero debemos indicar que tanto el nombre del directorio como los parámetros son optativos. Si no indicamos un nombre de directorio, ni parámetros siquiera, tal como en el ejemplo siguiente:

CAPÍTULO 7. PeCés: MS-DOS/Windows.

```
C:\EJEMPLO\>DIR_
```

LR-PeCé, como siempre, trabajará en el directorio actual y nos mostrará el contenido de C:\EJEMPLO\, que es donde nos habíamos quedado. Lo que veremos en nuestro caso será algo así como

```
C:\EJEMPLO\>DIR_
El volumen de la unidad C es GIGA_1
Directorio de C:\EJEMPLO
.      <DIR>    31/08/95  3:59
..     <DIR>    31/08/95  3:59
X      <DIR>    31/08/95  3:59
Y      <DIR>    31/08/95  3:59
Z      <DIR>    31/08/95  3:59
      5 Archivo(s) 182247424 bytes libres
```

Expliquémoslo. DIR nos muestra el contenido del directorio y aprovecha además la ocasión para informarnos del nombre del disco en que estamos trabajando y darnos un resumen que, entre otros datos interesantes, incluye uno de suma importancia: el espacio libre que nos queda en el disco en cuestión. Si miramos atentamente lo anterior veremos que, junto a cada fichero o directorio existente (en este caso sólo directorios), nos indica asimismo si son lo uno o lo otro, y la fecha y hora en que se crearon. Los directorios se indican con un “<DIR>” a continuación del nombre, y los ficheros con el tamaño que tengan, en bytes. Como por el momento no tenemos ficheros, pues no aparecen. En cuanto a lo del nombre del disco, la cuestión es que, aparte de “C:”, o “A:” o lo que sea, podemos ponerles los nombres que queramos a nuestros discos. La unidades también se denominan “volúmenes” en jerga, y se habla en consecuencia del “nombre del volumen”. No es que tenga demasiado interés la cosa, a la hora de usar MS-DOS, por lo que lo dejaremos estar.

Bien, si seguimos observando atentamente lo anterior, veremos dos directorios de nombre un tanto raro. Ni más ni menos que “.” y “..”. ¿Qué quiere decir eso?. Hemos aquí enfrentados a las abreviaturas que buscábamos. “.” indica el directorio actual. Existe y por lo tanto saldrá en el listado. Y “..” indica el directorio padre, el inmediatamente superior, que también existe a este nivel aunque no vamos a explicar porqué, y que por lo tanto también aparecerá. Bástenos con saber que si le decimos a LR-PeCé “CD ..”, no tendrá inconveniente alguno en devolvernos al directorio inmediatamente superior. La cosa resulta sumamente práctica, como puede verse si rehacemos el precedente ejemplo de desplazamientos estúpidos utilizando las habilidades tan recientemente adquiridas:

```
C:\EJEMPLO\X\>CD .._
C:\EJEMPLO\>CD Y_
C:\EJEMPLO\Y\>CD .._
C:\EJEMPLO\>CD Z_
C:\EJEMPLO\Z\>CD .._
C:\EJEMPLO\>CD .._
C:\CD EJEMPLO_
```

Y resulta ahora cuando menos tolerable.

Volviendo a “DIR”, tal vez se acuerde de que hace ya un montón de páginas introdujimos subrepticiamente y sin decir si valían o no para LR-PeCé, cosas como “DIR /W” y “DIR /P”.

Aclaremos inmediatamente que sí que valían y que los efectos de añadir estos parámetros a tan popular orden no son sino los que entonces indicamos, por lo que ahora no lo vamos a repetir. ¿Por qué?. Por una razón muy sencilla. No quiero perder el tiempo explicando parámetros. Ni hacérselo perder a Vd., querido LAO, ya que tenemos asuntos mucho más importantes entre manos. Con un solo tiro auténticamente genocida nos los cargaremos a todos ellos, para todas las órdenes de MS-DOS. Disparemos: el único parámetro que Vd. debe necesariamente conocer es “/h”, abreviatura de “help”, como todo el mundo sabe “ayuda” en inglés, (si está usando MS-DOS, aunque Novell DOS 7 también lo admite, la equivalencia es “/?”, abreviatura universal de “no tengo ni idea de qué hacer ahora”). Y es así pues ese, y no otro, es el parámetro de ayuda que hará que LR-PeCé nos muestre un resumen de la orden en cuestión, especificando sintaxis, parámetros que admite, y efecto de cada uno de ellos. Y no hay más que decir, aparte de poner un ejemplo. Si le decimos a nuestro genio, a estas alturas ya ligeramente hasta el gorro de nosotros, por pesados, algo así como:

```
C:\EJEMPLO\>DIR /H_
```

lo que obtendremos será el más o menos ya visto, :

DIR	Muestra los archivos de un directorio.
Sintaxis:	DIR /?
	DIR [espcom] [/L /2 /W] [/P /N] [/A /D /S] [/C /R]
/?	muestra un texto de ayuda
espcom	archivos para mostrar (admite comodines)
/A	muestra todos los archivos
/C o /R	convierten las opciones actuales en los valores por omisión
/D	muestra archivos sin atributo del sistema (por omisión)
/L	formato completo: tamaño, fecha y hora (por omisión)
/2	análogo al anterior, pero los archivos aparecen en dos columnas
/N	permite volver a la opción de paginación por omisión
/P	hace una pausa al final de cada página. La opción por omisión es no hacer pausa
/S	muestra archivos con atributo del sistema
/W	formato apaisado: sólo muestra los nombres de archivos y directorios
ninguna	sin opciones, muestra los archivos utilizando las opciones por omisión
Ejemplo:	DIR /C /W

información más que suficiente para cualquier necesidad y que está siempre a su disposición. O sea que nada, cuando quiera conocer posibilidades de una orden cualquiera, añádale un “/h” (o “/?”) y lea la pantalla. Recuerde la regla de buscar la información en primer lugar en la pantalla del ordenador, y que, al menos en lo referente a las órdenes MS-DOS, LR-PeCé sabe un montón de cosas que Vd. puede preguntarle con bastante comodidad.

Tras haber exterminado a los parámetros, liquidemos el otro punto pendiente que nos queda desde hace un rato. ¿Cómo decirle a LR-PeCé que trabaje en un directorio que no sea el actual?. Ya debería estar claro. Bastará con que especifiquemos el nombre completo del origen o destino de la orden a realizar.

Si quiere verlo de otra forma, podemos decir que a estas alturas de la evolución de MS-DOS, nuestro genio es lo suficientemente listo para suponer bastantes cosas (sumamente sencillas y en un ambiente muy controlado, eso sí). Cuando omitamos detalles, el los intentará suponer y funcionará con las opciones por defecto, por ejemplo trabajando en el directorio

CAPÍTULO 7. PeCés: MS-DOS/Windows.

actual. Cuando se los demos todos, no supondrá nada y hará exactamente lo que le digamos. De esta forma “CD” nos llevará desde cualquier directorio a cualquier otro, en un solo paso, “COPY” copiará cualquier cosa, desde cualquier sitio a cualquier otro, estemos donde estemos, y “DIR” nos mostrará el contenido de cualquier directorio, estemos o no en él. Si practicamos un poco:

```
C:\EJEMPLO\>DIR C:\_
```

nos mostrará el contenido del directorio raíz, donde tendremos por ahora nuestro CONFIG.SYS y CONFIG.CPY entre otros, mientras que

```
C:\EJEMPLO\>COPY C:\CONFIG.SYS C:\EJEMPLO\X\CONFIG.CPY_
```

creará una nueva copia de CONFIG.SYS en el directorio C:\EJEMPLO\X\. Y si por ventura volvemos al directorio raíz mediante:

```
C:\EJEMPLO\>CD.._
```

podremos, de un solo golpe y diciéndole a LR-PeCé:

```
C:\>CD C:\EJEMPLO\Z\_
```

pasar directamente a un territorio todavía no hollado por pie alguno. Una vez en él, es posible volver directamente al directorio raíz con otra abreviatura. A saber:

```
C:\EJEMPLO\Z\>CD \_
```

Y terminemos con esto recomendando que se lea la nota al pie⁶⁹.

Antes de volver a ocuparnos de “COPY”, a la que todavía le vamos a sacar bastante más partido, convendría que nos deshiciéramos de un último estorbo, de otro de esos conceptos fundamentales que deben dominarse. Ahora que andamos a vueltas con taquigrafías diversas podría ser el mejor momento. Por retomar a la vez otro hilo también reciente, recuerde lo que dijimos tiempo ha sobre los textos de ayuda que el ordenador podía mostrarnos. Su único inconveniente era que había que entenderlos. A estas alturas, con todo lo que hemos

69 Aunque la cosa depende en gran medida de la versión de DOS que estemos usando (o más exactamente del intérprete de órdenes con que trabajemos), LR-PeCé es capaz de completar las órdenes de diversas formas, no sólo suponiendo el directorio en que debe tener lugar una operación. También es capaz de apanárselas por su cuenta ante órdenes de copia en las que no se especifica el nombre del fichero de destino, y a veces hasta aquellas en las que no se detalla el origen, aunque no pondremos ejemplos de estas últimas. Si por ejemplo le decimos “C:\EJEMPLO\>COPY C:\CONFIG.CPY C:\EJEMPLO\X_”, la copia se realizará y C:\CONFIG.CPY se copiará, con ese mismo nombre (LR-PeCé supone que queremos seguirlo usando), en el directorio C:\EJEMPLO\X\. Hasta puede abreviarse todavía más. Así “C:\EJEMPLO\> COPY C:\CONFIG.CPY_” a secas, copiará CONFIG.CPY al directorio actual, y acabaremos con una copia en C:\EJEMPLO\, es decir con el fichero C:\EJEMPLO\CONFIG.CPY. Esta especie de taquigrafía viene muchas veces al pelo, pero exige, como cualquier virtuosismo, estar habituado a los conceptos en ella involucrados. Si no se quiere liar al principio, teclee un poquito más y especifíquelo todo, con lo que se evitará alguna sorpresa desagradable. Ya se irá soltando con la práctica.

avanzado, el texto que LR-PeCé nos ha mostrado tan obedientemente ante nuestra invocación con “DIR /H” debería ser casi totalmente inteligible. Si no lo ha leído con atención, hágalo ahora. Y reconozca que aparte de la forma en que se indica la sintaxis de la orden (por cierto, los corchetes indican siempre cosas opcionales) una sola cosa queda fuera de nuestro alcance. Tras “espcom”, (que también por cierto es el nombre abreviado que aquí le dan a nuestros “nombres de fichero”) aparece la desconcertante frase “admite comodines”. ¿Qué son los comodines, pregunta?. Ahí quería yo llegar.

Una de las mayores ventajas del uso del teclado para mandarle cosas al ordenador consiste en que es bastante sencillo, a nada que se pongan en juego una serie de convenciones, hacer que una orden afecte no sólo a un fichero, sino a todo un grupo de ellos. Por ejemplo, está bastante claro que “CONFIG.SYS” y “CONFIG.CPY”, tienen nombres bastante parecidos. Y si tuviéramos otros dos que se llamaran “HOLA.TXT” y “ADIOS.TXT”, también se parecerían. Y hasta “ADIOS_1.TXT” y “ADIOS_2 .TXT”. Bien, pues los comodines se encargan de que sea posible referirnos de un solo golpe a un conjunto de ficheros con nombres parecidos.

Los comodines, o “wildcards” en inglés, nombre que también se usa mucho, no son más que caracteres especiales que pueden incluirse en un nombre modelo de un fichero y que se sustituyen en el momento de ejecutar la orden que sea por todas sus alternativas posibles. Permiten usar “nombres patrón” para citar de un solo golpe a un montón de ficheros. En MS-DOS sólo existen dos, “?” y “*”. El primero indica que en esa posición puede aparecer cualquier carácter válido, pero sólo uno, y el segundo que cualquier número legal de caracteres legales nos sirve en el sitio en que lo coloquemos.

Por poner un ejemplo, que hace falta, hay que reconocerlo, podríamos nombrar los dos primeros ficheros anteriores mediante la abreviatura “CONFIG.*”, o, si lo preferimos “CONFIG.???”. Los dos siguientes se ajustarían al patrón “*.TXT”, y los dos últimos a “ADIOS_?.TXT”. Con tan pocos comodines que aprender⁷⁰, no es difícil cogerles el tranquillo enseguida, y todo usuario recién llegado suele apresurarse a mandarle a LR-PeCé cosas como:

```
C:\>COPY CONFIG.* C:\EJEMPLO\Z\_
```

que copiaría CONFIG.SYS y CONFIG.CPY al directorio citado como destino, o hasta:

```
C:\>COPY *.* C:\EJEMPLO\Y\_
```

que haría una copia de todos los ficheros no ocultos⁷¹ del directorio raíz a C:\EJEMPLO\Y⁷².

70 Otros sistemas operativos admiten más, y son más potentes pero también más complicados. Es posible en ellos decirle al ordenador cosas tales como “copia todos los ficheros que tengan un nombre cuyo tercer carácter sea una a, una b, o una x, pero ninguno más” o “copia todos los ficheros que tengan un nombre cuya segunda letra no sea una c”. La mayor parte de las veces, ya vale con los dos comodines de MS-DOS, por lo que no es necesario lamentarse demasiado.

71 No se acordaba de los atributos ¿verdad?. Pues están ahí para algo. Los ficheros ocultos, y ya que estamos, también los de sistema, son invisibles en condiciones normales y no se copiarán, ya que MS-DOS no nos deja tocarlos. Tampoco DIR nos los mostrará. Eso sí, si por casualidad

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Y ya que andamos con abreviaturas, abreviemos y pasemos a otra cosa, que con lo anterior ya vale. Volvamos a “COPY” para seguir sacándole partido. Hasta el momento nos hemos limitado a hacer copias de ficheros dentro de una misma unidad. Pero es exactamente igual de fácil copiar un fichero desde el disco duro, pongamos el C:, a un disquete, pongamos en A:. Raudo, veloz, y cada vez más suelto con su genio particular, todo LAO que se precie se precipitará al teclado a escribir algo así como:

```
C:\>COPY CONFIG.SYS A:\CONFIG.CPY
```

con lo que repetiría la primera operación de copia que hicimos pero dejando ahora CONFIG.CPY en el disquete. Eso si previamente ha colocado un disquete en el lector. ¿Que se le había olvidado?. Pero bueno, no debería hacer falta decirlo. ¿Que lo ha metido y lo único que le sale es algo así como “Error de unidad leyendo en la unidad A. Abandonar, Reintentar, Fallar ?” y no sabe qué hacer?. Henos ante uno de los mensajes de error de MS-DOS y ya de paso ante otra cuestión que tampoco debería hacer falta decir. Y es que recuerde que los disquetes, o en general cualquier disco, deberán estar formateados, tener grabada su correspondiente estructura para poder trabajar bajo MS-DOS. El disquete que Vd. tan precipitadamente ha metido en el lector no lo está, y para MS-DOS es como si no existiera. Antes de subsanar tan lamentable descuido, aclaremos lo de los mensajes de error.

El que ha aparecido antes es de los más frecuentes, y nos indica que la operación que intentamos realizar ha topado con un problema, no siendo posible leer la unidad en cuestión. Lo de “Abandonar, Reintentar, Fallar?”, quiere decir que MS-DOS nos deja que le aclaremos qué deseamos que haga en tan triste situación. Lo anterior no es más que un menú, en una de las presentaciones abreviadas que en su momento expusimos (¿ve como muchas cosas sirven para más que llenar páginas y páginas?). Las letras en mayúsculas son las que debemos pulsar para que LR-PeCé abandone, reintente o considere fallida la operación. “R”, reintentar, nos permite intentar corregir el problema sobre la marcha. Podríamos meter un disco ya formateado, pulsar “R”, y entonces LR-PeCé completaría la operación. Tanto “A” como “F” la darán por terminada y volverán al “prompt”, para que pidamos algo con más sentido la próxima vez.

¿Y si sólo tengo un disquete, el que ha provocado el error?. ¿Qué hago?. Pues nada hombre, hay que formatearlo. Se nos presenta una ocasión magnífica para añadir una orden más a las que ya conocemos. Dejaríamos el disquete en el lector, pulsaríamos la A, saldría de nuevo el “prompt”, y escribiríamos la adaptación correspondiente de la orden que presentamos a continuación, de nuevo en forma esquemática. Se trata de “FORMAT”, cuya traducción castellana podría ser “formatear”.

```
C:\>FORMAT UNIDAD /PARÁMETROS_
```

que en nuestro caso quedaría simplemente:

tenemos alguno marcado como de “sólo lectura” (atributo “r”) se copiará sin problemas.

- 72 No creo que deba terminar con los “wildcards” sin advertir que no es raro encontrar ciertas incoherencias en su funcionamiento. Si Vd. pretende realizar operaciones potencialmente peligrosas con ellos, especialmente si los usa para borrar selectivamente un grupo de ficheros, no estará de más que se asegure mediante un “DIR” de que los “wildcards” y Vd. están de acuerdo en el grupo de ficheros a manipular.

```
C:\>FORMAT A:_
```

Es probable que, tras escribir lo anterior y pulsar CR, nos enfrentáramos a una serie de instrucciones y advertencias en pantalla que, una vez seguidas y tras esperar alrededor de dos minutos a que la cosa acabara, ya que le cuesta lo suyo, condujeran a que termináramos con un disquete formateado y ya utilizable. Sobre el que podríamos situarnos en cualquier momento sin más que escribir su denominación como unidad. Así

```
C:\>A:_
```

nos conduciría a

```
A:\>_
```

y si allí hacemos

```
A:\>C:_
```

volveremos a C:⁷³.

Retomemos el hilo diciendo que es probable, pero no seguro, ya que también puede ocurrir que nos aparezca un mensaje de error aún no presentado pero también sumamente frecuente. Podríamos vernos ante algo así como: “No se encuentra el fichero”. ¿Pero cómo?. ¿De qué fichero habla?. No estamos copiando nada, ni hemos llamado a fichero alguno. ¿Qué demonios le pasa a COMMAND.COM que no es capaz de entender una orden expresada con total propiedad?. De nuevo, ahí quería yo llegar.

La cuestión es que COMMAND.COM es bastante limitadito, el pobre. Sabe cómo hacer DIR, CD, MD, COPY, y algunas cosas más, pero FORMAT no está entre sus habilidades. Entonces, ¿cómo es que FORMAT es una orden MS-DOS válida, si COMMAND.COM no la sabe ejecutar?. El truco de verdad, lo que hace que MS-DOS sea un sistema operativo y no un mero programa para gestionar ficheros con dificultad, está aquí. Su vocabulario no está limitado a un número predeterminado de órdenes. Es extensible. Se establece un mecanismo, sumamente simple, mediante el cual pueden añadirse a MS-DOS nuevos programas, que, invocados con nuevas palabras, llevarán a cabo nuevas tareas.

Las órdenes que sabe hacer solo, las que ya están incluidas en COMMAND.COM, se llaman “órdenes internas”. Son pocas y muy básicas, sólo aquellas que se emplean con mayor frecuencia. Todo lo que se salga de ahí, debe añadirse. ¿Dónde y cómo, pregunta?. Por supuesto, y ya que MS-DOS es un sistema operativo basado en disco, mediante los correspondientes programas en disco. Además de las órdenes internas, COMMAND.COM especifica un mecanismo de ejecución para las otras, que en justa consecuencia se llaman “órdenes externas”. Si yo le doy una orden que no se encuentra entre las que él conoce, buscará en el disco un programa que se llame igual. ¿Y cómo sabe COMMAND.COM, o nosotros mismos sin ir más lejos, qué ficheros contienen programas y cuales no?. Porque nos empebamos a barruntar que alguna diferencia debe haber.

73 Si Vd. se mueve entre varias unidades, observará que MS-DOS recuerda el directorio en el que se encontraba la última vez que estuvo allí. Así, si va desde C:\DOS\ a A:, cuando vuelva al disco duro mediante “A:\>C:_”, se hallará de nuevo en C:\DOS\.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

La respuesta es, de nuevo, absurdamente simple. MS-DOS usa las extensiones de los nombres de fichero como hemos dicho que debíamos hacer nosotros. Indica con ellas el tipo de información que contienen. De esta forma, COMMAND.COM supone que cualquier fichero con extensión .COM, .EXE o .BAT contiene programas ejecutables en MS-DOS. Cuando le pasamos una orden que no se encuentra entre las internas, busca un fichero con el mismo nombre y alguna de las extensiones anteriores y, caso de encontrarlo, lo carga en la memoria, le pasa gentilmente el control, y espera a que termine para retomar las riendas del ordenador⁷⁴. ¿Y si no lo encuentra?. Pues nos saca el mensaje de error ese de “Fichero no encontrado”, que quiere decir simplemente que no hay fichero .COM, .EXE o .BAT que se ajuste a la orden que hemos dado.

Una vez más, todo aprendiz de hechicero competente, habrá comenzado a oír el revolotear de una fantasmagórica mosca detrás de su oreja y a pensar que esto le suena a conocido. Ya hemos tratado de los problemas asociados a la búsqueda de ficheros, pero la solución que les dimos no parece suficiente. Por ahora, LR-PeCé busca en el directorio en que nos encontremos o en el que citemos expresamente. Pero cuando se trata de ejecutar programas de utilidad básica, que deberían estar siempre a nuestra disposición, no estaría de más independizar su ejecución de la ubicación que tengamos en un momento determinado, y tampoco conviene tener que teclear nada más que la orden. A ver porqué regla de tres vamos a tener que escribir “C:\PROGS\UTILS\GEN\FORMAT” simplemente para que LR-PeCé sea capaz de localizar el dichoso FORMAT.COM.

Nada, nada. Lo suyo sería que MS-DOS conservara una lista de direcciones donde buscar los programas, de modo que, cuando no indicáramos expresamente la ubicación de uno, mirara en el directorio actual, y si tampoco lo encontrara allí, pudiera tirar de lista y mirar por su cuenta en varios lugares adicionales. No es ni más ni menos que la solución general que hace unas páginas presentamos. Mediante algún mecanismo, tenemos que decirle a LR-PeCé que mire en Zaragoza, y no en toda, sino sólo en Delicias y Las Fuentes, cuando lo que busca no esté en la calle en que en ese momento se encuentre ni se le dé una dirección completa a la que acudir.

Como la tal lista debe estar siempre disponible, tiene bastante sentido que la dejemos en la memoria del ordenador. Un dato en memoria, con un nombre para que se le pueda localizar a su vez, era una “variable”. En MS-DOS, el problema se soluciona, ahora sí completamente, creando una variable que recibe el nombre de “PATH”, traducible como “ruta”, y que contiene la serie de directorios en los que efectuar la búsqueda de un programa cuando haga falta. Ni más ni menos que la ruta a seguir para localizar el dichoso programa que se obstina en no aparecer.

¿Cómo se crea la tal variable?. Por supuesto mediante una orden, que deberá ser necesariamente interna⁷⁵. Y como al diseñar el sistema operativo podemos elegir el nombre de las

74 .EXE es la abreviatura del inglés para “ejecutable”. A su vez, .COM es una abreviatura heredada de los tiempos del CP/M que indica que el tal fichero contiene el programa para añadir una orden (“command”). Y .BAT es la abreviatura de “batch”, que explicaremos más adelante. Y debo puntualizar que ficheros con otras extensiones pueden también contener programas, pero lo dejaremos estar.

75 La justificación de porqué es necesaria una orden interna se deja como ejercicio para el lector.

órdenes, nada más lógico que llamar “PATH” a la que debe gestionar la variable “PATH”. Su formato es:

```
C:\>PATH LISTA_ DE_DIRECTORIOS_
```

y bastará decirle a LR-PeCé algo así como:

```
C:\>PATH C:\DOS;C:\;D:\NWDOS_
```

para que, cuando digamos por ejemplo “FORMAT”, estemos donde estemos, LR-PeCé busque si es necesario el correspondiente fichero en todos los lugares indicados⁷⁶.

Bueno, ya que hemos aclarado porqué al decir “FORMAT”, podemos encontrarnos ante un mensaje de error un poco sorprendente, y completado aún algo más nuestra querida orden “COPY” con la posibilidad de operar entre unidades diferentes, podemos pasar a otras cosas, de nuevo mucho más importantes. Para que “PATH” tenga sentido, y en general para que el trabajo con el ordenador no se convierta en un caos, es necesario que las unidades de almacenamiento con que contemos se organicen adecuadamente⁷⁷. Si queremos que los directorios sirvan para algo, tendremos que usarlos para agrupar en lugares conocidos aquellos ficheros de naturaleza o utilidad similar. Por ejemplo, todos los programas relacionados con el sistema operativo deberían estar en el mismo directorio. Todos los ficheros relacionados con un programa determinado, tal como nuestro procesador de textos “TLW 4.0” deberían también encontrarse en un subdirectorio propio, donde fueran fáciles de localizar. Y por supuesto nuestros datos deberían estar donde queramos, organizados a nuestro gusto, pero ordenados de modo que la estructura de los directorios que nosotros hayamos montado sirva para agruparlos a nuestra conveniencia.

Desgraciada y afortunadamente a la vez, la cuestión de la organización del espacio de almacenamiento queda, es inevitable, bajo la responsabilidad del usuario del ordenador. De Vd. querido LAO. Debe ser cada cual el que lo monte como quiera, a base de emplear juiciosamente la orden “MD”. Tan sólo es posible indicar las convenciones al uso y dar algunas recomendaciones básicas. Vamos con ello.

En cuanto a las convenciones, decir que aparte de los ficheros que necesariamente deben encontrarse en el directorio raíz del disco de arranque, tales como los ya vistos COMMAND.COM y demás, no deberíamos dejar nada más en él, y todo programa, al instalarse, deberá colocarse en un directorio propio, de modo que los ficheros que lo componen queden agrupados y fáciles de localizar y manipular. Los programas de instalación realizan algo más que una mera copia de ficheros y suelen encargarse de ello. Así

Caerá en el examen final.

76 MS-DOS permite también especificar una lista de directorios donde deben buscarse no los programas a ejecutar, sino los datos que deben manipular. La orden correspondiente es APPEND, pero en mi opinión no tiene mayor interés en la actualidad.

77 “¡Organización!. ¡Organización!”, clamaba el personaje de un chiste ligeramente subido de tono. Es una pena no poder incluirlo por dicho motivo en lo que debe ser una lectura apta para todos los públicos, ya que subraya la necesidad de un mínimo de orden cuando se trata de llevar a cabo cualquier tarea.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

MS-DOS crea un directorio llamado justamente “DOS” en el que deja todos los ficheros .COM y .EXE que lo completan. DR-DOS hace lo mismo en un directorio “DRDOS”, y Novell DOS 7 no se aparta del esquema y crea un directorio “NWDOS” donde deja todo lo suyo. Esto permite que, añadiendo una entrada a la lista del PATH, todas las órdenes externas del DOS correspondiente resulten fácilmente localizables. Muchos programas de instalación modifican el PATH de este modo para poder ejecutarse desde cualquier sitio. Hay que precisar que esto último no es estrictamente necesario, y se suele preferir que pregunten al usuario si quiere hacerlo así. En general cualquier programa creará su propio directorio o directorios en el momento de instalarse y dejará allí todo lo necesario para funcionar correctamente. TLW 4.0 podría crear, como ya vimos en un capítulo anterior, un directorio “TLW4”.

Con este tipo de organización, convencionalmente aceptada, la ejecución de un programa en MS-DOS consiste siempre como máximo en dos pasos. Nos vamos al directorio que sea mediante “CD”, y una vez allí ejecutamos el programa, cuyo nombre corresponderá a un fichero .EXE (o .COM o .BAT) de ese directorio. No es raro que el nombre del programa a ejecutar y el del directorio en que se instala coincidan, o al menos se parezcan. Para ejecutar “TLW 4.0” haríamos algo así como:

```
C:\>CD TLW4_  
C:\TLW4\>TLW4_
```

con lo que TLW4.EXE (o .COM o .BAT), que debería encontrarse allí, se cargaría en memoria y se ejecutaría, y nosotros tendríamos un procesador de textos a nuestra disposición.

En cuanto a las recomendaciones, recordar una vez más que es necesario, absolutamente, que se tome la molestia de organizar sus datos. Le garantizo que resulta muy frustrante no poder localizar un fichero en el que se sabe con seguridad que estuvimos trabajando la semana pasada y que necesitamos con urgencia. Y lo de localizar nuestros datos es sólo una muy pequeña parte del total. Las operaciones relacionadas con el mantenimiento y la seguridad de los datos producidos con el ordenador son imposibles o muy difíciles de realizar sin una organización adecuada. Sin organización, el ordenador resulta improductivo. Hagamos un pésimo juego de palabras y digamos que el ordenador necesita orden para funcionar.

Lo de la productividad, ya le sonará, hemos dicho que lo dejábamos para otro capítulo. Que así sea. Baste con esto por ahora, pues seguimos tal vez más interesados por el momento en seguir aprendiendo cosas de MS-DOS. Volvamos a la orden “COPY”. “¿Otra vez?”, tal vez exclame. Pues sí, ya dijimos que la cosa daba para mucho. Pero casi hemos acabado, impaciente LAO. Y un hechicero debe tener paciencia, o sea que a ver si se corrige.

Empezamos nuestro recorrido por “COPY” hablando de que resultaba un modelo razonable del funcionamiento completo del ordenador. ¿Qué pasa si lo que queremos es crear de verdad un fichero?. No copiarlo, sino hacerlo nuevo, metiendo en él lo que nosotros queremos, escrito directamente desde el teclado. ¿Y qué pasa si queremos verlo en pantalla o sacarlo a papel?.

Recuerde que COPY podía coger y dejar datos desde y hacia cualquier fichero o dispositivo. Abramos nuestra mente y pasemos a hacer algo más que copiar desde y hacia vulgares

ficheros. Pasemos a enviar cosas desde el teclado a un fichero o a mandar ficheros a la pantalla.

Suponiendo que estemos en C:\EJEMPLO\ (ya debería ser capaz de colocarse en él solito si no está allí en este momento), y que queramos crear el fichero de texto "C:\EJEMPLO\HOLA.TXT", bastará para ello con conjurar a LR-PeCé mediante el siguiente hechizo:

```
C:\EJEMPLO\>COPY CON: HOLA.TXT_
```

El único resultado aparente de tal invocación consiste en que el cursor pasa a la siguiente línea de la pantalla. No ponerse nerviosos. Escribamos lo que queramos meter en el fichero. Por ejemplo, y puesto que se llama HOLA.TXT, algo así como:

```
Hola, Luis Ricardo.
```

```
-
```

ya que tampoco hay que complicarse mucho la vida. "COPY" nos muestra cómo va quedando el fichero de texto mientras lo vamos escribiendo. Observe que el cursor está en la segunda línea. Cuando escribimos "Hola, Luis Ricardo." y le damos a CR, el único efecto es el de pasar el cursor a la línea siguiente. CR, ahora un carácter más, se almacena en el fichero de texto y "COPY" me lo muestra como lo que convencionalmente es: un desplazamiento de carro.

¿Cómo hacemos para terminar?. Ya que nos podemos pegar horas escribiendo. Los caracteres de control vuelven a venir al rescate. Debe saber que, en MS-DOS, el carácter de control CTRL+Z (^Z o ASCII 26) actúa como un marcador de fin de fichero cuando de ficheros tipo texto se trata. Bastará por lo tanto con que pulsemos CTRL+Z y le demos a CR a continuación. LR-PeCé leerá la cosa como que ya está bien y quizá nos muestre un mensaje diciendo que se ha copiado un archivo.

¿Que queremos comprobar que de verdad la cosa ha funcionado?. Vaya, vaya, es Vd. realmente desconfiado, querido LAO. Por supuesto puede cerciorarse haciendo un "DIR". En la lista vería ahora una entrada más o menos parecida a:

```
HOLA      TXT      21 24/09/95 14:06
```

Aunque con esto valdría, espero que esté de acuerdo conmigo en que un vulgar "DIR" no está ya a nuestra altura. ¿Por qué no mirar dentro del fichero, hacer que LR-PeCé nos muestre qué es lo que de verdad ha guardado bajo ese nombre?. Usemos de nuevo "COPY", ahora en la forma:

```
C:\EJEMPLO\>COPY HOLA.TXT CON:_
```

y lo que veremos será, ni más ni menos, que

```
Hola, Luis Ricardo.  
1 Archivo(s) copiado(s)
```

Al ser CON: un dispositivo de entrada-salida, admite tanto que copiemos desde él a un fichero (lee el teclado) como al contrario (envía el fichero a la pantalla). Si estamos tan orgullosos que deseamos obtener una copia impresa y tenemos una impresora conectada en la puerta paralelo de nuestro PeCé, podríamos lograr tan simpár hazaña sin más que ordenar:

CAPÍTULO 7. PeCés: MS-DOS/Windows.

```
C:\EJEMPLO\>COPY HOLA.TXT LPT1:_
```

Nuestro cursillo acelerado de palabras mágicas para poner a LR-PeCé a nuestros pies está llegando a su fin. Describiendo “COPY” con un mínimo detalle hemos tenido ante todo ocasión de presentar un buen número de cuestiones básicas (búsqueda de ficheros, atributos, dispositivos, comodines...) que son las que de verdad importan. Porque lo de describir las distintas órdenes es algo que no merece la pena. Los manuales en línea para MS-DOS están ahí para algo, a nada que Vd. esté trabajando en un PeCé equipado con MS-DOS 6.2, DR-DOS 6 o Novell DOS 7⁷⁸.

Hace no demasiados párrafos, nos quitamos de encima los parámetros liquidándolos de un solo golpe. Cual si fuéramos Pedro Camacho en las postrimerías de su locura y estuviéramos intentando reconducir a trancas y barrancas nuestras radionovelas⁷⁹, provoquemos una nueva extinción en masa liquidando todas las órdenes restantes de MS-DOS presentando a aquella que nos va a llevar al manual en línea, donde encontraremos mucho más de lo que aquí tiene sentido incluir.

Para acceder a él, bastará que le digamos a LR-PeCé:

```
C:\EJEMPLO\>HELP80_
```

y pulsemos CR. A partir de este momento, lea la pantalla y siga las instrucciones.

Sólo vamos a salvar de la extinción total a aquellas órdenes relacionadas con cosas que en su momento prometimos decir cómo hacer y aquellas otras que necesitamos para explicar conceptos adicionales. Para que no se diga que no cumplimos con lo anunciado. Pero no serán muchas. MS-DOS tiene un repertorio bastante más amplio.

Acabemos el apartado con dos ejemplos. El primero va a consistir en ver cuantas órdenes más nos podemos encontrar en el manual en línea. Aparte de las internas que nos hayamos dejado, cualquier orden externa será un fichero .EXE, .COM o .BAT sito en el directorio C:\NWDOS (o C:\DOS si usamos MS-DOS). Bastará con decir:

```
C:\EJEMPLO\>DIR C:\NWDOS \*.BAT_  
C:\EJEMPLO\>DIR C:\NWDOS \*.COM_  
C:\EJEMPLO\>DIR C:\NWDOS \*.EXE_
```

y leer las listas correspondientes para que tengamos una idea bastante concreta del tajo que nos queda⁸¹. A título de ejemplo, mostremos la correspondiente a “*.EXE”, en un formato un poco especial que más adelante diremos cómo hemos obtenido:

78 Siento desconocer la situación en PC-DOS, la versión del DOS producida directamente por IBM.

79 Uno de los personajes de la novela “La tía Julia y el escribidor” de Mario Vargas Llosa. ¿Que no la ha leído?. ¿Y a qué espera, si puede saberse?.

80 Estrictamente hablando, “HELP” es la orden MS-DOS. En Novell DOS 7 es “DOSBOOK”, pero el más obvio “HELP” es también admisible.

81 Para evitar líos si Vd. se toma las cosas demasiado al pie de la letra, lo que no resulta nada conveniente en un texto como éste, advertir que en realidad no todo fichero .EXE, .COM o

```
C:\EJEMPLO\>DIR C:\NWDOS\*.EXE /4

Volume in drive C is GIGA_1
Directory of c:\nwdos\*.exe
append.exe      4K   attrib.exe    13K   chkdsk.exe     59K   cursor.exe     8K
dconvr2.exe    97K   debug.exe     40K   delpurge.exe  28K   delwatch.exe  19K
diskmap.exe   11K   diskopt.exe  159K   dosbook.exe   812K  doskey.exe    8K
dpmi.exe       7K   dpms.exe     47K   e.exe         22K   emm386.exe   176K
exe2bin.exe   10K   fbwx.exe     512K   fbx.exe       502K  filelink.exe 103K
find.exe      14K   join.exe      9K    lock.exe      36K   mem.exe       17K
move.exe      20K   netwars.exe  45K   nlsfunc.exe   3K    nwcachex.exe 36K
nwcdex.exe    20K   password.exe 14K   pnunpack.exe  16K   preview.exe   21K
rendir.exe     8K   replace.exe  15K   sconvert.exe  54K   screxec.exe   86K
script.exe    21K   sdefrag2.exe 221K  sdres.exe     44K   sdscan.exe   279K
serno.exe     10K   setup.exe    22K   setup2.exe   529K  setver.exe    19K
share.exe     5K   sort.exe     11K   stacker.exe   37K   subst.exe     8K
taskmgr.exe   47K   touch.exe    14K   undelete.exe 102K  uninstal.exe  32K
unstack2.exe  53K   winswap2.exe 27K   wsdres.exe   128K  wsdscan.exe  669K
xcopy.exe     17K   xdel.exe     14K   xdir.exe      23K

5,484,200 bytes in 59 files and 0 dirs
5,947,392 bytes allocated
93,945,856 bytes free
```

En ella puede encontrar los ficheros que implementan órdenes como ATTRIB, que le permitirá visualizar y modificar los atributos. Casi me permito recomendarle que sea la primera que consulte en su manual en línea y con la que experimente en cuanto pueda, pues el concepto es importante y no nos metimos en su momento en como manipularlos. También XDIR tiene interés para usuarios de Novell DOS 7 o DR-DOS 6, pues es un tipo de directorio que, entre otras mejoras, se toma la molestia de mostrar los atributos.

Y el segundo va a intentar remachar que todo lo relacionado con dispositivos, formas raras de “COPY” y demás temas “avanzados” están ahí para algo más que para hacer bulto. Si Vd. ya estaba versado en MS-DOS habrá estado pensando todo el rato que hay formas más fáciles y mejores de visualizar un archivo de texto en la pantalla (para eso existe una orden llamada “TYPE”) o de imprimirlo (orden “PRINT”) y hasta de crearlo (orden “EDIT”). Cierto. Pero para que vea que los conocimientos básicos de verdad pueden resultar sumamente útiles, expongamos un caso práctico.

Digamos que Pablo, el experto del capítulo 5, tiene montado en su ordenador, y sabe usar decentemente, un buen programa de diseño gráfico en Windows⁸². Pero no tiene una impresora en color. En cambio, su amigo Ricardo decidió entrar en esto de la informática a todo lujo, y entre lo que compró se encuentra el dichoso chisme. Por poner un ejemplo hipotético una Hewlett-Epson modelo Stylus 850C⁸³. Por motivos que no vienen al caso,

.BAT tiene sentido como orden DOS. Una única orden puede estar implementada mediante varios ficheros de programa.

82 Es sumamente probable que un programa decente de diseño gráfico para PeCés exija Windows.

83 Ejemplo totalmente hipotético. Epson y Hewlett-Packard están en este momento (septiembre de 1.995) disputándose el mercado de las impresoras color con tecnología de inyección de tinta.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Pablo necesita presentar un trabajo a todo colorín. Le pide por favor a Ricardo que le deje usar su impresora, y éste, por supuesto, le dice que no faltaba más.

Llevar tan simple propósito a la práctica exige concretar el procedimiento a seguir. La posibilidad A consistiría en que Pablo iría a casa de Ricardo, le desmontaría la impresora, la llevaría como pudiera hasta su casa (coche, taxi, autobús, o a pie), la conectaría a su ordenador, la usaría, y desharía lo hecho para acabar como al principio, con la Hewlett-Epson en casa de su legítimo propietario, quien de nuevo podría seguir infrutilizándola a su placer. No parece muy elegante.

La posibilidad B consistiría en que Pablo preparara el trabajo en su casa, con su programa, y cuando terminara fuera a ver a Ricardo no sólo con el disquete que lo contuviera, sino también con todos los necesarios para instalar el programa de diseño en su ordenador. Con esto podría terminar allí su trabajo, y acabar imprimiéndolo a color. Aunque algo menos chapucera que la posibilidad A, ante todo por el menor peso de los elementos a transportar entre ambos domicilios, esta alternativa puede ser irrealizable. Pongamos por ejemplo que el programa en cuestión necesita 40 Megabytes de espacio en disco duro para instalarse y al menos 8 Megabytes de RAM para poderse ejecutar⁸⁴, y que el ordenador de Ricardo no cumple alguno, o ninguno, de los dos requisitos. Además, incluso en el caso de poder llevarse a cabo, la instalación del programa puede ser cosa de una hora si todo va bien, y abriría además que retirarlo al terminar. Nos echamos al coleteo un par de horas adicionales de trabajo sin comerlo ni beberlo⁸⁵. Nada, nada, hay que buscar otra cosa.

La solución más adecuada se basa en realizar todo el trabajo, incluida la impresión, en el lugar en que tenemos el programa ya instalado, llevar a casa de Ricardo tan sólo un mínimo de disquetes con los datos a imprimir, y pasarlos allí a papel en su fastuosa impresora a color mediante algo que seguro que se puede hacer en su ordenador incluso sin entrar en Windows. Mediante un "COPY" desde MS-DOS.

Si Vd. quiere detalles, los pasos a seguir serían tal que así. Pablo configuraría su ordenador (Windows, para ser exactos) exactamente igual que si tuviera en su casa la Hewlett-Epson Stylus 850C. Es decir, instalaría los drivers de la impresora y configuraría su programa de diseño para que imprimiera en ella. ¿Y cómo va a poder imprimir, si en realidad la maldita impresora sigue en casa de Ricardo?. El truco consiste en "conectar" la impresora no a la puerta paralelo, donde Pablo tiene otra en blanco y negro, sino a un fichero, cosa que se hace con la misma facilidad que si se tratara de una instalación normal. Es decir eligiendo la correspondiente opción en el momento adecuado. En lugar de "LPT1:", se selecciona "FILE:". Trabajaría en su diseño hasta que estuviera contento con él, se aseguraría de que el resultado al imprimir va a ser correcto (para esto el WYSIWYG viene de perlas) y, para terminar, daría al programa la orden de imprimir. El programa generaría en este momento la serie de bytes que debería hacer llegar a través de la puerta paralelo y el correspondiente cable a una Hewlett-Epson Stylus 850C, y, en lugar de remitirlos a su destino natural dado que está imprimiendo hacia un fichero, nos pediría su nombre, le daríamos el que más nos

84 Estos requisitos son absolutamente normales hoy en día. No se exagera lo más mínimo.

85 Una vez más no sólo no exagero sino que tal vez me quedo corto. Como veremos a no mucho tardar, desinstalar (retirar del disco duro) un programa Windows no es coser y cantar.

apeteciera, y los facturaría allí. Pablo copiaría él o los ficheros con los datos de la impresión en uno o varios disquetes, iría a casa de Ricardo, le pediría que le dejara un momento el ordenador, teclearía el equivalente a:

```
C:\>COPY el_fichero_que_sea LPT1: /B
```

la impresora imprimiría todo lo rápido que su maquinaria le permitiera el tan deseado papelote, Pablo daría las gracias, y punto final. Si hemos de ser realistas, hay que calcular un tiempo adicional, que Pablo debería emplear necesariamente en contarle a Ricardo qué diablos había pasado⁸⁶.

Por si no ha quedado suficientemente claro, podemos decir que lo que Pablo ha hecho no ha sido otra cosa que representar el papel de cable de impresora. Ha montado la Hewlett-Epson Stylus 850C en el ordenador de su casa contando con que un “cable virtual” lo conectaba a la impresora de Ricardo⁸⁷. Como el tal cable no era real, ha guardado los datos que deberían haberlo atravesado en un disquete, los ha llevado él mismo, en persona, hasta casa de Ricardo, y allí, mediante una simple orden DOS, los ha terminado de conducir al destino para ellos previsto. Lo que acabamos de presentar no es sino la versión actual de un truco viejísimo conocido como “imprimir a fichero”. Viejísimo pero todavía imprescindible.

Debemos aclarar lo del parámetro /B de la orden “COPY”. No por el parámetro en sí, pues para eso ya tiene las ayudas, sino porque una vez más interesa como repaso de conceptos básicos. ¿Recuerda cómo acabamos la orden “COPY CON: HOLA.TXT”? Pulsando CTRL+Z. Dijimos que el ASCII 26 actuaba como un indicador de fin de fichero. Es así siempre, pero tiene sentido que así sea tan sólo si los ficheros que queremos copiar son de tipo texto. Un fichero que contenga datos para una impresora puede no estar en modo texto, y en ese caso no tiene sentido que se lea el ASCII 26 como “fin de fichero”. Ahora será un carácter más, tan poco especial como cualquier otro. Nosotros, que somos los encargados de entender el significado de los datos que el ordenador maneja, debemos evitar que LR-PeCé se confunda. Suplementamos pues su escaso conocimiento añadiendo el /B, con lo que le decimos que, en este caso particular, no queremos que mire el ASCII 26 como “fin de fichero”. Conseguimos de esta forma que, si el fichero son 50 Kilobytes, lleguen todos a la impresora aunque el cuarto byte sea justamente un ASCII 26⁸⁸.

86 ¿Que le parece que estamos engañando al ordenador?. Bueno, tal afirmación puede considerarse carente de sentido en términos absolutos. Y de ser cierta, nadie lo prohíbe. No es pecado, ni siquiera venial. En el capítulo 9 volveremos sobre lo de “engañar”, forzosamente entre comillas, al ordenador.

87 ¿A que es divertido y francamente biensonante eso de “cable virtual”? Pues nada, querido LAO, disfrútelo mientras pueda, porque con este tipo de cosas pasa como con lo de las abreviaturas.

88 Si Vd. prueba cosas como “COPY COMMAND.COM CON:”, o el equivalente “TYPE COMMAND.COM”, es decir si intenta visualizar como texto ficheros que no contienen este tipo de datos, verá que además de aparecerle en pantalla cosas bastante extrañas (resulta divertido las primeras veces), no parece que la cosa dure lo que debería. Así es. El proceso se interrumpe en cuanto se encuentra un CTRL+Z en el fichero en cuestión.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Tras esto, acabamos de pasar con nota, aunque sea incierta, la primera de las tres lecciones de que constará nuestro curso de hechicería para PeCés. Vayamos sin tardanza a por la segunda.

Abracadabra, pata de cabra

Hechizos y conjuros para LR.PeCé

Las palabras mágicas, de una en una, pues bueno. Pero lo que hacen los magos auténticos es enlazarlas, unirlas, formar con ellas conjuros y hechizos capaces de poner del revés lo que sea y de realizar auténticas maravillas.

Con la lección anterior y un poco de paciencia (la necesaria para leer las ayudas de las distintas órdenes y los manuales en línea) podemos cogerle el tranquillo a las palabras. Pero en MS-DOS hay más. Si Vd. es uno de esos magos ambiciosos y con tendencia a la codicia, digamos que hay maaasss....

Y como da la casualidad de que nos hemos preocupado por aclarar lo realmente fundamental, pues tampoco se crea que es nada particularmente raro. Un par de tontadas más que, eso sí, pueden darnos un magnífico rendimiento.

Por poner un ejemplo, continuemos con nuestro ya popular “COPY”⁸⁹. Vimos con él cómo podíamos mandar cosas a distintos sitios. Pero eso no cubre todas las necesidades, ya que si Vd. ya ha practicado y es de natural inquieto, le habrá apetecido por ejemplo que lo que “DIR” nos muestra en pantalla termine en la impresora. O en un fichero quizá. O que la visualización de un texto suficientemente largo en pantalla se realice página a página, esperando a que acabemos de leer una para seguir con otra.

Puede hacerse de varias formas que se imprima la salida de un “DIR”. La primera es usar la tecla “Impr. Pant.”. Y si es más grande que una pantalla, pues tirar de “Pausa” o de parámetros y de tecla “Impr. Pant”. O quizá le apetezca usar una posibilidad no muy conocida, que consiste en pulsar “CTRL+P”, con lo que activamos el “eco a impresora” de MS-DOS. A partir de ahí, todo lo que tecleemos nosotros o nos presente en pantalla LR-PeCé irá a la vez a la impresora, con lo que tendremos una copia en papel de toda la sesión. Cuando estemos hartos de gastar páginas, bastará pulsar de nuevo CTRL+P para volver al modo de trabajo normal, desactivando el eco⁹⁰. Pero lo suyo propiamente, lo que nos permite mandar casi cualquier cosa a cualquier sitio y de casi cualquier manera, son los filtros y las redirecciones⁹¹.

89 “iAaaaaarrrrrrjjj!”. “¡Por favor, basta de COPY!”. Gritos. Desmayos. LAOs mesándose los cabellos... no ponerse nerviosos, que es sólo un ejemplo.

90 Lo del eco a impresora es otro de los inventos que datan de los tiempos de CP/M, al menos. Como tal vez haya Vd. deducido por la forma en que se activa, a través de caracteres de control que, como puede ver una vez más, siguen estando ahí. Usado con moderación viene bien en bastantes ocasiones a pesar de su edad.

91 A las que por cierto no sé cómo denominar correctamente. ¿Redireccionamientos?. ¿Desviaciones?. ¿Desvíos?. ¿Reenvíos?. Vamos a quedarnos con lo de “redirecciones”.

Comencemos con los filtros. Dado que es muy frecuente que el usuario de un PeCé desee que una determinada información, la que sea, se le presente pantalla a pantalla de ser necesario, convenía que el sistema operativo respondiera decentemente a tan razonable deseo. Ya que lo que se pretende es simplemente leer con un mínimo de dignidad lo que el ordenador nos muestra. Algunas órdenes admiten parámetros que especifican este modo de visualización. Por ejemplo “DIR /P”. Pero no es una respuesta general al problema. Lo que más nos convendría es que cualquier cosa pudiera visualizarse así, no sólo el resultado de aquellas órdenes que admitieran el tal parámetro.

Un filtro no es más que un programa que se encarga de coger una serie de datos, los que sean, y de procesarlos para que se presenten en un formato determinado. El nombre viene de que los podemos imaginar justamente como eso, como una especie de filtro, o de molde, que ponemos a la salida de una orden para que el resultado final tenga una forma más acorde con nuestros gustos o necesidades.

Filtros de obligado conocimiento en MS-DOS tenemos un par. A saber, “MORE” y “SORT”⁹². Traducidos “ más” y “ordenar”. El primero tiene como objetivo hacer que los datos se presenten pantalla a pantalla, esperando a que acabemos de leer una para pasar a la siguiente. Y el segundo que cualquier texto producido por el ordenador, probablemente en forma de lista, sea susceptible de clasificación en orden alfabético.

Si son filtros y deben modificar una salida, habrá que conectarlos de algún modo con las órdenes que produzcan los datos. Normalmente, la conexión se realiza empleando un método que debería resultar bastante normal a estas alturas. Se usa un carácter especial, la barra vertical “|” (en la cara delantera de la tecla con el “1”) para conectar la orden que produce el texto a presentar y el filtro que debe formatearlo. Aunque puede seguir estando absolutamente seguro de que donde más información va a encontrar sobre estos temas es en el manual en línea de su DOS, que sigue donde siempre, pongamos algunos ejemplos.

Si le mandáramos a LR-PeCé, por decir algo “DIR C:\NWDOS*. * | MORE” lo que obtendríamos sería poco más o menos lo mismo que con “DIR C:\NWDOS*. * /P”, es decir, una presentación página a página de una lista probablemente muy larga. Por supuesto, use “DIR C:\DOS*. * | MORE”, si en su caso tiene un ordenador con MS-DOS. Y por cierto, a estas alturas ya no debería ser necesario insistir en estos detalles, o sea que ésta es la última vez que lo hacemos.

Pero si empleamos una orden nueva, capaz entre otras cosas de listarnos todos los ficheros por debajo de un cierto nivel, incluidos subdirectorios, y que responde al nombre de “TREE” (“árbol” en inglés, que podríamos traducir por “muéstrame el árbol de directorios”), podríamos obtener el mismo tipo de presentación, a pesar de que “TREE” no suele admitir el dichoso “/P”. Sin más que decir, por ejemplo: “TREE C:\WINDOWS\F | MORE”, veríamos una lista enorme que nos sería presentada decentemente. Por supuesto si tenemos Windows instalado en ese directorio en nuestro ordenador, precisión que de nuevo no debería ser necesario hacer y que no vamos a volver a realizar. Y si quiere saber lo que significa el “/F” ya sabe donde tiene que mirar.

92 Al menos en Novell DOS 7 existe un tercero, llamado FIND. El manual en línea de su DOS le informará de su situación al respecto.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Con “SORT” podríamos hacer que la lista de ficheros de Windows apareciera en orden alfabético. Bastaría con pasarle a LR-PeCé: “TREE C:\WINDOWS /F | SORT /+40” y de nuevo dejaremos los detalles para los textos de ayuda. Pero con lo anterior obtendríamos tan sólo, eso sí en orden alfabético, una enorme lista otra vez imposible de leer. Por lo que lo suyo sería añadir una nueva palabra a nuestra frase para decirle: “TREE C:\WINDOWS /F | SORT /+40 | MORE” con lo que finalmente llegaríamos a leer cómodamente tan descomunal retahíla.

Imaginemos que estamos tan contentos que queremos conservarla para la posteridad. ¿Cómo podríamos hacer que acabara en un fichero?. Las redirecciones entran en juego. Es posible llevar la salida de cualquier orden a casi cualquier sitio admisible en MS-DOS mediante un mecanismo similar al visto para los filtros⁹³. Y la forma de indicárselo a LR-PeCé implica, espero que ya lo sospechara, un carácter de nuevo un poco especial. El símbolo “” (al lado de la Z) hace que el texto asociado a una orden DOS se lleve al sitio que especifiquemos a continuación, en lugar de terminar por defecto en la pantalla. Si invocamos a LR-PeCé con el conjuro “TREE C:\WINDOWS /F | SORT C:\LISTA.TXT”, acabaremos con la lista correctamente almacenada en C:\LISTA.TXT, fichero con un nombre tan válido como cualquier otro. ¿Y si lo que queremos es ir construyendo la tal con algo de aquí y algo de allá?. ¿Sería posible añadirle por ejemplo la lista de ficheros de C:\NWDOS?. Lo es de hecho. Una pequeña variación del símbolo de redirección, que pasa a ser “”, permite añadir algo al final de un fichero ya existente. Bastaría, por tanto, con decirle a LR-PeCé “DIR C:\NWDOS*. * | SORT > C:\LISTA. TXT”.

Filtros y redirecciones permiten domesticar en buena medida la forma en que LR-PeCé nos dice las cosas. Igual que hemos hecho que algo acabe en un fichero, podemos por supuesto llevarlo a la impresora, o a un modem, o donde sea. Si a Vd. aún le parece poco, no desespere. Hay todavía más.

Para aprendices de brujo realmente codiciosos, existe incluso la posibilidad de añadir a MS-DOS las palabras que Vd. decida. Aquellas que, usando la fórmula que a Vd. más le apetezca, harán que LR-PeCé lleve a cabo la tarea que Vd. quiera. Dicho de otra forma, es posible programar con cierta facilidad un ordenador en MS-DOS.

Recuerde que las órdenes externas no eran más que ficheros que contenían programas ejecutables por el ordenador. Con un fichero .EXE, .COM o .BAT, es posible añadir a LR-PeCé lo que más nos apetezca. Los .EXE y .COM no tienen nada realmente especial. Son ficheros que contienen programas para LR-PeCé en el lenguaje que él mejor entiende, el denominado “código máquina”. El único problema con ellos es que desarrollarlos exige usar un lenguaje de programación, algo demasiado complicado para este texto. Pero los ficheros .BAT están ahí precisamente para que todo aquel que no desee programar de verdad tenga al menos un resquicio al que asirse a la hora de automatizar el funcionamiento de LR-PeCé. El nombre les viene de “batch”.

93 En realidad es posible hacer más que eso. Pueden enlazarse órdenes entre sí, de modo que la salida de una sirva como entrada de la siguiente, y el resultado pase a otra, y.... Nos limitaremos a lo más elemental y, una vez más, recordaremos que existen los manuales en línea. Y si Vd. posee un DOS decente, hasta tendrá un manual impreso más que correctamente en papel de buena calidad, donde se contarán ésta y otras muchas cosas.

El modo de trabajo “batch”, también conocido como “procesado por lotes”, es en principio una herencia de los tiempos en que los ordenadores eran algo lejano e inaccesible que se manejaba desde un terminal. El problema entonces era conseguir llegar a la CPU, normalmente de tiempo compartido, y habitualmente solicitada en exceso. Cómo no había forma de tener con ella un diálogo fluido, lo que se hacía era preparar la serie completa de órdenes que queríamos que realizara para que, cuando pudiera, las cogiera todas de vez y las llevara a cabo de una sola tacada. Por suerte, nuestra situación al tratar con un PeCé es mucho más comfortable⁹⁴, y ya no es este el fin primordial que tienen los ficheros .BAT, que siguen siendo tan útiles como antes aplicados a tareas de automatización del funcionamiento, o incluso programación, del PeCé.

Un fichero .BAT no contiene sino una serie de órdenes muy similares a las que daríamos directamente desde el teclado. La única diferencia es que se admiten algunas más⁹⁵. Para crear uno basta con almacenar la serie de órdenes que queremos que sean ejecutadas por COMMAND.COM en un fichero de tipo texto, en correcto ASCII de MS-DOS, y en ponerle el nombre que queramos con la extensión, he aquí el detalle importante, .BAT.

Supongamos que, fatigados por escribir las larguísimas órdenes de los ejemplos anteriores, y deseosos de realizar de vez en cuando una inspección de los directorios C:\WINDOWS y C:\NWDOS para ver qué contienen, queremos simplificar el manejo del PeCé en este punto. La cosa se resolvería con sencillez creando un fichero al que podríamos llamar por ejemplo VERLISTA.BAT que contendría las dos líneas siguientes, tal cual:

```
TREE C:\WINDOWS /F | SORT | MORE
TREE C:\NWDOS /F | SORT | MORE
```

y para crearlo bastaría con usar nuestro conocido “COPY CON: VERLISTA.BAT”, o, si hemos leído ya un poco las ayudas y sabemos que cualquier versión mínimamente decente de DOS incluye un elemental editor de textos (un programa para crear y modificar con comodidad ficheros en texto ASCII), algo así como “EDIT VERLISTA.BAT”

94 Yo me sentaba en el terminal. Supongamos que escribía “DIR” y le daba a CR. El resultado no era que me aparecía el directorio, sino que el terminal lo transmitía a la unidad central, donde la orden quedaba a la espera de ser procesada por la CPU. Cinco minutos después, la CPU quedaba libre durante dos segundos y yo obtenía el resultado. La siguiente orden que diera debía volver a esperar más o menos lo mismo. En este esquema de funcionamiento, en el que los tiempos de espera para acceder a la CPU son muy importantes, debe intentarse que, una vez que hemos llegado a ella, la CPU haga todo lo posible de una tacada. El “lenguaje batch”, el “procesado por lotes” era una necesidad. Esos dos segundos de CPU daban para que se procesara un fichero con 500 órdenes en lugar de un solo “DIR”. El modo de funcionamiento “interactivo” al que estamos tan acostumbrados (yo te digo, tu me contestas inmediatamente y vuelta a empezar) se basa en el diálogo con una CPU siempre a la escucha, y es sólo posible si de verdad tenemos una para nosotros solos, como es el caso en un ordenador personal.

95 Por elemental que sea un lenguaje de programación, y el lenguaje “batch” lo es, debe incluir una serie de cosas. Estructuras de control, gestión de variables, instrucciones mínimas de entrada y salida.... Son esas necesarias adiciones, en forma de las correspondientes órdenes, las que diferencian lo que podemos encontrar en un fichero .BAT de lo que se teclea directamente para que LR-PeCé lo ejecute de inmediato.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Una vez creado, bastaría escribir en el prompt:

```
C:\>VERLISTA_
```

y por supuesto darle a CR, para que se ejecutaran sucesivamente los dos anteriores mamotretos.

Lo único que ocurre al procesar un fichero .BAT es que COMMAND.COM coge la entrada de datos, la línea de órdenes, no desde el teclado como hace por defecto, sino del fichero ASCII donde están almacenadas la serie de órdenes que queremos que lleve a cabo. Un fichero .BAT puede verse como un ejemplo más de esas redirecciones que tan útiles resultan.

Aparte de los servicios que puedan prestarnos al usarlos en la forma en que nosotros deseemos, los ficheros “batch” están activamente implicados en el proceso de arranque del PeCé, un punto que aún no hemos terminado. Pero como eso pertenece al siguiente apartado, deberemos antes acabar con éste. Lo que hacemos ahora mismo, añadiendo simplemente que los lenguajes de tipo “batch”, a los que en general se conoce también como “scripts” o “procesadores de macros” cuando nos referimos a los propios de un programa o un sistema operativo que no sea MS-DOS y su COMMAND.COM, son notablemente importantes en asuntos relacionados con la productividad del ordenador, como veremos en el capítulo 9.

Vade retro, Satán

Exorcismos para LR.PeCé

Siguiendo con nuestro símil, la última parte que un hechicero necesita dominar antes de enfrentarse con confianza a incubos, súcubos, genios, espíritus, y ectoplasmas, es la relativa a qué hacer cuando vaya mal. Porque un hechicero competente debe ser asimismo razonable y tener la modestia de reconocer que en algún momento, por la razón que sea, se meterá la pata y algo no andará como debiera. Se necesita una colección de exorcismos, para reconducir al redil a la bestia desbocada cuando sea preciso. Vayamos a ello.

Y comencemos diciendo que también en este punto, el DOS de los PeCés ha recorrido un enorme camino desde que apareció. Algunos años luz, al menos. Si sabemos que son posibles tres o cuatro cosas, acabaremos por encontrar increíblemente atractivo a un sistema operativo insistentemente tachado de antipático y feo. Y nos encontraremos cómodos y a gusto con él sabiendo que no sólo es simple y bastante fácil de dominar, sino que también nos permite corregir situaciones potencialmente desastrosas. A nada que añadamos un par de detalles, configuremos cuatro cosas, y practiquemos una temporada no muy larga, MS-DOS acaba siendo un lugar sumamente acogedor⁹⁶.

96 No estoy tratando de venderle nada. Pero sí que creo que MS-DOS tiene ahora mismo una mala fama heredada del pasado que ya no se merece. Cierto, siempre debió ser así, y la mala fama se la ganó a pulso en su momento. Pero es triste que un usuario no sepa valorar lo que tiene a su alcance cuando es algo que merece la pena. Muchos problemas de gente que usa un PeCé y no quiere salir jamás de Windows se resuelven conociendo mínimamente MS-DOS. Y lo que quiero dejar claro es que el pavor a MS-DOS está ahora mismo absolutamente injustificado. Y estuvo injustificado mucho antes si en lugar de MS-DOS nos referimos a DR-DOS.

El primer exorcismo a conocer responde a la siguiente pregunta: ¿qué hago cuando quiero organizar mi disco duro, o un disquete?. O si lo desea de otra forma: ¿cómo puedo borrar, cambiar de nombre, o mover ficheros y directorios?.

Tras un mínimo trasteo con su PeCé observará que se comienzan a acumular en el disco duro un montón de ficheros que no tenemos el más mínimo interés en conservar. Nuestro HOLA.TXT no tiene mayor sentido una vez que hemos sacado de él las lecciones necesarias. Porque nos barruntamos que si dejamos en el disco duro todo lo que entre, se acabará llenando, o cuando menos los ficheros interesantes de verdad serán más difíciles de localizar, sepultados bajo montañas de basura⁹⁷. No hay problema. La necesidad de borrar es tan perentoria que se encarga de llevarla a cabo una orden interna. Su nombre es “DEL”, abreviatura del inglés “delete”, que como ya sabemos es posible traducir por “borrar”. Existe incluso un doble, la llamada “ERA”, abreviatura del inglés “erase” y traducible del mismo modo⁹⁸. Por supuesto es posible obtener multitud de detalles sobre ambas en los lugares habituales, es decir a través del parámetro “/h” y el manual en línea. Advertencia igualmente aplicable a las órdenes que citemos a continuación y que ya no volveremos a hacer. Tenemos que aplicarnos el cuento, no queda más remedio que predicar con el ejemplo, y espero que coincida conmigo en que lo hemos repetido hasta la extenuación.

Tampoco el cambiar el nombre a archivos y directorios es en la actualidad mayor problema. La orden “REN”, abreviatura del inglés “rename”, traducible por “renombrar”, permite cambiar con facilidad el nombre de un fichero; en tanto que “RENDIR”, al menos en Novell DOS 7, hace lo propio con los directorios⁹⁹. El nombrecito es una abreviatura de “rename directory”. Y ésta es una de las cosas que no existían en las primeras versiones de MS-DOS, en las cuales era necesario comenzar por crear un directorio nuevo, continuar copiando penosamente los ficheros y subdirectorios correspondientes al directorio a renombrar, y terminar borrando el original no menos penosamente, para conseguir llevar a cabo tan elemental operación.

Porque borrar un directorio era otra cosa también difícil en aquellos tiempos. Si Vd. presta un poco de atención verá que plantear semejante tarea lleva de inmediato a una interesante cuestión: ¿qué hacemos con los subdirectorios y los ficheros de cada uno de ellos que puedan encontrarse en él?. La respuesta inmediata es borrarlos también. Aunque el riesgo asociado a tal respuesta podía proporcionar memorables momentos de disfrute para espíritus intré-

97 Así es, en efecto. Los discos duros muestran una irrefrenable tendencia a la saturación, y aunque sobre sitio, un fichero o directorio que no hace falta es siempre un estorbo. Complica la organización del disco duro, aspecto que hay que cuidar en grado sumo. No deje basura dando vueltas en su ordenador, del mismo modo que, si pudiera evitarlo, no tendría su escritorio sepultado bajo innumerables papelotes inservibles.

98 Que haya dos órdenes distintas para hacer lo mismo puede parecer una extravagancia. La razón de que existan ambas es que “ERA” proviene de los tiempos del CP/M, y es de suponer que hay gente que aún la usa. Cómo no estorba demasiado, pues ahí sigue.

99 Otras versiones de DOS pueden hacerlo de distinta forma. Por ejemplo extendiendo el ámbito de aplicación del “REN” normal, o mediante el uso con “REN” del correspondiente parámetro. MS-DOS 6.22 permite hacerlo con la orden “MOVE”. Y no debería haberlo dicho, pues el manual en línea le hubiera permitido también aclarar este punto.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

pidos, era de suponer que un novato no iba a estar muy conforme con que pudiera perderse el contenido de un disco en su totalidad manifiesta (como diría Forges) sin más que borrar, tal vez por error, su directorio raíz. En consecuencia “RD”, orden así bautizada a partir de “Remove Directory” abreviado, traducible por “quitar directorio”, y que es con lo que se cuenta inicialmente para este tipo de operación, no permite que se borren directorios que no estén ya vacíos de todo contenido.

Este enfoque alternativo primaba la seguridad. Tampoco se podían borrar con facilidad todos los ficheros de un directorio a base de “DEL *.*”. El ordenador nos preguntaba, y aún sigue haciéndolo, si estábamos seguros de querer tal barbaridad, y sólo tras la confirmación de que sí, que de verdad lo queríamos hacer, procedía con la masacre. Era una forma de abordar la cuestión bastante adecuada cuando los discos duros eran pequeños y su estructura sencilla.

Pero conforme fueron aumentando de capacidad, complicando su organización, y, porqué no decirlo, los usuarios de MS-DOS fueron cada vez más capaces de hacerse cargo de sus propios problemas, hubo un cierto clamor pidiendo que se incluyeran en MS-DOS algunas bombas atómicas para, cuando hiciera falta, cargarse con facilidad cualquier cosa.

Si Vd. desea borrar de verdad directorios, eliminando de paso todo lo que contengan, es posible en Novell DOS 7 mediante “XDEL”, cuyo nombre no es sino una cierta forma de abreviar algo así como “eXtended Delete”, traducible por “borrar ampliado”¹⁰⁰. En MS-DOS 6.22 es posible hacer lo mismo mediante “DELTREE”, otra abreviatura de “borrar árbol”. Y en cuanto a lo de mover ficheros y directorios, ni más ni menos que la asociación de copia y borrado del original, existe asimismo una orden que ya lo hace solita y que se llama, vaya casualidad, “MOVE”. Al menos en Novell DOS 7¹⁰¹.

¿Y dónde ha quedado la seguridad, pregunta?. Podríamos decir que Vd. puede seguir con “DEL” y “RD”, si lo desea, y ya sería una respuesta bastante correcta, pero la realidad es aún mejor. MS-DOS añadió en sus versiones modernas una serie de mecanismos que permiten recuperar, cuando sea preciso, aquellos ficheros o directorios que se hayan borrado previamente. La orden para hacerlo es “UNDELETE”, traducible por “deshacer borrado”.

¿Que cómo es posible tal maravilla, pregunta una vez más?. Buena pregunta. Ya hemos dicho muchas veces que no está de más conocer los mecanismos detrás de las cosas. Hay al menos

100 XDEL permite también borrar ficheros ocultos, de sistema, de sólo lectura... o sea que borra lo que le digan. Eso sí, especificando los correspondientes parámetros y tras confirmación. De la misma familia son XDIR y XCOPY, que permiten entre otras cosas listar o copiar cualquier tipo de fichero. Si no dispone de equivalentes, aplique la lógica. Un fichero de “sólo lectura” no se puede borrar, ni renombrar, ni modificar su contenido. La presencia del atributo “r” lo impide. Para poder hacer algo de lo anterior, habrá por tanto que quitar en primer lugar tal atributo. Lo que es perfectamente factible con un uso adecuado de “ATTRIB”. Y a partir de aquí podemos aplicarle DEL, o REN o EDIT. Y podemos rematar la faena colocando de nuevo el atributo “r” para que siga siendo un fichero de sólo lectura, aunque distinto. No debe ver los atributos como una propiedad definitiva de un fichero. Son algo temporal, tan susceptible de ser cambiado como cualquier otra cosa.

101 Y en MS-DOS 6.22. Donde también se emplea, como ya dijimos, para renombrar directorios.

tres formas de hacerlo. Espero que recuerde que un fichero era una entrada en la FAT con el nombre correspondiente y una lista de los sectores que contenían sus datos. Hay que empezar por decir que MS-DOS borra un fichero de forma algo rara a primera vista. Lo único que hace es marcar la correspondiente entrada del directorio como vacía y los sectores a ella asociados como libres. Con este método se gana ante todo en velocidad. Basta con ir a la FAT y cambiar muy pocos bytes para completar el proceso¹⁰². Y por si fuera poco, mientras no escribamos nada en el disco, los datos siguen estando allí, con lo que son susceptibles de recuperación. Y si escribimos algo, siempre serán recuperables aquellos ficheros que no hayan tenido la desgracia de ver cómo uno de sus antiguos sectores se empleaba para grabar un nuevo fichero. Ya que entonces los datos sí que se escriben en los correspondientes sectores.

El primer método de “desborrado” se basa en usar la información existente en la FAT sobre un fichero que se ha borrado para recomponer la entrada. Al borrar, se elimina también el primer carácter del antiguo nombre del fichero, por lo que, al deshacer la operación, es necesario suministrarlo de nuevo. El ordenador nos lo pide en su momento.

El segundo mecanismo no es más que una mejora del anterior. Basta con almacenar, de alguna forma¹⁰³, copias adicionales de la FAT o de lo que sea preciso, para que el desborrado pueda hacerse automáticamente sin necesidad de pedir ni siquiera que se complete el nombre. Es tal vez el sistema más recomendable. En Novell DOS 7 se activa mediante la orden “DISKMAP”, traducible por “Guarda un mapa del disco”.

Y el tercero es aún más drástico. Se cambia el mecanismo de borrado, de modo de cuando decimos “DEL”, el ordenador pasa simplemente el correspondiente fichero a la categoría de “oculto”, y añade el espacio que en realidad todavía sigue ocupando a la cantidad que nos muestra como disponible. En Novell DOS 7 puede ponerse en funcionamiento con la orden “DELWATCH”, traducible por “Borrado supervisado”. Aunque el más seguro de todos, este método puede conducir a situaciones chocantes para el usuario no advertido. En operaciones masivas de borrado y copia, puede resultar que aunque el ordenador nos diga que tenemos 80 Megabytes disponibles, se niegue a realizarlas de inmediato o añada misteriosos mensajes (que reflejaran la necesidad de deshacerse de verdad de los correspondientes ficheros) cuando queramos copiar 5 o 10 Megabytes. Y si lo desactivamos, un montón de ficheros que creíamos haber eliminado, volverán a la vida. Suele ser además el método más lento, por lo que no resulta tan interesante como pudiera parecer¹⁰⁴.

102 Y si lo que Vd. quiere no es simplemente borrar, sino desintegrar un fichero, de forma que no quede ni rastro de que alguna vez estuvo allí (algo que resulta interesante en entornos profesionales en los que importa la seguridad), sepa que hay programas que permiten hacerlo. Suele ser uno de los servicios que se incluyen en las “utilidades” o “herramientas”, que aparecerán en el próximo capítulo.

103 Vamos a adelantarnos un poco a los acontecimientos. Normalmente consiste en una simple orden añadida a los ficheros que automatizan el proceso de arranque. Ficheros que como pronto veremos se llaman CONFIG.SYS y AUTOEXEC.BAT, ambos en el directorio raíz.

104 Si ha oído Vd. hablar de la famosa “papelera” del Apple Macintosh, también presente en el reciente Windows 95 (del que pronto nos ocuparemos aunque sea un poco), sepa que consiste en poco más que en este último sistema de recuperación. Si quiere se lo digo más claro. También DOS, o Windows 3.1, tienen su “papelera”. Tan sólo tiene que activar el “DELWATCH” o

CAPÍTULO 7. PeCés: MS-DOS/Windows.

La seguridad en los PeCé ha aumentado hasta el extremo de que es incluso posible recuperar los datos de un disquete que se ha formateado por error. El mecanismo involucrado en tan simpár maravilla añade a lo visto con DISKMAP una astuta maniobra para acelerar el proceso de dar formato a un disco. Vimos que podíamos echarnos un par de minutos al coletito con cada disquete de alta densidad que formateáramos. ¿No habría forma de hacerlo más rápido?. En el caso sumamente frecuente de que lo que queramos sea formatear de nuevo un disco ya previamente formateado, sí que lo es. Basta con simplificar el proceso. En lugar de ir comprobando si cada sector del disquete está todavía utilizable, y de ser así borrándolo y añadiéndolo a la lista de los utilizables para terminar escribiendo una FAT vacía que lo haga accesible a los ojos del DOS, proceso en el que desde luego nos cargamos de forma irremediable todo lo que en el disco hubiera, lo que podemos hacer es reescribir directamente una FAT vacía. Evitamos la comprobación del estado de todos los sectores del disquete, que es lo que de verdad lleva tiempo, y además, si el disco no estaba totalmente lleno, aún podemos dejar una copia de la vieja FAT y de todo lo que haga falta para poder deshacer el proceso y recuperar todos sus ficheros en caso de que nos demos cuenta de que por error hemos formateado el disquete que no era. Nuestro viejo conocido “FORMAT” se toma automáticamente la molestia de elegir el “formato rápido” o el normal en función del disquete que queramos formatear¹⁰⁵. Es preciso añadir que en ocasiones el “formato rápido” no funciona bien. Un parámetro al uso permite entonces indicar a DOS que lo haga como Dios manda.

Otra operación básica a la hora de organizar un disco es la copia de ficheros. No debería hacer falta decir cual es la orden DOS de que disponemos para llevarla a cabo, pero tal vez convenga añadir que hay más. Es posible también copiar directorios y subdirectorios con sus ficheros correspondientes (“XCOPY”). Y, por supuesto duplicar un disquete (“DISKCOPY”), algo fundamental por motivos de seguridad¹⁰⁶.

Para ver si de una vez se aclara con lo que estamos diciendo recientemente, añadamos algunas cosas sobre esta última orden. Duplicar un disquete, pensará Vd., es ni más ni menos que eso, coger un original y acabar con una copia idéntica. Y aunque no lo diga pensará también que el proceso se desarrollará necesariamente de forma inmediata y entre disquetes. Pues no tan necesariamente. En Novell DOS 7, DISKCOPY puede también guardar la “fotocopia” de un disquete en un fichero normal y corriente, donde queda almacenada hasta que la ocasión lo requiera. El tal fichero contendrá la FAT, los sectores, y todos los datos necesarios para, cuando así lo decidamos, terminar el proceso con la grabación del disquete duplicado¹⁰⁷

cualquier utilidad similar, ya que muchos programas de “herramientas” (de los que también pronto nos ocuparemos aunque sea un poco) hace tiempo que incluyen éste y otros inventos aún más maravillosos. A pesar de todo, me sigo quedando con el segundo sistema.

105 Puede que Vd. crea que lo del “UNFORMAT”, que es ni más ni menos que la orden que deshace un formateo indeseado, le va a venir de perlas. Hombre, no está mal que exista. Pero, por si le puede servir, yo jamás la he usado. La razón no es ni más ni menos que uno suele asegurarse de qué disquete emplear antes de empezar. Además, suelen grabarse de inmediato cosas en él. Y entonces la recuperación de los antiguos ficheros es mucho más problemática.

106 En el capítulo 9 trataremos con mayor detalle este tipo de asuntos.

107 El DISKCOPY de MS-DOS no lo admite. Pero hay multitud de “herramientas” que duplican

Cuando tengamos organizado el disco, podremos empezar jovialmente a usar el ordenador de verdad, produciendo con el devenir de los días y los meses un montón de datos con o sin sentido. Y añadir programas, a ser posible conservando una organización decente, para hacer nuevas cosas que produzcan aún más datos con o sin sentido. Saldremos del enfebrecido paroxismo de nuestra obcecación productora cuando nuestro PeCé, casi exhausto, nos presente un mensaje en pantalla diciendo que no queda sitio en el disco duro. ¿Qué hacer en ese momento?. ¿Qué mágico exorcismo podemos usar para salir del paso?. Ya que lo suyo propiamente es o bien vaciar en lo posible el disco que tan pequeño se nos ha quedado pasando los datos a otro sitio, o bien aumentar la capacidad de almacenamiento de nuestro PeCé a base de añadir o sustituir por otros más capaces los periféricos que deseemos o que nuestra capacidad adquisitiva nos permita. ¿Y si no podemos hacer nada de eso?. Puede que tengamos la cuenta a cero y no podamos comprar otro disco duro, o que se nos hayan acabado los disquetes y no podamos vaciar el disco ya lleno, o que simplemente no dispongamos del tiempo necesario para nada de lo anterior. Cambiar o añadir un disco es cosa de un día al menos, si todo va bien, y vaciar un disco duro pueden ser varias horas. Nuestro furor productivo nos pide soluciones inmediatas, ya, para continuar trabajando mientras resolvemos el problema de manera más definitiva.

Cuando tengamos organizado el disco, podremos empezar jovialmente a usar el ordenador de verdad, produciendo con el devenir de los días y los meses un montón de datos con o sin sentido. Y añadir programas, a ser posible conservando una organización decente, para hacer nuevas cosas que produzcan aún más datos con o sin sentido. Saldremos del enfebrecido paroxismo de nuestra obcecación productora cuando nuestro PeCé, casi exhausto, nos presente un mensaje en pantalla diciendo que no queda sitio en el disco duro. ¿Qué hacer en ese momento?. ¿Qué mágico exorcismo podemos usar para salir del paso?. Ya que lo suyo propiamente es o bien vaciar en lo posible el disco que tan pequeño se nos ha quedado pasando los datos a otro sitio, o bien aumentar la capacidad de almacenamiento de nuestro PeCé a base de añadir o sustituir por otros más capaces los periféricos que deseemos o que nuestra capacidad adquisitiva nos permita. ¿Y si no podemos hacer nada de eso?. Puede que tengamos la cuenta a cero y no podamos comprar otro disco duro, o que se nos hayan acabado los disquetes y no podamos vaciar el disco ya lleno, o que simplemente no dispongamos del tiempo necesario para nada de lo anterior. Cambiar o añadir un disco es cosa de un día al menos, si todo va bien, y vaciar un disco duro pueden ser varias horas. Nuestro furor productivo nos pide soluciones inmediatas, ya, para continuar trabajando mientras resolvemos el problema de manera más definitiva.

También para esto hay solución en el mundo del DOS. Ha llegado el momento de comprimir el disco. Ejecutando un programa, nuestro disco lleno de 200 Megabytes va a ver su capacidad casi duplicada, y a tener todavía otros 200 Megabytes disponibles. Hasta que los llenemos, debería darnos tiempo incluso de ahorrar para un disco duro más gordo. Una vez más nos estaremos preguntando cómo es posible tal maravilla y cómo ponerla en práctica.

discos y en la actualidad la mayor parte de tales programas sí lo permiten. Esta opción le da al proceso una notable flexibilidad, que permite salir del paso en ciertas ocasiones comprometidas. Si no tenemos ahora mismo disquetes vacíos, o si aquellos de los que disponemos no son del mismo tamaño que los originales, o si tan sólo queremos una copia de seguridad que eventualmente nos permita reconstruirlos no sabemos cuando y estamos hasta el gorro de almacenar miles de disquetes y tenemos sitio en el disco duro, lo suyo propiamente es hacer una copia de disquete a fichero.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

¿Se acuerda del ejemplo sobre distintos formatos de texto que pusimos hace algunos capítulos? ¿Recuerda aquello de “Qué bonito es Candanchú con nieve, y qué bonitas vistas tiene”? Pues conseguir duplicar la capacidad del disco es poco más que eso. La información, exactamente la misma información, puede codificarse de formas muy diferentes, y como no podía ser menos unas son más compactas, otras son más legibles, y así sucesivamente. El hecho es que la codificación normalmente empleada a la hora de almacenar datos en general en un PeCé (programas, textos, números, y algún otro etcétera) no está optimizada para ahorrar espacio. Si cambiamos el código de almacenamiento de datos y vamos a uno que, aunque quizá más complicado, aproveche mejor el espacio, nuestro disco duro “casi lleno” va a pasar a “bastante vacío”. En general el proceso se conoce como “compresión de datos”, y en los últimos 5 años ha dejado de ser una curiosidad académica y se ha convertido en algo cotidiano, y en consecuencia de necesario conocimiento¹⁰⁸.

Para que nos hagamos una idea más clara del proceso, pongamos un ejemplo de cada uno de los tres métodos más conocidos de compresión (el nombre correcto en informática es “algoritmo”¹⁰⁹). Quizá el más sencillo es el conocido como RLE, abreviatura del inglés “Run length encoding”, traducible tal vez por “codificación valor -frecuencia”. No puedo sino pedir, y hasta suplicar, que no se le ocurra jamás usar la traducción que acabo de proponer. Consiste en que cuando tenemos datos formados por valores discretos que se repiten, es posible sustituirlos por una indicación del valor y el número de repeticiones. De esta forma la secuencia “AAAAAAAABBBBBBXXXXHHHHHHHHH” podría codificarse como “A8B5X4H9”. En ocasiones es incluso posible prescindir del valor. Así, si sabemos que un dato que represente lo que sea está formado por ristas de ceros y unos que se suceden alternativamente, es decir algo como “000000000000011111111111111111100000-0111000000000000000111111111111111111”, podríamos dejarlo simplemente en “14 21 6 3 15 21”, y espero no haberme equivocado contando.

El método de compresión de Huffman es de aplicación más general. Si empleamos el código ASCII para codificar algo, digamos un texto, estamos empleando un código de longitud fija. Cada carácter ocupa un byte. Si tenemos un texto que contiene muchísimas “A” y muy pocas “Z”, es posible comprimirlo si cambiamos el código ASCII por otro de longitud variable, en el que los caracteres más frecuentes (la “A” de antes) se codifiquen con dos bits, por decir algo, y los menos frecuentes (la “Z” de antes) con los 8 bits que ocupa cada carácter en el código ASCII. El código Huffman se construye a la medida de cada conjunto de datos, analizando la frecuencia de los bytes que en él aparecen, con lo que se logra una optimización bastante decente.

108 ¿Porqué, pregunta?. Ya digo que existía antes, y ya entonces se empleaba en los PeCés. Pero la popularización de Windows, el aumento desaforado en tamaño experimentado por los programas, la proliferación de datos de tipo gráfico, y otros fenómenos parecidos llevaron a que la optimización de la capacidad de almacenamiento disponible pasara a ser una necesidad. La aparición y posterior generalización de los compresores de disco, de los que estamos tratando, acabó por ponerla a disposición de todo el mundo. Las mismas razones que justifican la compresión de datos, justifican el vertiginoso crecimiento en capacidad experimentado por los discos duros en este mismo período.

109 En el glosario se explica un poco más lo del “algoritmo”. Y los tres algoritmos de compresión que se citan.

Y finalmente podemos recurrir a algo parecido a lo del código Huffman pero haciendo que cada elemento de nuestro código signifique no un carácter sino una “frase”. En textos muy repetitivos, donde aparezca muchas veces la expresión “Pablito clavó un clavito”, por decir algo una vez más, será posible sustituir la tal frase por un único byte, pongamos por caso. También ahora el código correspondiente se construye a la medida, analizando los datos a almacenar. Este es el algoritmo de “Lempel-Ziv-Welch”, los tres individuos que lo desarrollaron, y se conoce universalmente como algoritmo LZW.

Conocidos los algoritmos que hacen posible que nuestra preciada información ocupe menos sitio, veamos como se conseguirá automatizar la cosa en un PeCé para que el DOS, solito, sea capaz de sacarle el máximo partido a nuestro disco duro. Se crea un fichero normal y corriente, eso sí muy gordo, ya que puede llegar a ocupar casi todo el disco duro, y en él se almacenan comprimidos todos los ficheros que teníamos antes sin comprimir. Ese fichero gordo debería aparecer ante nosotros como una unidad normal. Si hemos comprimido el disco C:, lo mejor sería que siguiera apareciendo como tal. Lo único que hace falta es un programa que le enseñe a DOS como guardar y recuperar los datos de ese fichero y lo convierta en la nueva unidad C:¹¹⁰. A estas alturas es obvio que bastará montar un driver adecuado, que oculte la información asociada a los detalles del proceso (coger del fichero los datos comprimidos y dejar en la RAM los datos sin comprimir, como estaban originalmente) y lo haga presentarse ante nosotros con la apariencia adecuada. DOS permite que se le añada el tal driver, igual que un ratón, o un CD-ROM, y ya está. En la práctica, todo consiste en ejecutar un programa, cuyo nombre depende de la versión concreta de DOS que empleemos, que se encarga de crear el fichero, comprimir los datos, y añadir el driver al sistema operativo. En DR-DOS 6.0 se llamaba SuperStor, en Novell DOS 7 se llama Stacker, y en MS-DOS 6.22, DoubleSpace. Excepto DoubleSpace, los otros han estado disponibles como programas independientes antes de ser incluidos como una parte más en distintas versiones de DOS. Y no hay más, o sea que no es para tanto¹¹¹.

Acabaremos con la compresión de discos con algunas notas. En primer lugar, decir que su eficacia depende del tipo de datos de que partamos. El proceso no se puede reiterar hasta el infinito, y aquellos datos que se hayan almacenado inicialmente en un formato ya suficientemente compacto, no serán susceptibles de compresión posterior. Como regla general, para un disco lleno con datos variados, lo más que podemos esperar es que su capacidad se

110 Lo que sucede en realidad es que la unidad C: de verdad pasa a ser la D:, pongamos por caso, y la nueva unidad comprimida se queda con el nombre antiguo.

111 No es para tanto teóricamente. En realidad tiene muchísimo mérito programar un buen sistema de compresión de discos. Debe ser sólido, pues la pérdida de datos en el proceso puede ser catastrófica. Debe ser rápido, a ser posible tanto que no se note que el disco está comprimido. Debe ser pequeño, para no llenar la preciosa zona de los primeros 640 KB de RAM del PeCé con un driver enorme. Y hay más trucos que el empleo de un algoritmo de compresión. Por ejemplo, y no lo explicaremos, los sectores del disco duro se hacen también de tamaño variable para optimizar aún más el almacenamiento. Con según que sistemas de compresión la cosa funciona de verdad muy bien, y por ejemplo la velocidad de operación no se resiente en forma apreciable. Hay que comprimir y descomprimir los datos sobre la marcha, pero el tiempo necesario para hacerlo se compensa con el menor volumen de datos, ahora comprimidos, que se mueven entre el disco duro y la RAM. Y con un buen sistema de caché, la cosa marcha admirablemente.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

duplique¹¹². En segundo lugar, que los algoritmos antes citados conservan toda la información existente en los datos originales, y son capaces de reconstruirlos exactamente, hasta el último bit. No puede ser de otra forma cuando se comprimen programas. No todos los algoritmos de compresión son así, y en ocasiones se prefiere perder parte de la información a cambio poder comprimir aún más, especialmente cuando se procesan datos de tipo gráfico (ver “MPEG” y “JPEG” en el glosario). En tercer lugar, debemos también precisar que existen programas para comprimir no ya todo un disco en bloque, sino sólo aquellos ficheros que indiquemos. Son una más de esas “herramientas” tan necesarias para hacer un montón de cosas útiles y son sumamente populares en la actualidad. Y finalmente que no hay que engañarse. La compresión del disco no hace sino posponer un problema de falta de espacio de almacenamiento. En algún momento tendremos que vaciar el disco, o añadir o incrementar el que ya tenemos. Al final, los datos de uso menos frecuente van a tener que pasar a un medio de almacenamiento secundario, ya sea un disquete, una cinta o un magnetoóptico. Y entonces nos encontraremos que el volumen de datos a transferir es el doble del que teníamos antes. Los compresores de disco nos dan algo de tiempo para que preparemos una solución definitiva, pero nada más.

Como puede ver, querido LAO, el disco duro es sometido a multitud de manejos dudosamente honorables. Y conviene que esté en buena forma. No es raro que, con el tiempo, algunas de las entradas de la FAT no reflejen la realidad, y por ejemplo existan sectores sin dueño conocido. Ni son utilizados por un fichero ni aparecen como libres. O que los ficheros estén sumamente fragmentados, con todos sus sectores esparcidos aquí y allá, con lo que el disco pierde velocidad ya que es preciso desplazar mucho las cabezas de lectura. Y en ocasiones queremos cambiar en profundidad su organización, fragmentándolo por el motivo que sea en una o varias unidades lógicas. También DOS proporciona exorcismos adecuados para estas ocasiones. “CHKDSK” se encarga de comprobar, y en su caso reparar, la FAT del disco. “DISKOPT” de compactarlo o desfragmentarlo, es decir de juntar todos sus ficheros y de poner en sectores contiguos los datos de cada uno de ellos, de modo que no queden sectores libres en medio¹¹³. Y “FDISK” permite actuar sobre las particiones. Más que suficiente. Un consejo. No enrede con FDISK mientras no esté seguro de lo que hace. Puede encontrarse, entre otros posibles desastres, con que ya no es posible acceder a su disco duro, o con que ha borrado, de verdad, todos los datos que contenía¹¹⁴.

112 No citamos antes a los datos de tipo gráfico, y fue intencionado. Aunque muy propicios para la compresión, siendo posible reducir su tamaño en un 80 o 90% en ocasiones, hace ya tiempo que se han desarrollado formatos gráficos en los que los datos se almacenan ya comprimidos. Con ellos no es posible ganar mucho. Lo mismo puede decirse de otros tipos de datos producidos con algunos programas. Lo de antes no es, por tanto, sino una estimación general para datos normales.

113 Me parece que el equivalente en MS-DOS 6.22 es “DEFRAG”. Y existe también un “SCANDISK” que viene a ser un CHKDSK ampliado.

114 FDISK es otra de las bombas atómicas que hay que manipular con cuidado. Lo que no quiere decir que no sea necesaria. Y para que vea que es conveniente saber como funcionan las cosas, pongamos otro ejemplo avanzado. Vd. monta finalmente un nuevo disco duro en su ordenador, tras haber estado trabajando un tiempo con el original comprimido en parte. Sus programas están repartidos entre la unidad C:, la comprimida, y la D:, el resto sin comprimir. Desgraciadamente, la presencia del nuevo disco de verdad modifica necesariamente la asignación de unidades en DOS, y su unidad D:, justo donde están los programas que más utiliza, aparece tras la instalación del nuevo como unidad E:. Sus programas ya no funcionan, por ejemplo porque Windows sigue

En toda la historia que estamos contando hay una tarea tan importante como la organización y mantenimiento del disco duro de la que aún no nos hemos ocupado, y que es incluso anterior a ella. Hemos supuesto que estamos trabajando, y hasta que lo hemos hecho durante una cantidad de tiempo apreciable. Hemos avanzado la necesidad de añadir nuevos drivers, por ejemplo para comprimir el disco, o para manejar un ratón o un lector de CD-ROM. Y de hacer cosas tales como activar sistemas avanzados de recuperación de ficheros borrados. Pues bien, para todo ello es preciso configurar el PeCé. ¿Que dice que Vd. no va a hacer nada de eso y que por lo tanto no va a necesitar configurar?. No me sea ingenuo, hombre. Aún así la configuración es inevitable, algo de lo que resulta imposible escapar. Una configuración adecuada no sólo hace su ordenador más cómodo. También puede hacerlo mucho más seguro y rápido. Muchas veces hace falta simplemente para que ese programa que nos interesa se digne a correr¹¹⁵. Y por si fuera poco, una buena configuración es la mejor medicina preventiva para evitar desastres. Pero al configurar topamos de frente con una auténtica fuente de problemas potenciales, como ya hemos dicho en algún momento (ver glosario).

Desgraciadamente, no hay orden DOS que lleve a cabo por nosotros tan titánica tarea¹¹⁶. Incluso cuando la instalación de nuevas aplicaciones, de nuevos elementos de hardware, o la propia modificación de la configuración existente se realiza mediante un programa que intenta automatizar el proceso¹¹⁷, los resultados pueden no ser los deseados. Y muchas veces no lo son. Se requiere entonces la intervención humana, en forma de alguien lo suficiente experto como para meterse a cambiar a mano los parámetros adecuados. Para salir victorioso del empeño ante un PeCé que no marcha como debiera, es necesario saber con detalle cómo arranca, qué parametros son modificables y cual es la manera de hacerlo y, lo más importante de todo, ser capaz de detectar la causa de los problemas que lo aquejan, para lo que suelen hacer falta además unas cuantas herramientas adicionales en forma de programas adecuados, bastante experiencia, y una importante cantidad de tiempo para realizar un montón de probatinas diversas.

creyendo que están en D:. No es fácil modificar Windows para que sepa que ahora están en E:. La situación de partida puede variar (dos particiones sin comprimir en un único disco duro, por ejemplo) pero el resultado es el mismo. ¿Qué hacer si no tenemos tiempo para reorganizarlo todo?. De nuevo es cosa de varias horas, al menos, y nosotros tenemos que sacar un trabajo urgente, ya. Como su nuevo disco está vacío, la mejor solución sería usar FDISK para borrar su partición. Eliminamos la nueva unidad, deja de estar en DOS, restablecemos la situación inicial y, cuando tengamos tiempo, ya la volveremos a montar, de nuevo con FDISK, y modificaremos o copiaremos todo lo que sea preciso para que las cosas marchen.

115 Hasta se da el caso, por suerte cada vez más raro, de que distintos programas exijan configuraciones mutuamente incompatibles entre sí. También esto tiene solución en las modernas versiones de DOS. Con ellas y mediante diversos mecanismos, o recurriendo a “herramientas” adicionales si así lo deseamos, se pueden almacenar distintas configuraciones y seleccionar la apropiada a nuestros fines en el momento de efectuar el arranque del ordenador.

116 Es un problema gordo, pero no se apresure a denigrar aún más a DOS. Que yo sepa, tampoco lo hace ningún otro sistema operativo de ningún ordenador.

117 En Novell DOS 7 hay un programa que intenta hacerlo y que se llama “SETUP”. Ya existía en DR-DOS 6.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

No es siquiera concebible el pretender que alguien que comenzó a leer hace algunos cientos de páginas este texto teniendo muy poca idea de lo que un ordenador pudiera ser, acabe dentro de algunos cientos más siendo capaz de dejar fino como la seda cualquier PeCé. Hay que conformarse con proporcionar algunos detalles más de obligado conocimiento y poner algunos ejemplos elementales a ser posible no exentos de interés.

Comencemos por completar la descripción del proceso de arranque en MS-DOS y ver en qué puntos es posible actuar para configurar el ordenador. Volviendo al principio, cuando enchufamos nuestro PeCé lo primero que hace es ejecutar el programa de arranque de la BIOS. Y ya aquí es posible intervenir. Las opciones que se encuentran en este nivel se refieren casi exclusivamente a la configuración del hardware, como era de esperar¹¹⁸. Si controlamos los mensajes que aparecen en pantalla, alguno de ellos nos dirá algo parecido a “Pulse DELETE¹¹⁹ para entrar en el SETUP”. Lo de “SETUP” es traducible más o menos simultánea e indistintamente por “configurar” o “instalar”. Si le damos a la tecla correspondiente, lo que aparecen son una serie de menús relativos a si la CPU y la placa base deben funcionar así o asá, si el disco o discos que tenemos son de este tipo o de aquel, y cosas similares¹²⁰. Hay pocas cosas de interés para un novato, y son las únicas que vamos a ver. En primer lugar es aquí donde, si es preciso, deberíamos poderle decir a nuestro PeCé que queremos que arranque con el bloque numérico del teclado activado. Como recordará, dijimos que era muy conveniente asegurarse de que el teclado funcionara como debe¹²¹. En segundo lugar y en ordenadores modernos, encontrará también aquí la posibilidad de activar los modos de desconexión automática de componentes en caso de inactividad del ordenador. Y en tercer lugar y para terminar, pues casi no debería tocar nada más, verá también en algún sitio que es posible proteger con una clave el acceso tanto al ordenador como a la rutina de configuración que estamos usando. Actívela si es preciso¹²² y prepárese para ir saliendo. Eso sí, dese unas vueltas por todo para que las cosas le vayan sonando, y no se olvide de grabar los valores modificados para que en efecto entre en funcionamiento la nueva configuración.

118 También la justificación de este aserto caerá en el examen final.

119 Es una de las formas más normales de acceder a la configuración de la BIOS. Por supuesto, es posible que la tecla o combinación de teclas a usar sea diferente. En mi ordenador, por ejemplo, lo es. Permanezca atento a la pantalla, ya sabe que es allí donde va a encontrar las indicaciones pertinentes para un montón de cosas.

120 Si quiere detalles, aunque le resulten incomprensibles, aquí van algunos. Es posible establecer si la CPU debe usar las cachés interna y/o externa, los ciclos de espera para acceder a la memoria, cuestiones sobre direccionamiento de IRQs, precisar las características de los discos duros (número de cabezas, sectores por pista y demás) y muchos otros etcéteras de parecido cariz.

121 Si Vd. tiene un ordenador viejo de verdad, quizá su BIOS no cuente con tal opción. No todo está perdido. A veces KEYB, que luego citaremos, permite activar el teclado numérico. Y si no es así, hay programas a los que se puede recurrir para hacerlo, aunque pueden no ser sencillos de localizar para un novato. Al igual que KEYB, como también veremos pronto, se colocan en AUTOEXEC.BAT con las correspondientes opciones y ya está. En MS-DOS 6.22 es posible activarlo con una orden (“NUMLOCK”) en CONFIG.SYS.

122 Cuidado con las claves a todos los niveles. Si se le olvidan, va a tener problemas. Úselas tan sólo si de verdad hace falta y asegúrese de poder recordarlas.

Nos despediremos con algunas notas. En primer lugar, no le tenga miedo a configurar la BIOS. Que no se lance a toquitearlo todo no quiere decir que haga falta tener un miedo cerval a darse unas vueltas por allí. Las BIOS de hoy en día permiten, si hemos enredado demasiado, restaurar con facilidad los valores de configuración por defecto, y eso garantiza que siempre vamos a poder volver a recuperar el control del equipo. Y además son mucho más civilizadas que las antiguas, siendo capaces de detectar automáticamente un buen montón de cosas. Por ejemplo, las características de los discos duros¹²³. Limítese por lo tanto a no tocar aquello que no sepa qué es. Tampoco viene mal enterarse del tipo exacto de BIOS con que cuenta su ordenador. Aunque lo más probable es que se trate de un chip de ROM, y por lo tanto inmutable, no es tan raro que, especialmente en ordenadores de marca, el fabricante haya optado por almacenar la BIOS en un tipo de memoria permanente pero regrabable (ahora mismo “memorias tipo flash”, ver glosario). Esto permite que, si es necesario, un programa en disquetes pueda cambiar por completo el BIOS de un ordenador, dándole tal vez un nuevo margen de vida o solucionando algunos problemas con los que no se contaba. Si Vd. tiene la suerte de poseer uno de estos equipos, entérese también de cómo conseguir las nuevas versiones del BIOS. Y para terminar, aclaremos cómo es posible cambiar valores en un componente de ROM. Por supuesto, los nuevos valores que elijamos en la configuración no pueden almacenarse en la ROM, que está “fundida” de una vez por todas. Se guardan en realidad en una zona especial de RAM, de bajo consumo y susceptible de mantenerse siempre alimentada con una batería situada en la propia placa, que se denomina CMOS-RAM o “memoria CMOS” a secas. También se almacenan allí, por ejemplo, la fecha y la hora, y puede estar seguro de que Vd. se topará inevitablemente con ella si alguna vez se decide a sumergirse en las interioridades del PeCé.

El proceso de arranque continúa tras la lectura de los características del hardware que se encuentran en la BIOS y la CMOS-RAM del modo que vimos hace ya algunas hojas. Pero no termina como entonces se dijo. Tras cargar nuestro viejo COMMAND.COM, el programa de interpretar y ejecutar órdenes MS-DOS, no aparece el prompt y a funcionar. Y como vamos a ver, por muy buenas razones. La mayor parte de la configuración está aún por hacer, debe completarse antes de ponernos a trabajar, y a ser posible de forma automática. Podría dejarse que fuera el usuario el que, cada vez que arrancara el ordenador, fuera añadiendo cosas a través de las correspondientes órdenes del teclado. Pero sería cuando menos incómodo, ya que el proceso de arranque se alargaría y complicaría sobremanera¹²⁴. La respuesta correcta es almacenar la secuencia de órdenes que el PeCé debe ejecutar para terminar la configuración en unos ficheros que se ejecuten automáticamente cada vez que el ordenador arranque¹²⁵. Al fin y al cabo los ficheros eran para guardar cosas. Y nada mejor

123 Algunas BIOS incluso tienen a bien estar en castellano, contar con textos de ayuda razonables, y hacer correctamente un buen montón de autodiagnósticos. Son más frecuentes en ordenadores de marca, en los que la mayor integración de la placa base permite además que den acceso a otras opciones de interés. Por ejemplo, establecer las frecuencias de refresco a que debe trabajar la tarjeta gráfica.

124 Es más, sería imposible, ya que según que drivers no permiten su instalación una vez que COMMAND.COM ha entrado en contacto con el usuario. El CD-ROM por ejemplo no puede instalarse en su totalidad de esta forma (aunque sí en su parte final).

125 ¿Ficheros? ¿Por qué no uno sólo?. Si se deja todo en un único fichero, se corre el peligro de que acabe siendo demasiado gordo. Y en él se mezclarán probablemente órdenes de propósito muy diferente. Agrupándolas de acuerdo con su finalidad en ficheros distintos se gana en claridad.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

que guardar un montón de órdenes tal vez bastante complejas y siempre necesarias para evitar el tener que estar a vueltas con ellas cada vez que arranquemos.

¿En qué formato se guardan y en qué ficheros?. A la primera parte de la pregunta una respuesta que Vd. debería ser capaz de dar solo. En ficheros .BAT, que eran justo los que guardaban órdenes DOS para que el ordenador las ejecutara cuando fuera preciso. Y a la segunda parte una respuesta que vuelve a traer a la palestra a un viejo conocido. En MS-DOS, se reparte el trabajo de configuración durante el arranque entre dos ficheros diferentes. El primero en ejecutarse es el espero que ya familiar CONFIG.SYS, que se encarga ante todo de añadir y configurar drivers adicionales. Y tras él le llega el turno a AUTOEXEC.BAT, donde se encuentran las órdenes para adaptar el PeCé a nuestro gusto¹²⁶. No debería ser necesario decir que, para que COMMAND.COM pueda hallarlos con facilidad, ambos deberán encontrarse en el directorio raíz del disco de arranque. Es por eso por lo que empleamos a CONFIG.SYS como ejemplo en las operaciones de copia, sabiendo que estaría allí.

Como seguro que se muere de ganas de ver la pinta que pueden tener tales ficheros, y puesto que conviene que al menos los conozca un poquito, además de que muchos puntos que hemos dejado pendientes requieren de ellos para su explicación, veamos un ejemplo de cada uno.

Empezaremos por CONFIG.SYS. La extensión .SYS, un poco especial, tan sólo indica que su cometido consiste ante todo en cargar drivers adicionales, que es un fichero “de sistema”. Por lo demás es un fichero “batch” normal, con órdenes en ASCII¹²⁷, tal como puede verse en el siguiente ejemplo. El CONFIG.SYS del ordenador en el que estoy escribiendo esto, es cual sigue¹²⁸:

Además puede establecerse un orden de ejecución, de modo que las operaciones de configuración más próximas al hardware (adición de nuevos drivers, configuración de la memoria y otros etcéteras) se ejecuten necesariamente en primer lugar. MS-DOS no es muy partidario de cambiar las cosas de sitio en la memoria una vez que se han cargado y conviene, mejor dicho es necesario, que los drivers de monten y se dejen en paz en lo sucesivo.

126 Como veremos pronto, la separación no es ni mucho menos tan drástica. Algunos drivers se montan desde AUTOEXEC.BAT, por ejemplo.

127 Empero, y como tal fichero “de sistema”, las órdenes que contiene no suelen poderse ejecutar directamente desde el teclado.

128 Ahora sí, debo reiterar que estoy en un ordenador con Novell DOS 7. El conjunto de drivers y sus parámetros no son directamente utilizables en MS-DOS. Si desea información exacta para su equipo, deberá recurrir al manual de su DOS.

Vade retro, Satán

```
DEVICE=C:\NWDOS\EMM386.EXE FRAME=NONE E=C000-CBFF ROM=NONE
      rem DEVICE=C:\NWDOS \EMM386.EXE /W
DEVICE=C:\NWDOS\DPMS.EXE
DEVICE=C:\NWDOS\SETVER.EXE
DEVICE=C:\NWDOS\DBLBUF.SYS

DEVICE=C:\ADAPTEC\ASPI4DOS.SYS /D
DEVICE=C:\ADAPTEC\ASPICD.SYS /d:MSCD001

DOS=HIGH,UMB
SHELL=C:\COMMAND.COM C:\ /E: 512 /P
BREAK=OFF
BUFFERS=60
FILES=60
FCBS=10,10
LASTDRIVE=Z
HISTORY=ON,1024,OFF

COUNTRY=34,,C:\ NWDOS\COUNTRY.SYS
      rem DEVICEhigh=C:\NWDOS \STACHIGH.SYS
DEVICEHIGH=C:\NWDOS\ANSI.SYS
      rem STACKS 9,256
```

No se deje intimidar por el fiero aspecto de lo anterior. Vd. ya sabe que “eso” deben ser ante todo drivers. Como puede ver, muchas líneas empiezan con “DEVICE” o “DEVICEHIGH”, seguido de un nombre de fichero con la extensión .SYS o .EXE. Tales órdenes no hacen sino cargar un driver. Tal vez el más importante de ellos en la actualidad es el primero que aparece, EMM386¹²⁹. Y lo es porque se encarga de manejar la memoria, operación fundamental donde las haya y la más proclive a crear conflictos. El montón de parámetros de que va seguida debería hacerle sospechar que hace falta que se configure bien para que las cosas funcionen correctamente, y en efecto así es. Verá que hay tres versiones más que empiezan con “rem”. Por ahora ignórelas.

Tras EMM386, otras dos órdenes “DEVICE” se encargan de montar dos drivers un poco especiales. El primero, “DPMS”, permite el acceso a los “servicios en modo protegido de DOS” (en inglés “DOS protected mode services”), y el segundo, “SETVER” (abreviatura de “set version” en inglés, es decir “establece la versión del DOS”), se encarga de engañar cuando haga falta a aquellos programas que controlan la versión del DOS sobre la que se están ejecutando.

Los tres “DEVICE” siguientes están relacionados con la existencia de un CD-ROM. Si quiero que el mío funcione correctamente, deben estar allí. El primero monta un sistema de “doble buffer”. El segundo, puesto que está conectado a un bus SCSI, instala el driver ASPI, de utilidad general en tales buses y que ya citamos de pasada en su momento. Y el tercero, usando dicho driver general, monta a su vez el propio de un CD-ROM. La instalación del CD-ROM, de todos modos, no ha terminado aún.

Desde “DOS...” hasta “FCBS...” lo que hay son órdenes relacionadas con el funcionamiento básico de DOS. Se cargan la mayor parte de los programitas que lo componen en memoria

129 En MS-DOS 6.22, el papel del EMM386 de Novell DOS 7 se reparte entre dos drivers distintos. A saber: HYMEM.SYS y EMM386.EXE.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

alta¹³⁰ para dejar lo más libre posible la zona de los 640 KB iniciales, se selecciona COMMAND.COM como nuestro programa de comunicación con el PeCé, se definen cuantos ficheros vamos a poder tener abiertos, y poco más.

La orden "LASTDRIVE" (último disco) establece que, en nuestro caso y luego veremos porqué, DOS debe admitir letras de unidad hasta la Z. A su vez, "HISTORY", traducible como "historial", activa en Novell DOS 7 la posibilidad de recuperar órdenes dadas con anterioridad desde el teclado y de editarlas con facilidad, al igual que antes lo hacía en DR-DOS. Como dijimos en su momento, es necesario si uno no se quiere desesperar al trabajar con DOS¹³¹. "COUNTRY", que traducido al castellano sería "país", le indica a nuestro PeCé que está en España y debe intentar adaptarse al sol, la paella, y especialmente a los usos y costumbres sobre horas, fechas y monedas, entre otros. Y "ANSI", abreviatura de "American National Standards Institute", le enseña, quien lo diría, cómo sacar caracteres en colorines o desplazar el cursor con soltura por la pantalla. ¿Sorprendido por el nombre, acaso?.

No hay más, aparte de una serie de líneas que empiezan con "rem" y que están indentadas. ¿Para qué sirven?. La configuración, ya se ha dicho, requiere hacer probatinas. Uno coge un editor de texto adecuado y modifica CONFIG.SYS para ver si soluciona un problema. Pero no sabe de antemano si el cambio va a solucionar algo o no. En lugar de borrar una orden, es mejor decirle al ordenador que la ignore. Así no tendremos que volver a escribirla con todos sus parámetros si al final resulta que es mejor dejarla. Cualquier línea que empiece con "rem" (abreviatura del inglés "reminder", traducible como "nota") es para nuestro COMMAND.COM un comentario, y la ignorará, como dijimos que también debía hacer Vd. en un principio.

Como puede ver, mi CONFIG.SYS tiene una serie de probatinas, o mejor dicho alternativas, que por ahora están desactivadas a base de un "rem" inicial. Lo de ponerlas indentadas es una costumbre para separarlas claramente de la parte de CONFIG.SYS que sí se ejecuta en este momento. Si me veo en la necesidad de modificar mi configuración, Dios no lo quiera, es cuestión de activar unas líneas y desactivar otras, o como mucho añadir alguna nueva. El agrupar líneas en bloques es otra costumbre para asociar visualmente las órdenes relacionadas entre sí. A mí me ayuda, tal vez a Vd. no. Cuando se meta a modificar sus ficheros de configuración, es bastante probable que también Vd. acabe desarrollado sus propias manías.

Tras toda la batería anterior, ejecutada automáticamente por el PeCé en el momento de arrancar, tenemos un PeCé potencialmente más útil, pero todavía no en orden de marcha. Aún no hemos acabado con los drivers, y no está de más adaptarlo a nuestros gustos en lo posible. Ha llegado la hora de AUTOEXEC.BAT. Veamos uno, de nuevo el del ordenador en el que escribo, pues no hay porqué ir más lejos:

130 La "memoria alta" (high) y los "upper memory blocks" ("umb" o bloques de memoria superior, por decir algo) son dos de los múltiples recursos puestos a nuestra disposición por EMM386, que se ha cargado con anterioridad. No entraremos en detalles. Baste saber que se trata de cargar los drivers y el núcleo de DOS en zonas de memoria alejadas de los primeros 640 KB, los que necesitamos para los programas que vamos a usar para hacer cosas de verdad.

131 En MS-DOS se hace con "DOSKEY", que se coloca en AUTOEXEC.BAT o se ejecuta directamente desde teclado.

```
@ECHO OFF
      rem C:\ADAPTEC\MSCDEX /D:MSCD001 /M:12 /L:R
C:\NWDOS\NWCDEX.EXE /L:R /V /D:MSCD001

:NWDOSBEG
PATH C:\NWDOS;C:\NWCLIENT; C:\;C:\WINDOWS
PATH=%PATH%;C:\GMKW ;D:\PCTOOLS;D:\PKZ ;C:\CPACK;C:\PDL
PATH=%PATH%;C:\ODAPI
SET PIMS=C:\CPACK
SET GMKW5=C:\GMKW
SET WINFORTH=D:\4TH\LMI

SET LMOUSE=C:\MOUSE
C:\MOUSE\CLOAKING.EXE
C:\MOUSE\mouse.exe
      rem C:\MOUSE\mouse.exe AHIGH S09 DUAL
KEYB SP+

      rem SHARE /L:20
SHARE /L:500 /F:4096
      rem SHARE /L:500 /F:5120

VERIFY OFF
PROMPT [$P]
SET TEMP=C:\NWDOS\TMP
IF NOT DIREXIST %TEMP% MD %TEMP%

DISKMAP C: D: E: F:
SET NWDOSCFG=C:\NWDOS\CFG

      rem C:\CPACK\DC95.EXE
loadhigh C:\CPACK200\LLRA1

      rem NWCACHE 6144 1024 /LEND=ON /DELAY=ON
NWCACHE 2048 2048 /LEND=OFF /DELAY=ON
:NWDOSEND
      rem WIN
```

La primera línea le resultará un pelín sorprendente. Lo que le dice a COMMAND.COM es que tenga a bien no mostrar en pantalla la orden del fichero AUTOEXEC.BAT que va a llevar a cabo, empezando por ésta. Así aparecen tan sólo los mensajes producidos por dichas órdenes, con lo que normalmente ya basta¹³². La siguiente, “NWCDEX”, acaba de cargar los drivers del CD-ROM. El nombre viene de “Novell CD Extensions”, y es más o menos equivalente a “MSCDEX”, abreviatura a su vez de “Microsoft CD Extensions”. Tras ejecutar esta orden, el CD-ROM está en condiciones de funcionar, o sea que aparece como una unidad más del DOS. El parámetro “/L:R” obliga justamente a que aparezca como unidad “R:”. Reconozco que es una manía, me gusta que el CD-ROM se llame así¹³³, y al fin y al

132 Si Vd. está en la fase de probatinas diversas (“depuración”, se llama a eso), tal vez le interese que aparezcan las órdenes. Un “rem” ante el “@ECHO OFF”, y problema solucionado. Y si Vd. quiere engañar a alguien y hacerle creer que su PeCé arranca directamente en Windows, añádale a todas las líneas un “mayor que NUL:” y acabe su AUTOEXEC.BAT con un “WIN”. La pantalla no dirá esta boca es mía mientras se ejecute AUTOEXEC.BAT.

133 Elijo la “R” de CD-ROM para designar la unidad. La “C” y la de “D” no se pueden usar, como

CAPÍTULO 7. PeCés: MS-DOS/Windows.

cabo AUTOEXEC.BAT está para poner el ordenador a nuestro gusto. ¿O no?. Esto explica, siquiera en parte, la presencia en CONFIG.SYS de la orden “LASTDRIVE=Z”¹³⁴.

Verá a continuación una serie de líneas que no hacen más que definir el “PATH”, la serie de directorios en que DOS va a buscar órdenes él solito. Ya hablamos de la importancia de la orden y es aquí, en AUTOEXEC.BAT, donde conviene establecerlo para que siempre que arranque el ordenador, nos encontremos a LR-PeCé sabiendo donde están los programas de uso más inmediato. Tal vez le sorprenda lo de “PATH=%PATH;...”. No quiere decir otra cosa que añada los nuevos directorios a los que ya conoce. Resulta bastante práctico.

El grupo de “SET lo que sea” no es sino un bloque que define variables propias de una serie de programas, y que dichos programas usan para localizar sus componentes. Aparecen luego varias líneas con “MOUSE”. Se encargan de montar el ratón en DOS¹³⁵. En realidad es la línea “C:\MOUSE\MOUSE.EXE” la que instala el driver. La primera es un SET sin otro fin que el ya dicho, y la del “CLOAKING” se dedica de nuevo a poner en marcha algunos trucos para que el driver del ratón no ocupe mucha memoria en la zona de los primeros 640 KB.

“KEYB SP+” es otra de las órdenes pendientes que teníamos por ahí, dejadas de la mano de Dios. Como tenemos un teclado en castellano, hay que decírselo a DOS, para que cuando pulsemos la “ñ” no nos salgan “:”. Si Vd. no incluye el equivalente para su DOS, ya que la versión de Novell o DR-DOS es un poco particular, su teclado no funcionará como teclado castellano. Es obvia la necesidad de meter tal orden en AUTOEXEC.BAT.

“SHARE” se encarga de activar la posibilidad de que varios programas accedan a la vez a un mismo fichero, “VERIFY OFF” le indica a LR-PeCé que no compruebe las escrituras que haga en disco, para no perder tiempo y porque no suele hacer falta, y “PROMPT” es otra de las cosas de las que dijimos que hablaríamos.

Por supuesto, se encarga de establecer la forma del “prompt” en DOS. Bástele saber que es necesario, como dijimos, que LR-PeCé nos muestre siempre el directorio en el que estamos, y que eso se logra con el parámetro “\$P”, que no es sino una abreviatura de “path”. Tal cual está aquí, el “prompt” aparece tal que así:

```
[directorio actual]_
```

Una vez más, a mí me gusta así. Lo más normal es que a Vd., en cambio, le salga algo como:

```
directorio actual>_
```

debería ser obvio.

134 Por defecto es “LASTDRIVE=E”, y en ese caso no puede haber otras unidades que de la “A:” a la “E:”.

135 Windows monta automáticamente el ratón para su propio uso, esté o no instalado en DOS. Lo del ejemplo es para poder usar el ratón en programas DOS, que también los hay que lo emplean, no se vaya Vd. a creer.

y eso quiere decir simplemente que su orden “PROMPT” irá seguida ni más ni menos que por el parámetro “\$P\$G”. Si Vd. quiere enredar con el prompt y hacer que el ordenador se ponga a sus pies cada vez que le pida una orden, mostrándole algo similar a:

```
[directorio actual] Ordena y obedezco. ¡Oh, amo!: _
```

la cosa es tan simple como cambiar la orden para dejarla tal cual sigue: “PROMPT [\$P] Ordena y obedezco. ¡Oh, amo!: ”¹³⁶.

Las dos órdenes siguientes crean una nueva variable y, caso de no existir previamente, un directorio. También LR-PeCé necesita usar ficheros cuando debe trabajar. Con estas órdenes establecemos su directorio de trabajo, de modo que deje allí los ficheros temporales (de usar y tirar) que haga falta. El “IF NOT DIREXIST...” es un ejemplo de las órdenes de control de ejecución que todo lenguaje de programación necesita incluir. Lo que hace es conminar a LR-PeCé a crear el directorio temporal indicado en la anterior orden “SET” en caso de que no exista previamente.

La orden “DISKMAP” la tratamos no hace mucho, en relación con la recuperación de ficheros borrados. Es mi método favorito, no debería repetirlo, y lo aplico a todas las unidades de disco duro presentes en mi equipo. No, no tengo cuatro discos. Son dos discos duros con particiones, para definir cuatro unidades.

De lo que queda, poco en realidad, sólo debemos comentar un “LOADHIGH” y otro par de cosas. Una vez más se trata de un truco para desocupar la zona de los primeros 640 KB, que como ve hay que mimar. El programa que carga es otro driver, cuyo cometido no viene a cuento¹³⁷. “NWCACHE” es fundamental para el rendimiento del ordenador. Instala un caché de disco duro más que razonablemente eficaz con el fin de hacer las lecturas y escrituras de datos mucho más rápidas¹³⁸ que no es ni más ni menos que la orden para entrar en Windows. Tras pulsar el interruptor y cacharrear un rato, el ordenador nos dejaría confortablemente enfrentados a la interfase gráfica de usuario típica de los PeCés y con la configuración hecha. Ni veríamos el prompt. También esto es adaptar el ordenador a nuestros gustos, que es el cometido principal de AUTOEXEC.BAT. En este momento (otras veces no ha sido así, y por eso tengo lo del “rem win”), yo prefiero acabar en DOS. No es tan grave escribir “WIN” y darle a la tecla gorda de la derecha esa.

136 Todos hemos enredado con estas cosas. El asunto pierde interés cuando uno se da cuenta de que LR-PeCé no es más sumiso por mostrarnos semejante “prompt”, y de que, en caso de engañar a alguien, nos estamos engañando a nosotros mismos. Suele acabarse volviendo a un “prompt” normalito, que nos muestre el directorio actual y no incordie mucho. Como puede ver, incluso en el “prompt personalizado” anterior, le recomiendo que no olvide los detalles de auténtico interés en la práctica.

137 Ya sé que le gustaría saber para qué es. Resulta que por motivos de trabajo estoy en un grupo de gente que usa pequeños PeCés de bolsillo, en concreto los HP-95LX y HP-200LX. El driver en cuestión permite conectar en cualquier momento y con suma comodidad un HP-200LX a mi ordenador de mesa a través de una puerta serie, con el fin de pasar datos de uno a otro.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Una vez más, casi casi hemos acabado con DOS. Tan sólo, de nuevo como casi siempre, algunas notas, tal vez muchas en esta ocasión. Por ejemplo: ¿qué pasa si por descuido o por lo que sea, no tenemos un CONFIG.SYS y un AUTOEXEC.BAT en nuestro directorio raíz?. No se preocupe, el ordenador todavía arrancará. Eso sí en un estado no configurado, por defecto, y donde no tendrá ratón, ni CD-ROM, ni PATH... ni siquiera teclado en castellano. Es en este momento cuando deseará conocer donde están en un teclado USA (el seleccionado por defecto) los caracteres interesantes en DOS, a saber “\”, “/”, “*”, “:”, “-”, “_”. Ya lo vimos, tiempo ha, y los necesitará para decirle a LR-PeCé lo que sea, entre otras cosas algo así como “C:\NWDOS\KEYB SP+” con lo que el teclado entrará en razón rápidamente. Tampoco tendrá un historial de las órdenes que desde el teclado vaya dando. No le vendrá mal saber que al menos hay algunas teclas a su disposición, incluso en tan triste estado. Por defecto, “F3” recupera la orden anterior (sólo la última, no hay más), cursor izquierda o la tecla de retroceso llevan el cursor hacia atrás borrando para corregir si hace falta, y cursor derecha recupera el resto de caracteres. Aunque poco, algo es algo.

Normalmente, cuando se arranca de esta forma, DOS nos pide que confirmemos la fecha y la hora antes de mostrarnos el “prompt”. Esto no hace sino confirmar lo que en su momento dijimos. Conviene que se asegure de que la fecha y la hora son correctas antes de trabajar. En caso de querer incluir en su versión particular de AUTOEXEC. BAT las órdenes correspondientes para hacerlo así, sepa que son “TIME” y “DATE”.

¿Y qué pasa si me pongo a enredar con CONFIG.SYS y AUTOEXEC.BAT y lo que dejo es tan intragable que el ordenador no llega a arrancar?. Aclaremos que es un caso perfectamente posible, especialmente si toca según que cosas de EMM386.EXE. Hay varias respuestas para situación tan desesperada. La clásica consiste en recurrir a los discos del sistema operativo. Inserte el disquete de arranque en la unidad A: y aunque tenga un verdadero desastre en C:, su PeCé se pondrá en marcha desde el A:. ¿Ve para que valen los disquetes de arranque?. La clásica reformada consiste en hacer lo mismo con un disquete de emergencia a nuestra medida, donde hayamos dejado previamente todo lo necesario para resolver el chandrío. Y la tercera usa un truco de reciente aparición. Si pulsa “F5” mientras el ordenador arranca, el DOS ignorará a CONFIG.SYS y AUTOEXEC.BAT y terminará en el mismo estado que si no estuvieran, es decir la configuración por defecto. Pero podrá volver a editar sus ficheros de arranque y recuperar el control. Si Vd. quiere saber cual es la orden concreta que está creando los problemas, pulse “F8” en lugar de “F5”. Con esto, COMMAND.COM sí que leerá CONFIG.SYS y AUTOEXEC.BAT, pero lo hará línea a línea, mostrándosela y preguntándole si quiere que la ejecute o no. Lo de estas dos teclas de función es otra bendición de los cielos de reciente aparición. Vaya nuestro agradecimiento para Microsoft, que es la casa que creo que lo introdujo en su MS-DOS 6.0¹³⁹.

Y ya que hablamos de modificar ficheros tipo texto, podríamos precisar un poco cómo se hace. Hace ya tiempo que cualquier versión presentable de DOS incluye un programa al menos decente para ello. Suele llamarse “EDIT”, o algo similar. Por mi parte sigo utilizando el que se incluía en DR-DOS 6. Se llamaba “EDITOR .EXE”, y siempre me pareció un

139 Desconozco la gama del PC-DOS de IBM, y por tanto no puedo asegurar que tal opción no haya estado disponible desde la noche de los tiempos en sus versiones del DOS, o que no sea algo tomado de OS/2 o de cualquier otro programa de cualquier otro fabricante. Lo que sí que es seguro es que no estaba en DR-DOS 6. Por suerte, las buenas ideas se difunden con rapidez.

nombre demasiado largo. En consecuencia, hace ya bastante tiempo que le cambié el nombre y lo dejé en “E.EXE”. Novell DOS 7 lo cambió por un “EDIT.COM” que trabaja en modo gráfico y soporta ratón. Yo copié mi “E.EXE” al directorio correspondiente¹⁴⁰ y sigo modificando los ficheros en ASCII con una orden similar a “E CONFIG.SYS”. Todo esto viene a cuento de que nadie le obliga a renunciar a esos programas que Vd. usa con agrado cuando sale una nueva versión del sistema operativo. Y que, puesto que pueden añadirse órdenes a DOS, no hay inconveniente en que lo haga a su gusto y en función de sus necesidades.

Puede tener también interés explicar porqué prefiero mi E.EXE. Aunque de aspecto rústico, es más que sobradamente potente para lo que debe hacer¹⁴¹. El ratón no hace falta para nada a este nivel, ni las pantallas multicolores. Tiene, eso sí, unas ayudas adecuadas. ¿Cuántas y cómo son?. ¿Recuerda lo del “CTRL+J para ver la ayuda” del capítulo 5?. ¿Y los ejemplos de textos de ayuda para desplazamiento del cursor por la pantalla?. Pues estaban tomados ni más ni menos que de él. Por supuesto, permite emplear las teclas de cursor para moverse en pantalla, no sólo reconoce caracteres de control. Y una de sus grandes ventajas es que ocupa muy poco sitio, tan sólo 21 KB. Si queremos montar un disquete de emergencia a nuestra medida, debe contener un buen montón de cosas. Los ficheros básicos de sistema (IBMDOS.COM, IBMBIO.COM, COMMAND.COM, el MBR...), los drivers necesarios para que nuestro ordenador arranque al completo quizá con excepción del ratón (KEYB.COM, EMM386.EXE, compresor de disco, los drivers relacionados con el CD-ROM...), todo un conjunto de órdenes externas capaces de recomponer un disco duro (FDISK.COM, SYS.COM, FORMAT.COM, tal vez DISKOPT.EXE...), más órdenes externas para actuar sobre ficheros y directorios a cualquier nivel (XDIR.EXE, XCOPY.EXE, XDEL.EXE...), por supuesto los correspondientes CONFIG.SYS y AUTOEXEC.BAT convenientemente adaptados a arrancar desde disquete, y algunas cosas más, tal vez por ejemplo un programa de detección y a ser posible eliminación de virus informáticos¹⁴². Todo eso ocupa sitio, no tenemos más que un disquete de 1'44 MB para meterlo, y no es posible renunciar a un editor de textos adecuado. En resumen, E.EXE cumple sobradamente, y por eso lo sigo usando.

Pues nada, hombre. Ya hemos terminado nuestro cursillo acelerado de hechicería DOS. Espero que al menos sepa algo más que cuando empezamos. Si quiere algún tipo de prueba, una especie de proyecto fin de carrera o tesina de licenciatura que le permita asegurarse de que es Vd. ya un casi experto en DOS, la cosa es sencilla. Cuando se sienta con fuerzas, confianza, y sabiduría suficientes como para meterse a modificar Vd. solito y con buenos resultados la configuración de la BIOS, el CONFIG.SYS, y el AUTOEXEC.BAT de su PcÉ, podrá considerar que ha terminado con nota. Y no se preocupe, se dará cuenta inmediatamente y sin error posible de que sabe lo necesario.

140 ¿Cual será?. Un positivo para el que lo adivine. Lea atentamente el AUTOEXEC.BAT... ¿Aún no?. Bueno, otra pista. Fíjese atentamente en la orden PATH... ¿Qué, todavía no lo adivina?.

141 Por ejemplo, es posible incorporar el contenido de un fichero cualquiera al texto que estamos editando, operación sumamente interesante. El “EDIT” de MS-DOS 5 no dejaba hacerlo, aunque, eso sí, tenía muchos más colorines y aceptaba el ratón.

142 Ya hablaremos de ellos en su momento. O en el glosario.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

He aquí algunas pistas para ayudarle en su empeño. Los problemas, especialmente en el caso de imposibilidad de correr determinados programas, falta de espacio en los primeros 640 KB, cuelgues imprevistos y similares, suelen muchas veces deberse a una configuración incorrecta de EMM386.EXE, es decir de la gestión de memoria. La orden “MEM” le ayudará a diagnosticarlos. Por cierto que es con esta orden, en concreto en su forma “MEM /A”, con la que sacamos el listado de los drivers en memoria de nuestro PeCé, que mostramos un par de veces en capítulos anteriores. Las teclas “F5” y “F8” le ayudarán asimismo muchísimo a diagnosticar problemas de configuración. Los drivers adicionales (ratón, CD-ROM, etc...) son otra fuente potencial de incompatibilidades. No se olvide de cargar un buen programa de caché de disco si quiere que su PeCé funcione con rapidez, y de activar algún sistema de recuperación de ficheros borrados para aumentar su seguridad. Y si tiene problemas con un programa concreto, recuerde que muy probablemente existe un “Read.me” o “Leeme” o similar en alguna parte. Léalo, es probable que incluya información interesante.

Y un último consejo para terminar de verdad el apartado. El DOS de los PeCés, en cualquiera de sus versiones actuales, no es una maldición. Al contrario. Es un sistema operativo simple, fácil de comprender y más que razonablemente capaz a nivel personal. A nada que, como con los directorios, lo mire con cariño y le dedique un poco de tiempo para aprenderlo con algún detalle, le demostrará que como mínimo es un sólido refugio donde acudir en caso de apuro. No es tan fiero el teclado como lo pintan. Y si la interfase de usuario de su DOS no le acaba de convencer, la solución no es olvidarse de él, sino cambiarla por otra. Tal vez otra versión de DOS, o simplemente otro intérprete de línea de órdenes mejor dotado que su COMMAND.COM¹⁴³, basten para que Vd. no tenga que renunciar a usarlo.

La sustitución de COMMAND.COM por otro interfase de usuario es práctica habitual en la actualidad. Si de verdad quiere alejarse lo más posible del teclado, ha llegado el momento de comunicarse con LR-PeCé a través de la ventana...

Voglio entrare per la finestra

LR-PeCé y sus ventanas

A partir de determinado momento en la historia del PeCé, tal vez coincidente con la aparición del Apple Macintosh, ésta fue la consigna, muchas veces explícita, de un montón de usuarios¹⁴⁴. Por si no se apaña ni mínimamente en italiano, lo traduzco: “Quiero entrar por

143 Seguro que Vd. se ha dado cuenta hace rato de que COMMAND.COM es el programa que implementa la IU del DOS. Es perfectamente posible sustituirlo por otro programa más acogedor. Me permito recomendarle para tal fin a 4DOS.COM, un programa distribuido a través de shareware (en el próximo capítulo veremos qué es eso) y muy difundido en la actualidad. Está en inglés, pero se encarga de demostrar en la práctica y de forma irrefutable que una interfase de línea de órdenes puede ser un buen sitio para trabajar. El directorio a cuatro columnas del apartado “Abracadabra” es una ínfima muestra de un montón de posibilidades adicionales. El DIR de 4DOS.COM admite un parámetro /4 que muestra el directorio en este formato.

144 “Voglio entrare per la finestra”, de la ópera “Leonora o el amor con juglar” de Giovanni Corporcorto. Versión del tenor Sergio Tulian, coro y orquesta del Cotelengo de Santa Eduvigis, bajo la dirección del autor. El juglar Ludovico trata de escalar el muro de su amada Leonora y cantarle su aria de amor... él sólo quiere cantarle su aria de amor y solamente eso... aunque cada

la ventana”. En realidad no eran sólo las ventanas. Lo que los sufridos seguidores del PeCé querían decir era que tal vez hasta asesinarían a alguien a cambio de un GUI, una interfase de usuario moderna que al menos les permitiera mantener el decoro cuando discutieran con los afortunados poseedores de un Mac sobre las respectivas virtudes de sus ordenadores. La IU del PeCé era entonces realmente espantosa, ni más ni menos que una línea de órdenes de lo más rural, y el GUI del Mac no estaba pero que nada mal.

No nos extenderemos ahora con un análisis comparativo de ambos tipos de IU. Ya lo hicimos en un capítulo anterior, y probablemente Vd. recordará que, tras una larga espera, los deseos de los poseedores de un PeCé se hicieron realidad. En 1.990 apareció Windows 3.0, y poco más de un año después y tras ser profundamente revisada para dar paso a Windows 3.1, se hizo finalmente posible que un PeCé fuera probablemente tan cómodo de usar como un Macintosh¹⁴⁵. Como también dijimos entonces, es aquí donde nos encontramos en este momento, y procede por lo tanto acabar el capítulo sobre utilización de un PeCé viendo qué pasa con Windows. En realidad, ya no estamos exactamente en Windows 3.1. Hace cosa de un par de años salió Windows 3.11, y en agosto (E.E.U.U.) o septiembre (Europa) de 1.995 se comenzó a vender la última encarnación de Windows por el momento, conocida como Windows 95. Ahora mismo (diciembre de 1.995) coexisten todas las versiones citadas y otra serie de GUIs incorporados a otros sistemas operativos igualmente disponibles para PeCé, entre los que debemos necesariamente citar a OS/2 y Windows NT (ver glosario). Dado que por diversos motivos Windows 3.1 sigue siendo probablemente la más popular, y que como interfase de usuario es casi idéntica a Windows 3.11 (la segunda en difusión tal vez) y Windows NT, nos centraremos en ella. Eso sí, comentaremos algún aspecto de la que se propone como su sucesora, Windows 95.

Antes de meternos en faena (que, como verá, podría por diversos motivos ser calificada en términos taurinos como “de aliño”) conviene precisar categóricamente algunos puntos, y ese es ni más ni menos que el objetivo principal de este apartado. Cabe la posibilidad de que Vd. tenga el atrevimiento de creer que, puesto que estamos cómodamente instalados desde hace un lustro en el mundo de los GUIs, ningún interés tiene el ocuparse del mundo del DOS sin Windows. Quizá tenga la osadía de opinar, tal vez hasta en alta voz, que lo que se ha tratado de explicar y hacer inteligible con sumo esfuerzo e inciertos resultados en las numerosas páginas ya consumidas en el presente capítulo, no es sino una lamentable pérdida de tiempo que demuestra irrefutablemente, en caso de no estar ya sobradamente establecidas, la insanía mental y tremenda pesadez del autor. Tal vez crea Vd. que DOS ha muerto y nada sobre él debe conocerse. Quizá hasta se haya aventurado, con intrepidez y falta de conocimiento sin límite, a saltarse el montón de páginas anteriormente citado y venir directamente aquí para aprender lo que se diga de Windows y nada más.

Si ese es el caso, debo decir tan sólo que parece confirmarse una vez más que el dicho popular “¡Qué atrevida es la ignorancia!” está cargado de razón. Sepa, atrevido amigo, que la

vez piensa menos en cantar y más en “solamente eso”. “Les Luthiers”, por supuesto.

145 Esta afirmación da todavía pie a interminables discusiones entre ambos bandos en lo que sin duda es una guerra de religión (ver “Fervor religioso” en el glosario). Los devotos del Mac suelen opinar que de eso nada. Los del PeCé afirman fieramente que se queda corta ya que Windows es mucho mejor.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

estructura de este texto no está pensada para saltarse capítulos enteros y comenzar o continuar la lectura donde mejor parezca, sino para seguirlo línea a línea. Sepa, osado lector, que Windows no es más que una de las múltiples interfases de usuario de que se ha dotado a DOS y por extensión a los PeCés, a lo largo de su historia, y que sin conocer el otro 50% de la estructura del software del PeCé, lo que nosotros hemos denominado globalmente “sistema de ficheros”, crudo lo tiene para aclararse con un chisme de estos por mucho que conozca Windows. Al fin y a la postre Windows puede verse como un sustituto más, gráfico eso sí, de nuestro viejo conocido COMMAND .COM. Sepa, impaciente aprendiz de hechicero, que si de verdad quiere aprender Windows con un mínimo de sentido, hay que empezar por dominar el DOS, que es por donde se ha tratado de empezar aquí, y vuelva en consecuencia al lugar que tan imprudentemente abandonó para continuar allí su lectura. Sepa finalmente, poco advertido y tal vez futuro colega informático, que DOS no ha muerto a pesar de lo que diga la gente, y que bastará con que intente realizar el equivalente a un “CHKDSK” o incluso un “UNDELETE” desde Windows, para que se encuentre con un mensaje que le diga que mejor debería hacerlo desde DOS. Cierto, algunos programas permiten hacer un “UNDELETE” desde Windows, incluso en todas sus versiones, pero esos mismos programas le mandarán a DOS cuando intente hacer otras cosas. En resumen, no es por esnobismo por lo que Windows permite abrir “ventanas DOS”, es decir acceder al ordenador en la forma clásica a través de la línea de órdenes aunque sea en una ventana desde Windows. Están ahí porque son necesarias. En mi opinión, todo lo anterior es estrictamente aplicable asimismo a Windows 95¹⁴⁶.

En resumen, con un GUI, por moderno que sea, no estamos sino añadiendo un componente más al sistema operativo. Puesto que las adiciones usan lo ya existente, un GUI depende en buena medida de la base sobre la que se instala, y querer limitarse a aprender lo relativo a un GUI sin querer enterarse de lo que existe por debajo es pretender hacer una casa sin cimientos. Una llamada a un inevitable desastre.

Si Vd. quiere alguna vez poder decir que “sabe manejar PeCés” considere que tal expresión supone bastante más que mover el puntero del ratón por el escritorio de Windows (ya veremos qué es eso). En el capítulo 9 nos ocuparemos del tema con mayor detalle, pero piense por el momento en sus posibilidades de conseguir empleo en el gremio si en una entrevista ante su futuro jefe, que casualmente tiene informatizado su negocio con una red de 50 PeCés sobre DOS y Novell Netware¹⁴⁷ funcionando a completa satisfacción desde hace dos años y con su coste ya amortizado, afirma algo similar a: “Lo que yo domino es Windows. El DOS ya no tiene ningún sentido. Debería Vd. en primer lugar actualizar su

146 La propaganda afirma lo contrario, pero Windows 95 no parece haber cambiado en realidad la estructura base del software del PeCé. Sigue siendo Windows corriendo sobre DOS, aunque la separación entre ellos sea mucho más confusa que antes. Ahí están todavía IO.SYS, MSDOS.SYS, CONFIG.SYS y AUTOEXEC.BAT, junto a alguno más. Y las “ventanas DOS” y los modos “DOS puro” ahora multiplicados en distintas opciones en el momento de arrancar. Y por cierto, en mi opinión, Windows 95 ya no es un buen ejemplo de ocultamiento de información y simulación de apariencias. Es más bien un claro ejemplo de su uso en exceso. Una pequeña muestra es que IO.SYS agrupa ahora lo que antes eran los dos ficheros de sistema, mientras MSDOS.SYS se ha convertido en un fichero de configuración análogo a CONFIG.SYS e innecesariamente oculto. Volveremos a ello en su momento.

147 El estándar en sistema operativo para redes de PeCés.

Más historia, ahora de Windows

parque de ordenadores y renovar el software. Total, con unos 20 millones de pesetas y en cosa de medio año, haríamos lo mismo pero de forma mucho más elegante”.

No hay nada raro en que Windows se monte sobre DOS. No es DOS el único sistema operativo que separa su interfase gráfica de su núcleo. Es posible así disponer en modo texto, con unos requisitos de equipamiento mucho menores, de la mayor parte de los servicios que el ordenador pueda proporcionar. Citemos por ejemplo a Unix, un clásico modelo de sistema operativo potente¹⁴⁸. Lo único extraño, si acaso, es el montón de cosas que hace Windows en un PeCé. ¿Que cuantas hace exactamente?. Para aproximarnos un poco, hace falta conocer cómo surgió. Una vez más, conviene tener en cuenta la Historia.

Más historia, ahora de Windows

De cómo LR-PeCé se dotó de un GUI

Para conocer los orígenes de Windows es preciso retroceder muchos años, casi al principio de los PeCés. Tanto como para tener que aclarar que entonces no se hablaba de GUIs, sino de “entornos integradores”. Tanto como para tener que presentar a Visicorp, el fabricante de la primera hoja de cálculo¹⁴⁹ para microordenadores que tuvo éxito. Se llamaba VisiCalc y se desarrolló inicialmente para el Apple II, lo que debería contribuir a situarnos, siquiera mentalmente, en tan pretéritas eras. Pues bien, Visicorp anunció en noviembre de 1.982 y distribuyó allá por el 1.983-84 un “entorno integrador” para PeCés al que denominó VisiOn. Se vendía a 495 dólares de los de entonces, requería al menos un PC-XT, 256 KB de RAM, un disco duro de 5 MB y un ratón, y pasó a mejor vida con mucha más pena que gloria. Visicorp desapareció poco después. Para que se haga una idea de verdad sobre cuán lejos nos hemos ido, digamos que intentaba imitar al Apple Lisa (ver glosario). El Lisa no era sino el “papá” del Apple Macintosh, ya que este último todavía no había aparecido¹⁵⁰.

Tan temprano furor para desarrollar “entornos integradores” requiere la existencia de poderosas razones de naturaleza poco menos que universal en el mundo informático. Así es, en efecto. Los primeros entornos integradores y todos sus sucesores, entre ellos Windows, se desarrollaron ante todo para intentar dar respuesta a dos problemas graves: la integración de los distintos programas que se usan en el ordenador, y la unificación de su interfase de usuario. ¿Qué es todo eso, pregunta?. Vamos a ello.

148 Unix es multitarea, multiusuario, y tiene un núcleo que trabaja en modo texto. Sólo con eso se pueden hacer un montón de cosas. Si Vd. quiere, y tiene memoria, CPU, tarjeta gráfica y monitor suficientes, puede montar X Windows, su GUI. O mejor dicho, el núcleo de su GUI. Pues sobre él se monta a su vez un gestor del GUI, que es el que termina de definirlo. Y quizá un conjunto de bibliotecas adicionales, como Motif, por poner algún ejemplo. Eso sí, todo modular, con cometidos definidos y claros.

149 Un tipo de programa. En el capítulo siguiente veremos cosas de esas.

150 Tal vez se le ocurra pensar que el atribuir la aparición de Windows a un mero intento de imitar al Macintosh puede pecar de simplismo. Es innegable que existe una cierta imitación, por evidentes motivos comerciales. Pero los GUIs no nacieron con el Mac. Ni siquiera con el Lisa. Y se justifican ante todo como un intento de solucionar graves problemas generales en la utilización de los ordenadores, por lo que no es extraño que cada familia desarrolle el suyo. Ver “PARC” o “Lisa” en el glosario.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Pues bien, la cuestión era la siguiente. Supongamos que, por esas fechas, alguien quería escribir un informe en el que debían aparecer simultáneamente algunas páginas de texto, tres o cuatro tablas numéricas y quizá, de propina, dos gráficos. Los acontecimientos para llevar a cabo con su microordenador tan elemental encargo se desarrollaban como a continuación se indica. El programa TLW 1.0 (fíjese en la versión) le permitía crear el texto, y hasta imprimirlo con un aspecto sumamente rústico y a cambio de un ruido ensordecedor en la impresora matricial de 8 agujas de su mesa. Para manejar tablas de números, Vd. necesitaba otro programa. Digamos que la editora de software que producía con sumo éxito la familia de procesadores de texto TLW, y a la que denominaremos TLSoft en lo sucesivo, tenía también en su gama a TLS, abreviatura de “The last spreadsheet” o “La última hoja de cálculo” (paciencia hasta el próximo capítulo), nombre de nuevo algo tétrico pero que podría valer. Equipado con TLS 1.0, nuestro sufrido usuario del PeCé de aquellos tiempos hubiera preparado sus cuentas. ¿Cómo las hubiera metido en el texto?. Bueno, TLW 1.0, al igual que todos sus coetáneos, no era capaz de leer directamente los ficheros producidos con una hoja de cálculo. Nuestro sufrido informático desconocido hubiera debido recurrir probablemente a exportar los datos para crear otro fichero que contuviera tales cuentas en estricto texto en código ASCII, algo que todo procesador de textos es capaz de leer, e incorporarlas de este modo a su informe. Tras lo cual hubiera debido proceder a darles un aspecto digno, probablemente con bastante dificultad, en TLW 1.0. Los gráficos, por su parte, completaban la odisea. No sólo TLG 1.0 (por supuesto “The last graphics 1.0”, o “El último programa para gráficos 1.0”) resultaba un tanto crudo de uso y requería asimismo que se le pasaran los datos de TLS previamente exportados en ASCII, sino que para conseguir llevarlos a papel hacía falta un cuarto programa, encargado de imprimirlos penosa, lentamente, y con una calidad final más que cuestionable. Y en el documento final, lo único que había de ellos era el nombre del fichero gráfico a incluir en el texto, y un espacio reservado en blanco para pegar la hoja correspondiente.

¿Qué se ganaba con usar todos los programas de la misma casa?. Que los datos exportados en ASCII de un programa fueran utilizables sin mayor manipulación por otro. Si Vd. usaba el programa de gráficos de otro fabricante, podía ser necesario entrar a reestructurar incluso el fichero ASCII exportado desde TLS 1.0, o, simplemente, no había forma humana de pasar los datos. Además de que en TLS 1.0 “F1” daba acceso a las ayudas y en el programa de la competencia se usaba para, inmediatamente y sin ningún tipo de pregunta adicional, borrar todos los datos que hubiéramos podido crear hasta el momento. O sea que los programas se manejaban cada cual como Dios le daba a entender, sin que lo aprendido para uno de ellos sirviera de mucho para los demás¹⁵¹. Puede que lo anterior no sea tan espantoso como lo que le ocurría a nuestro asimismo desconocido usuario de los ordenadores con paneles de conexión de hace algunos capítulos, pero era todavía un triste destino.

Los entornos integradores buscaban ante todo facilitar algo tan necesario como el intercambio de datos entre programas. Que la elaboración de mi informe pudiera hacerse cogiendo una tabla de aquí, un texto de allá y un gráfico del otro lado sin tener que perder tres meses, y con la garantía de que el resultado final fuera digno.

151 “¡Hombre!, igual que ahora”, tal vez se apresure a pensar. Ni mucho menos, querido LAO. Puede que en este momento la microinformática no sea perfecta, pero se ha avanzado bastante.

Más historia, ahora de Windows

Lo de la unificación de la interfase de usuario era dar un paso adicional. Puesto que resolver el problema anterior requería cierto esfuerzo, ¿por qué no aprovechar para que nuestro entorno integrador ofreciera además una serie de recursos que dieran a todos los programas el mismo aspecto general, de modo que todos se manejaran de forma parecida?. Así se reduciría mucho el tiempo de aprendizaje de programas nuevos, ya que los recursos generales y el aspecto global de todos sería análogo. Todos tendrían menús, colocados en el mismo sitio, y si en uno “F1” daba acceso a las ayudas, en los demás también.

A partir de aquí, si se conseguía un entorno integrador adecuado, sería posible manejar de forma similar el procesador de textos de la casa A, la hoja de cálculo de la casa B, y el paquete de gráficos de la casa C, y además intercambiar fácilmente datos entre ellos para acabar con un informe final decente¹⁵². Hacía falta dejar que cada cual seleccionara a su gusto sus componentes y el entorno integrador proporcionara el pegamento para hacer que todos los programas (o “módulos”) se parecieran lo necesario e intercambiaran datos sin dificultad.

En realidad, el problema del intercambio de datos no se resolvió inicialmente en los PeCés mediante el desarrollo de entornos de integración universales, sino con programas “integrados”, que ofrecían en un solo programa la posibilidad de hacer de todo. Es decir que poco después apareció TLOP 1.0, o “The last and only program 1.0” (traducido “El último y único programa v1.0”) que hacía textos, gráficos, y manejaba hojas de cálculo todo a la vez, y era capaz de montar nuestro informe de una sola tacada. No tuvieron éxito a largo plazo¹⁵³, principalmente porque eran demasiado gordos para los tiempos que corrían. Los discos duros de cualquier capacidad eran entonces poco menos que un lujo asiático, la RAM, escasa, y la potencia de las CPUs, miserable. Los integrados no eran la solución real al problema, y a la larga resultaron intragables por lentos e inoperantes, además de que podía ser que un buen módulo de proceso de textos fuera acompañado de habilidades de manejo de gráficos y cálculo punto menos que indecentes. Pero, eso sí, dieron tiempo a que se desarrollaran otros entornos integradores tal vez mejores que, sobre equipos muchísimo más potentes, acabaron por hacer al menos en parte lo que se esperaba de ellos en un principio.

La lista de entornos integradores desarrollados para PeCés es notablemente extensa. Ya en 1.985 podemos citar al menos tres de ellos. A saber, TopView, de IBM (la misma del PC-DOS y el IBM PC); GEM, de Digital Research (la misma del DR-DOS); y Windows, de Microsoft (la misma del MS-DOS). Si deduce de lo anterior que existía una cierta relación entre sistema operativo y entorno integrador, no me queda sino felicitarle, querido LAO. En efecto, en el mundo del PeCé los entornos integradores se plantearon, como no dejaba de ser lógico, como extensiones al sistema operativo base. En resumen, principalmente como extensiones a MS-DOS o alguno de sus clónicos.

152 Y hasta cabía la posibilidad de añadir al invento una cierta independencia del ordenador en concreto. Hacer que los programas desarrollados para VisiOn funcionaran igual sobre un PeCé que sobre un Atari ST, un ordenador no compatible y posterior (ver glosario), siempre que se desarrollara la correspondiente versión de VisiOn para él. Podía intentarse hacer al invento “multiplataforma”, y algo de esto había en las primeras hornadas de GUIs. La historia sólo ha llegado hasta ese extremo, tal vez, en el mundo Unix con X-Windows.

153 Los programas integrados es algo de lo que hablaremos en el próximo capítulo. Como veremos entonces, la anterior afirmación debe matizarse muchísimo.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Aunque TopView trabajaba en modo texto (eso sí, con ventanas), tanto GEM como Windows eran entornos gráficos, ya que pronto pareció evidente que si se querían manipular e integrar gráficos lo suyo era trabajar de ese modo. Así también sería posible dar un aspecto WYSIWYG a los documentos que se produjeran. Sólo por dar una idea de lo que se vendía en aquellos tiempos, fijémonos un poco en GEM en su versión 1.0. Trabajaba en modo gráfico, se parecía notablemente en su aspecto al Apple Macintosh ya que permitía abrir cualquier tipo de ventana en cualquier posición y usaba iconos si así se deseaba, tenía una barra general de menús de “acceso rápido” en la que bastaba con colocar el puntero del ratón para que los menús se desplegaran automáticamente, y requería muy pocos recursos ya que era capaz de funcionar dignamente por ejemplo en un Olivetti M24, un equipo con un i8086. Al igual que el Mac, era monotarea, y no permitía cargar más de un programa simultáneamente. Digital Research lo vendía a fabricantes, no a usuarios finales. Es decir, que Vd. compraba un “ordenador con GEM”, no se lo añadía por su cuenta a su PeCé. Windows, en su versión 1.0, no admitía más que ventanas en modo “mosaico”, es decir no superpuestas, aunque se pretendía multitarea y era capaz de cargar varios programas a la vez.

¿Por qué no se popularizaron inmediatamente, pregunta?. Con esta situación inicial, no es descabellado pensar que la historia de los GUIs para PeCé debería continuar diciendo que se comenzaron a usar con rapidez y que se estableció una sana competencia entre ellos que con el tiempo, y reflejando la mejor evolución de uno u otro y sus mayores ventas, condujo al estándar actual.

Pues de eso, casi nada. La realidad se complicó más que notablemente debido a muy diversas razones que pasamos a enumerar. TopView se descartó casi de inmediato. De los dos entornos restantes, GEM parecía inicialmente el mejor situado. Tenía un aspecto más agradable, requería muy poco ordenador para hacerlo correr, y estuvo disponible rápidamente no sólo para el PeCé sino también para otros ordenadores de la época. El Atari ST (ver glosario), un equipo planteado como una alternativa directa al Apple Macintosh¹⁵⁴, lo incorporaba en ROM. A Apple no le gustó en absoluto la broma y, amenazando a Digital Research con acciones legales basadas en la copia no autorizada de la interfase del Macintosh, la obligó a rediseñar el aspecto de GEM. La versión del Atari ST conservó la apariencia original, pero ya en mayo de 1.986, la versión para PeCés había sido modificada. GEM 2 era menos atractiva, a pesar de lo cual tuvo cierto éxito. La incluía por ejemplo el Amstrad PC1512, el primer clónico en Europa con bajo precio y gran aceptación, que se lanzó hacia finales de 1.986.

Windows 1.0 se proponía mientras tanto como la “alternativa para PeCés de alto de gama”. Microsoft estaba entonces siendo casi la única editora importante que producía software para el Macintosh y entre esto, una serie de acuerdos con Apple para compartir código, y que Windows 1.0 tenía una apariencia menos sofisticada, no sufrió en un primer momento la cólera de Apple.

Pero sí en un segundo. La cuestión es que la situación se complicó aún más con la aparición de los PeCés basados en el Intel 80386. Más o menos por esa época, IBM decidió lanzar sus PS/2. Para dotarlos de un sistema operativo más potente, IBM y Microsoft se embarcaron

154 GEM cumplió bastante con la voluntad antes citada de construir un entorno “multiplataforma”.

Más historia, ahora de Windows

en el desarrollo conjunto de OS/2, que debía incluir un GUI (vamos a volver a llamarlo así) al que se denominó “Presentation Manager”. Como primera aproximación a su desarrollo, Microsoft rediseñó Windows. La versión 2 se dotó de ventanas superpuestas, y Hewlett-Packard propuso un conjunto adicional de extensiones llamado “New Wave” que aumentaba su parecido con el Macintosh¹⁵⁵.

Apple demandó inmediatamente ante la justicia americana a Microsoft y Hewlett-Packard por infringir sus derechos de “copyright” sobre la interfase de usuario del Macintosh. Pedía ni más ni menos que una compensación económica por daños y perjuicios, la inmediata paralización de la distribución de Windows 2.03 y New Wave, y la destrucción de los disquetes originales de ambos productos. Su magnanimidad se limitaba a no extender las acciones legales a todo aquel programador que hubiera desarrollado productos para Windows¹⁵⁶.

Aunque finalmente sin efecto, pues la demanda no prosperó, la actitud de Apple contribuyó a la paralización del mercado de GUIs para el PeCé. Pero en mi opinión, no es ni mucho menos la causa más importante a la hora de explicar la lentitud de su despegue, todavía lejano por esas fechas.

Puede que Digital Research con su GEM, Microsoft con Windows, y Microsoft e IBM al alimón con su futuro OS/2, estuvieran en efecto dando una solución a los problemas inicialmente planteados. Pero no eran los únicos. Los problemas pueden resolverse de muchas maneras. El más importante, el intercambio de datos, sólo requería que se desarrollaran formatos estándar en los que todos los fabricantes estuvieran de acuerdo. Y una cierta normalización podía al menos paliar los inconvenientes causados por la falta de unificación de la interfase de usuario. El PeCé ha sido siempre magnífico desarrollando estándares. El intercambio de datos se hizo posible en cuanto hubo la más mínima posibilidad de declarar algún formato particular como universal. Los programas incluyeron por su cuenta un montón de cosas que se suponía que debían esperar a los “entornos integradores”, tales como gestores de ficheros y posibilidades ampliadas de impresión. Los programas gráficos se desarrollaron usando el modo gráfico del PeCé sin más adiciones. La interfase de usuario del PeCé mejoró mucho, y se unificó en parte, al usar todos los programas las teclas de función de forma similar e incorporar textos de ayuda bastante completos.

Digamos que los editores de software para PeCé decidieron resolver los problemas anteriormente citados para que sus programas pudieran comunicarse entre sí y con los más populares de la competencia de modo que la solución fuera aceptable en los equipos de la época. Y tuvieron éxito. La gente usó sus programas, trabajó con ellos, creó colecciones

155 Y que añadía otras capacidades, tales como un cierto soporte para objetos (cosa ésta que veremos mucho más adelante) y la posibilidad de automatizar su operación mediante un lenguaje interno gestionado por “agentes” (esto último lo dejaremos estar, aunque como se verá en el capítulo 9, no es ninguna tontería).

156 Si nos encaminamos al armario de las revistas de la época, es posible encontrar alguna justificación o detalles adicionales. Por ejemplo, en el “Science et Vie Micro” de Mayo de 1.988 y junto a lo ya expuesto, se citaba a Del Yocam, uno de los directivos de Apple, que manifestaba: “El aspecto del Mac, protegido por copyright, está en la base de su éxito comercial. Este éxito parece haber animado a algunos a copiar nuestro interfase visual, más que a inventar el suyo”.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

enormes de datos que los necesitaban, y no podía, ni quería, renunciar a los programas con los que se ganaba los garbanzos a cambio de un “entorno” por muy “integrador” y “gráfico” que fuera.

O dicho de otra forma, tanto GEM como Windows requerían nuevos programas, desarrollados expresamente para ellos. No los había (GEM por ejemplo sólo consiguió un programa importante, el Xerox Ventura), y no eran capaces de usar el software normal basado en DOS. El resto de editores, como suele ser habitual, desarrollaron para “el mínimo común denominador”, es decir un PeCé de medio pelo sin GEM ni Windows, y la gente usó los programas de la mayoría. Sin programas que integrar y cuya interfase de usuario unificar, un entorno integrador, de nuevo por muy integrador y gráfico que fuese, resultaba simplemente inútil. Para que un GUI tuviera éxito debía resultar compatible con los programas DOS de la mejor manera posible. OS/2 incluyó el “compatibility box” (por traducirlo de algún modo “caja de compatibilidad”), para que pudieran ejecutarse desde él los programas DOS que la gente usaba. Esto al menos permitía seguir trabajando y cogiéndole el aire al nuevo GUI mientras se esperaba a que aparecieran los programas específicos que lo hicieran útil de verdad. Aún así, OS/2 no despertó gran entusiasmo, entre otras cosas porque su GUI se retrasó muchísimo y sus requisitos de hardware eran tales que no podía usarse sino en los PeCés de mayor capacidad, es decir en muy pocos.

Cuando finalmente se lanzó un OS/2 completo, la versión 1.2 con Presentation Manager y todo, la situación siguió exactamente igual. A pesar del considerable empuje de Microsoft e IBM y de las cantidades derrochadas en publicidad para promocionar el PS/2 y su vertiente software que ahora nos ocupa, el OS/2, la enorme mayoría siguió usando Lotus 123 y WordPerfect (dos de los programas estándar para DOS aparecidos mientras tanto) o cualquiera de sus equivalentes sobre uno u otro DOS.

IBM se empezó a poner nerviosa. Ni los PS/2 ni OS/2 despegaban. Microsoft se empezó a poner nerviosa. Ni Windows ni OS/2 despegaban. Digital Research hacía tiempo que había dejado de ser una alternativa a considerar a medio plazo. No le iba mal entre los Amstrad PCW y su CP/M, el DR-DOS, el GEM del Atari ST y algún otro que aún se vendía, y no parecía una compañía particularmente codiciosa. Sin embargo, era cada vez más obvio que, tras casi 7 años de espera, los usuarios del PeCé estaban más dispuestos que nunca a equipar sus PeCés con un GUI.

Se empezó a hablar de que Microsoft iba a saltar en marcha del tren de IBM e iba a sacar un nuevo Windows. Al fin y al cabo su última versión en aquellos tiempos, llamada Windows 386 por estar específicamente dirigida a PeCés basados en el Intel 80386¹⁵⁷, había recibido buenas críticas y podía ser una buena base para una nueva edición que corriera en ordena-

157 La evolución del hardware seguía mientras tanto avanzado a todo trapo. El PeCé base de entonces tenía un Intel 80286 como CPU, pero ya no eran raros los equipos que incorporaban un 80386. Y la diferencia entre uno y otro era muy notable. El 80386 era el primer microprocesador de Intel realmente capaz de manejar la multitarea y cantidades grandes de memoria. Se esperaba de él que proporcionara un soporte adecuado a cualquier evolución significativa en los sistemas operativos para PeCé. Y cumplió con las expectativas. Aún seguimos en esta fase. Recuerde lo que en el capítulo 6 dijimos sobre las distintas generaciones de PeCés. Si quiere fechas, el lanzamiento de Windows 386 se produjo hacia 1.988. La versión anterior, Windows 2.0, apareció en 1.987.

Más historia, ahora de Windows

dores menos potentes que los necesarios para OS/2. Que tuviera un público potencial más amplio que OS/2, vamos. Por supuesto, a IBM no le gustó. Si su mayor socio en el desarrollo del “sistema operativo de los noventa” daba la espantada ¿quién iba a creer en OS/2?. De todos modos eran rumores, porque, al menos de forma oficial, Microsoft estaba firmemente al lado de IBM.

En el Byte de Enero de 1.990 se cita una rueda de prensa al alimón de Bill Gates (patrón entonces y ahora de Microsoft) y Jim Cannavino (entonces el encargado de la división de “sistemas personales” de IBM), realizada en el curso de un Comdex (una feria americana de informática), en la que se intentó clarificar la situación. Bill Gates afirmó tajantemente que el desarrollo de OS/2 2.0, la versión de OS/2 “32 bits” dedicada a ordenadores con procesadores 80386 o 80486 y un mínimo de 4 Megabytes de RAM, continuaba imparable y se empezaría a distribuir durante ese año. OS/2 1.2 seguiría disponible para los ordenadores de gama media basados en el 80286 y con un mínimo de 3 Megabytes de RAM. Quizá se consiguiera que funcionara en ordenadores con tan sólo 2 Megabytes. Jim Cannavino, por su parte, afirmó que Windows seguiría siendo el punto de entrada a los GUIs para PeCés de baja gama, con menos de 2 Megabytes de RAM. Nunca se le incorporarían las capacidades reservadas a OS/2. Los dos a coro aseguraron que a mitad de año se comenzarían a distribuir nuevos programas en modo gráfico, en primer lugar para el Presentation Manager de OS/2 y con posterioridad, si acaso, para Windows.

La realidad fue que a mitad de año Microsoft lanzó Windows 3.0. La nueva versión mejoraba muchísimo su aspecto. Tenía un equivalente aceptable del “compatibility box” para que se pudieran ejecutar desde él bastantes programas DOS. Necesitaba unos 10 o 15 Megabytes de disco duro pero corría en un 80286 con 2 Megabytes de RAM, aunque prefería claramente un 80386 con 4 Megabytes. Incluía tres o cuatro programas de uso general para resultar mínimamente útil de entrada. Se vendía barato, por debajo de las 10.000 pesetas, creo recordar¹⁵⁸. Inmediatamente quedó claro que el PeCé había encontrado su GUI y se vendió como rosquillas. No es que fuera perfecta. Un año después, más o menos, Windows 3.1 redondeó las aristas más molestas del producto. Tampoco era perfecta, ni mucho menos, pero era claramente preferible. Poco después, los PeCés eran mucho mejores que dos años antes y demostraban ser capaces de usar Windows 3.1 sin problemas significativos. Desde luego, nuestro PeCé típico del capítulo anterior estaba perfectamente dotado para emplearla, y hasta puede usarse sin problemas en bastante menos. En dos años, se desarrollaban más programas para Windows que para DOS. Un PeCé ahora mismo debe casi obligatoriamente ser capaz de ejecutar Windows para ser considerado aceptable¹⁵⁹.

Por terminar cuanto antes la historia del GUI de los PeCés, añadamos que hacia 1.994 salió un Windows con capacidades, siquiera modestas, de operar en red. Se le llamó Windows 3.11, y el hecho de que se diferencie de Windows 3.1 en el segundo decimal debería indicar que ambas se encuentran muy próximas. Ya dijimos, y repetimos, que el asunto de las redes

158 Desgraciadamente, una vez más mi memoria demuestra ser más floja de lo que debiera. No me acuerdo con exactitud del precio, pero creo que por ahí iban las cosas.

159 Lo que no quiere decir que un PeCé sólo con DOS esté obsoleto. Algunos portátiles, en especial los más pequeños, pueden resultar sumamente útiles, y a la vez cómodos, sin necesidad de trabajar en Windows. Además un GUI no es en absoluto necesario para hacer un montón de cosas, por lo que sigue habiendo PeCés que, incluso pudiendo, no lo emplean.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

se sale claramente del ámbito de este texto, por lo que no haremos distinción entre ambas. En 1.993, Microsoft lanzó Windows NT, un sistema operativo absolutamente nuevo para PeCés (y en principio también para ordenadores de otros tipos) que usaba el GUI de Windows 3.1. Era el “alto de gama” de Microsoft en sistemas operativos, pensado ante todo para ordenadores de empresa y que, una vez más, se sale del ámbito que hemos marcado. Y en agosto-septiembre de 1.995 apareció la última entrega, Windows 95, a la que se conocía como “Chicago” en su período de desarrollo. Esta sí que se lanzó como una puesta al día de Windows 3.1, y por lo tanto, algo diremos de ella. Todas han respetado en buena medida el principio sagrado de los PeCés, la compatibilidad, de modo que una inmensa mayoría de los programas desarrollados para Windows 3.1 siguen funcionando en las nuevas versiones. Y como antes dijimos, aquí estamos.

¿Qué pasó con OS/2, pregunta?. Microsoft e IBM rompieron, por supuesto, su idilio en el momento en que apareció Windows 3.0, e IBM se quedó sola con OS/2. Completó por su cuenta la versión 2.0 y lo intentó imponer, con escaso éxito, en dura pelea con Microsoft¹⁶⁰. Le sucedieron con rapidez múltiples versiones mejoradas, algunas de ellas (OS/2 Warp) pensadas para “añadirse” más que para sustituir a Windows. En la actualidad me parece que va por la 3.0, aunque no es fácil seguirle la pista a su nomenclatura. En ningún momento ha dejado de ser una alternativa marginal si nos referimos a usuarios particulares, aunque ha tenido algo más de suerte en el mundo empresarial.

Lo anterior ha sido una pequeña historia, muy resumida y sin entrar en excesivos tecnicismos. Porque hubo otros “entornos integradores”, “sistemas operativos con interfase gráfica”, o “simplemente GUIs”, que ni tan siquiera hemos nombrado y que gozaron de cierto éxito. Como por distintos motivos no está de más que nos suenen algunos, citaremos unos cuantos. Desqview, de un fabricante llamado Quarterdeck, era uno. GEOS, de Geoworks, fue otro. Ambos eran muy interesantes técnicamente, y ambos estuvieron disponibles antes de que Windows se hiciera con los PeCés. Y aún hubo más. Al igual que probablemente pocos se acordaban de VisiOn, es casi seguro que pocos hayan oído hablar siquiera de Freestyle, un interfase gráfico que se manejaba mediante una tableta gráfica y un micrófono, y que el fabricante Wang propuso en diciembre de 1.988 para sus equipos de la serie 200/300 y PeCés de tipo AT.

¿Por qué Windows se quedó con los PeCés?. ¿Qué tenía él que no tuvieran los demás?. ¿Qué inconvenientes tuvieron que soportar sus usuarios una vez adoptado?. ¿Qué aspecto tiene y qué añade en concreto a un PeCé con DOS?. Lo veremos en el próximo apartado, ahora que conocemos un poco más sus orígenes.

160 Analizar las razones del relativo éxito (o relativo fracaso) de OS/2 frente a Windows es un ejercicio que puede resultar interesante, pero que no realizaremos en profundidad. Aunque sólido e innegablemente avanzado en muchos aspectos, baste con insinuar que OS/2 siempre ha arrastrado una considerable falta de drivers, y que resultaba muy frustrante no poder trabajar nada más que en VGA cuando Windows permitía poner el monitor en 1024x768; o que OS/2 volvió a mostrar aquellos enigmáticos mensajes de error típicos del CP/M, similares a “System error #xx” cuando se pretendía arrancar con un disquete no adecuado.

Claqué y ventanas

LR-PeCé imita a Fred Astaire

Clic, clic, quiticlic, clac, clac... . El ruido de los ratones al pulsarlos evoca la banda sonora de las películas de Fred Astaire. A partir de 1.990, se convirtió en algo habitual en toda sala informática en la que hubiera un mínimo de dos PeCés. Los GUIs se habían adueñado de la situación. Una consecuencia inmediata para los sufridos usuarios que, tras arduos esfuerzos, habían conseguido aprender para qué servía cada tecla de su teclado tipo AT y se empezaban a sentir a gusto con sus programas DOS fue que debieron volver a la escuela elemental para que les fueran inculcadas nuevas habilidades, nuevas formas de ver las cosas, nuevos reflejos. La interfase de usuario del PeCé había cambiado y había que cambiar con ella. Ya no basta con saber que “F1” significa ayuda¹⁶¹, hay que desenvolverse ante expresiones tan pintorescas como “escritorio”, “doble clic”, “arrastrar y soltar”, “ventana DOS”, “menú contextual”, y otras muchas tal vez mucho más crípticas.

Es exactamente en esta fase donde debería encontrarse todo LAO realmente recién llegado a la informática que haya guiado sus primeros pasos por este opúsculo. Asumiendo las responsabilidades que nos corresponden por haberlo traído hasta aquí, habría que aclararle cuanto antes lo que es preciso saber para pasar sin excesivo dolor de DOS a DOS con Windows. Pero una vez más, digámoslo ya para evitar decepciones, el objetivo no va a ser explicar con detalle el manejo de Windows o sus componentes. Un GUI, como en su momento se dijo, está pensado para seducir, dejar que el usuario juegue, y así aprenda de forma casi automática. Los GUIs son bonitos y divertidos, contienen montones de “tutoriales” interactivos para enseñar lo más básico (el anglicismo es tal vez traducible por “cursillos”), y por si fuera poco se han publicado cantidades ingentes de libros que, estos sí, persiguen el objetivo citado. Fieles a nosotros mismos y a nuestra probada cortedad, nos dedicaremos a lo más básico y limitaremos nuestro papel a recomendarle que juegue con Windows siempre que pueda y a puntualizar aquellas obviedades que, curiosamente, no suelen aparecer demasiado en los manuales habituales de introducción a los PeCés. Vamos a ello.

Meter Windows 3.0 en un PeCé en 1.990 era sinónimo de tener que lidiar con numerosos problemas. No sólo había que aprender un montón de nuevas habilidades. Como se dijo en su momento, un GUI consume recursos con voracidad insaciable, por lo que había además que sacrificar en el altar del monstruo una buena parte del disco duro disponible y soportar que nuestro PeCé de entonces (con un 80386SX, unos 2 Megabytes de memoria y un disco duro de 40 Megabytes), que tan raudo funcionaba con programas DOS, se convirtiera por arte de birlibirloque en una auténtica tortuga. Aún así, muchos usuarios lo compraron, lo instalaron en sus PeCés, y lo usaron. Ya que Windows 3.0 se vendía suelto, para que todo el que quisiera se lo colocara a su ordenador. Esto fue un gran acierto de Microsoft. Ya no se compraban PeCés con Windows, sino que cualquiera podía, si quería, añadirlo por su cuenta.

161 Por suerte, sigue siendo cierto. No todo está perdido al pasar a Windows. Como se ha recalado unas 20.000 veces aproximadamente, la mayor parte de lo que se ha visto sobre el uso del PeCé con un teclado es tan general que sigue funcionando en Windows, y en OS/2 y en casi cualquier otro sistema operativo para PeCés.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

“Alguna razón debía haber para que Windows se vendiera, no todo serían problemas”, estará pensando nuestro querido LAO. Si habían podido esperar 7 años para tener un GUI, los usuarios del PeCé hubieran podido esperar alguno más si Windows 3.0 no hubiera significado un cambio a mejor en un número suficiente de aspectos. Apúntese de nuevo un positivo en su libreta de notas.

Ya vimos, al hablar de la historia de Windows, que los dos problemas clave que todo GUI debe resolver se habían paliado mucho en el mundo de los PeCés recurriendo a otro tipo de soluciones. Pero el mero desarrollo de su software y hardware llevó a que otros aparecieran. Uno de los principales fue el que podríamos denominar “unificación de los drivers”. Consistía en que uno se compraba por ejemplo una impresora nueva y, como no deja de ser lógico, albergaba el temerario propósito de usarla con todos sus programas. Pero resultaba que la impresora necesitaba, para poderle sacar partido, un driver diferente para cada programa con el que fuera a usarse. Pongamos un ejemplo concreto con programas que ya hemos citado. Supongamos que alguien estaba usando Lotus 1-2-3 y WordPerfect. Se compraba una Hewlett-Packard DeskJet. Para que funcionara bien había que instalar en el directorio de Lotus el driver de la HP DeskJet para Lotus 1-2-3 y en el directorio de WordPerfect el driver, totalmente distinto del anterior, de la HP DeskJet para WordPerfect. Sin ellos, tan sólo podían aprovecharse una mínima parte de las capacidades de la impresora. Aunque Hewlett-Packard suministraba una colección razonable de drivers, en la que se incluían casi todos los programas más populares, era imposible abarcarlos todos. Con Harvard Graphics¹⁶², por decir algo, nuestra DeskJet podía no funcionar. Muy pocos fabricantes, tanto de hardware como de software, daban un soporte adecuado a los cientos de periféricos y programas existentes en el mercado.

Y había más. Si uno usaba un procesador de textos, no era extraño que deseara imprimir un documento larguísimo y a la vez seguir trabajando en otra cosa, tal vez en una carta personal. Se llama a esto, en jerga, “imprimir en tarea de fondo”. No resultaba agradable tener que esperar tres horas a que la impresora acabara de imprimir, bajo control de nuestro procesador de textos, para poder empezar un documento nuevo. Ya que quizá no teníamos tres horas por delante, sino que en media hora debíamos irnos, y lo que nos interesaba era preparar la carta, mandarla imprimir, e irnos tras dejar a la impresora con sus cosas para recoger las páginas a la vuelta, cuando volviéramos del cine.

Y todavía más. Al hacer que los programas DOS hicieran “todo lo necesario para ser útiles”, se les habían añadido montones de opciones, haciendo que su tamaño creciera hasta límites casi insostenibles. Por ejemplo, uno siempre quiere abrir ficheros desde un programa, pero también quiere cambiarles el nombre, y hasta borrarlos o moverlos de un directorio a otro. Salir al DOS para hacerlo, teniendo que regresar al programa para continuar trabajando, era una lata. Muchísimos programas añadieron auténticos gestores de ficheros que permitían hacer este tipo de operaciones sin tener que salir de ellos.

DOS pedía a gritos que se le añadieran capacidades multitarea (recuerde, poder correr varios programas a la vez), se unificaran los drivers del nuevo hardware y software que fuera apareciendo, y se proporcionaran servicios básicos para uso de todos los programas, entre ellos que se aclarara por fin la gestión de la memoria del PeCé por encima del primer

162 Un programa de representación gráfica de datos numéricos bastante popular.

megabyte de RAM. En un entorno multitarea es posible ejecutar simultáneamente el programa que sea y un gestor de ficheros, y pasar de uno a otro según convenga. No haría falta incorporar un gestor de ficheros, un gestor de impresión y un gestor de drivers a cada programa. Estos podrían hacerse más pequeños, eliminando componentes redundantes. Siempre se podrían manejar los ficheros desde el mismo gestor de ficheros, se podría imprimir desde el mismo gestor de impresión, y bastaría con un driver, el del sistema operativo central, para que un nuevo chisme funcionara a pleno rendimiento con absolutamente todas las aplicaciones que tuviéramos.

Pues bien, Windows 3.0 permitió un avance importante en todos estos campos. Incorporaba una cierta capacidad multitarea, proporcionaba un servicio central de recogida de drivers para los dispositivos más diversos, especialmente tarjetas gráficas e impresoras, incluía un gestor de ficheros en modo gráfico que no estaba mal para la época, un gestor de impresión que ponía la impresión en modo gráfico y/o en tarea de fondo al alcance de cualquier programa, permitía trabajar con memoria en gran cantidad y, por supuesto, era un GUI de apariencia bastante bonita que resultaba más que razonablemente seductor. Por si no bastara, también había una “caja de compatibilidad” que permitía acceder a los programas DOS, algo que, como hemos dicho, era tal vez más importante de lo que pudiera parecer. Y desde luego incluía también todos los adminículos que se dan por supuestos en un GUI, es decir ventanas, ratones, menús desplegables, y otros etcéteras. Astutamente, además, Windows permitía, y aún permite, acceder a muchas de sus capacidades desde el teclado, mediante las correspondientes combinaciones de teclas.

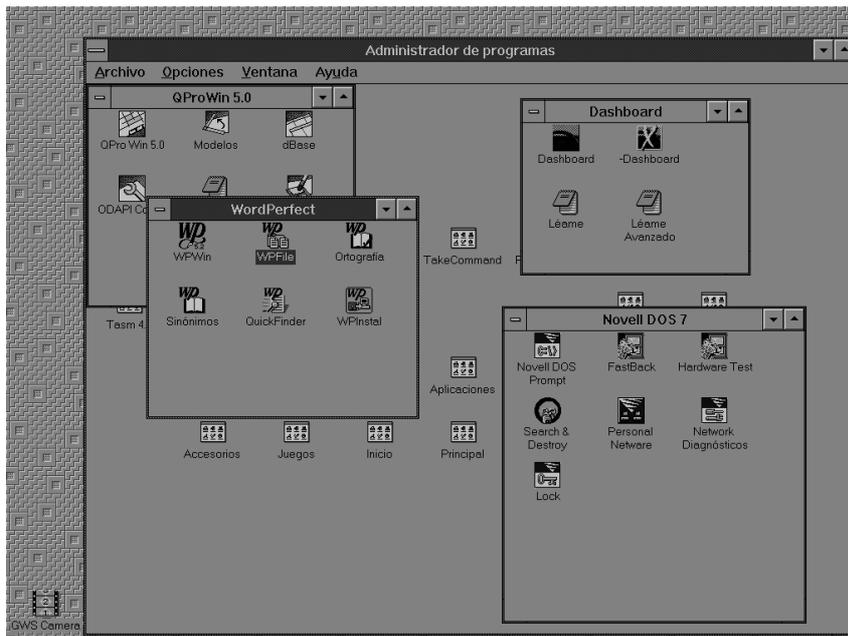
Aunque no hubiera al principio muchos programas para Windows, aunque el ordenador corriera menos y emplearlo supusiera renunciar a buena parte del disco duro, parecía claro que el nuevo GUI merecía la pena y que prometía mejores programas, de apariencia más agradable, más prácticos, más cómodos, más potentes y tal vez hasta más pequeños. No es extraño que Windows 3.0 se popularizara, especialmente tras ser sustituido por Windows 3.1, ya que esta versión mejoraba significativamente la impresión de textos al incluir tipos de letra escalables y con espaciado proporcional (el llamado True Type, ver glosario), era marginalmente más rápida, completaba las opciones de compatibilidad hacia el DOS con la posibilidad de abrir múltiples “sesiones DOS” en sendas ventanas, e incluía algunas otras mejoras adicionales en absoluto despreciables. Más adelante entraremos en detalles sobre algunas de estas nuevas facultades que adornan a nuestro LR-PeCé al vestirse con ropajes de Windows. Dejémoslo así por ahora, y pasemos a ocuparnos de cosas más urgentes.

Bueno, ya tenemos Windows 3.1 (vamos a saltar de versión) instalado en nuestro ordenador. ¿Cómo lo ejecutamos, a qué nos enfrentamos, y de qué recursos disponemos a partir de entonces?. ¿Qué vemos al entrar y cómo “leemos” las pantallas que se nos presentan?.

Para llegar a Windows será necesario darle al ordenador una orden en estricto DOS, ya que es ahí donde nos encontraremos al entrar en un PeCé. La orden es “WIN”, se la pasaremos a LR-PeCé como siempre hacemos con cualquier orden, es decir escribiendo “WIN” en el “prompt” y dándole a la tecla gorda de la derecha esa. Esto hará que se ejecute el fichero “WIN.COM”, sito en el directorio WINDOWS, si tenemos el tal directorio incluido en nuestro “path”. Por cierto, admite parámetros. Entre ellos un “/?”, a través del cual conseguir explicaciones sobre cuales son y qué hace cada uno. Si no especificamos ninguno, Windows se ejecutará (en un 386) en lo que se denomina “modo extendido”, el modo por defecto, que es el más potente y el que supondremos a partir de ahora.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Todo lo anterior debería resultar lógico y escasamente sorprendente. Del mismo modo que no debería sorprender el decir (ya lo dijimos de hecho) que, mediante la oportuna modificación del AUTOEXEC.BAT, puede hacerse que el PeCé acabe su proceso de arranque dejándonos directamente en Windows. Pero a Windows se entra y se sale con frecuencia, y si Vd. empieza por ignorar la forma en que en realidad nuestro PeCé ha terminado allí, mostrando tan maravillosas pantallas multicolores, está una vez más abocado al desastre. Ya dentro de Windows, lo que vemos puede ser algo más o menos como lo que sigue:



o sea ventanas, tal vez dentro de otras ventanas, y pequeños dibujitos con una etiqueta debajo esparcidos por todas ellas. A tales dibujitos se los llama “iconos”.

Si tenemos en cuenta que, en la noche de los tiempos, a los GUI se los llamaba “entornos WIMP”, tal apariencia no debería resultar extraña. Recuerde que lo de WIMP corresponde a la abreviatura de “Windows” (ventanas), “Icons” (iconos), “Mouse” (ratón) y “Pull-down menus” (menús desplegables). Pues bien, Windows 3.1 pone a nuestra disposición ingentes cantidades de ventanas y de iconos, para que no se diga. El ratón se da por supuesto, y los menús desplegables también están ahí, enseguida veremos donde.

Para explicar lo que estamos viendo y poder decir cómo sacarle partido, hay que introducir algunos conceptos. El manejo de un GUI se basa en lo que se denomina “metáfora del escritorio”. Lo que traducido no es más que un nuevo caso de simulación de apariencias a cargo de nuestro ordenador. Se supone que la pantalla va a intentar imitar una superficie de trabajo, como una mesa, sobre la que vamos a disponer aquello con lo que queramos trabajar. A dicha mesa metafórica se la denomina “escritorio” y puede identificarse burdamente con el fondo de la pantalla, que en la anterior figura aparece con un tramado en gris que simula una especie de baldosas cuadrangulares. Sobre ella dispondremos las ventanas en cuyo interior trabajaremos. Cada ventana será en principio el espacio de trabajo, la

“pantalla virtual”, del programa correspondiente. En un entorno multitarea, en el cual podemos ejecutar varios programas a la vez o varias veces un mismo programa (se habla entonces de distintas “sesiones” del mismo), las ventanas son fundamentales a la hora de separar el espacio de trabajo de cada uno.

Cada ventana tiene vida propia, una entidad particular, y está controlada por un programa determinado. Para desplazarnos de una a otra, lo único que hay que hacer es mover el cursor del ratón (una flecha que desgraciadamente no se aprecia en la figura) al interior de aquella en que queramos situarnos y darle una vez a su botón izquierdo. El borde de la ventana cambiará de color, indicándonos con ello que a partir de ese momento es a esa ventana (a su programa) a la que va a llegar aquello que tecleemos. Llámase a la ventana en ese estado “ventana activa”, y sólo puede haber una¹⁶³. Lo de “activo” no quiere decir que el resto de programas se detengan en sus ventanas correspondientes, sino que continuarán trabajando sin mirarnos. En caso de que necesiten algún dato de nosotros, esperarán pacientemente a que volvamos a ocuparnos de ellos.

Si hemos digerido lo anterior, acabamos de aprender varias cosas. En primer lugar que podemos mover el ratón por la mesa sin que pase cosa alguna distinta a que el puntero a él asociado en la pantalla también se mueva. Llámase a eso “desplazar o mover el puntero”, como es natural. En segundo lugar, que cualquier acción implica al menos confirmar la posición del puntero mediante la pulsación del botón izquierdo del ratón. Colocar el puntero sobre algo (una ventana, un icono) se llama “señalar” y pulsar el botón se llama “hacer clic”. Ejecutando la totalidad del proceso anterior (señalar y hacer clic) lo que hacemos es “seleccionar” un algo, sea por ahora ventana o icono, o en su momento fichero, elemento gráfico, o cualquier otra cosa. Una vez seleccionado, es posible manipular el algo en cuestión.

El botón izquierdo del ratón resulta ser el equivalente en el gremio de los roedores de nuestra tecla gorda de la derecha esa, es decir del retorno de carro. El proceso de selección, debo de llamar su atención sobre ello, es igualmente posible desde una línea de órdenes. Basta con mostrar al usuario un listado de elementos disponibles, dejarle la posibilidad de escribir “Selecciona el elemento tal” y de darle a la tecla gorda de la derecha esa, y acabar pasando la orden a un hipotético COMMAND.COM que se encargue de considerar al tal elemento como el objeto con que vamos a trabajar a partir de ese momento y hasta de realizar con él la acción que hayamos indicado. ¿Qué hemos estado haciendo si no al teclear cosas tan familiares como “DIR C:\” seguido de “COPY CONFIG.SYS A:”? Quiérese decir con lo anterior que el truco para aclararse rápidamente con Windows y poder sacarle mucho más partido que la mayoría de los mortales, reducidos a una triste manipulación de una metáfora de algo para ellos desconocido, consiste tan sólo en conocer previamente DOS y ser capaz de traducir las acciones con que nos desenvolvemos en el nuevo entorno a aquellas que el ordenador realiza de verdad. En ser capaz de ver lo que hemos llamado “sistema de ficheros” a través de la “interfase de usuario” de la que en ese momento nos servimos. No es más que

163 El borde de la ventana cambiará de color si Windows está configurado adecuadamente. Por mucho Windows que tengamos, seguiremos teniendo que configurar. De hecho, con Windows tendremos muchísimas más cosas que configurar que antes, entre ellas los colores de los distintos elementos de las ventanas en los distintos estados en que pueden aparecer. No debería ser necesario decir más para que vaya viendo que un GUI puede mejorar unas cosas y empeorar otras.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

lo que siempre hemos estado haciendo, ya fuera con una línea de órdenes, unos menús o, en este momento, un GUI. Para poder aprovechar un ordenador, la interfase de usuario debe ser un cristal coloreado a través del cual sepamos ver qué ocurre, y no un decorado opaco que deforme u oculte aspectos esenciales de su funcionamiento. Volvamos, tras tanta metafísica, a ocuparnos de cómo manejar el ratón y los otros elementos recién incorporados al patrimonio de nuestro PeCé.

Hemos explicado antes cómo seleccionar una ventana. Una ventana tiene cuatro lados, y el superior es más gordo. En él, de izquierda a derecha, pueden distinguirse cuatro zonas. La primera es cuadrada, pequeña, y muestra un signo menos. La segunda es grande y contiene un rótulo. Las dos últimas vuelven a ser pequeñas y contienen normalmente un triángulo hacia abajo y otro hacia arriba. La zona gorda con el rótulo contiene el nombre de la ventana, y las otras tres son “zonas activas” que permiten cerrar la ventana (signo menos), minimizarla reduciéndola a un icono con una etiqueta igual a su nombre (triángulo hacia abajo), o maximizarla para hacer que ocupe toda la pantalla (triángulo hacia arriba). Se cierra llevando el puntero al signo menos y dando dos clics en rápida sucesión, maniobra a la que se denomina “hacer doble clic”. Cerrar una ventana es equivalente a acabar de ejecutar el programa que la controla. Si ya no necesitamos un programa, tampoco necesitamos reservar un área de trabajo para él, o sea que no hace falta su ventana, y por tanto ambas operaciones se asocian. Se maximiza o minimiza poniendo el puntero en la zona correspondiente y dando un solo clic, acción a la que se llama “pulsar”. ¿Porqué uno, si antes eran dos?. Digamos por ahora que son idiosincrasias de Windows. Si se maximiza, el dibujo cambia a uno que incluye los dos triángulos, queriéndonos así indicar que pulsando allí devolveremos la ventana a su tamaño original. Hay que resaltar que minimizar una ventana, reducir su tamaño a un icono, no es lo mismo que cerrarla. Minimizamos cuando no queremos acabar con un programa pero necesitamos sitio en la pantalla para trabajar con otro. El programa que controla la ventana minimizada sigue ejecutándose, aunque no veamos por el momento el fruto de su trabajo. Como necesitamos sitio para hacer otra cosa, lo quitamos del medio por el momento, con la clara intención de volver a él cuando sea preciso.

El nombre de la ventana activa suele estar en otro color que el de las demás. En la figura anterior mostramos una ventana algo especial, correspondiente a un programa algo especial que pronto explicaremos, y dentro de ella tenemos 4 ventanas más con diverso número de iconos en cada una. Si se fija atentamente en la figura verá que, sabiendo esto, es posible deducir con facilidad que la ventana activa es el “Administrador de programas” y dentro de él la ventana “WordPerfect”. Si se fija en el interior de ésta última verá que el segundo icono, el que está entre “WPWin” y “Ortografía”, tiene su etiqueta de otro color, si intentamos usar una terminología más apropiada diremos que está en “video inverso”. Puede deducir de este hecho que está seleccionado. Es, por así decirlo el “elemento activo” en este momento.

Terminemos cuanto antes las lecciones de claqué para poder pasar a cosas más serias. Fíjese que hay dos ventanas superpuestas, las llamadas “QProWin 5.0” y “WordPerfect”, y dos que no lo están, las llamadas “Dashboard” y “Novell DOS 7”. ¿Cómo hacer que “QProWin 5.0” y “WordPerfect” dejen de estar pisándose?. Una vez seleccionada, es posible no sólo cerrar, minimizar o maximizar una ventana, sino también desplazarla y cambiar su forma y tamaño, acciones asimismo fundamentales a la hora de repartir el espacio de trabajo de que disponemos a nuestro gusto y conveniencia. Para desplazarla, basta colocarse sobre el nombre y pulsar, sin soltarlo, el botón izquierdo del ratón. Con esto, podemos decir que hemos sujetado aquello que queramos desplazar, que ahora es una ventana pero podría ser otra

cosa, tal como un icono o un elemento gráfico de un dibujo, o un fichero. Sin soltarlo, movemos el puntero, y veremos que aquello que hayamos sujetado se desplaza con él. Lo dejamos doquiera que nos apetezca, soltamos el botón izquierdo, y ahí que se quedará. Ésta complicada maniobra ratonil se denomina “arrastrar”. Lo de cambiar el tamaño de las ventanas es bastante espectacular. Si nos colocamos exactamente sobre el límite de una ventana, veremos que el puntero del ratón cambia de forma, pasando de ser una flecha orientada hacia la esquina superior izquierda, a ser una flecha bidireccional. Si ahora “arras-tramos” el límite según el método antes indicado, cambiaremos el tamaño de la ventana¹⁶⁴.

Tan sólo queda un paso de claqué clásico por aprender, aunque en realidad ya está casi explicado. Lo que hemos hecho con la ventana para cambiarla de posición ha sido más “arrastrar y soltar” que otra cosa, ya que hemos enganchado al objeto a manipular y lo hemos soltado en otra posición, aquella que hemos considerado más conveniente. La expresión se aplica con mayor propiedad a cualquier elemento que no sea una ventana, especialmente a ficheros. En jerga, y viste mucho decirlo o sea que será mejor que se lo aprenda, se usa lo de “arrastra y soltar” en su lengua vernácula, y se dice “drag and drop”. Se arrastra algo a un “área activa” y se suelta allí. El cursor cambia de forma sobre la marcha, apareciendo un “prohibido” cuando estamos en zonas en las que no tiene sentido dejarlo. Cada “área activa” implica una acción determinada que se va a realizar automáticamente sobre el elemento allí depositado en cuanto lo soltemos de nuestra mano. Pero antes de entrar por estos vericuetos del GUI, es mejor que acabemos de pasar revista a cosas más familiares.

Sigamos con los menús. En un GUI se encuentran con profusión, normalmente en su variedad desplegable. Todo programa que tenga la bondad de poner a nuestra disposición una serie de acciones a realizar cuando lo consideremos conveniente, nos va a mostrar un menú principal para poder acceder a ellas que va a estar en la línea inmediatamente por debajo del nombre o título de la correspondiente ventana. El “Administrador de programas” de la figura anterior, muestra uno. Puede leerse allí “Archivo Opciones Ventana Ayuda”, y cada palabra corresponde al encabezamiento de un menú que da en este caso la posibilidad de realizar operaciones con los archivos asociados a lo que se ve, de fijar opciones relacionadas con el programa en cuestión, de trabajar con distintos aspectos de sus ventanas, o de acceder a ayudas. Convencionalmente, la primera entrada del menú principal es siempre “Archivo” y la última “Ayuda”. Para ver los menús completos basta con colocar el puntero sobre cualquiera de las opciones y hacer clic. El menú asociado para esa entrada se despliega hacia abajo (en tiempos se hablaba de “menús persiana”), nos muestra las acciones relacionadas con el tema de que se trate, y es posible elegir una repitiendo el proceso, es decir haciendo clic sobre cualquiera de ellas. Si queremos ver lo que oculta cada una de las entradas del menú principal, es posible recurrir a un truco. Si hacemos clic pero no soltamos el botón izquierdo, el menú se despliega, pero entonces podemos, sin soltarlo, mover el puntero por el resto de las entradas del menú principal, y los submenús correspondientes se irán desplegando a nuestro paso, mostrándonos sus interioridades para que las examinemos a voluntad. Si seguimos sin soltar el botoncito, podremos también desplazar el puntero en vertical para ir seleccionando opciones de los submenús. Se ejecutará la acción que esté seleccionada, y

164 Algunas ventanas no permiten realizar todas las operaciones que se han indicado. Hay, por decirlo así, ventanas de tamaño fijo, que no es posible minimizar, ni maximizar ni redimensionar. En consecuencia, mostrarán menos dibujos al lado del nombre.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

que aparecerá siempre en video inverso durante todo este teje maneje, cuando finalmente tengamos a bien soltar el dichoso botón. Los menús se cierran en cuanto acabamos de usarlos, volviendo a mostrar aquello que hubieran podido ocultar al desplegarse. ¿Y si no queremos hacer nada, cómo cerramos el menú sin ejecutar acción alguna?. Lleve el puntero a cualquier zona de la ventana que no sea el dichoso menú, o póngalo sobre el escritorio, y haga clic. O suéltelo, si aún no lo ha hecho, allí.

Los menús tienen algunas otras particularidades en Windows que conviene conocer. En primer lugar, si Vd. odia los roedores, puede activarlos y seleccionar y ejecutar opciones usando tan sólo el teclado¹⁶⁵. Si pulsa la tecla “Alt” (justo a la izquierda de la barra espaciadora), el menú principal se activa, las teclas de desplazamiento horizontal del cursor le permiten moverse entre sus entradas, las de desplazamiento vertical despliegan el submenú correspondiente y le dejan desplazarse por sus opciones, y la tecla gorda de la derecha esa le permite ejecutar la que se encuentre seleccionada. ¿Y si de nuevo no quiero hacer nada, sino salir una vez más de allí?. La tecla “Escape”, como no, viene a nuestro rescate. Es también posible ejecutar directamente cualquier entrada de un menú pulsando la tecla del carácter que aparece subrayado en la pantalla. “Alt” activará el menú principal, “O” desplegará el menú “Opciones”, y “G” seleccionará la opción de “Guardar la configuración al salir”, por poner un ejemplo.

Cuando haga todo lo anterior, que seguro que lo ha hecho ya pues resulta bastante llamativo, verá que a veces aparecen entradas en los menús un poco raras. Están en un color más claro, como difuminadas o borradas en parte, aunque pueden leerse normalmente. Son opciones existentes pero que se encuentran desactivadas. La cuestión es que los menús en un GUI suelen ser no sólo desplegables, sino también “contextuales”, es decir que en función de la situación en que estemos nos mostrarán tan sólo como activas aquellas funciones que tengan sentido en ese momento. Si no tenemos un elemento seleccionado no lo podremos copiar, por ejemplo, y entonces la opción “Copiar” nos aparecerá desactivada, en color más claro. Eso nos permite saber que una opción existe, aunque sea utilizable de inmediato.

Otras opciones muestran al lado, además del carácter subrayado, una combinación de teclas. Es posible ejecutarlas también mediante dicha combinación. Y algunas nos enseñan orgullosas unos puntos suspensivos o un triángulito. Nos quieren decir con eso que son la entrada a un cuadro de diálogo u otro submenú, que se mostrarán cuando las seleccionemos.

Tras las ventanas y los menús, les llega el turno a los iconos. ¿Qué es ese montón de dibujitos etiquetados que se esparcen sin prejuicios por todas partes de la figura anterior?. ¿Para qué sirven elementos tan ubicuos?. Si no se ha perdido hace rato, cosa bastante probable, todo LAO habrá observado un hecho chocante. Las ventanas son las zonas de trabajo de los programas que usamos en Windows. Los menús nos dan acceso a las acciones asociadas con ellos. ¿Pero cómo diablos hacemos para ejecutar un programa y abrir la ventana que ponga

165 No es razonable ir en contra del espíritu de una interfase de usuario, sea la que sea. Es decir, que si va a emplear un GUI, es mejor que se acostumbre a usar el ratón. De todos modos, no está de más poder recurrir al teclado, porque en ocasiones, por ejemplo cuando el GUI está pésimamente configurado, o tenemos conflictos graves con los drivers del ratón que monte nuestro equipo, o en otras mil situaciones, el ratón no funcionará decentemente, o no veremos el puntero, y el teclado nos vendrá de perlas para salir del embrollo.

a nuestra disposición, a través de su barra de menús desplegable, todas las acciones que le son propias?. Los iconos son la puerta de acceso que buscamos. Un programa ejecutable en Windows está representado por un icono, y haciendo doble clic sobre él, lo pondremos en marcha. Dentro de la ventana “WordPerfect” hay un icono rotulado “WPWin”. Dos clics sobre él, y se pone a funcionar el procesador de textos con que se está escribiendo esto. Dos clics sobre el icono “NovellDOS Prompt” en la ventana “Novell DOS 7”, y se abre una ventana con nuestra familiar línea de órdenes de DOS. Y así sucesivamente. A partir de ahí podemos acceder a menús desplegables, redimensionar ventanas, y disfrutar de las ventajas de un GUI, haciendo aquello que se supone que debe hacer un ordenador, es decir procesar textos, copiar ficheros, recorrer directorios, y demás. Aprovecho para machacarle de nuevo que el lanzamiento de los dos programas anteriores podría hacerse tal como se ha indicado, con lo que los tendríamos corriendo simultáneamente y podríamos pasar de uno a otro cuando quisiéramos. Por algo estamos en un entorno multitarea.

Las etiquetas de los iconos pueden tener hasta cuarenta caracteres incluyendo espacios, al fin y al cabo son etiquetas que, como los títulos de las ventanas, no están sometidas a ninguna restricción significativa. Es razonable esperar por tanto que muestren mensajes algo más explícitos que en el ejemplo anterior, de modo que nos indiquen sin posibilidad de error a qué programa dan acceso¹⁶⁶.

No todos los iconos representan programas ejecutables. Ya dijimos que el “Administrador de programas” era una cosa algo especial. Si vuelve a fijarse en la figura verá, dentro de su ventana, una serie de iconos que intentan representar a otras ventanas minimizadas con nombres tan variados como “Accesorios”, “Juegos”, “Principal”. Dos clics sobre cualquiera de ellos tiene como único efecto que el icono correspondiente se despliega, abriendo una ventana en la que encontramos iconos de verdad, de los que permiten ejecutar programas. Es decir que hemos llegado a la figura anterior tras pulsar los iconos tipo ventana minimizada llamados “QProWin 5.0”, “WordPerfect”, “Dashboard” y “Novell DOS 7”. Dentro del “Administrador de programas”, los iconos tipo ventanita representan grupos de programas ejecutables. Se habla en consecuencia del “Grupo Novell DOS 7”, del “Grupo WordPerfect” o del “Grupo Principal”, por citar unos cuantos”. Windows crea automáticamente algunos de ellos (Principal, Juegos, Accesorios...), y otros los crea el usuario a su conveniencia ya veremos como.

Por acabar con los iconos por el momento, llamaré su atención sobre ese que está sobre el escritorio, rotulado “GWS Camera”. Corresponde a un programa minimizado, y por lo tanto en ejecución. En concreto aquel con el que se ha capturado la imagen visualizada en la pantalla de mi monitor en un momento determinado y que ha terminado impresa en una hoja de papel como la figura anterior. Si hace clic una vez sobre un icono minimizado, se abre un menú sobre él que le permite entre otras cosas cerrar el programa asociado sin necesidad de maximizarlo de nuevo. Este mismo truco, es decir que un solo clic en un elemento que normalmente se manipula con dos dé acceso a un menú, funciona también con el signo menos ese que nos permite cerrar una ventana. Y quizá funcione en alguna otra ocasión similar pero no me atrevo a decir que sea algo general. Traducido: en la duda,

166 Como veremos más adelante, Windows es muy adaptable a los gustos del usuario de turno. Yo prefiero los nombres cortos, que le vamos a hacer.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

pruebe a dar un clic sobre cualquier cosa, o intente redimensionarla, o experimente con algún otro tipo de manipulación. Tal vez pase algo, o tal vez no.

Nos queda por tratar tan sólo uno de los recursos principales de Windows, o en general de los GUIs, para terminar con esta especie de inmersión salvaje que estamos llevando a cabo. Tal vez se acuerde de lo que se dijo con anterioridad sobre las teclas de función y su moderno sustituto en los GUIs, los botones. Aquellas acciones más frecuentes se ponen a disposición del usuario no mediante menús, siempre farragosos de recorrer, sino mediante botones, dibujitos que, pulsados con el ratón, ejecutan inmediatamente la correspondiente acción. En la figura anterior tenemos dos a la vista, repetidos varias veces, y nos permitirán aclarar un poco una irregularidad que hemos señalado. Tienen en Windows, a partir de la versión 3.0, un cierto aire tridimensional, queriendo indicarnos que pueden pulsarse. Los botones que podemos ver son justamente los correspondientes a maximizar y minimizar una ventana, y es por eso por lo que basta un solo clic, en lugar de dos (lo que no está claro ahora es por qué “cerrar ventana” no es un botón) para llevar a cabo la tarea. Con otros programas distintos al “Administrador de programas”, nos aparecerán en mayor profusión, y con dibujos encima de lo más variado. Si recordaba que existían los botones, recordará también lo que se dijo entonces sobre la eficacia de dichos dibujitos a la hora de darnos a entender qué acción representan, por lo que no le sorprenderá que en ocasiones contengan no un dibujo, sino alguna palabra que indique con mayor precisión qué diablos van a hacer.

El porqué de la rotulación de los botones, los cambios de forma en el puntero del ratón en operaciones de “drag and drop” o la desactivación de opciones de los menús, nos conducen inexorablemente a otro tema metafísico, que ya podemos abordar al haber acabado con casi todo lo que vamos a decir de las manipulaciones posibles en un GUI. Sí, querido LAO, eso es todo, o casi, en cuanto a pulsaciones de ratón y manejo de ventanas y menús. Hay cosas más importantes que tratar y poco espacio para hacerlo, ya llevamos demasiadas hojas en este capítulo. Y una de ellas es cómo “leer” la información de la pantalla en un entorno gráfico.

Una interfase de línea de órdenes con ayudas decentes proporciona al usuario una serie de guías que son sumamente claras a nada que se sepa cómo llegar a ellas y entender lo que allí puede leerse. Un GUI intenta mostrar en todo momento una información de ayuda apropiada al contexto de la acción que se pretende realizar, y lo hace preferentemente en forma gráfica. Para decirle que no puede llevar un elemento a determinada posición del escritorio, cambia la forma del puntero por un símbolo de prohibición. En el momento en que dicho elemento llega a una ubicación permitida, la forma del puntero vuelve a ser la habitual, y se sabe así que es posible dejarlo allí. Un menú colorea de forma diferente las opciones que están activas. Un GUI emplea con profusión los recursos habituales de comunicación que citamos hace ya tiempo y que deberían resultar familiares, es decir paneles de diálogo, menús, campos de entrada controlada de datos, editores de línea y de pantalla completa, pero les añade la capacidad de seleccionar elementos y de desplazarlos entre áreas activas, cada una con significados concretos. Curiosamente, los significados de cada área no suelen indicarse, y el usuario debe deducirlos en función de la naturaleza del programa a que pertenece el espacio sobre el que pretendemos operar. Es preciso aprender a leer la información gráfica de un GUI, y para ello ante todo hay que saber qué hay que buscar.

El problema al intentar usar eficazmente un GUI consiste en que la representación de las acciones a realizar, el estado de los elementos en ellas implicados, las posibilidades existentes en un momento dado, o casi cualquier otro aspecto de la interfase de usuario, siguen sin ser

obvias y de hecho su elección es arbitraria. Podemos resumirlo diciendo que no hay razones para que algo requiera un clic, o dos, o ninguno o una acción de “arrastrar y soltar” y no una selección en un menú. Y el problema se agrava al intentar el GUI mostrar simultáneamente toda la información posible en la escasa cantidad de espacio disponible en una pantalla de ordenador, pues es preciso recurrir entonces a expresiones sumamente condensadas, extremadamente sucintas de dicha información, que son aún menos obvias. Un GUI requiere atención a los detalles. Un cambio de color en el rótulo de un icono, o por supuesto su forma, indica algo; que una zona de la pantalla sea más oscura o tenga un recuadro alrededor, también, y así sucesivamente. Y al contrario de lo que ocurre con el lenguaje escrito, tal vez menos compacto, el significado de los símbolos gráficos no está unificado y no siempre es claro. Quédese con la idea general del comentario anterior, y recuerde que, por suerte para los usuarios del PeCé, en informática no es frecuente que se tiren los muebles viejos por la ventana. Windows ha heredado la tendencia de los programas del PeCé a incluir extensos textos de ayuda donde se intenta explicar su manejo. Y “F1”, nuestra vieja amiga, sigue funcionando. Recorra a ella siempre que haga falta.

Windows es un entorno gráfico con otra particularidad. Es extremadamente redundante, y las cosas pueden hacerse simultáneamente de varias formas diferentes. “F1” le llevará a las ayudas, pero también, como vimos, puede llegar a ellas a través de los menús que muchas ventanas le mostrarán, o pulsando, cuando lo vea, el correspondiente botón rotulado “Ayuda” o “Help”. Y acabaremos la digresión repitiendo un consejo que ya dimos. Sea coherente al principio, y no se empeñe en usar simultáneamente las tres o cuatro formas diferentes de hacer lo mismo que Windows ofrece. Use con preferencia los menús, o el teclado, o los botones, o aquello que más le plazca, para hacer lo que desee, y vaya poco a poco explorando nuevas alternativas. Es mejor no liarse que querer correr demasiado.

Acabaremos de explorar la interfase de usuario de Windows en sus aspectos más inmediatos viendo cómo salir de nuevo a DOS y cómo podemos, no podía ser de otro modo, copiar nuestro CONFIG.SYS a A:. Vamos a ello.

Lo primero es fácil. Para salir de Windows seleccione “Salir” en el menú “Archivo” del “Administrador de programas”. Convencionalmente, “Salir” es siempre la última opción del menú “Archivo” en cualquier programa, y hace exactamente lo mismo que un doble clic sobre el botón de cierre de su ventana (el signo menos al lado del título), es decir terminar su ejecución. ¿Va cogiendo lo de la redundancia?.

Lo de copiar un fichero es más complicado, y no vamos a poner una figura para hacerlo más sencillo. Aplíquese el cuento y lea las ayudas, o busque pistas en lo que vaya viendo en su monitor, o en fin, Vd. sabrá. Se supone que un GUI es fácil de usar, y si no lo logra con la explicación que vamos a dar, tal vez tenga una visión más objetiva de lo que significa usar un GUI y entienda mejor el porqué de las consideraciones “metafísicas” que se han hecho.

La cuestión es que para copiar un fichero hay que empezar por saber el programa que debemos usar. Hay un “Administrador de archivos” en Windows que justamente sirve para estas cosas. Búsquelo dentro del grupo de programas llamado “Principal”. Dos clics sobre él lo pondrán en marcha, y verá una ventana con su menú correspondiente, una serie de botones que representan las distintas unidades de almacenamiento que se encuentren en su ordenador (al menos A: y C:, seguro), y una ventana que tiene como nombre el del directorio actual. Está dividida en dos, y en la parte izquierda verá un árbol con los directorios de la unidad en que se encuentre ubicado por defecto, y otra ventana en la que se muestran los

CAPÍTULO 7. PeCés: MS-DOS/Windows.

ficheros del directorio actual, que aparecerá también en video inverso en la ventana de los directorios.

Es muy probable que la tal ventana tenga un borde derecho y uno inferior un tanto sorprendentes. Son anchos, como el título, y tienen un cuadrado en medio, más o menos, y unos botones con flechas en los extremos. Acaba Vd. de ser presentando a los ascensores. Cuando una ventana no basta para contener todo lo que en ella debería verse, se activa este mecanismo, que permite desplazar su contenido de modo que tengamos acceso a todo. Pulse los botones de dirección y verá que la lista de ficheros se mueve hacia donde sea mostrando más cosas, es decir más ficheros o más directorios. El cuadrado del medio, el cajetín del ascensor, se desplaza a lo largo de sus guías, indicando la posición relativa en que nos encontramos, mientras que su tamaño nos da una idea global de la relación existente entre la información que nos está siendo presentada y la total. También es posible arrastrar directamente el tal cajetín, con lo que el desplazamiento se hace más rápido, o hacer clic dentro de las guías, con lo que el contenido de la ventana se desplaza “página a página”, por decirlo rápidamente.

Muévase con los ascensores hasta que localice el directorio raíz (probablemente Vd. se encontraba en C:\WINDOWS, con lo que deberá ir hacia arriba en el árbol de directorios) y haga clic sobre él. ¡Caramba!, la ventana con la lista de ficheros cambia automáticamente para mostrarle el contenido del directorio raíz. Muévase de nuevo, ahora en la ventana con la lista de ficheros, hasta encontrar CONFIG.SYS, que debe seguir donde estaba. Y aquí viene lo bueno: inserte un disquete ya formateado en la unidad A:, seleccione el fichero CONFIG.SYS, arrástrelo sobre el botón que representa la unidad A: y suéltelo allí. Le aparecerá una ventana para preguntarle si quiere copiar el tal fichero a A: con dos botones rotulados “SÍ” o “NO”, déle al de “SÍ” y más o menos ya está. Para asegurarse de que la copia ha tenido lugar puede, si lo cree necesario, pulsar (un solo clic) el botón que representa la unidad A:. Le aparecerá su directorio, y podrá comprobar que CONFIG.SYS está allí de verdad.

Si por un casual da dos clics en lugar de uno, se le abrirá una ventana adicional situada en la unidad en cuestión, y esto debería desplegar un amplio abanico de posibilidades ante todo LAO que se precie. Lo dejaremos así, para que disfrute en lo sucesivo de su propia autonomía.

La gestión de ficheros desde el “Administrador de archivos” es un buen ejemplo de todo lo que hemos estado diciendo. Me permito resaltar que lo anterior puede ser más divertido que teclear “COPY C:\CONFIG.SYS A:”, pero no está claro que sea más evidente. Y casi acabamos con el apartado haciendo notar que, por mucho Windows que tengamos, hay que seguir mirando a los ficheros, que siguen donde estaban. Es decir en un directorio determinado de una de las unidades DOS con que contábamos, y con el nombre, tanto completo como abreviado, que siempre han tenido.

Antes de salir de aquí, completaremos un poquito lo anterior. ¿Qué pasa si queremos copiar todos los ficheros del directorio raíz de C:\ a A:?. Sabemos que “COPY C:*.* A:”, lo hacía. Pero ¿nos dejará seleccionar el “Administrador de archivos” más de un fichero para arrastrarlo a A:?. Si pulsamos sin más sobre un fichero diferente a CONFIG.SYS, el nuevo se selecciona, pero a costa de dejar al otro suelto. Si Vd. pulsa “CTRL” (es decir la tecla de control), mientras “pincha” ficheros adicionales, la selección de los anteriores se mantiene, y Vd. pasa a poder seleccionar un grupo cualquiera. Pinchar un fichero ya seleccionado lo

Bailes modernos

libera, y de nuevo se abre un amplio abanico de posibilidades ante nosotros. Pero es mucho clic para tan poca selección, y tal vez barruntemos que debe haber una forma más fácil. En efecto, si Vd. tiene ya seleccionado un fichero, puede desplazar el puntero, y, manteniendo pulsada la tecla “SHIFT” (sí, la tecla de alzamiento de mayúsculas), hacer clic sobre otro, con lo que seleccionara también todos los ficheros que estén en medio. Tras abrir éste último abanico de posibilidades, acabamos con la historia, rogándole que, de ser necesario, vuelva a leer el párrafo anterior.

El hecho es que la gestión de ficheros desde un administrador de archivos de tipo gráfico tiene sus ventajas y sus inconvenientes. Es más rápido para desplazarse por unidades y directorios, para “navegar” por el sistema de ficheros, y es una ventaja el poder seleccionar grupos que no responden a un patrón de comodines determinado¹⁶⁷. Pero en muchas ocasiones es más sencillo abrir una ventana DOS, es decir ejecutar COMMAND.COM desde Windows (el doble clic en “Novell DOS Prompt” que antes citamos) y teclear allí lo que sea. Al final, se recurre a uno u otro método en función de las necesidades e incluso el humor del momento.

Y ya vale de Windows elemental. Estamos preparados para cosas interesantes de verdad y eso es lo que vamos a hacer en el apartado siguiente. Terminemos una vez más con un consejo. Si puede, practique todo lo anterior a la menor ocasión. Repetimos que es más difícil de explicar que de hacer. Debe tener en cuenta la idea del escritorio, pensar que debe disponer sobre él los elementos que quiera usar, y hacerles un hueco para trabajar. Lo que nos lleva, como comentamos al hablar de la configuración que debería tener un PeCé para Windows, a la necesidad de enormes pantallas. 1024x768 es lo conveniente, y 800x600 el mínimo casi irrenunciable. Y también considerar como trabaja Windows en realidad bajo su seductora apariencia, la forma en que se organiza o puede organizarse, y otros asuntos. Justo lo que vamos a ver a continuación.

Bailes modernos

LR-PeCé se mueve mejor con el “twist”

Tras la sofisticada apariencia creada por botones, ventanas, paneles gráficos de diálogo, menús desplegables e iconos varios, todo GUI esconde tan sólo aquello que nos queda cuando salimos de él, es decir un sistema operativo más o menos simple con unos mecanismos de ejecución de programas y de organizar ficheros bastante elementales.

Tras Windows se esconde nuestro viejo DOS, con las mismas particularidades de siempre. Sacar provecho a Windows, orientarse sin dudas por un PeCé que nos lo muestra como interfase de usuario, requiere ser capaz de conectar iconos con programas, conocer qué nuevos elementos organizativos pretende añadir Windows a los ya existentes, qué recursos adicionales aporta que no consistan en un mero maquillaje de los que ya teníamos, y cuales de las cosas que se daban por establecidas han sido cambiadas. Eso es lo importante de verdad y a eso vamos ahora. Tras darle un rápido repaso, acabaremos el apartado, y prob-

167 Si usamos 4DOS.COM como intérprete de órdenes DOS, sí es posible. Cuenta con una orden al efecto, llamada “SELECT”, que pone una vez más de manifiesto que puede existir una diferencia de comodidad o de aspecto al usar una interfase de usuario u otra, pero en realidad es posible hacer lo mismo con todas ellas si se diseñan correctamente.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

ablemente el capítulo, ocupándonos de ver cómo ha ido evolucionando Windows en base a la experiencia acumulada en sus ya numerosos años de vida y haciendo un balance final de sus virtudes y defectos.

MS-DOS usa una línea de órdenes como interfase de usuario y diferencia y organiza datos y programas empleando tan sólo el sistema de ficheros. Los directorios nos permiten agrupar en la forma que deseemos aquellos ficheros que correspondan a programas ejecutables o a datos. Podemos mezclarlos todos en un cajón de sastre enorme, para lo que habilitaríamos un gigantesco directorio al que podríamos llamar “CAJON” (o peor aún, podríamos dejarlos todos en el directorio raíz), o podemos recurrir a una organización más convencional y mucho más conveniente en la que separemos cada programa en un directorio propio y agrupemos los ficheros de datos en una serie de directorios al uso que reflejen su clasificación según criterios de nuestra elección.

Windows es ligeramente diferente. Sus diseñadores decidieron que era conveniente añadir a lo anterior una serie de artificios para visualizar, agrupar y organizar los programas como una entidad separada. ¿Qué mecanismo se estableció para ello?. A estas alturas no debería sorprender que la única figura que por el momento hemos mostrado sobre Windows refleje justamente tal mecanismo. El “Administrador de programas”, ese programa que calificamos de algo especial, es el encargado de poner en práctica el encargo. Los diseñadores de Windows 3.0 (y 3.1 y 3.11 y...) decidieron en su momento que los programas se representaran mediante iconos que permitieran su ejecución y que pudieran clasificarse en grupos representados por ventanas dentro de aquella que abre el propio “Administrador de programas” al ponerse en marcha. Es por supuesto ese mismo “Administrador de programas” el encargado de proporcionarnos herramientas para crear, borrar o modificar “Grupos de programas”, y para llenar tales grupos con los correspondientes iconos, para lo cual podemos también crearlos o borrarlos a voluntad o moverlos de un grupo a otro¹⁶⁸.

En realidad, tal paso dista bastante de ser gratuito. Podemos decir que un GUI no está totalmente definido hasta que no establecemos un mecanismo de acceso, sea el que sea, a los programas que nos interesa usar, o, si lo preferimos así, que la interfase de usuario en un GUI consta de unos recursos comunes (menús desplegables, ventanas, etc.) y un administrador de programas que se ejecuta necesariamente al ponerla en marcha y que acaba de definir su comportamiento. Las ventanas y demás recursos solos, sin un administrador de programas adicional, no servirían nada más que para mostrarnos nuestra vieja línea de órdenes, eso sí en modo gráfico. Y para que cuando tecléamos en ella la orden correspondiente para ejecutar un programa cualquiera, por ejemplo:

```
C:\>CD TLW4_  
C:\TLW4>TLW4_
```

y le diéramos a la tecla gorda de la derecha esa, se abriera la ventana correspondiente que nos permitiera trabajar con él. Puede considerarse que sería una mejora, pero está claro que, de dejarlo así, la interfase de usuario seguiría básicamente igual que antes¹⁶⁹.

168 El número máximo de grupos de programas, al menos en Windows 3.1, es de cuarenta. Tal vez por el momento le parezcan muchos.

169 No se si debería poner esto, aunque desde luego es un ejemplo claro de lo que se quiere decir. En fin, habrá que recomendar que salga de aquí si no quiere liarse o que olvide lo que sigue

Por ir dando detalles de cómo funciona Windows en realidad, que al fin y al cabo es de lo que se trata, aclaremos que lo que sucede cuando tecleamos “WIN” y le damos al retorno de carro no es sino que el PeCé pasa a modo gráfico, se cargan una serie de programitas encargados de gestionar memoria, impresoras, menús, ventanas y otro montón de cosas y, finalmente, cuando todo está preparado, el control se pasa a un programa a través del cual nos comunicaremos preferentemente con el ordenador a partir de entonces. Ni más ni menos que lo que pasaba con “COMMAND.COM” en MS-DOS pero con más recursos a nuestra disposición. Tal programa existirá, deberá ser localizable como un fichero ejecutable, y no está de más que lo busquemos y lo veamos listado en su directorio, para ir perdiéndole el respeto a Windows y quitarle lo antes posible la magia que pudiera quedarle. El directorio es aquel en el que hayamos instalado Windows, normalmente “C:\WINDOWS”¹⁷⁰, y el programa encargado de administrar a los demás, nuestra vía preferente de comunicación en Windows, no es otro que el fichero “PROGMAN.EXE”. Cuando entramos en Windows, PROGMAN.EXE acaba por ejecutarse, y es él el que nos muestra las ventanas de los grupos de programas y el que interpreta dos clics sobre un icono como la orden para que se ponga en marcha el programa correspondiente.

Como con cualquier cuestión de organización en el ordenador la última palabra debe tenerla el usuario, “PROGMAN.EXE” se limita a establecer unos criterios, unas imágenes (grupos de programas representados por ventanas dentro de las cuales hay iconos), y a ofrecernos unas herramientas para adaptar tales imágenes a nuestros deseos. En última instancia somos nosotros los responsables de quitar, poner o redistribuir los programas, sus iconos y sus grupos de la forma que queramos.

Es sencillo hacerlo. Para cambiar un programa de grupo basta con arrastrar el icono correspondiente de una ventana a otra. Y para copiar un icono de un grupo a otro basta con mantener pulsada la tecla “CTRL” mientras llevamos a cabo la operación anterior. El menú principal del “Administrador de programas” nos permite crear grupos nuevos con el nombre que queramos, o nuevos iconos, o investigar que hay detrás de cualquiera de ellos. Nada que deba sorprender a un usuario de Windows mínimamente advertido, o que no se pueda descubrir por uno mismo si se tiene algo de paciencia, curiosidad y se consultan las ayudas. Por tanto, no entraremos a describirlo en detalle, pero debemos destacar el hecho de que al usar Windows ya no sólo tenemos que mantener los directorios organizados, sino también los grupos de programas. Tenemos un poquito más de trabajo.

inmediatamente si no acaba de entenderlo. La cuestión es que Windows no es el único GUI, ni mucho menos; que existen diseños mucho más claros en cuanto a la identidad de sus componentes; que ya citamos a Unix con anterioridad; y que debemos intentar exponer las ideas generales que sirven para cualquier ordenador con cualquier sistema operativo, evitando encerrarnos en DOS, pues ningún LAO sabe donde puede acabar. En Unix, montar un GUI exige lanzar X-Windows, su sistema de ventanas, y una vez allí ejecutar un “gestor de programas”, por llamarlo de alguna manera, para acabar de completar el invento. Sin el segundo, la interfase de usuario de que se disfruta es ni más ni menos que la versión Unix de lo que se acaba de narrar.

170 No hay problema en instalar Windows en otro directorio, o en otra unidad, por lo que lo anterior no pasa de ser de la situación más común.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Al igual que la instalación de un programa en MS-DOS intentaba normalmente hacer las cosas por nosotros creando los directorios necesarios donde dejar los ficheros correspondientes, la instalación de un programa Windows creará además con frecuencia el grupo de programas y los iconos para acceder a él o a los distintos módulos de interés que lo forman. Es decir que no debemos preocuparnos demasiado por lo que se nos viene encima, pero conviene conocer qué está pasando y como podemos intervenir, ya que en ocasiones se nos preguntará en qué grupo queremos dejar los iconos, o si hace falta crear uno nuevo o usar uno ya existente, o cualquiera otra de un centenar de posibilidades que requieren que sepamos que existe una cosa que se llama “grupos de programas” y para qué sirven.

Puestos a ello, podemos plantearnos cómo ejecutar un programa de instalación desde Windows, ya que obviamente no es posible esperar que exista el icono correspondiente. Este es otro de los servicios que el “Administrador de programas” debe proporcionar, y el procedimiento para hacerlo entra dentro de lo razonable. Una opción de su menú “Archivo”, reza justamente “Ejecutar”, y si la seleccionamos podremos teclear en la línea de órdenes que nos presenta el nombre del programa correspondiente. O, lo que es muy frecuente y resulta francamente cómodo si no sabemos ni su nombre concreto, pulsar el botón “Examinar” y movernos por los directorios en que esperamos se encuentre para seleccionarlo allí. Busque, si está instalando un programa distribuido en disquetes, algún fichero que se llame “A:\INSTALAR.EXE”, o “A:\INSTALL.EXE” o quizá “A:\SETUP.EXE”. Si duda, es probable que el manual del programa, en el apartado “Instalación”, sea lo suficientemente explícito como para que la operación pueda realizarse sin traumas¹⁷¹.

Otra opción del menú “Archivo” del “Administrador de programas” que debe ser descrita sin discusión posible es “Propiedades”, pues nos mostrará inequívocamente qué diablos está pasando con Windows y donde ha quedado nuestra línea de órdenes de DOS, aquél antediluviano invento, ahora tan lejano al parecer, en el que debíamos teclear cosas como “C:\TLW4\TLW4” y quizá añadir diez o doce parámetros separados con guiones o barras, o cualquier otro carácter más o menos aberrante, para lograr poner en marcha un programa. Abra uno de los grupos de programas, uno cualquiera, y seleccione (un clic encima) cualquier icono allí presente. A continuación, elija “Archivo” en el menú principal, y una vez abierto, complete el proceso activando “Propiedades”. Pongamos un punto y aparte para darle algo de suspense a la cosa.

Lo que verá será un panel de diálogo que, sin delicadeza alguna, le demostrará que la línea de órdenes y sus parámetros siguen allí. Esperemos un poco para que se reponga de tan desagradable sorpresa, y pasemos a analizar su contenido. Lo más cruel son cuatro zonas de texto, manipulables mediante un editor de líneas, rotuladas “Descripción”, “Línea de comando”, “Directorio de trabajo” y “Tecla de método abreviado” que contienen respectivamente la etiqueta del icono (probablemente el elemento que querremos cambiar con

171 Es normal que el programa de instalación le informe también de cualquier error que pueda producirse durante su operación. También es normal que cada instalación cree su propio grupo de programas si no se le dice lo contrario, en caso de que nos deje. Eso lleva a que el límite de cuarenta grupos como máximo impuesto por Windows no sea tan inalcanzable, y el correspondiente grupo no aparezca cuando Vd. lo busque. Por cierto, intentar crear el grupo cuarenta y uno no siempre provoca un error, al menos esa es mi experiencia, y las consecuencias de todo lo anterior son persistentes quebraderos de cabeza hasta que se descubre qué ha pasado, y la necesidad de tomarse la molestia de organizar a mano los programas y sus grupos.

mayor frecuencia), la orden en estricto DOS que se ejecuta cuando le damos dos clics, el directorio que el programa emplea para trabajar y, de haberla, la combinación de teclas que permite acceder a él directamente. Debajo de ellas se muestra el icono asociado por el momento al programa en cuestión y un cuadradito que podemos seleccionar si queremos que el programa se ejecute minimizado, y a la derecha una serie de botones que nos dan acceso a cosas tales como cambiar el icono, por si el actual no nos gusta, o el ya familiar “Examinar” por si queremos que pase a estar asociado a otro programa. Como en su momento se dijo, la tecla de tabulación, o el ratón pulsado en uno u otro lugar, nos darán acceso a todas las opciones. Por si necesitamos más explicaciones para usarlo correctamente, también hay un botón de “Ayuda” aunque conviene recordar que “F1” sigue estando en el mismo lugar del teclado en que la dejamos la última vez.

Las explicaciones que nos deben ocupar ahora no son otras, sin embargo, que las que nos recuerden que la situación a que nos hemos enfrentado no es en absoluto distinta a aquella que describimos a la hora de hacer funcionar un programa con interfase basada en menús, hace ya algunos capítulos. Por mucho “PROGMAN.EXE” que usemos, repetimos una vez más, su cometido principal no va a ser sino el de conectar nuestros iconos y pulsaciones de ratón con lo que en realidad un ordenador basado en DOS entiende, es decir órdenes tipo texto con sus parámetros, directorios y demás. Si ahora vemos y manipulamos iconos, en algún lugar deben estar, y en algún sitio debe encontrarse asimismo el mecanismo de asociación de unos y otros. El ordenador jamás hace nada solo, y todo lo que nos muestre debe estar almacenado en algún sitio. La única cuestión que se plantea es que las interioridades sean más o menos accesibles al usuario.

Ahora que hemos visto lo que se oculta tras el mecanismo de ejecución de un programa Windows, podríamos aprovechar para ver qué hay tras la posibilidad de ejecutar programas DOS desde Windows. Ya dijimos que la compatibilidad con ellos fue algo sumamente importante para el éxito de Windows 3.0 y sus sucesores y sigue estando a nuestra disposición. Un programa Windows, con sus iconos predefinidos y demás, cuenta con trabajar bajo Windows y exige que se le lance desde allí. Si intentamos ejecutarlo desde DOS, a través de la línea de órdenes, lo único que lograremos será que nos aparezca un mensaje diciendo “Este programa requiere Windows para funcionar” o algo parecido. Los programas DOS son accesibles desde Windows porque nuestro GUI para PeCés se toma la molestia de ser respetuoso con las condiciones de trabajo que estos requieren. De hecho, Windows no hace sino ampliarlas, sin cambiarlas sustancialmente. Pero será preciso decirle a Windows de alguna forma una serie de cosas sobre el programa DOS a ejecutar. Habrá que aclarar si necesita o no memoria EMS, cuanta de la memoria de Windows queremos reservar, si queremos que se ejecute en una ventana dentro de Windows o si preferimos que ocupe toda la pantalla, dejando de mostrarnos la de Windows¹⁷², y en fin, una buena cantidad de detalles similares.

La solución al problema consiste en ampliar lo visto con las “Propiedades” de los iconos. Windows admite un nuevo tipo de ficheros ejecutables. Además de las extensiones ya conocidas que permiten distinguir los programas de los ficheros normales (.EXE, .COM, y

172 En cuyo caso una combinación de teclas, en concreto ALT+Tab, nos permitirá pasar de una pantalla a otra. Windows nos muestra un texto de ayuda al lanzar aplicaciones DOS en pantalla completa para darnos a conocer o recordarnos este tipo de detalles.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

.BAT), aquellos ficheros que muestren la extensión “.PIF”, serán también ejecutables, y corresponderán a programas DOS en los que se ha definido su comportamiento bajo Windows. En concreto, “PIF” es la abreviatura de “Program Information File”, o “Fichero de información sobre el programa”. Muchos programas DOS incluyen en la actualidad un fichero .PIF para lanzarlos en las mejores condiciones posibles desde Windows. Para acceder a él con facilidad, podremos mediante el “Administrador de programas” asociarle un icono, que también se incluye muchas veces como un fichero con la extensión “.ICO”, e incluirlo en un grupo de programas o crear uno propio para su uso exclusivo. Aunque hay otras posibilidades que pronto veremos. Y podremos también modificar los parámetros asociados a su ejecución, si es que nos atrevemos o resulta necesario, mediante el programa correspondiente. En concreto el “Editor PIF”, que se encuentra en el grupo “Principal”, uno de los que Windows monta automáticamente con las utilidades que se incluyen en nuestro GUI para PeCés.

Acabaremos con el asunto de los programas DOS en Windows con un par de notas. No es posible lanzar en una ventana Windows aquellos programas DOS que trabajen en modo gráfico, y creo, aunque ya no estoy muy seguro de ello, que el puntero no funciona con aquellos programas en modo texto que, siendo posible ejecutarlos en una ventana, emplean el ratón. Al menos, ese es el comportamiento a esperar con las versiones estrictas de Windows 3.0, 3.1 y 3.11. Existen, y son cada vez más populares, adiciones a Windows que evitan estos problemas y dejan que casi cualquier programa DOS se beneficie de las ventanas y el ratón en Windows. Son las denominadas “extensiones 32 bits” o Win32s. El único problema que queda entonces es la pérdida de velocidad que se experimenta al hacer que los programas DOS en modo gráfico se ejecuten en una ventana Windows, pero en fin, Vd. verá. Por supuesto, dichas extensiones están incluidas, y ampliadas, en Windows 95¹⁷³.

Cada vez nos queda menos para acabar el capítulo, querido LAO. ¡Ánimo!. Un último esfuerzo y pasaremos a tratar temas tal vez menos áridos y quizá hasta más divertidos. Los que nos quedan por tratar son, de todos modos, ineludibles. Y el siguiente nos devuelve al sistema de ficheros y a considerar cómo se integra en Windows, y cual es su importancia para desenvolverse bajo tan popular GUI.

Pues bien, hay que comenzar por hacer notar que todo lo dicho sobre ficheros “.PIF”, propiedades de los iconos o programas Windows y demás, debería haber puesto claramente de manifiesto que el sistema de ficheros sigue estando allí. Que el “Administrador de programas” es tan sólo la carrocería Windows, pero que el motor sigue siendo nuestro cúmulo de directorios y ficheros DOS de toda la vida. No sólo el sistema de ficheros no cambia en absoluto en Windows (afirmación que no puede extenderse a Windows 95 o NT, tema sobre el que ya hablaremos cuando le toque) sino que en realidad el “Administrador de programas” de Windows simplemente complementa al “Administrador de ficheros”, y que mucho de lo que hemos visto puede hacerse directamente desde éste último, o implica una interacción entre ambos en la cual el “Administrador de ficheros” es, por así decirlo, el que de verdad corta el bacalao.

173 Yo tengo instaladas las Win32s hace tiempo, y es por eso por lo que ya no me acuerdo con detalle de como eran las cosas sin ellas. Valen para mucho más que para arreglar el ratón. Y también existen otras extensiones más recientes. Por ejemplo las WinG, que aparecieron en marzo de 1.996 y abren la posibilidad de aumentar la velocidad de visualización en Windows 3.1.

El “Administrador de ficheros” es necesario para copiar ficheros, duplicar, crear o borrar directorios, acceder a unidades y cualquier otra tarea de las que le son propias. pero además, si sabemos cual de esos ficheros que tan alegremente nos muestra corresponde al programa que queremos ejecutar (será un .EXE, probablemente), bastarán dos clics sobre él para ejecutarlo directamente. No necesitamos los iconos para nada. O, si queremos añadir un programa a un grupo, es posible hacerlo abriendo éste último a través del “Administrador de programas”, lanzando el “Administrador de ficheros”, localizando en él el fichero que contiene el programa a añadir (insistimos de nuevo, tal vez un .EXE, quizá un .COM o .PIF o tal vez .BAT) y arrastrándolo hasta la ventana del grupo donde queramos dejarlo¹⁷⁴. En buena medida, el “Administrador de ficheros” puede sustituir al “Administrador de programas”.

Si hemos de ser precisos con lo que decimos, se trata en realidad de mostrar el hecho de que si bien la presencia de un “Administrador de programas” en un GUI no es gratuita, la elección de su forma de representar los programas y acceder a ellos, la manera en que completa la interfase de usuario del GUI, esa sí, es totalmente arbitraria. Igual que Windows usa iconos y ventanas, podría emplear un “Administrador de ficheros” ampliado para hacer lo mismo tal vez mejor.

De hecho, y tras el deslumbramiento inicial, se descubre rápidamente que el esquema de organización impuesto por el PROGMAN.EXE de Windows 3.0, 3.1 y 3.11 es escasamente práctico, y que puede que el único motivo racional que justificara semejante diseño fuera proporcionar ventanas en abundancia a los frustrados usuarios del PeCé de 1.990, haciendo que Windows (“ventanas” en inglés) hiciera honor a su nombre incluso en exceso. Y también que el “Administrador de ficheros”, aparte de otras deficiencias, peca por el mismo lado. En 1.993, o incluso antes, los usuarios del PeCé ya no necesitaban ver infinitas ventanas para compensar sus traumas frente a los propietarios de otros ordenadores, y aparecieron otras formas de organizar el funcionamiento de Windows, otros PROGMAN.EXE, igual que antes habían aparecido alternativas a COMMAND.COM para mejorar la interfase de DOS. Pero esto es volver al problema de la configuración de los ordenadores, ahora en Windows. Lo dejaremos para un poco más adelante, como corresponde a un tema apropiado al chismorreo, o, si se prefiere así, altamente subjetivo.

Liquidemos algunas tuercas antes de darnos al cotilleo. Windows impone algunos cambios profundos en una serie de aspectos firmemente establecidos en DOS. En primer lugar, el conjunto de caracteres, nuestro familiar ASCII del DOS al que tanta importancia hemos dado porque se la merece, se sustituye en Windows por otro conjunto diferente, el denominado ANSI, abreviatura de “American National Standards Institute”. Los primeros 128 caracteres ASCII se respetan, pero existen diferencias en la parte del “ASCII extendido” es decir entre los caracteres 128 a 255, como era bastante fácil de prever. No es que debamos desmayarnos, o arrancarnos los cabellos a mano ante el desaguizado, pero debemos permanecer atentos a algunas aberraciones que pueden aparecer cuando accedamos a ficheros creados en DOS desde programas Windows. Algunos caracteres, especialmente eñes, vocales acentuadas, o aperturas de admiración o interrogación entre otros, pueden ser sustituidos por símbolos de libra esterlina, cuadraditos, o cualquier otra cosa. Muchos programas

174 La maniobra, realizada con un programa Windows, creará el correspondiente icono. ¿De donde ha salido?. Los ficheros .EXE de los programas Windows suelen contener uno o varios de ellos.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Windows, de todos modos, se toman la molestia de devolver la cordura a nuestros textos si les indicamos, cuando nos lo pregunten, que van a trabajar con ficheros codificados en “ASCII del DOS”.¹⁷⁵

Existen también algunos cambios nimios, más bien anecdóticos, pero que pueden acabar siendo molestos. O quizá a Vd. le resulten una bendición, quien sabe. Su número y naturaleza depende de la versión concreta de Windows que emplee. Por ejemplo, Windows 95 ha añadido un par de teclas más al ya clásico esquema del teclado AT, un poco en la línea de los Apple Macintosh. Uno que a mí me resulta particularmente difícil de asimilar es la insistencia en hablar de “carpetas” en lugar de directorios cuando se está en Windows¹⁷⁶. Los directorios siguen estando ahí aunque no queramos verlos, y el resultado neto de tan audaz operación de maquillaje a base de brochazos de Titanlux es que ahora hay que llamarlos de otra forma más y se hace más difícil conectar lo que pasa en Windows con lo que pasa en DOS. Admito que se hable de “ficheros” o de “archivos” indistintamente, pues ambos términos son traducciones correctas del inglés “files”, pero hablar de “carpetas” en un PeCé sólo conduce a que un LAO bien intencionado y con ganas de aprender pueda estar buscando banamente la diferencia entre “carpeta” y “directorio” durante varios meses.

Pero hay cambios que no por sutiles dejan de ser importantes. Uno de ellos quizá es inherente a todo entorno gráfico, sea el que sea¹⁷⁷. En un GUI se invierte el mecanismo para especificar las acciones a realizar con respecto al normal en una interfase de línea de órdenes. En ésta última estamos acostumbrados a una forma de decir las cosas que responde al esquema “acción a realizar seguida del elemento sobre el que llevarla a cabo”, por ejemplo “DIR C:\WINDOWS/W”. Resulta, debe reconocerse, bastante natural. Es más o menos lo mismo que cuanto le decimos a alguien “Vete al cuerno”, una expresión que muchos LAOs desearían probablemente poder dirigir al autor en este momento, hartos ya de tanto rollo. Especificamos la acción (“ir”, en este caso), y luego el sujeto que debe realizarla (“tú”, supongamos que sea el autor). Podemos incluso completarla con algunos parámetros que resulten adecuados (“al cuerno”, en el ejemplo anterior, o “/W” en el caso del directorio). En un GUI es preciso seleccionar primero el elemento con que vamos a trabajar y luego especificar la acción, por ejemplo eligiéndola en un menú, que debe serle aplicada. La cosa se completa precisando los parámetros adecuados para llevarla a efecto, por ejemplo mediante un panel de diálogo que se nos muestra a continuación. Sería más o menos lo mismo que decirme “Tú ir”, a lo que yo contestaría “¿Dónde?”, para que se me indicara “Al cuerno”.

175 Ya que estamos, hay que aprovechar para aclarar también que a nivel informática en general existe cierta tendencia hacia la sustitución del ASCII por un sistema de codificación en 16 bits, ya no en 8, que responde al nombre de Unicode. Creo que las más modernas versiones de Windows, y en particular Windows 95, lo usan, aunque no debería hacerme mucho caso sobre este punto en concreto. La documentación de Windows 95 o Windows NT, si Vd. usa alguno de ellos, debería ser mucho más categórica que yo sobre este punto y otros muchos.

176 El primero me parece una estupidez, y el segundo es, desgraciadamente, una tendencia general en el mundo de los GUIs. En mi opinión, claro.

177 Quiérese decir que es probable que lo que sigue sea aplicable también a OS/2, Macintosh, X-Windows, o cualquier otro GUI.

El asunto no es realmente muy grave, pero conviene tenerlo en cuenta porque muchas de las opciones de los menús que aparecen en un GUI, dan por supuesto el elemento con que van a trabajar si no existe alguno ya seleccionado. Si una acción se muestra como disponible en un GUI, es que va a aplicarse sobre algo.

Y lo anterior nos lleva a otro tema importante, como todo LAO que lo sea debería estar esperando hace un par de párrafos. Al principio de los tiempos se contaba con que los GUIs resolvieran el problema del intercambio de información entre programas, y a sus diseñadores no se les olvidó a pesar del tiempo transcurrido desde que se comenzaron a proponer hasta que tuvieron éxito. De hecho, en Windows existen al menos tres mecanismos que buscan resolverlo a distintos niveles. El primero, el que más nos debe interesar conocer por ser el de uso más frecuente es el denominado “portapapeles”. Es posible en Windows pasar datos de un programa a otro empleando una zona de almacenamiento en memoria a la que se denomina así. Podemos imaginarla como una especie de hoja en sucio en la que podemos escribir lo que sea desde un programa, para leer su contenido desde otro. O como un fichero sin nombre ni ubicación en nuestros discos, cuyo única identificación es la un ideal “portapapeles”, en el que dejar gráficos, trozos de texto o bloques de cuentas, para leerlos desde otros programas cuando haga falta¹⁷⁸. La metáfora que se emplea en general para manipular el contenido del portapapeles es la denominada “cortar y pegar”. Nos imaginamos que cortamos un trozo de aquello con lo que estamos trabajando y lo pinchamos en un lugar conocido. Abrimos otro programa y pegamos en el documento que con él estemos manipulando el trozo de papel virtual que habíamos almacenado. El lugar concreto en que se incorpora el contenido del portapapeles es la posición que marquemos con el puntero, y el modo de indicarle a LR-PeCé que lo haga es elegir la opción “Pegar” del menú “Editar”. En el mismo menú veremos un “Cortar” que llevará, para su uso posterior, un bloque que hayamos marcado (arrastrando sobre él el puntero del ratón, por ejemplo) al portapapeles, borrando a la vez cualquier dato que pudiera contener y quitándolo del documento con que trabajemos, y un “Copiar” que hará lo mismo pero no lo retirará del documento en uso. En resumen, si elegimos “Pegar” querrá decir que algo podrá ser pegado, y veremos aparecer en nuestro documento el contenido del portapapeles, ya que toda acción en un GUI se realiza sobre algo previamente seleccionado o que se da por supuesto. Si queremos saber si hay algo en el portapapeles en un momento determinado, hay un programa para ello en el grupo “Principal”, denominado justamente “Visor del portapapeles”. Debe hacerse notar asimismo que en Windows 3.11 el portapapeles se ha convertido en una especie de “libreta de notas”, con capacidad para almacenar más de un elemento (y, por favor, lea la nota)¹⁷⁹.

178 Es justamente a través de ficheros, en este caso nada virtuales y con sus atributos de formato bien precisados, como decidieron comunicarse los programas DOS. Tal mecanismo sigue disponible en Windows, y siempre podremos recurrir a grabar los datos que queramos en un fichero concreto de un disco concreto desde un programa, par luego leer ese mismo fichero desde otro programa que entienda ese mismo formato. Muchas veces interesa hacerlo así, aunque el portapapeles nos evita la molestia de elegir la ubicación del fichero, su nombre y su formato. Windows se ocupa de los detalles automáticamente, pero no siempre los procesos automáticos producen el efecto exactamente deseado.

179 Aunque en la exposición estamos hablando de intercambio de datos entre programas, las operaciones con el portapapeles son también posibles dentro de un documento único, y de hecho éste es el ámbito en que se emplean con mayor frecuencia. “Cortar” y “Pegar” servirá por ejemplo para cambiar de sitio un párrafo en un texto, en tanto “Copiar” y “Pegar” nos permitirá

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Los otros dos mecanismos de intercambio de datos que Windows proporciona son más sofisticados y se salen del ámbito del texto. Responden a las iniciales “DDE” y “OLE”, abreviaturas de “Dynamic Data Exchange” y “Object Linking and Embedding” que podríamos traducir por “Intercambio dinámico de datos” y “Enlace e inclusión de objetos”. Aunque el DDE es anterior, el mayor énfasis en la actualidad se hace sobre OLE, que ya va por su versión 2. Se habla de “programas Windows que soportan OLE 2.0”, y nos conformaremos por el momento con que le suene el acrónimo, fácil de recordar en España por sus implicaciones raciales. En el último capítulo es probable que digamos algo más sobre los “objetos”, que debe reconocerse que están bastante de moda.

Ocupémonos de las tuercas que nos quedan de Windows antes de acabar el capítulo con unas pocas figuras más y los chismorreos ya prometidos. ¿Qué directorios componen Windows, para qué sirven, y qué más debemos saber de sus interioridades?. En principio, los componentes de Windows irán a parar a la unidad y el directorio que indiquemos en el momento de instalarlo. Por defecto, su ubicación será C:\WINDOWS. Dentro de él, al menos existirá un directorio \SYSTEM (pueden ir apareciendo más conforme vayamos instalando programas), y los nombres deberían indicarnos que en el segundo se agruparán los componentes encargados de hacer el trabajo más sucio. Debemos saber que el funcionamiento de los programas Windows se basa en buena medida en una serie de pequeños trozos de programa utilizables por todos ellos que se llaman “bibliotecas dinámicas”. Los ficheros que los contienen lucen una extensión “.DLL”. Traducido, cuando se instala un programa Windows es posible que añada algún fichero “.DLL” al directorio \WINDOWS\SYSTEM, y que emplee cuando se ponga a funcionar algunos de los que ya estaban allí. Conforme se instalan más y más programas se van copiando más ficheros “.DLL”, más ficheros con programas no siempre de nombre muy claro, creando más ficheros con grupos de programas, y en buena parte van a parar a \WINDOWS y \WINDOWS\SYSTEM. Es decir que dichos directorios empiezan a crecer, y crecer, y crecer... y no es fácil quitar un programa porque en el proceso nos podemos cargar, si lo hacemos a mano, algún fichero “.DLL” que sea de uso común, con lo que un programa que deseemos conservar ya no podrá ejecutarse.

También contribuyen en buena medida al aumento de tamaño de los directorios de Windows los ficheros que contienen tipos de letra adicionales. Es casi inevitable que todo usuario neófito en el uso de un PeCé se deje deslumbrar por las posibilidades tipográficas de Windows. Como casi siempre, el furor se apacigua en buena medida al descubrir que los efectos inmediatos de añadir 200 tipos de letras adicionales a Windows consisten ante todo en hacerlo algo aún más lento de lo que por su natural ya es y desperdiciar algunas decenas de megabytes más de nuestro disco duro, nunca lo suficientemente capaz.

Lo de los tipos de letra es algo sobre lo que tal vez volvamos más adelante, pero baste por el momento con indicar que al ser los tipos “True Type” de Windows escalables, es decir que la forma general de una letra puede usarse prácticamente en cualquier tamaño, es

repetir un bloque ya existente, quizá para obtener la versión final de la copia a base de cambiarla ligeramente. Muchos programas ofrecen métodos propios para hacer más ágiles estas modificaciones internas de un documento. Es por ejemplo frecuente que, una vez marcado, un bloque de datos pueda ser “arrastrado” directamente a otra posición, con lo que nos evitamos una operación de “Cortar” y “Copiar”. Las ayudas de los programas que Vd. use, o sus manuales, deberían resultarle de gran ayuda a la hora de descubrir este tipo de cosas.

suficiente con un tipo “con serif” y otro “sin serif” que resulten agradables a la vista. Sólo hay que añadir las fuentes con símbolos raros (matemáticos, por ejemplo) que se puedan necesitar. A no ser que se ocupe Vd. de temas de diseño gráfico, en cuyo caso qué le vamos a decir que Vd. no sepa. Sabrá ya, entre otras cosas, que un tipo “con serif” es aquel que incorpora trazos de grosor diferente en el diseño de los caracteres. Como éste en que se está escribiendo habitualmente, sin ir más lejos. En uno “sin serif” el trazo es de grosor constante, como los listados de directorios que hemos mostrado en alguna ocasión. Sabrá también que hay tipos con espaciado proporcional y otros con espaciado fijo, y que cada tipo tiene un nombre propio que no está de más conocer. Por ejemplo y siguiendo sin ir muy lejos, este tipo de letra que está leyendo ahora mismo se llama “Garamond”.

¿Que Vd. desconocía lo del “serif” y creía que estábamos hablando de alguna película del Oeste y tampoco tiene muy claro qué es eso del “espaciado proporcional”? He aquí otro asunto al que quería llegar, también bastante general, y sobre el que de nuevo profundizaremos más adelante. La cuestión es que hacer las cosas bien en el campo que sea exige conocimientos especializados, y el tema de la impresión es casi un coto privado de gente extraña con raros saberes, un lenguaje propio, y hasta cierta sensibilidad artística. Para moverse en él es preciso saber que el tamaño de una letra se denomina en realidad “cuerpo”, que “Times Roman” es el nombre de un tipo de letra (más correctamente deberíamos hablar de una “fuente”) bastante común y dotada de “serif” al igual que “Arial” lo es de una sin tal atributo, y que para que algo quede impreso con corrección el espacio que ocupa una “m” debe ser mayor que el que ocupa una “i”, y la separación entre caracteres debe ser variable. Los tipos capaces de lograr lo anterior son “de espaciado proporcional”, en tanto aquellos en los que las letras ocupan un espacio fijo y tienen una separación constante son “de espaciado fijo”. Todo eso y muchas cosas más, tales como que determinados grupos de letras no se separan tanto como para que no se superpongan parcialmente (y henos aquí ante el concepto de “kerning”), deberemos aprender si queremos usar a fondo un invento como Windows que nos proporciona con largueza un montón de recursos para lograr imprimir documentos con aspecto realmente profesional. En resumen, cualquier herramienta informática, por bien realizada que esté, exige del usuario cierta formación para sacarle provecho, y cuanto más potente sea, mayores serán las exigencias. No es lo mismo llevar un Vespino que una Ducati 900. O sea que su nivel de conocimientos no debe ser menor en Windows que en DOS sino al contrario, debe ser notablemente mayor, ya que Windows le da más posibilidades. Por cierto “Times Roman” y “Arial” son dos de las fuentes que Windows monta por defecto, y su empleo juicioso debería bastar para cualquier usuario normal. Lo que quiere decir que no se complique demasiado la vida.

Porque ya se encargará Windows de complicársela bastante, no crea. Al hilo de lo anterior, estamos bordeando otro tema fundamental a la hora de usar la informática con provecho, y aunque nos ocuparemos con mayor profundidad de él en otro capítulo, no estará de más por ahora que lo vayamos presentando. MS-DOS es más bien pequeño y sencillo en su estructura, en tanto Windows es innegablemente más grande y mucho más complicado. Para que el ordenador sea más cómodo y seductor es preciso pasar de un sistema formado por 145 ficheros y que ocupa unos 7 megabytes a otro que necesita 1.044 ficheros y llena 50 megabytes para hacer básicamente lo mismo. Las cifras corresponden a mi ordenador en este momento. Y la mayor complejidad ocasiona un número ciertamente mayor de problemas potenciales. Es posible entrar con cierta facilidad a configurar DOS, editando uno mismo CONFIG.SYS y AUTOEXEC.BAT, pero es notablemente más difícil hacer lo mismo en Windows. No es fácil corregir problemas en Windows, ni sencillo hacer algo tan elemental como desinstalar un programa a mano, ya dijimos que se corre el riesgo de cargarse algún

CAPÍTULO 7. PeCés: MS-DOS/Windows.

fichero “.DLL” de uso general. Por fortuna, las fuentes se ponen y quitan con mayor sencillez, usando un programa al efecto que podemos encontrar una vez más en el grupo “Principal” y que se llama “Panel de control”. Luego volveremos sobre él.

En resumen, que debe contar con que Windows no va a dejarse manipular con tanta sencillez como DOS, debido a su mayor complejidad. ¿Que no entiende de qué estoy hablando, dice?. Pongamos un ejemplo. Un buen día, tras instalar un programa cualquiera tal vez simplemente para probarlo, o cambiar, pronto veremos cómo, la resolución y el número de colores de su pantalla en Windows, descubre que su impresora le imprime sólo la mitad superior de cada cinco líneas de texto cuando emplea otro programa que llevaba varios meses trabajando sin fallo¹⁸⁰. No es evidente cómo quitar el nuevo programa, además de que no es tan seguro que sea realmente el culpable del chandrío aunque varios dedos le apunten, y pasar al modo de pantalla anterior no resuelve nada. Tal vez hace un mes Vd. cambió alguna cosa más, de la que ya no se acuerda, o quizá lo que ocurre es que el nuevo programa ha actualizado algún fichero “.DLL” de uso general que resulta incompatible con su driver de toda la vida para su impresora en Windows. Aunque tengamos la suerte de identificar el problema inequívocamente, la solución no siempre es fácil. ¿Cómo restauramos la vieja DLL, por ejemplo?. Ante tal situación, lo más que puede hacer un novato es adoptar una serie de precauciones y poco más. Expongamos las más obvias.

Lo primero es que, hasta que realmente se aclare sobre cómo funciona Windows, deje que sean los propios programas Windows los que se ocupen de hacer las cosas. Por ejemplo, no borre a mano (quizá desde el “Administrador de ficheros”) aquellos ficheros que Vd. cree que contienen tipos de letras que ya no necesita. Use el módulo correspondiente del “Panel de control” que ya citamos, que para eso está, y elija la opción que le permite eliminar los ficheros del disco. Y para suavizar el problema de desinstalar una aplicación, no estará de más que se tome la molestia de indicarle en su momento al correspondiente programa de instalación que la monte en un directorio propio. Al menos así podrá borrarlo en su mayor parte si es menester sin temer efectos colaterales imprevistos (nos olvidaremos de aquello que haya podido dejar en \WINDOWS o \WINDOWS\SYSTEM, qué le vamos a hacer). Si el programa crea un icono de “Desinstalación”, cosa por suerte cada vez más frecuente en la actualidad, úselo preferentemente si alguna vez quiere quitarlo¹⁸¹. Y en general, enrede lo menos posible y asegúrese una vía de retorno sólida por si algo sale mal cuando no le quede más remedio que hacer probatinas. Más adelante concretaremos más cómo hacerlo, tampoco es tan difícil.

Una idea bastante correcta del nivel de complejidad a que nos enfrentamos en Windows puede obtenerse sin más que visualizar dos ficheros bastante importantes que aún no hemos nombrado y que nos van a llevar en volandas hacia ese ansiado final del capítulo que venimos reiteradamente anunciando. Ya vimos que al arrancar el ordenador en DOS, CONFIG.SYS

180 No me lo invento. Me pasó exactamente eso hace cosa de año y medio, más o menos. Sin embargo, el diagnóstico y su posible solución no fueron como se cuenta a continuación. Más adelante pondremos algún ejemplo totalmente ajustado a la realidad.

181 Quitar programas de Windows es realmente un problema. Tanto que se han desarrollado programas especiales que se encargan sólo de eso. No me atrevo a recomendarlos, sin embargo, ya que carezco de experiencia en su uso y lo poco que he visto tampoco es que me haya impresionado demasiado.

y AUTOEXEC.BAT se encargaban de completar la configuración de LR-PeCé y ponerlo a nuestro gusto. Pues bien, también Windows tiene sus equivalentes. Al menos dos ficheros se ejecutan al entrar en él para establecer una configuración que funcione y adecuarlo a nuestras preferencias. Se llaman "WIN.INI" y "SYSTEM.INI", y están en el directorio \WINDOWS. Con esos nombres debería estar claro que SYSTEM.INI es más o menos el equivalente a CONFIG.SYS en DOS, mientras WIN.INI es el sosias de AUTOEXEC.BAT. ¿Cómo puede verlos, pregunta, ya que se está muriendo de curiosidad?. Abra el "Administrador de ficheros", y busque en el directorio \WINDOWS\SYSTEM un fichero llamado "SYSEDIT.EXE". Dos clics sobre él lo pondrán en marcha (no, no existe el correspondiente icono en el "Administrador de programas") y verá cuatro ventanas con los contenidos de CONFIG.SYS, AUTOEXEC.BAT, WIN.INI y SYSTEM.INI. Recórralas un poco. Si es la primera vez que se enfrenta a ellos, la cabeza le dará vueltas ante WIN.INI y SYSTEM.INI¹⁸². Terminemos diciendo que no son los únicos ".INI" que Windows mira de vez en cuando. Hay más. Cerraremos temporalmente el asunto precisando que existe una diferencia fundamental entre los ficheros ".INI" y los ".BAT". Ambos son ficheros texto, pero los segundos permiten automatizar el funcionamiento del ordenador, programarlo vaya, en tanto los primeros nos dejan tan sólo colocar los valores adecuados en una serie de variables que controlan el modo de operación de Windows. Windows, para hacer cualquier tipo de trabajo, requiere que nos sentemos ante él y empecemos a mover el ratón y pulsar botones, en tanto que los ficheros ".BAT" de nuestro denigrado DOS abren una vía para que, aunque sea con cierto esfuerzo previo, podamos irnos tranquilamente con la novia (o novio) mientras dejamos al ordenador trabajando en automático¹⁸³.

Aplicándonos el cuento recién enunciado, aunque los ficheros ".INI" citados especifiquen la configuración de Windows, mejor será cambiarla a través del programa que corresponda. De hecho existen dos programas de obligado conocimiento para hacerlo. Como el asunto a resolver es importante (la configuración siempre lo es), se encontrarán ambos en el grupo "Principal".

El primero, el que se ocupa de las cosas más normalitas y menos problemáticas, ya lo hemos citado. Es el "Panel de control". Como seguro que vamos a acabar ante él, será mejor que tengamos cuanto antes una idea de su aspecto. Es más o menos como se ve en las dos ventanas de la parte superior izquierda de la figura siguiente.

El segundo, dedicado a asuntos más conflictivos y que afectan profundamente al funcionamiento de Windows, es el llamado "Instalar Windows". En él podemos establecer el tipo de pantalla, de ratón, y de teclado que vamos a usar, y el tipo de red local al que conectarnos.

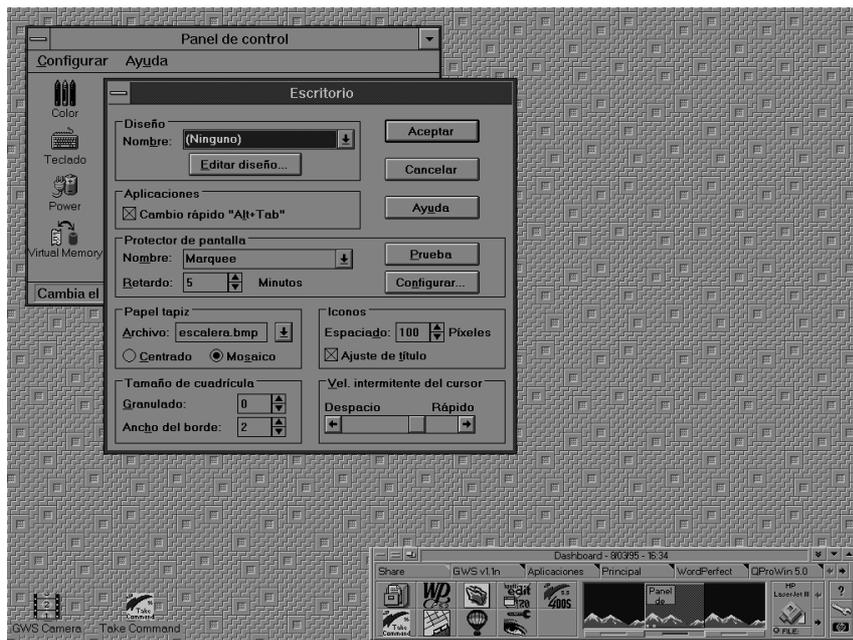
182 Lo anterior es estrictamente aplicable a Windows 3.1. No me atrevo a decir que valga para Windows 3.11 y mucho menos para Windows 95. Éste último en concreto, se ha diseñado con criterios más "cerrados" cara al usuario final. A Windows 95 realmente no le apetece mucho que se hurgue en sus interioridades.

183 Por supuesto que se han desarrollado para Windows una serie de lenguajes análogos al "batch" de DOS. Otros sistemas operativos (por ejemplo OS/2 y su REXX) los incluyen directamente. Pero, por una serie de motivos que no entraremos a analizar, no se usan demasiado. Al menos en Windows.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Henos aquí, una vez más, enfrentados a un proceso de selección de drivers diversos, por traducir lo anterior a una terminología que ya debería resultarnos familiar. Lo de los drivers afecta tanto a la forma en que trabaja cualquier programa, que Windows incluye el equivalente DOS de “Instalar Windows” y se recomienda usarlo en según qué ocasiones. De hecho ofrece más posibilidades que su hermano. Para ponerlo en marcha salga de Windows, colóquese en el directorio \WINDOWS (esto último es fundamental), y teclee “INSTALAR”. Complete el conjuro, como siempre en DOS, con un golpe seco en esa tecla gorda de la derecha que tanto hemos citado.

Mientras “Instalar Windows” tiene reservado, en cualquiera de sus versiones, el desagradable papel de lidiar con drivers y otros entes que es mejor permanezcan lo más alejados posible del común de los mortales, el “Panel de control” es un enchufado con un destino mucho más dulce. Aprovechando el trabajo sucio realizado por su colega menos afortunado, él se encarga de facilitar la selección de los colores a emplear cuando se visualizan las ventanas, la velocidad de respuesta del puntero a los desplazamientos del ratón, de poner o quitar tipos de letra, seleccionar el fondo del escritorio, y otro centenar de pequeños detalles que no afectan en principio al funcionamiento de Windows (qué le importa a él que el fondo de las ventanas sea gris o amarillo chillón) pero que son muy de agradecer por el usuario. O sea un trabajo nada complicado y que el cliente siempre agradece. Un chollo, vamos. No es extraño que el “Panel de control” sea tan popular.



Volviendo a su aspecto, podemos ver en la figura que la ventana principal del “Panel de control” contiene un buen número de iconos. Cada uno de ellos da acceso, a través del típico doble clic, al panel de diálogo correspondiente al elemento que pretendemos configurar. El que aparece desplegado en la figura es el del “Escritorio”, y en él puede verse que es posible regular cosas tan nimias como el ancho del borde de las ventanas Windows, o la separación entre iconos que se asume por defecto.

Nos detendremos tan sólo a comentar un par de entradas. En el apartado de “Papel tapiz” puede leerse “escalera.bmp”, a la vez que aparece un botón a su derecha con una flecha hacia abajo y un par de circulitos, uno de los cuales, el rotulado “Mosaico”, está seleccionado con un punto. Lo que quiere decir todo lo anterior es que podemos visualizar como fondo del escritorio de Windows la imagen contenida en cualquier fichero gráfico que use el formato “.BMP” (dejaremos los detalles sobre este punto hasta el próximo capítulo), que el que se muestra en la figura corresponde justamente al fichero llamado “escalera.bmp”, que como es más bien pequeño podemos hacer que cubra toda la pantalla usando una visualización en plan mosaico, y que hay otros modelos a nuestra disposición cuya lista podemos ver sin más que pulsar el botón de la flechita. A través de la entrada “Protector de pantalla” es posible indicarle a Windows que, cuando no exista actividad perceptible en el teclado o el ratón durante un tiempo determinado (lo que normalmente querrá decir que no estamos trabajando con el ordenador y no necesitamos que intente comunicarse con nosotros), tenga a bien no dejar la imagen allí para evitar que el tubo del monitor pueda dañarse¹⁸⁴.

No todas las posibilidades del “Panel de control” son banales. Al menos las entradas “Impresoras”, “386 extendido” y “Puertos”, entre otras, son fundamentales. No comentaremos lo de “Impresoras” ya que parece obvio para qué sirve, pero lo del “386 extendido” es la puerta normal de acceso a asuntos tan importantes para el rendimiento de Windows como el tipo y tamaño de la memoria virtual, y “Puertos” es básico a la hora de configurar las comunicaciones de su ordenador a través de las puertas serie. Y por favor lea la nota¹⁸⁵.

Tanto el “Panel de control” como “Instalar Windows” no hacen sino modificar automáticamente los ficheros “WIN.INI” y “SYSTEM.INI” que dijimos eran los encargados de establecer la forma en que Windows debe operar. No está nada mal que sea así, ya que si algún día nos atrevemos a entrar a modificar directamente dichos archivos mediante cualquier editor (el llamado “Bloc de notas”, en el grupo “Aplicaciones” es el que Windows usa

184 Una imagen de alto brillo, inmóvil, y que persista durante mucho tiempo, puede hacer que el monitor termine por visualizar imágenes fantasma. Este efecto puede apreciarse en bastantes cajeros automáticos y se debe al desgaste de los elementos electroluminiscentes (“fósforos”) del tubo de imagen. Como cualquier ordenador produce preferentemente imágenes estáticas, y un monitor vale sus dineros, muchos sistemas operativos controlan la actividad al otro lado de la pantalla para evitar tal riesgo. Un entorno multitarea puede hacer sin problemas este tipo de cosas, y Windows se toma la molestia de ponerlo a nuestra disposición. Para evitar que se quemase el tubo basta con desconectar la visualización, es decir poner toda la pantalla en negro. Pero también puede lograrse el mismo efecto pasando a mostrar imágenes en movimiento. Esto último es mucho más divertido, y como un GUI debe tener su toque de distracción, los “salvapantallas” permiten mostrar un acuario animado, o un caleidoscopio, o incluso tostadoras volantes. Es libre de elegir el motivo que prefiera, pero debería activar el “salvapantallas”.

185 Ya hablamos de la memoria virtual, es decir de la posibilidad de usar una parte del disco duro como si fuera RAM. No es lo mismo que la RAM de verdad, pero algo es algo. Windows la usa, y es obligatoria si monta las extensiones de 32 bits. Puede establecerse su tamaño máximo y su tipo, y es quizá sobre éste último sobre el que deberíamos decir algo. Si Vd. valora ante todo la velocidad de operación, fíjela en “Permanente” y active el “acceso de 32 bits”. Pero si puede prescindir del aumento marginal de velocidad que tal situación proporciona y valora la solidez ante posibles desastres, elija “Temporal”. No entraremos en más detalles técnicos.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

por defecto), siempre resulta mucho más fácil realizar pequeños cambios sobre lo que ellos han hecho que pretender crearlos partiendo de cero.

Como el “Panel de control” es más tranquilizador y cómodo de usar que “Instalar Windows”, existe cierta tendencia a que se le vayan añadiendo iconos. No es extraño por ejemplo que la instalación de los drivers de una nueva tarjeta de video cree el suyo para que puedan cambiarse la resolución y el número de colores de la pantalla, aunque no siempre se deje el nuevo icono en el “Panel de control”.

Un GUI debe ser cómodo, cuanto más mejor, y la configuración de Windows no acaba en el “Panel de control”. Igual que la solución más completa a los problemas de la interfase de usuario del DOS era sustituir COMMAND.COM por un intérprete de órdenes avanzado, lo mejor que puede hacerse para curar algunos de los males de la interfase de usuario de Windows es cambiar PROGMAN.EXE por algún programa alternativo que se adapte mejor a nuestras preferencias.

Justamente en la última figura, como era de esperar, puede verse uno de ellos. Pero antes de entrar a analizarlos, aunque sea someramente, debemos llegar a un acuerdo sobre el vocabulario a emplear. El hecho es que todos ellos modifican lo que hasta el momento hemos venido llamando “esquema de organización de Windows”, “metáfora propuesta para acceder a los programas”, u otro montón de expresiones pintorescas que Vd. no tendrá dificultad alguna en encontrar en las páginas anteriores si las busca. Como en realidad, aunque de vez en cuando intente disimularlo levemente, pretendo escribir algo calificable como “de alto nivel intelectual” para que la gente pueda decir “mira qué listo es ese señor”, no puedo desaprovechar la oportunidad de simplificar tan rocambolescas expresiones sustituyéndolas por un único vocablo, que debe ser necesariamente biensonante y más bien culterano, aunque nos pasemos un poco al aplicarlo. Pues bien, cuando tengamos que tratar ese tipo de cosas hablaremos del “paradigma” propuesto por el PROGMAN.EXE en cuestión, y por cierto, espero que se haya dado cuenta de que vengo usando hace rato el nombre del fichero que contiene el “Administrador de programas” de Windows por defecto como un genérico para referirme de una sola vez a un montón de programas que hacen lo mismo y que, por supuesto, se llaman de forma totalmente diferente.

Pasando a meternos en harina, reiteremos una vez más que el paradigma propuesto por Windows de acceder a los programas a través de iconos que pueden organizarse mediante grupos visualizados por ventanas, no es el único posible. De hecho presenta graves inconvenientes de manipulación. De todas las maniobras a realizar con el ratón, es quizá el “doble clic” la menos obvia y más complicada de poner en práctica. Y para acceder a un icono en un grupo hay que usarla no una, sino dos veces. Parafraseando una expresión que ya utilizamos con los directorios, podríamos decir que es “demasiado doble clic para tan poco movimiento”.

Sustituir un paradigma por otro no es fácil, aunque existan motivos para hacerlo. La nueva propuesta debe aportar ventajas innegables, debe ser fácil de comprender, y, ya que el otro estaba allí antes, debe respetarlo lo necesario como para no causar problemas, lo que puede traducirse en que lo englobe de alguna manera. En la práctica, no conviene hacerse falsas ilusiones, el sustituto proporciona más un compromiso razonable que una clara mejora en todos los aspectos citados.

Pasando a citar ejemplos alternativos, quizá el más radical y que menor esfuerzo requiere para acceder a él, ya que tan sólo exige que aprendamos algunas posibilidades adicionales del “Administrador de ficheros” de Windows, es el que podríamos denominar de “acceso a los programas desde los datos”.

La idea básica en que se fundamenta es la de asumir que a la gente le importa un bledo el programa que haya que emplear o se haya empleado en su momento para crear un documento cualquiera, sean cuentas, textos o dibujos, y podríamos extender la lista a la totalidad de las actividades posibles con el ordenador. Lo que la gente está acostumbrada a manejar, lo que “ve” o “querría ver” en su GUI, son sus informes al consejo de administración, la redacción del niño, las cuentas del banco, la conexión a Internet, o el cartel que dibujamos para intentar vender nuestro piso la última vez que nos mudamos de casa. Es perfectamente posible ocultar los programas detrás de los datos que vayan creando sin más que asociar cada tipo de fichero, en función de su formato y contenido, a un programa capaz de trabajar con él. Como las extensiones de los nombres de ficheros DOS, los que tenemos en nuestro PeCé, se emplean justamente para indicar el tipo de datos que contienen y su formato, bastará con enlazar una extensión propia de un tipo de datos con el fichero correspondiente al programa para manipularlos. ¿No establecemos una relación entre un icono determinado y un programa si usamos PROGMAN.EXE?. Pues es incluso más fácil asociar cualquier fichero con extensión .BMP, un gráfico como hemos dicho, con el programa que queremos usar para manipularlo habitualmente. Tal vez con Paintbrush, uno de los que incluye Windows y que se encarna en el fichero PBRUSH.EXE. El “Administrador de archivos” ofrece justamente una opción “Asociar” para ello. Una vez definidas las asociaciones, bastarán dos clics sobre cualquier fichero de datos para que el programa correspondiente se abra automáticamente.

En resumen, podemos usar únicamente el “Administrador de ficheros” como nuestra puerta de acceso a los programas Windows. La gran ventaja de este paradigma es que da la máxima importancia a lo que en realidad la tiene, es decir a nuestros datos, ya que el ordenador no sirve para otra cosa que para producirlos y mantenerlos. Pero tiene varios inconvenientes. En primer lugar muchas veces tenemos varios programas para manipular el mismo tipo de datos e interesa usar uno u otro en función de lo que queramos hacer en concreto, y eso dinamita la posibilidad de establecer asociaciones “una extensión-un programa”. En segundo que un enfoque lo suficientemente radical, o si se quiere lo suficientemente coherente, como para no exigir un conocimiento real de qué programas están detrás, exige bastante más que las asociaciones citadas y acabamos bordeando la necesidad de hablar de “objetos” en lugar de ficheros, un tema que está a la espera de que se desarrollen sistemas operativos que los usen como su ladrillo de construcción y que no sólo no está resuelto sino que no está claro que se vaya a resolver rápidamente. Tal vez el binomio Next-Nextstep, sea el mejor exponente por el momento de este enfoque (ver glosario). Ya dijimos que lo de los objetos lo dejábamos para otra ocasión¹⁸⁶. Otro problema es que este paradigma ignora los

186 New Wave, las extensiones propuestas por Hewlett Packard para Windows y que fueron también objeto de la demanda judicial de Apple anteriormente citada, iban en esta dirección, añadiendo “objetos” y “agentes” a un Windows que hubiera podido beneficiarse de ellos. El mismo “OLE” de Microsoft va también en el sentido de añadir un cierto toque “orientado a objetos” a Windows. New Wave puede considerarse difunto, aunque no funcionaba mal, y la importancia de OLE es por el momento más bien discutible. Los ficheros siguen reinando en los PeCés, y tal vez no sea mala cosa.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

grupos de programas que Windows establece por defecto¹⁸⁷. Y finalmente, aunque el problema es fácil de solucionar¹⁸⁸, hacen falta nombres de ficheros más largos que los que permite la norma 8.3 para poder ver un fichero que se llame “Informe a la dirección”.

En consecuencia, las asociaciones datos-programas a través del “Administrador de ficheros” acaban siendo más una posibilidad adicional de Windows que un cambio de paradigma. Por cierto, muchos programas establecen automáticamente las suyas cuando se instalan. No tiene más que ver el fichero WIN.INI para comprobarlo. Aquellos ficheros que en Windows tengan establecida una asociación mostrarán en los listados del “Administrador de ficheros” un icono no vacío, una especie de hoja con líneas, lo que nos indicará que dos clics sobre él servirán para lanzar el programa asociado.

Ya que estamos hablando de mejoras a la interfase de usuario de Windows, y hemos topado con el “Administrador de ficheros”, no podemos dejar pasar la ocasión de decir que la comodidad de Windows también aumenta mucho si se sustituye por otro mejor y precisar que existen multitud de alternativas¹⁸⁹.

Otras modificaciones al paradigma propuesto por Windows respetan su estructura base pero la visualizan de una forma diferente para proporcionar un acceso más rápido a los programas. Un excelente ejemplo de este enfoque es “Dashboard”, justo el responsable de la cosa rara esa que aparece en la última figura ocupando su parte inferior derecha. La traducción de “Dashboard” es “Cuadro de mandos”, y pretende ser una especie de tablero de un coche que agrupa todo lo necesario para controlar los aspectos más esenciales de Windows. Los grupos de programas se visualizan como entradas de un menú principal que en la figura está dispuesto en horizontal, justo encima del cúmulo de dibujitos. Verá por ejemplo que su cuarta entrada empezando por la izquierda es “Principal”, y eso debería permitirle entender la idea perfectamente. Pues bien, un solo clic, y se despliega un menú en vertical mostrando como entradas de menú adicionales aquellos programas incluidos en el grupo que sea. Otro clic, uno solo, y el programa se ejecuta. Se gana notablemente en rapidez de acceso. la ganancia es aún más apreciable si se tiene en cuenta que Dashboard se pretende un tablero completo, e incluye entre otras cosas una serie de botones (en la figura los siete a la izquierda, dispuestos en dos filas) que permiten acceder directamente a los programas usados con mayor frecuencia. El primero, por ejemplo, lanza el “Administrador de ficheros”. Dashboard es un desarrollo original de Hewlett-Packard, que lo regalaba con sus ordena-

187 Bueno, sería posible asociar los ficheros que contienen los grupos (extensión .GRP) a PROGMAN.EXE, pero el enfoque carece de coherencia ya que no hace sino sacar a la palestra unos programas que se pretende ocultar.

188 Por ejemplo 4DOS, una de las alternativas disponibles a COMMAND.COM, añade un cierto soporte a nombres largos de ficheros incluso en DOS (hay que hablar más de “descripciones” de ficheros que de nombres largos) mediante el sencillo expediente de crear en cada directorio un fichero oculto llamado DESCRIPT.ION donde se guardan hasta 40 caracteres adicionales dedicados a explicar qué es lo que contiene cada uno de los ficheros del directorio.

189 Mi favorito siempre ha sido el “Administrador de ficheros”, en jerga “File manager”, de las “PC Tools” para Windows, un conjunto de herramientas de lo más popular. Que, por cierto, también incluyen un sustituto del “Administrador de programas” con bastantes de las posibilidades que luego han aparecido en Windows 95.

dores, y ha tenido mucha más influencia que su otro desarrollo anterior en la misma línea, el denominado New Wave. El hecho es que, en mayor o menor medida, este tipo de diseño general ha aparecido en todo GUI posterior a 1.994, y pueden encontrarse análogos en OS/2, Windows 95 y hasta X-Windows¹⁹⁰.

Otras modificaciones al paradigma propuesto por Windows respetan su estructura base pero la visualizan de una forma diferente para proporcionar un acceso más rápido a los programas. Un excelente ejemplo de este enfoque es “Dashboard”, justo el responsable de la cosa rara esa que aparece en la última figura ocupando su parte inferior derecha. La traducción de “Dashboard” es “Cuadro de mandos”, y pretende ser una especie de tablero de un coche que agrupa todo lo necesario para controlar los aspectos más esenciales de Windows. Los grupos de programas se visualizan como entradas de un menú principal que en la figura está dispuesto en horizontal, justo encima del cúmulo de dibujitos. Verá por ejemplo que su cuarta entrada empezando por la izquierda es “Principal”, y eso debería permitirle entender la idea perfectamente. Pues bien, un solo clic, y se despliega un menú en vertical mostrando como entradas de menú adicionales aquellos programas incluidos en el grupo que sea. Otro clic, uno solo, y el programa se ejecuta. Se gana notablemente en rapidez de acceso. la ganancia es aún más apreciable si se tiene en cuenta que Dashboard se pretende un tablero completo, e incluye entre otras cosas una serie de botones (en la figura los siete a la izquierda, dispuestos en dos filas) que permiten acceder directamente a los programas usados con mayor frecuencia. El primero, por ejemplo, lanza el “Administrador de ficheros”. Dashboard es un desarrollo original de Hewlett-Packard, que lo regalaba con sus ordenadores, y ha tenido mucha más influencia que su otro desarrollo anterior en la misma línea, el denominado New Wave. El hecho es que, en mayor o menor medida, este tipo de diseño general ha aparecido en todo GUI posterior a 1.994, y pueden encontrarse análogos en OS/2, Windows 95 y hasta X-Windows¹⁹¹.

Igual que podemos visualizar los programas como entradas en un menú, podemos también dar un mayor énfasis a los botones, o quizá incluso proponer el acceso a Windows a través de nuestro viejo paradigma de la línea de órdenes. No es nada descabellado. Por ejemplo “TakeCommand”, un programa del fabricante de 4DOS, usa éste último enfoque.

En resumen, si PROGMAN.EXE y el “Administrador de archivos” no le gustan, y debo reconocer que existen motivos racionales para ello, cámbielos por otros. Windows se hará de golpe mucho más cómodo.

¿Qué pasa si tengo instalado por ejemplo Dashboard y quiero, por el motivo que sea, acceder puntualmente al antiguo “Administrador de programas” de Windows?. La solución a tan tremendo problema es tan sencilla como abrir el “Administrador de archivos” y dar dos clics sobre PROGMAN.EXE. Por cierto, algunos de los administradores de programas alternativos permiten ser usados como meros complementos o como auténticos sustitutos de

190 Algunos indicios me hacen creer que Dashboard para Windows no es sino la adaptación al GUI de los PeCés de un producto análogo de HP para el HP-UX (versión Unix de HP) de sus estaciones de trabajo. Su origen por tanto estaría en Unix y X-Windows.

191 Algunos indicios me hacen creer que Dashboard para Windows no es sino la adaptación al GUI de los PeCés de un producto análogo de HP para el HP-UX (versión Unix de HP) de sus estaciones de trabajo. Su origen por tanto estaría en Unix y X-Windows.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

PROGMAN.EXE, siendo obviamente en este último modo de funcionamiento donde más posibilidades ofrecen.

La necesidad de rediseñar profundamente el interfase de usuario de Windows, cambiando el paradigma, ha llevado a que Windows 95 incorpore ya otro, bastante mejorado. Se articula alrededor de una “Barra de tareas” o “Taskbar” y, como se ha dicho, tiene ciertas reminiscencias de “Dashboard” y hasta algún aire de Macintosh o de OS/2 (entre otros los llamados “accesos rápidos”). De todos modos, incluso en Windows 95, los antiguos grupos de programas siguen estando allí, y existe una vía de acceso a ellos.

Como tanto Windows 95 como otros muchos programas en Windows 3.1 y sus sucesores se benefician de algunos recursos adicionales que todavía no hemos citado, pasemos a presentarlos. Ha llegado la hora de conocer al botón derecho del ratón, a los “menús objeto”, y a las etiquetas automáticas, tres inventos de suma utilidad y cada vez más generalizados en los programas para Windows.

En algún momento a lo largo de los últimos años, un nuevo benefactor de la fracción del genero humano que usa ordenadores debió hacer para sí la siguiente reflexión, cuyo tono debería resultar familiar: “Un ratón de Pecé tiene al menos dos botones. De hecho algunos tienen hasta tres, pero el estándar Microsoft o los PS/2 sólo incorporan dos, por lo que mejor ignoramos el tercero. En Windows sólo se emplea el izquierdo. Tal vez pudiera usarse el botón derecho para algo que mereciera la pena”. Acababa de nacer el botón derecho del ratón. Qué hacer exactamente con él debió ser objeto de profundas reflexiones. Quizá conviniera asignarle una función similar a la de la tecla “Escape”. O tal vez hacer que diera un acceso directo al “Administrador de programas”. O quizá... . Durante cierto tiempo no estuvo nada claro qué hacer con él. Pero de nuevo en algún momento, dio con una posibilidad que parecía interesante. Se dijo: “Muchas veces en Windows sabemos que podemos hacer algo con aquello que tenemos seleccionado, pero no sabemos en qué menú está la opción correspondiente. Podríamos hacer que el botón derecho nos permitiera acceder a un menú que contuviera la mayor parte de las acciones más corrientes con ese tipo de objeto. Si tenemos seleccionado un fichero podría desplegarse a su lado un menú para renombrarlo, copiarlo o borrarlo, por ejemplo”. Habían nacido los “menús del objeto”, que no están vinculados a ninguna barra de menús principal sino a cada elemento seleccionable en Windows, y que suelen aparecer justamente al darle al botón derecho del ratón mientras colocamos sobre ellos el puntero¹⁹². Por cierto es a través del botón derecho como se llega en Windows 95 a los grupos de programas.

Todavía se podía mejorar otro punto particularmente desagradable. Siguió su reflexión con un tema que hemos reiterado repetidamente: “Lo de los botones es muy decorativo y permite un acceso rápido a programas, por lo que deberían conservarse, pero el problema es que representan la información de forma nada obvia. Un texto es feo pero su significado es

192 Creo que los “menús objeto” aparecieron en Windows con una hoja de cálculo denominada “Quattro Pro”, que lo incorporó en su versión 1.0. El editor de tal producto es una casa denominada Borland, que muestra en su línea de productos una notable “orientación al objeto”, por lo que la cosa no resultó demasiado sorprendente. Pero sobre programas y sus editores hablaremos en el próximo capítulo. Lo que sí debemos decir ahora es que el invento se popularizó con notable celeridad.

descifrable de inmediato. De todos modos, una vez sabemos qué significa un icono, tal vez porque lo leamos en alguna parte, se le puede identificar sin duda en lo sucesivo. Podríamos añadir una especie de etiquetas que se desplegaran automáticamente cuando el puntero estuviera sobre ellos más tiempo del estrictamente necesario. El usuario los leería, se quedaría con su significado, y los podría buscar directamente a partir de entonces. Para llamar su atención podríamos ponerlas en amarillo, como los ‘Post-It’, y visualizarlos al lado del botón correspondiente. Podrían ser una especie de etiquetas de la interfase Hewlett-Packard en versión Windows y asociadas a los botones”. Dicho y hecho. Todo lo anterior ha ido apareciendo por la interfase Windows cada vez con mayor frecuencia, contribuyendo a hacerla aún más confortable.

Quizá el último invento aparecido sean los menús de despliegue automático. No renuncio a presentarlo porque es propio de Windows 95 y hace falta poner el contenido del texto al día, y porque son un maravilloso ejemplo de que es mejor contemplar las cosas con cierto sentido del humor. ¿Se acuerda de lo que dijimos, tiempo ha, sobre GEM?. En aquel primigenio GUI para PeCés, bastaba con colocar el puntero sobre un menú para que éste se desplegara, sin tener que hacer clic. Los menús ofrecen así un acceso más rápido a sus opciones y ganan en dinamismo, con lo que resultan más divertidos todavía. Desgraciadamente, la idea no prosperó. Pues bien, los cambios en la interfase de usuario incorporados a Windows 95 han puesto un énfasis mucho mayor en los menús, y para mejorar su rapidez y darles un aspecto más seductor, se ha vuelto a hacer que se comporten como lo hacían en GEM. Se abren con sólo poner el cursor encima. Ya se dice que no hay cosa peor en esta vida que tener razón antes de hora. Realmente es como deberían haber funcionado siempre.

Y bien, esto es todo. A partir de aquí, no volveremos a decir nada sobre el uso y abuso del software básico del PeCé, es decir sobre DOS y Windows, aunque antes de terminar, no estaría de más, hacer algunas recomendaciones sobre cómo configurar éste último para que resulte práctico y agradable.

Ya dijimos que no se pasara con las fuentes y que sustituyera lo antes posible el “Administrador de programas” y el “Administrador de archivos” por otros más capaces. También que, si su monitor se lo permite, use una resolución elevada asociada a una frecuencia de refresco alta y un número de colores suficiente. 1.024 x 768 puntos en 256 colores con un refresco de 70 Hz no entrelazado es un magnífico modo de pantalla si Vd. va a pasar varias horas al día ante su ordenador en Windows. Si no tiene dinero para un monitor que llegue hasta ahí (si va a comprar un ordenador nuevo es mejor que se asegure de que sí que llega pues uno de al menos 15 pulgadas capaz de eso y de más ya no es tan caro), deberá conformarse con una resolución de 800x600, 256 colores, y por supuesto un modo no entrelazado a la máxima frecuencia de refresco posible. No creo que se pueda trabajar cómodamente en Windows o en cualquier otro GUI por debajo de eso. Algunos sustitutos del administrador de programas, como por ejemplo Dashboard, proporcionan artificios para aumentar la superficie de trabajo, lo que nunca viene mal.

En cuanto a los colores con los que debe pintar Windows, se ruega que tenga en cuenta que el aspecto debe ser agradable, pero es una consideración secundaria frente a la ergonomía. Evite combinaciones chillonas, no porque sean de mal gusto, sino porque fatigan muchísimo la vista. El gris claro es un buen fondo sobre el que disponer, empleando matices intensos pero no agresivos, los otros colores que deben diferenciar los distintos elementos. Windows 95 ha mejorado muchísimo en el aspecto ergonómico, y las combinaciones de color que nos ofrece siguen este tipo de pautas.

CAPÍTULO 7. PeCés: MS-DOS/Windows.

Tampoco conviene renunciar a darle un toque personal a nuestro Windows. Al fin y al cabo tenemos un ordenador para nosotros solos. Un salvapantallas a su gusto puede hacer más relajante una pausa para pensar, o un fondo de pantalla agradable contribuir a disminuir su fatiga. De lo que se trata es de conseguir que cinco horas seguidas ante la pantalla de nuestro PeCé no sean el equivalente a una tortura insoportable que nos deje los ojos y el cerebro al borde de la desintegración, sino un rato más bien largo de trabajo productivo en el que disfrutemos de la mayor comodidad y nos enfrentemos al menor número de interferencias posible .

Para que se haga una idea de lo que puede querer decir lo anterior, la siguiente figura muestra, más o menos, el aspecto del Windows que vengo usando hace ya cosa de un año. El fondo de pantalla se reveló para mí como un elemento particularmente trascendental, y fue objeto de numerosas probatinas hasta que llegué a uno que me convenciera plenamente. Por cierto, antes solía preferir el “escalera.bmp” que aparece en las otras figuras y que también está bastante bien. Y por si también le interesa y como eso no se ve en la figura, aclararé que empleo un salvapantallas más bien anodino, en concreto el denominado “Marquesina”, que no pasa de ser un simple mensaje deslizante.



Si quiere terminar la configuración haciendo que al entrar en Windows se pongan en marcha automáticamente algunos programas, lo tiene fácil. Basta dejar una copia de los iconos correspondientes en el grupo “Inicio”.

Sólo nos queda resumir las ventajas e inconvenientes de DOS y Windows para terminar dignamente. Debería bastar con puntualizar que cómodo, bonito, y útil no son sinónimos de obvio y sencillo. Es posible que se le puedan aplicar a Windows los tres primeros calificativos, pero el coste a pagar por él incluye al menos el olvidarse de los dos últimos. Un GUI implica una complejidad adicional más que importante, y eso ocasiona numerosos problemas, como veremos más adelante. Mi consejo es que intente aprovechar todo lo que

el PeCé le da actualmente a nivel de sistema operativo y no ignore a DOS, que sigue ahí. Al menos el 75% de la capacidad de saber manejar un ordenador, por muchos GUIs que tengamos, depende de saber orientarse en su sistema de ficheros.

El coste en realidad es mayor que una mera renuncia a dos calificativos. Es preciso insistir en la exigencia de un hardware potente impuesta por Windows, que a cambio no llega nunca a proporcionar, con ordenadores más costosos, una velocidad o solidez análogas a las que DOS proporciona con total discreción. Puede argumentarse que Windows debería tener una estructura diferente, ya que está a medio camino entre un GUI auténtico y un sistema operativo e intenta hacer lo mejor que puede tareas que en realidad no le corresponden¹⁹³. Por ejemplo la multitarea, por citar algo, que debería estar incorporada en DOS.

En mi modesta opinión, el precio más oneroso a pagar es el importante aumento de complejidad derivado del uso de Windows, pues va en detrimento de la solidez del sistema operativo del PeCé. Uno tiende más a maravillarse de que Windows se cuelgue poco (cuando está bien configurado su solidez es bastante aceptable), que a preocuparse y buscar la solución cuando de hecho se cuelga. Al fin y al cabo, nunca debería ocurrir, e indica un defecto en alguna parte.

La imagen elegida como fondo de mi escritorio es una buena metáfora de lo que en mi opinión significa trabajar con Windows o en general, porqué no decirlo, con cualquier sistema informático. Muestra una foto de 1.932 titulada “Lunchtime atop a skyscraper”, lo que en castellano vendría a ser “Hora de almorzar en la cima de un rascacielos”. En ella aparece un grupo de obreros de la construcción disfrutando de lo que adivinamos es su pausa para almorzar. Sus actitudes son normales, mucho más propias del animado coloquio que parecen mantener que de una situación que, objetivamente considerada, debería provocar una cierta inquietud en su ánimo. Supongo que no hacen falta más comentarios en cuanto a su significado.

Por cierto, al ser colores predominantemente grises (blanco y negro) no resulta en absoluto molesta a la vista. Los aspectos decorativos no deben perjudicar la ergonomía. Tampoco deben perjudicar a la velocidad de trabajo, y probablemente la perjudican en este caso. Pero hay otras formas de hacer que Windows corra más, y también las uso, y estoy dispuesto a renunciar a una parte de las mejoras logradas a cambio de ver asiduamente una foto tan magnífica. En resumen, usted sabrá querido LAO. Usted debe ser el que valore los pros y los contras. Si pretende que se lo den todo mascado con los ordenadores, no llegará muy lejos.

A modo de resumen global, nos hemos ocupado en este capítulo de ver el sistema operativo completo de los PeCés, es decir DOS y Windows. En conjunto es algo bastante útil a nivel personal, suficientemente extensible (véase si no su colección de drivers y la posibilidad del DOS de llegar a Windows) y más que razonablemente sencillo de aprender. ¿Quiere esto decir que la situación vista desde el lado del software base del PeCé sea perfecta o pueda darse por definitiva?. Nada más lejos de la realidad. No es demasiado difícil encontrar los límites de Windows 3.0, 3.1, 3.11 o Windows 95. Muchas de las promesas iniciales de los

193 Porqué Microsoft no propuso un nuevo DOS con su GUI es un interesante tema teórico. Quizá sólo criterios de mercado justifiquen el diseño actual. Será mejor que lo dejemos estar...

CAPÍTULO 7. PeCés: MS-DOS/Windows.

GUI en su versión PeCé no se han cumplido. Un caso particularmente flagrante es aquello de que podían contribuir a hacer los programas más pequeños. Los programas Windows son muchas veces descomunales, e incluso el famoso tema del intercambio de datos sigue siendo una asignatura pendiente, como demuestra el hecho de que su más potente herramienta, el llamado OLE, vaya por su segunda versión y aún no termine de convencer, por lo que muchas veces se prefiere seguir recurriendo a los ficheros. Estos al menos no requieren ingentes cantidades de memoria para funcionar (olvídense de OLE con menos de 16 Megabytes de RAM) y son predecibles en su comportamiento.

Hay un cierto deseo latente (o explícito, lo mismo da por ahora) de algo manifiestamente mejor, un multitarea de verdad que maneje sin problemas cantidades enormes de memoria y de almacenamiento en disco, sea más rápido y potente, y ofrezca mayores garantías de permanencia cara a un futuro incluso lejano.

Eso sí, también hace falta de nuevo que mantenga la máxima compatibilidad con lo existente, ya que entre otras cosas es fácil que los programas para Windows sean los más comprensivos para con el usuario medio que se hayan diseñado jamás (si exceptuamos, tal vez, los de unos pocos GUIs de la competencia). Hasta viene bien no tener que renunciar a los programas para MS-DOS. Pero en todo usuario avanzado aparece cierto tono verde de envidia cuando oye cosas como “Unix”, o “Windows NT” tal vez, o quizá “VMS”. El hecho es que el desarrollo de nuevos sistemas operativos mejorados para PeCés continúa. Es un campo abierto en el que por el momento no hay un candidato claro a tomar el relevo. El movimiento es incesante y los cambios en un futuro más o menos próximo, bastante probables. Volveremos sobre ello, como siempre decimos. Y si cree que estoy desaprovechando una magnífica oportunidad para ponerme a hablar de Windows 95, no se preocupe. También volveremos sobre él un poco más adelante, aunque quizá no en la forma que Vd. espera.

El capítulo ha sido enorme, lo reconozco. Tal vez no me crea, pero he intentado que fuera lo más corto posible, aunque manteniendo eso sí el tono general hasta ahora empleado en cuanto a explicación de los conceptos que creo importantes. Tómese un descanso. Si ha leído atentamente hasta aquí, se lo ha ganado a pulso varias veces. Cuando se haya repuesto, espero que vuelva. En el capítulo 8 nos ocuparemos de un tema inevitable y que constituye la auténtica fuerza de los PeCés: sus programas de aplicación. Tan inevitable que de hecho ya ha ido saliendo, pues nos hemos visto en la necesidad de nombrarlo, incluso poniendo ejemplos concretos, para poder seguir adelante en muchos momentos.

Si necesita que se le anime un poco más, sepa que, palabrita del niño Jesús, lo peor ha pasado. Lo que nos queda es mucho más divertido, si bien no menos necesario de conocer.