

## CAPÍTULO 8

*PeCés: Programas*Un secreto a voces¿Porqué LR-PeCé liga tanto?

<b>Distribución a librerías</b> Mira Editores Concepción Arenal, 22 50005 Zaragoza Tel: 976 354165 Fax: 976 351043 e-mail: lcentral@clv.es	<b>Distribución a particulares</b> Luis Membrado Giner Andador Anayet, 4, 6º C 50015 Zaragoza e-mail: lmg00009@telefonos.es lmg00009@telefonos.es
---	--

Antes de nada, antes incluso de comenzar el capítulo mínimamente en serio, es preciso hacer una puntualización. Seguro, querido LAO, que se ha dado cuenta de que conforme avanzamos en el texto los aspectos técnicos van cediendo poco a poco su lugar a opiniones y juicios de valor. Hablamos cada vez más de “no parece conveniente la forma en que se hacen algunas cosas en Windows” y cada vez menos de “para copiar un fichero en DOS teclee COPY y pulse RC”. No hay nada raro en ello. Es lo habitual al ir avanzando en el conocimiento de cualquier materia. Pero es preciso subrayarlo porque a partir de este momento los temas técnicos van a ocupar un clarísimo segundo plano. Como no creo posible, ni conveniente, intentar abstraerme por completo de los aspectos subjetivos asociados a todo juicio de valor, debe tener en cuenta que los criterios con que se juzgarán las cosas son los míos, y que los suyos pueden ser completamente diferentes.

Intentaré, eso sí, explicar los motivos en que me baso, pero no será tanto para justificarme (es difícil hacer cambiar de opinión a un aragonés convencido, o al menos eso dicen) como para que Vd. pueda reelaborar los juicios a su gusto y conveniencia. Al igual que decíamos que sabrá perfectamente cuando domina DOS en el momento en que se atreva a modificar los ficheros de configuración con buenos resultados, podemos decir que notará que se está comenzando a aclarar en esto de la informática en general cuando sea capaz de llegar a sus propias conclusiones. Aísle los datos, elabore sus criterios, y rehaga los juicios a su gusto. Lo que a mí puede parecerme un problema, para Vd. puede ser una virtud.

Se trata de aprender a pensar y valorar el fenómeno informático y sus implicaciones para nuestro caso particular, y no de recitar como un loro un montón de datos técnicos inconexos o, aún peor, de asumir como propios y hasta defender con furia juicios de valor totalmente inapropiados para nuestra situación, necesidades, y gustos personales. Aprenda a pensar, le garantizo que es un ejercicio bastante placentero en términos generales y, en informática, hasta productivo en el aspecto económico. Y ya vale, pasemos a ver cosas sobre los programas de aplicación del PeCé, que es de lo que debemos ocuparnos en este capítulo.

No hace tanto, al fin y al cabo sólo han pasado unos siete u ocho años, aún se planteaba la cuestión de si PeCé sí o Pecé no cuando uno se iba a comprar un nuevo ordenador. La variedad de oferta era mucho mayor entonces, y equipos como el Atari ST, el Commodore Amiga, y el Apple Macintosh, merecían al menos entrar en la selección final. Los PeCés todavía no habían llegado a su mejor momento, y considerando sólo el sistema operativo y el hardware de cada uno, difícilmente el PeCé figuraba en primera posición en el momento de decidir. Pongamos como ejemplo, sin ir más lejos y exponiendo sólo lo más elemental<sup>1</sup>, las cuentas que yo me hacía en aquellos tiempos.

1 En el glosario puede encontrar algún detalle adicional sobre estos equipos.

© 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, compire 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 Pts (30'05 euros)

## ***CAPÍTULO 8. PeCés: Programas***

Había que comenzar reconociendo que el Apple Macintosh podía ser muy agradable, pero era un equipo notablemente cerrado, sin apenas documentación técnica disponible, y que resultaba muy caro si ya no eras un estudiante que pudiera beneficiarse de los descuentos para educación propios de la universidad. Y yo ya no lo era, por lo que mejor olvidarse del Mac.

El Atari ST no estaba mal. Era una especie de Macintosh popular, con un montón de interfases. Pero a pesar de incluir hasta una SCSI, parecía faltarle un cierto toque de seriedad. No se acababa de vender del todo bien y quizá no sobreviviera demasiado tiempo. Además se esperaba una versión perfeccionada que debía incluir un coprocesador gráfico (el Atari Mega ST, y luego el Atari TT) por lo que quizá conviniera esperar a ver si terminaba de cuajar o mejoraba un poco.

El Commodore Amiga tenía sobre el papel unas características espectaculares. Tres coprocesadores lo convertían en un as de la pantalla y el sonido, y su sistema operativo multitarea lo era de verdad, lo que siempre resultaba interesante. Era más caro que el Atari ST, sin embargo, y también era posible que le faltara un hervor. No era tampoco obvio que fuera a durar demasiado.

Todos los equipos anteriores equipaban CPUs de la familia Motorola 68000 y uno u otro GUI. Ambos puntos había que contabilizarlos como claramente a su favor. Frente a ellos estaba el PeCé, que se estaba vendiendo francamente bien y era el estándar más claro desde la aparición de la microinformática, pero que no contaba sino con un Intel 80286 y un sistema operativo más bien espantoso. Los equipos con un 80386, bastante mejores, aún eran caros, y los GUIs para PeCé no terminaban de cuajar. Comprarlos sólo por la estandarización que suponía y sus posibilidades de supervivencia a medio y largo plazo era algo que no nos acababa de entrar en la cabeza a muchos. Aún así... había que reconocer que tal vez era la mejor opción de compra.

¿Porqué?. La razón fundamental es que, como se vio en los primeros capítulos, un ordenador no sirve estrictamente para nada sin su software de aplicación. Hacen falta programas que permitan hacer cuentas, procesar textos, crear gráficos o llevar una facturación. El hardware y el software base (sistema operativo y GUI) no hacen sino poner un ordenador en blanco en condiciones de hacer algo encima de una mesa, pero, sin nada más, la utilidad del total sigue estando muy cerca de cero. El PeCé tuvo desde muy poco después de su aparición una biblioteca de software que hacía palidecer de envidia a cualquiera. Era sin duda alguna el equipo potencialmente más útil, lo que compensaba con creces la mayor parte de sus fallos.

Frente a algún procesador de textos para el Atari ST o el Amiga, no sencillos de conseguir además, el PeCé contaba con varios, todos ellos con fama de sólidos y potentes y diferentes orientaciones. Uno permitía ante todo ocuparse de crear documentos de tipo técnico, el otro era el más compatible con todo lo anterior por el momento, el de más allá estaba especialmente pensado para desenvolverse bien en diferentes idiomas. No había color. Para sacarle partido a un ordenador, la inversión más sólida era el PeCé. Por su colección de programas de aplicación, ya que había de todo y bueno, aunque fuera algo más difícil de usar que los demás.

Es tan cierto lo anterior que la difusión final del Macintosh, del Atari ST y del Amiga, quedó reducida a una serie de nichos en los que su software, basado en características propias que sólo ellos poseían, no tenía competencia. El Macintosh era el más cómodo de usar por gente

## ***Tipos de programas***

que no quería saber nada de informática y era tal vez el más dotado para aplicaciones gráficas exigentes. Se quedó como un equipo para estudiantes, profesionales con alto nivel de ingresos, y gente dedicada a las artes gráficas. El Atari ST era el único que incorporaba un interfase MIDI (ver glosario) por lo que acabó siendo el ordenador de aquellos que querían hacer música computerizada. El Amiga ofrecía alta velocidad de proceso de imágenes animadas y posibilidad de sincronizar las generadas por el ordenador con las procedentes de otras fuentes, por ejemplo de la televisión, y acabó siendo el equipo típico de los que querían trabajar en efectos especiales para vídeo.

No es posible, en consecuencia, hablar del PeCé sin ocuparnos en buena medida de lo que le daba y aún le da mucho de su atractivo. Y, hablando en términos generales, es imposible introducirse con fundamento en la informática sin considerar aquello que justifica la existencia de los ordenadores, es decir el software de aplicación. Nos ocuparemos de ello en las siguientes páginas

### **Tipos de programas**

### **De compras para LR-PeCé**

Para no hacer el pardillo a la hora de elegir software de aplicación para nuestro PeCé necesitamos una serie de conocimientos bastante variados. Hay que saber qué fabricantes existen, qué productos tienen en su catálogo, sus precios, estar al tanto no sólo de lo que han hecho hasta el momento sino de lo que se supone que van a hacer en un futuro próximo, conocer las distintas formas de distribución de software que existen, tener una idea clara de lo que queremos hacer con nuestro PeCé y de lo que debemos exigirle a un programa a la hora de estudiar si es adecuado... y vamos a dejarlo aquí porque no merece la pena continuar. Responder lo mejor posible a la pregunta aparentemente banal de “¿cual es el programa que más me conviene para hacer lo que yo quiero hacer con mi PeCé y dónde y cómo lo puedo conseguir?” exige en realidad una familiaridad nada banal con bastantes aspectos del negocio informático.

Por supuesto, es Vd. muy libre de no complicarse en absoluto la vida. Desembarcar en unos grandes almacenes y comprar lo que allí le quieran vender es una receta que puede funcionar en bastantes casos. Pero tal vez algunos lectores no tengan el dinero necesario para ponerla en práctica y quieran saber si hay otras alternativas. Y al fin y al cabo Vd. está leyendo un libro para saber más, no para que le den las recetas más rápidas que le permitan poner su ordenador en marcha sea bien o mal. Y normalmente, a base de recetas de ese tipo, será más mal que bien.

No es difícil orientarse, siquiera mínimamente, en el negocio del software para PeCé. E ir de compras en términos generales suele resultar divertido. Tras el maratón del capítulo anterior, quizá nos convenga de verdad un poquito de descanso. Imagínese que quedamos una tarde para irnos a ver escaparates y mientras tanto nos dedicamos a charlar sobre qué tienda es la que mejor producto tiene, dónde se puede conseguir eso mismo pero en mejores condiciones de compra o incluso a menor precio, y entre escaparate y escaparate, y parada para tomar una cerveza y paseo, nos vamos poniendo al día sobre como está el mercado del software para PeCés.

Bueno, hemos quedado en algún lugar popular, nos hemos encontrado a la hora prevista y, con toda la tarde por delante, nos encaminamos a nuestro primer destino que bien puede ser un hipotético gran almacén donde pasaremos revista a los productos que allí encuentre-

## ***CAPÍTULO 8. PeCés: Programas***

mos. Para que la visita resulte provechosa y no nos hagamos un lío fenomenal al llegar, tal vez lo primero que haya que contarle sea qué tipos de programas existen para el PeCé.

Podríamos solventar el expediente con un lapidario “Para el PeCé se pueden encontrar programas de todas las clases”, pero eso nos dejaría con un largo camino por recorrer hasta los grandes almacenes sin tener un tema de conversación a mano, lo que resultaría bastante aburrido. Mejor será que demos algún detalle.

Si agrupamos los programas existentes para el PeCé (o en general para cualquier ordenador) según su finalidad, podemos distinguir varios grandes grupos. Tal vez los más populares y obvios son los que se dedican a juegos, al procesado de textos, a la manipulación de datos numéricos, a la gestión de bases de datos y las aplicaciones gráficas. Pero por supuesto hay más. Por ejemplo en la actualidad están muy de moda las aplicaciones relacionadas con la comunicación entre ordenadores. En particular ahora mismo lo de conectarse a Internet es una fiebre que evidentemente necesita programas especiales para poderla apaciguar (y sobre la que, como solemos decir, volveremos en su momento). Y siempre ha sido necesario complementar el sistema operativo de nuestro ordenador con algunos pequeños, o no tan pequeños, programitas para hacer que funcione todavía mejor, es decir con los llamados “paquetes de herramientas”. O quizá nos apetezca hacer nuestras propias aplicaciones, o hasta no queramos saber nada de MS-DOS y Windows y busquemos ardientemente un sistema operativo diferente. Para lograr todo eso, y muchas cosas más, existen programas dentro del mundo del PeCé.

Como puede verse, la lista, aun muy parcial, es larga. Y si queremos ir preparando nuestra visita a los grandes almacenes, quizá debamos comenzar por decir que la relación es tan amplia que difícilmente encontraremos toda la gama de productos posibles en un único tipo de comercio. Los canales de distribución del software para PeCé dependen claramente de su grado de especialización. Encontraremos los tipos más habituales en cualquier sitio. Un procesador de textos, un programa para juegos, una hoja de cálculo, es lo que vamos a ver en los grandes almacenes. Pero no es fácil que tengan otros sistemas operativos, o lenguajes de programación. Para poder comprar esas “cosas raras” habrá que ir a comercios especializados, o quizá incluso recurrir a la venta a través de catálogo. Quizá hasta haya que conseguir la dirección del fabricante, tal vez extranjero, y establecer contacto con él para comprarlo directamente por correo.

Y la separación es normalmente todavía mayor cuando agrupamos los programas según tengan una vocación de uso general o vayan dirigidos a colectivos muy particulares. Un procesador de textos o una base de datos (aunque todavía no sepamos qué son con exactitud), pueden ser de utilidad para casi todo el mundo. Son programas “abiertos”, muy adaptables a necesidades concretas diferentes. Pero un programa creado para informatizar empresas de transportes, o comercios de ultramarinos, o ayudar en la gestión de la consulta de un dentista, es algo notablemente más cerrado, que sólo tiene interés para un pequeño grupo de gente. Se habla en ocasiones, y no son términos demasiado populares ni que me gusten mucho aunque vamos a emplearlos durante un rato, de software “horizontal” y “vertical”. Es por supuesto el “software horizontal”, el de uso general y que vale un poco para todo, el que más universalmente se distribuye y más fácilmente se encuentra. El “vertical” suele exigir que se le busque en empresas especializadas en servicios informáticos, que son las que los desarrollan, los venden o los alquilan quizá incluso en un paquete con ordenador y todo, y los mantienen, es decir se encargan de adaptarlos a los cambios en el



## ***Tipos de programas***

gremio objeto de sus desvelos y de responder a las quejas o solucionar los problemas de sus clientes. Por supuesto cobrando.

Porque otro punto en el que los ordenadores personales han supuesto una cierta revolución es en el de la relación del cliente con su software. Al principio de los tiempos informáticos, cuando el ordenador era una máquina casi inaccesible, no tenía sentido comprar programas. Los ordenadores sólo los sabían manejar unos pocos, e incluso el sistema operativo era algo que se hacía a medida. Lo que se compraban eran “servicios informáticos”. Yo le pagaba tanto a fulanito de tal, normalmente una empresa, y su personal se encargaba de montarme todo el sistema, enseñar a mis empleados a usar el programa para hacer lo que yo quería hacer con él y punto final. Una cantidad adicional al año me daba derecho a que, cuando surgiera algún problema, un empleado de la compañía viniera a solucionarlo. Y hasta a que, cuando fuera necesario, el programa se modificara para seguir cumpliendo su cometido. Si por ejemplo se trataba de un sistema para realizar la facturación de mi negocio había que ir adaptándolo a los cambios en la legislación que fueran produciéndose, tales como un nuevo tipo del IVA por ejemplo. En el límite, la empresa que proporcionaba los servicios informáticos ni siquiera vendía el programa. Lo que yo podía comprar era una licencia de uso y un contrato de soporte y mantenimiento. Mi dinero me daba sólo el derecho a usar un programa y la atención necesaria para hacerlo con provecho. El programa, por decirlo pronto, estaba en alquiler. Cada año era necesario pagar de nuevo por su uso.

El enfoque anterior es aún bastante frecuente, con más o menos matices, en el mundo de lo que hemos llamado software vertical. Y tiene bastante sentido que así sea. Pero en el mundo de los ordenadores personales, que es el que debemos tratar preferentemente, no sirve. El usuario de un ordenador personal comienza por comprar un equipo a precio bastante moderado, y no está dispuesto a pagar cantidades importantes cada año para que se le deje usar y se le mantenga un único (o varios) programas. Lo que él pretende es emplear el ordenador para cualquier cosa que se le ocurra gastando la menor cantidad de dinero posible en el empeño. Y para conseguirlo está dispuesto a ser él el que se encargue de instalar, aprender a manejar y mantener el software necesario para llevar a efecto sus deseos. Tal vez Vd. no se lo haya planteado nunca de esta forma, pero esa es la realidad. En este nuevo escenario, los productores de software pasan a vender copias de sus programas a todo aquel que desee usarlas, y a ofrecer un servicio de soporte bastante limitado que consiste la mayor parte de las veces en unos manuales y textos de ayuda incluidos en el precio y quizá la posibilidad de realizar un pequeño número de consultas telefónicas. Como las necesidades de los posibles clientes son bastante variables, es lógico que los programas para PcCs tengan un diseño más abierto, sean más generalistas en su campo de acción, de modo que cada usuario los pueda adaptar a la tarea concreta que quiera realizar con ellos.

En realidad, el fabricante de software para PcCs sigue manteniendo la propiedad del programa de aplicación, y continúa desarrollándolo por su cuenta para evitar que se quede obsoleto. Puede decirse con bastante objetividad que, en términos de licencia de uso, lo que Vd. compra es una licencia de uso ilimitada en el tiempo y con un mínimo soporte inmediato. Es normal también, de todos modos, que los fabricantes intenten conservar sus clientes, por lo que las nuevas versiones que van apareciendo de un programa suelen ofrecerse a los usuarios de versiones antiguas a un precio reducido. Se habla entonces de “actualizaciones”.

Un concepto relacionado con el anterior es el de “actualización competitiva”. Supongamos que estamos hablando de procesadores de textos, y por ir avanzando materia digamos, como de hecho así es en este momento, que los tres programas más populares para Windows en

## ***CAPÍTULO 8. PeCés: Programas***

este campo son Microsoft Word 6.0, WordPerfect 6.1 y Lotus WordPro. Pues bien, Microsoft podría vender a 35.000 pesetas su Word 6.0 a un nuevo usuario, en tanto que el precio de actualización para alguien que compró en su momento la versión anterior del mismo producto podría ser de 22.000. Para ganar clientela quitándosela a la competencia, Microsoft vendería el mismo producto a 25.000 pesetas a cualquier usuario de Lotus WordPro o WordPerfect, y a eso es a lo que se denomina “actualización competitiva”. Los condicionales de la frase anterior se deben tan sólo a que los precios son hipotéticos. Por lo demás la situación descrita es totalmente real.

Para tener derecho a beneficiarse de ofertas de actualización, un usuario de software para PeCé debe estar registrado, es decir, debe haber enviado en el momento de realizar la compra una “tarjeta de registro” del producto, aunque en ocasiones, especialmente en el caso de las actualizaciones competitivas, basta con la factura o se pide uno de los disquetes originales del producto a actualizar. Estar registrado es también necesario para acceder al soporte, poco o mucho, incluido por el vendedor en el precio de compra.

La adquisición y el uso de software está lleno de legalismos en los que no entraremos en profundidad. Baste indicar que la compra de software comercial para PeCé no suele dar más derecho que el ya indicado de su uso ilimitado por parte de su comprador y de nadie más, aunque no siempre es exactamente así. Al abrir el paquete de un programa suele aparecer, además de la correspondiente tarjeta de registro que conviene enviar cuanto antes a la dirección que se indique, una nota repleta de letra pequeña en la que se puntualizan los derechos (normalmente pocos) y deberes (habitualmente muchos) del comprador.

Uno de los derechos que debe incluir necesariamente es la posibilidad de realizar copias de seguridad del software. Normalmente el programa se distribuirá en un número indeterminado, y probablemente y por desgracia en la actualidad bastante elevado, de disquetes. No lo hemos dicho todavía pero no son así como muy fiables a largo plazo. Como ya dijimos en su momento, el programa que hemos comprado en disquetes deberá ser instalado en el disco duro de nuestro ordenador para poderlo usar. La operación consiste simplemente en ejecutar un programa que normalmente se hallará en el primero de ellos e ir respondiendo a nuestra conveniencia a las preguntas que nos haga. Hasta aquí, todo va bien. Pero recuerde que trabajar con un ordenador era más o menos como desayunar en un andamio, algo sometido a contingencias imprevisibles, y tal vez un mal día dentro de algunos meses y por motivos que no vienen al caso nuestro programa deje de funcionar. Habrá que volverlo a instalar, y unos meses de almacenamiento han podido ser fatales para alguno de los disquetes originales que recibimos. Para asegurarse de poder salir del paso es necesario que lo primero que haga con ellos, incluso antes de instalarlos, sea duplicarlos. El DISKCOPY de MS-DOS y un montón de disquetes nuevos suele bastar. Con dos copias del conjunto de disquetes tenemos una probabilidad razonable de poder reinstalar cuando haga falta. Y si Vd. es particularmente paranoico, siempre puede hacer una tercera<sup>2</sup>.

Los temas relacionados con la seguridad en el trabajo con el ordenador nos ocuparán parte del próximo capítulo, por lo que lo dejaremos estar por ahora. Y en cuanto a temas derivados

---

2 Es cada vez más frecuente que el software se distribuya en CD-ROM. Es un soporte más fiable que los disquetes y con mucha mayor capacidad, por lo que no es tan necesario, ni tan sencillo, realizar una copia de seguridad de los programas distribuidos de esta forma.

### ***Como forrarse vendiendo programas***

del uso legal o ilegal de los programas y las acciones que los fabricantes emprenden en ocasiones para limitar éste último, puede encontrar material adicional en el glosario bajo los títulos “Protección contra copia” y “Piratería”, por lo que también lo dejaremos estar. Tenemos otros asuntos más urgentes de los que hablar, ahora que acabamos de entrar en los grandes almacenes...

### **Como forrarse vendiendo programas**

### **LR-PeCé y sus proveedores**

Tras atravesar la metafórica puerta de nuestros grandes almacenes y de ascender no menos metafóricamente a la planta en la que se encuentre la sección de “Informática” que buscamos tan ansiosamente, podremos finalmente comenzar a recorrer de forma asimismo metafórica un montón de estanterías llenas de programas para PeCé.

Y en ese momento nos enfrentaremos a un hecho que deberíamos haber previsto pero que aún no hemos tratado. Al igual que cuando queremos comprar o estudiar la adquisición de cualquier otra cosa, un coche pongamos por caso, también ahora deberemos elegir entre distintos productos fabricados por diferentes marcas que globalmente hacen más o menos lo mismo. Pero que no necesariamente son iguales. No es sencillo razonar con alguien sobre la conveniencia de comprar un Citroën Xantia en lugar de un Renault Laguna, un Opel Vectra, un Audi A4, o un Seat Toledo. Como todo LAO sabe, en la elección de uno u otro modelo acaban por influir decenas de factores, algunos de ellos muy poco racionales. La imagen de marca es uno, cuya importancia no vamos a entrar a valorar.

Un recién llegado al mundo de la informática, como se supone que Vd. era cuando comenzó a leer hace ya algunos cientos de páginas, no tiene por qué estar enterado de cuales son los fabricantes de software para PeCé ni mucho menos sobre la imagen que cada uno de ellos ha conseguido asociar a su marca a base de largos años de presencia en el mercado y numerosas campañas de publicidad. Conviene, de todos modos, que se vaya acostumbrando a conocerlos y a crearse su propia opinión sobre ellos, ya que estas cosas van a acabar por influir también en Vd. cuando tenga que decidir a quién le compra su procesador de textos, o su hoja de cálculo.

Intentar dar algunas pinceladas sobre el perfil de cada marca me pone en una situación un tanto incómoda. Por una parte me comprometí de algún modo con el lector a ser todo lo crítico que fuera menester para tratar cualquier tema con rigor, y quizá ahora mismo algún que otro LAO espere ver sangre a borbotones. Pero por otra parte, las opiniones del tipo “la marca tal no hace un coche decente desde 1.920” suelen ser muy poco justificables de forma objetiva, lo que las condena al mero lenguaje coloquial. Y por si fuera poco, todo fabricante de software tiene su amor propio y no es honrado herirlo gratuitamente de forma pública. Una descalificación global, si procede, debe estar sólidamente fundamentada. En este tipo de situación, una antigua receta diplomática aconseja tratar únicamente los aspectos positivos y fiar el resto a la innegable habilidad de la gente inteligente para leer entre líneas. Intentaré salir del apuro aplicándola, y narraré por tanto historias de éxitos pasados que hicieron que una marca determinada gozara en su momento de un puesto entre los grandes, puesto que, por cierto, muchas aún conservan.

A la hora de tratar de quién es quién en el mundo del software para PeCés y de dar a conocer a los fabricantes importantes del ramo, no cabe duda alguna de quién ocupa el primer lugar. Microsoft, la compañía de Bill Gates, es el más o menos incontestable número uno. ¿Cuales

## ***CAPÍTULO 8. PeCés: Programas***

son sus poderes?. Podríamos resumirlo con una frase que yo he leído múltiples ocasiones en muchas revistas y que Vd. irá encontrándose de cuando en cuando en su camino si sigue en esto. Mucha gente dice que “Microsoft significa negocio”, y es probable que sea un buen resumen de una compañía que tiene la clara voluntad de ser el punto de referencia inexcusable en todo tipo de software para PeCé y que crea la base que soporta la actividad de muchas otras empresas.

Microsoft comenzó como una pequeña compañía que vendía un lenguaje de programación para ordenadores personales de 8 bits bastante antes de que empezara lo de los PeCés. El lenguaje de programación era un Basic que gozó de bastante aceptación<sup>3</sup>. Eran los tiempos en los que CP/M era el sistema operativo estándar de los micros de 8 bits y Digital Research el indiscutible jefe del cotarro. La gran habilidad de Microsoft, donde se basó su gran éxito, fue negociar acertadamente con IBM su contrato como suministrador del sistema operativo de los PeCés. Microsoft supo ver la importancia de un mercado todavía inexistente y tomó posiciones a tiempo para dominarlo a medio y largo plazo. No es descabellado decir que esa ha sido siempre, y aún es, su principal línea de acción. Por ejemplo, repitió posteriormente la jugada con Windows 3.0, enfrentándose para ello nada menos que a IBM.

Otra de sus características diferenciales consiste en aplicar la idea de que el software dirigido a un consumo masivo debe ser de apariencia sencilla. Parecer simple de manejar y serlo en lo posible. Más que dar un acceso completo a la mayor parte de los aspectos técnicos de sus programas, lo que busca Microsoft es que queden lo más alejados posible de un usuario que en principio se supone inexperto. Y debe reconocerse que consigue al menos que sus programas sean fáciles de instalar y de apariencia agradable.

Microsoft disfruta de dos grandes ventajas. Un nivel de ingresos mínimos enorme como consecuencia de su situación claramente dominante en el sector de los sistemas operativos y GUIs para PeCé (cada uno que se vende con DOS y Windows deja un dinero en sus arcas) y la voluntad, claramente manifestada muchas veces, de emplear dichos recursos para seguir siendo el punto de referencia en todo lo relacionado con su software. Eso explica su capacidad para aceptar impasible lo que podrían ser grandes fracasos para compañías menos sólidas en la confianza, muchas veces confirmada, de que sus productos acabarán por imponerse.

La gama de productos de Microsoft es en la actualidad enorme. Y la importancia de conseguir en el mundo de los PeCés la etiqueta de “estándar”, innegable. No es arriesgado decir que, en mayor o menor medida, cualquier producto de Microsoft es ya, o puede llegar a serlo, el estándar correspondiente para PeCés.

Microsoft no domina por igual, no podía ser de otro modo, todos los tipos de programas. Es especialmente fuerte en software base (DOS y Windows), lenguajes de programación, y software de aplicación para Windows (hojas de cálculo, procesadores de texto, y algún otro).

---

3 Ya veremos más adelante qué es eso de los lenguajes y algo más en concreto el Basic. Por ahora nos debería bastar con saber que son programas que sirven para desarrollar otros programas, por lo que son muy importantes. Y lo eran aún más en aquellos tiempos en los que el software de aplicación para microordenadores casi no existía y cada cual debía crear por su cuenta los programas necesarios para hacer lo que fuera con su ordenador

### *Como forrarse vendiendo programas*

Pero, en resumen, la compra de un producto de Microsoft tiene la garantía de aquel capaz de hacer y deshacer los estándares del PeCé. En ese sentido es probablemente la apuesta más segura.

Si la historia de Microsoft tiene el atractivo de los ejércitos casi invencibles, la de Borland es más romántica. Es la del guerrillero que, con fuerzas mucho más escasas, se las ingenia para lograr victorias importantes aunque sean parciales. Phillipe Kahn, el fundador y patrón durante mucho tiempo de Borland, es un francés especializado en Matemáticas, se supone que aplicadas, que desembarcó un buen día en Estados Unidos con la loable y aparentemente descabellada intención de vender sus programas en competencia con Microsoft, ya por entonces un gigante de escala más que respetable. Por si no bastara con tamaña insensatez, el producto de Borland era otro lenguaje de programación, en este caso un Pascal<sup>4</sup>.

Ocurría que ya en aquellos tiempos Microsoft dominaba los lenguajes de programación para PeCés y, además de su archifamoso Basic, tenía un Pascal en sus catálogos. Era un producto que se pretendía profesional, como corresponde a un lenguaje potente, y que se vendía en número más bien escaso pero a precio más bien elevado. Phillipe Kahn observó simplemente que el Pascal podía convertirse en un lenguaje de masas ya que al fin y al cabo estaba bastante de moda entonces, y que el gran obstáculo a vencer no era otro que el precio. La versión desarrollada por Borland era mucho más barata, más pequeña, y más rápida. Más sencilla tal vez pero más que suficientemente potente como para ser sobradamente útil. El Turbo Pascal, que así se llamaba el producto, se vendió en cantidades que nadie creía posibles para un lenguaje profesional, y demostró entre otras cosas que los usuarios del PeCé estaban dispuestos a absorber productos en teoría no propios de aficionados a la informática siempre que su precio fuera moderado y su calidad alta.

Borland siguió durante mucho tiempo cultivando el mercado de los lenguajes de programación, y aún sigue siendo el gran especialista del tema junto con Microsoft. Se diversificó con otros productos, y tiene asimismo una gran implantación en el sector de las bases de datos para PeCés (como veremos, algo en realidad bastante relacionado con los lenguajes de programación). También ha tenido éxito en el campo de las hojas de cálculo, primero para DOS y luego para Windows, con su gama Quattro Pro.

Phillipe Kahn ya no está en Borland. Hace poco fundó una compañía nueva llamada Starfish Software que entre otros productos distribuye en la actualidad Dashboard (el sustituto del administrador de programas que citamos en el capítulo anterior) y la versión para Windows de Sidekick, otro clásico de Borland, que tal vez conviniera comentar. Se trataba de un conjunto de pequeñas utilidades para DOS que permanecía en memoria siempre listo para ser llamado cuando hiciera falta, sin necesidad de salir del programa que estuviéramos usando. Una vez más un producto pequeño, barato, y ante todo útil. Por cierto, recientemente ha aparecido una versión ampliada para Windows. Borland ha sido durante mucho tiempo sinónimo de software de calidad a buen precio.

Otra marca a conocer es Lotus. Su especialidad, lo que le dio su fama y sus ingresos, fueron las hojas de cálculo para DOS, y su producto principal el llamado Lotus 123 (se lee un-dos-

---

4 Solo nos interesa por ahora saber que sirve para lo mismo que el Basic, es decir para hacer otros programas, y que en principio es más potente que aquel.

## ***CAPÍTULO 8. PeCés: Programas***

tres). También Microsoft tenía una, llamada Multiplan. Pero Lotus desarrolló un producto más completo, tanto que, como veremos más adelante, se propuso inicialmente como un paquete integrado, una de esas cosas que hacían de todo y con las que se intentó resolver en primera instancia lo del intercambio de datos en los PeCés. Ya podemos decir que hacía un algo de gráficos, un pelín de base de datos, y ante todo una más que razonable cantidad de hoja de cálculo, que era su fuerte. De ahí su nombre de un-dos-tres<sup>5</sup>. En consecuencia, no resultaba una base de datos capaz, ni un maravilloso programa para realizar gráficos, pero como hoja de cálculo era la mejor porque esos toques de gráficos y base de datos venían muchas veces de perlas.

Además de que, eso debería darse por descontado, era un programa de muy buena calidad general. Era rápido, potente, y aunque no podía considerarse pequeño resultaba todavía de tamaño tolerable para los tiempos que corrían<sup>6</sup>. Y su manejo era muy coherente y razonablemente cómodo, con una interfase de “menús horizontales autoexplicados” por llamarla de alguna manera que consistía en una primera línea con las distintas entradas del menú, de una sola palabra, y una segunda línea en la que automáticamente se detallaba la opción seleccionada en aquel momento. Lotus 123 tuvo tanto éxito que fue durante mucho tiempo la aplicación estrella de los PeCés, y estandarizó por completo el sector de las hojas de cálculo.

Ashton Tate es otra de las compañías que fue capaz de imponer un estándar sin llamarse Microsoft. Su producto era una base de datos de los tiempos del CP/M, y no está tampoco de más saber como se logró. Se llamaba inicialmente Vulcan, me parece recordar, y al principio no tuvo demasiado éxito. También me parece recordar que Ashton Tate la compró y lo primero que hizo fue cambiarle el nombre pasando a denominarla dBase II, abreviatura de “data base II” o “base de datos II”. No es que hubiera habido nunca un dBase I, pero eso le daba un cierto aspecto adicional de producto acabado y experimentado. Podríamos decir que la argucia funcionó tan bien que el producto se vendió en cantidades ingentes y se creó el correspondiente estándar en el sector de las bases de datos para DOS, pero probablemente pecaríamos de exageración manifiesta. Es probablemente más correcto decir que fue la primera base de datos disponible y con garantías para PeCés y aprovechó su ventaja inicial. A partir de ahí aparecieron las versiones III, IV, y en la actualidad hasta versiones para Windows. Pero no busque Ashton Tate en las estanterías de nuestro hipotético gran almacén. Ya no existe como tal, por motivos que pronto veremos.

WordPerfect es otro caso interesante. Inicialmente era una compañía radicada en Utah, que por cierto es un estado norteamericano con notable afición a la informática. Su producto era un procesador de textos para DOS que lucía el nombre de la empresa. Y que también fue capaz de imponerse frente a otros procesadores de texto para DOS, tales como el Word de Microsoft o el más antiguo WordStar, ya desaparecido y que constituyó el primer gran estándar en los tiempos del CP/M y primeros años del DOS.

---

5 Y a partir de aquí el nombre de Quattro para la hoja de cálculo de Borland, ya que fue lanzada al mercado para ser “más que Lotus 123”.

6 Aquellas versiones de Lotus 123 “no pequeñas pero aún de tamaño tolerable”, son, en relación con la elefantiásica cosecha actual de programas, calificables con total propiedad de minúsculas.

### *Como forrarse vendiendo programas*

Lo curioso de WordPerfect es la forma en que logró vender su producto. ¿Se acuerda de una nota al pie en la que se comentaron hace ya muchas páginas algunas aberraciones, por suerte ya pasadas, de su sistema de ayudas?. Pues si se acuerda, tendrá alguna pista sobre una chocante característica del producto. WordPerfect 4, por irnos a una versión ya madurita, tenía un respeto casi nulo por las convenciones del mundo del PeCé. Mientras que en todos los demás programas “F1” daba acceso a las ayudas, en WordPerfect se llegaba a ellas a través de “F3”, me parece recordar, y no eran sensibles al contexto como era de rigor, sino que estaban ordenadas alfabéticamente. En lugar de usar la tecla “Escape” para salir de donde no queríamos estar, había que emplear “F7”. Por su parte “Escape” no solo no escapaba, sino que reiteraba las veces que fuera una acción. Su interfase de usuario estaba basada hasta el paroxismo en el uso de combinaciones de teclas, completadas cuando era necesario por casi infinitos menús.

En resumen, emplear WordPerfect sin conocerlo a fondo y haber sido presentados previamente era una juega en la que se subvertían todas las suposiciones con las que un usuario de PeCé ya experimentado contaba de antemano. El programa era potente y muy sólido, eso sí, y tenía la gran virtud de ser casi sin discusión el que mejor sabía sacar partido a cualquiera de las 10.000 impresoras distintas con drivers no unificados a las que había que enfrentarse en aquellos tiempos. Y lo de la interfase de usuario a base de combinaciones de teclas tenía su razón de ser, pues los ratones aún eran raros y ya vimos que, una vez aprendidas las correspondientes combinaciones, lo más rápido es tener todas las funciones importantes a distancia de una tecla, por muy combinada que esté, especialmente si se trata de un procesador de textos y debemos usar el teclado necesariamente. Pero algo más debía haber para que un producto similar, además nada barato, pudiera llegar a imponerse.

La gran baza de WordPerfect era su soporte al usuario. Había un teléfono a disposición de cualquier cliente registrado del producto que, al menos en Estados Unidos, permitía consultar 24 horas al día y 7 días a la semana a un personal más que razonablemente capaz cualquier duda de uso o problema que pudiera surgir. Esto matizaba mucho el problema de una interfase de usuario nada estándar.

A partir de la versión 5.1 para DOS, WordPerfect dulcificó un poco su radical postura y pasó a respetar en alguna medida las convenciones, incluyendo menús desplegables y ayudas más o menos sensibles al contexto. En la actualidad, WordPerfect sigue siendo un actor importante en el sector de los procesadores de texto para PeCés. Luego volveremos sobre ello también.

Otra compañía a conocer con sede en Utah es Novell, que se encargó de demostrar hace ya tiempo que un recién llegado le podía plantar cara a cualquiera en cualquier sector. Incluidos Microsoft e IBM y sectores tan claramente estratégicos como el de los sistemas operativos para redes. No nos extenderemos demasiado ya que, una vez más, tenemos que renunciar a tratar un asunto que se sale del marco de este texto. Baste con decir que, simplemente, Novell propuso un muy buen producto, tal vez el mejor, y su Novell Netware se quedó con la mayor parte del negocio de los sistemas operativos DOS para red. Podemos decir que sigue conservando alrededor del 50% del negocio en la actualidad.

Nuestra galería de famosos particular debe incluir también necesariamente a Corel, quizá la incorporación más reciente al grupo de los grandes del software para PeCé. Su especialidad, sin discusión posible, son los programas de diseño gráfico para Windows. Su primer producto famoso, que data de cuando Windows estaba casi en pañales, es el llamado Corel

## ***CAPÍTULO 8. PeCés: Programas***

Draw, en principio un programa de dibujo. A partir de la versión 3, se convirtió, mas que en un programa, en un conjunto de programas que permitían cubrir dignamente todas las necesidades del diseño gráfico en un PeCé con Windows. De hecho, si intentamos caracterizar el enfoque que Corel da a las aplicaciones deberíamos hablar ante todo de su interés en proporcionar soluciones completas.

Por ejemplo, la realización de trabajos de diseño gráfico en cualquier ordenador lleva asociada la necesidad de disponer de equipos muy potentes en todos los sentidos y que en particular deben tener acceso a enormes capacidades de almacenamiento. En el mundo de los PeCés eso significó durante mucho tiempo recurrir a equipos dotados de interfase SCSI, cuando todavía no estaba demasiado desarrollada. Corel incluyó en su gama un conjunto de herramientas, drivers, y hasta “kits”, para añadir un CD-ROM basados en la tecnología SCSI, en los tiempos en que tales cosas parecían más curiosidades académicas que productos de consumo general, y fue de las primeras compañías que distribuyó su software utilizando el CD-ROM como soporte. No era una concesión a la moda, pues de acuerdo con este enfoque de vender “productos completos”, Corel incluyó también hace ya mucho tiempo con su Corel Draw enormes colecciones de gráficos ya realizados o tipos de letra listos para usar, lo que lo hace aún más útil.

Una última característica que distingue a Corel en la actualidad es su novedoso enfoque sobre las versiones antiguas de sus productos. Corel Draw 3 es un auténtico veterano, muy superado por las versiones posteriores que han ido apareciendo. La 4 era mejor, y la 5 no digamos. En la actualidad va por la versión 6, para Windows 95, que añade posibilidades de realizar gráficos tridimensionales. En muchas otras compañías, la aparición de una nueva versión lleva aparejada la inmediata desaparición de la anterior, que deja de distribuirse. No está claro que tal comportamiento tenga demasiado sentido. Es frecuente que la nueva versión sea más potente y mejor pero requiera equipos mucho más capaces para poderle sacar partido, y mucha gente no necesita las posibilidades adicionales. Pues bien, cuando apareció Corel Draw 4, se propuso a un precio ligeramente inferior al que tenía Corel Draw 3, pero ésta no se retiró del mercado sino que se abarató considerablemente. Corel ha estado vendiendo durante mucho tiempo de forma simultánea sus Corel Draw 3, 4, y 5 con precios escalonados entre las 10.000 y las 60.000 pesetas.

Existen por supuesto otro montón de marcas, más o menos especializadas en sectores concretos del software para PeCés. No podemos terminar sin citar a Central Point Software y sus herramientas para DOS y Windows, las durante mucho tiempo clásicas “PC Tools”, o a su gran competidor en el ramo, las “Norton Utilities”. Pero debemos poner algún límite, y probablemente tenemos ya más que suficiente con lo anterior.

La multitud de marcas indica que nadie es perfecto o que el campo a cubrir es mucho, o ambas cosas. Desde luego, en cuanto a lo primero todas han tenido enormes resbalones. Los éxitos de una son los fracasos de las demás, y que WordPerfect triunfara quiere también decir que a Microsoft Word para DOS, Borland Sprint, o Lotus Manuscript, les fue bastante mal. Y lo segundo es innegable, por lo que la respuesta más probable es la tercera.

Hay una característica de la producción de software que no debe perderse de vista. Una compañía ya instalada, grande y con muchos clientes, no necesariamente produce de forma automática un producto cualquiera mejor que un recién llegado. Es más, suelen ser las compañías más pequeñas, incluso de una o dos personas, las que son capaces de desarrollar los productos más innovadores, coherentes y capaces. Todas las grandes de hoy en día



### *Como forrarse vendiendo programas*

empezaron como pequeñas aventuras con un producto tirando a revolucionario. La situación por lo tanto es más abierta de lo que pudiera parecer y los francotiradores siguen teniendo un papel muy importante que jugar.

Pero tampoco podemos acabar el apartado sin destacar la clara tendencia que existe a la absorción de unas marcas por otras, dirigidas a lograr la consolidación de auténticos gigantes del software que puedan competir en igualdad de condiciones frente a compañías como Microsoft. Para que se haga una idea, expondremos algunos de los manejos en que se han visto envueltas las marcas que hemos presentado.

Ashton Tate ya no existe porque fue comprada, hace mucho tiempo, por Borland, que de este modo se convirtió en un actor principal en el segmento de las bases de datos, especialmente porque ya contaba con otro producto importante del ramo, una base de datos llamada Paradox. La misma Borland se deshizo de uno de sus productos estrella, la hoja de cálculo Quattro Pro en versión Windows, vendiéndosela a Novell, que en aquel momento estaba sumamente interesada en montar un conjunto de programas integrados para Windows (luego volveremos sobre esto). Para lograrlo, antes Novell compró WordPerfect, que tampoco existe ya como tal<sup>7</sup>, y luego añadió Paradox, al menos parcialmente. No era tampoco su primera compra, pues antes había hecho lo propio con Digital Research, la del DR-DOS que por eso acabó llamándose Novell DOS, y con algún productor de Unix, otro sistema operativo. Lotus tampoco es una compañía real en la actualidad, pues fue absorbida por IBM, aunque se decidió conservar la marca. Fue el modo que IBM encontró para tener por fin un cierto peso en la vertiente software de los Pecés, algo que se le resistía con empecinamiento. El producto de mayor interés para IBM tiene, una vez más, relación con las redes y se llama Lotus Notes. Central Point y Norton fueron absorbidas a su vez por Symantec, que se convirtió así de golpe en el especialista monopolístico de las herramientas para PeCé. Pero Novell perdió interés hace no demasiado en su aventura integradora, ya que un cambio en su dirección llevó al aparentemente inevitable cambio de política, y decidió volver a concentrarse en las redes, por lo que colocó el letrero de “se vende” sobre WordPerfect, Quattro Pro y algún otro. Corel los compró hace cuatro días (estamos en Marzo de 1.996 al escribir esto), en una decisión que parece razonable pues se comentaba hace ya tiempo su intención de extender su gama más allá del software para gráficos.

En consecuencia, la lectura que debemos hacer de este apartado tiene varios niveles. Nos interesan los datos concretos pero hay que tener presente que tan sólo como punto de partida para orientarnos en una situación sumamente cambiante. Conviene conocer la política de cada fabricante más como un dato histórico que como algo que pueda darse por sentado de forma inmutable en toda la gama de productos de una compañía determinada. Al fin y a la postre las compañías se compran y se venden y hasta cuando permanecen pueden cambiar de política con tanta facilidad como nosotros de camisa.

---

7 Si Vd. es uno de los millones de españoles que ha seguido con asiduidad el Tour de Francia en los últimos años, tal vez recuerde que en el 94 corría el equipo WordPerfect y al año siguiente lo sustituyó otro llamado Novell. Ahora puede explicarse el porqué de tales movimientos. Como puede ver, la informática tiene incluso consecuencias deportivas. Es este tipo de cuestiones, obviamente mucho más importante que el saber como se copia un fichero a un disquete, lo que innegablemente demuestra que la informática para PeCés es un tema a conocer en el mundo actual.

## ***CAPÍTULO 8. PeCés: Programas***

Quizá lo más concreto que nos quede sea la posibilidad recién adquirida de saber un buen montón de cosas que se le pueden exigir a cualquier fabricante porque forman parte de la cultura general. Toda marca deberá ofrecer un producto a ser posible tan estándar y accesible al usuario inexperto como los de Microsoft, tan rápido, barato y útil como los de Borland, con ese algo más que hizo de Lotus 123 un éxito memorable, con un soporte al usuario tan ejemplar como el de WordPerfect, tan revolucionario y claramente mejor como los de Novell, y tan completo como los de Corel. Y si es necesario, el fabricante puede ayudarse en su empeño de imponerlo al mundo con alguna ingeniosa argucia, como en su momento hizo Ashton-Tate.

### **Otras formas de ganar pasta, aunque menos**

### ***Luis Ricardo se va de “wares”***

Todo el comentario que la visita a nuestro hipotético gran almacén en su sección de “Informática” puede arrancar de nuestro también hipotético compañero de compras es un simple: “El software serio para Pecés es muy caro. Cualquier programa comercial de marca conocida vale al menos unas 10.000 pelas. Antes de rascarme el bolsillo me gustaría enredar un poco, a ver si me termino de orientar en DOS y Windows y luego, si el invento me convence, ya decidiré qué compro. ¿No hay una forma legal de conseguir programas baratos para poder ir haciendo algo útil?”.

Debe reconocerse que nuestro hipotético LAO está en su perfecto derecho de realizar tal observación y que su actitud es cuando menos calificable de prudente. La triste realidad es que la compra de un paquete comercial no está exenta de riesgos, y conviene andar con pies de plomo. Al menos en España, y salvo que el comercio donde hayamos adquirido el producto proponga garantías adicionales, uno suele enfrentarse con cierta asiduidad cuando mira la caja que contiene un programa recién comprado a notas del tipo: “No se admitirá ninguna devolución una vez abierto”. Ante lo cual sólo queda comprobar si en alguna parte del exterior se indica claramente que es ése el programa que queríamos y viene en el soporte y formato deseados, y resignarse ante la fatalidad y proferir un “Que sea lo que Dios quiera” en el momento en que lo abrimos para instalarlo en nuestro ordenador con el fin de comprobar si también, además, hace lo que nosotros queríamos que hiciera.

Traducido a términos más familiares es como si Vd. comprara un televisor, lo sacara de la caja, lo enchufara a la red, se encontrara con unos colores espantosos y un sonido horrible y, al ir al comercio a hacer valer su garantía, recibiera por toda respuesta un: “Lo sentimos mucho pero como ya lo ha puesto en marcha no podemos aceptar ninguna responsabilidad”. ¿A que suena absurdo?.

Si no “a ciegas”, la adquisición de software comercial en nuestro país es un proceso que se desarrolla al menos “a tuertas”. En otros países es otra cosa. Si lee revistas americanas verá con cierta asiduidad notas bastante diferentes, del tipo “30 days money back guarantee” que quiere decir que Vd. tiene treinta días para devolver el programa que acaba de comprar si no llega a convencerle. Y por supuesto con la caja abierta<sup>8</sup>.

---

8 Existen motivos racionales que justifican las restrictivas políticas de distribución puestas en práctica por algunos productores de software. Se arguye muchas veces para justificar la imposibilidad de devolver paquetes abiertos que alguien podría comprar un programa, copiarlo, y devolverlo para recuperar el dinero. Sin dejar de ser cierto, es discutible la fuerza real de tal

### *Otras formas de ganar pasta, aunque menos*

¿Qué mecanismo legal existe para que un futuro usuario compruebe si el programa tal de la marca cual es efectivamente lo que anda buscando antes de soltar la tela?. La solución que suele proponerse son las llamadas “demos”, abreviatura de “demostración” o “programa de demostración”. Consisten en versiones normalmente gratuitas que intentan darnos una idea del funcionamiento del programa comercial. Su contenido varía entre una mera sucesión de pantallas laudatorias del producto, sin mayor valor operativo, y una copia limitada del programa de verdad que, aunque por ejemplo no nos deje grabar los datos que con ella vayamos produciendo o se desactive automáticamente 30 días después de su instalación, al menos sí que nos permite ver la cosa tal cual es.

La idea de las demos no está mal en teoría, pero su concreción en la práctica ha llevado a la aparición de chistes tan clásicos como el que sigue. Dícese que un buen día llega la hora de un usuario de productos de informática personal. Sus muchos sufrimientos en este terrenal valle de lágrimas lo llevan directamente al cielo, pero al llegar y ser presentado a San Pedro, nuestro héroe saca a relucir la desconfianza adquirida a lo largo de muchos años de chascos microinformáticos y dice: “Oye, si no te importa, antes de quedarme me gustaría ver lo que hay en el infierno y el purgatorio. Me parece que en esos sitios también me admitirían y quizá no estén mal del todo. Y ya que estoy aquí, podría aprovechar para echar una miradita”. San Pedro accede y nuestro protagonista se da un paseo por el cielo. Lo que ve no le acaba de convencer. La gente está contenta, o al menos eso parece, pero no se aprecia demasiado entusiasmo. “Bueno, me voy a ver las otras alternativas”, dice. El purgatorio es mucho más anodino, y eso hace que se encamine al infierno. Allí le repite la historia a Pedro Botero: “Querría ver que tal pinta tiene esto, por si me interesa quedarme”. Pedro Botero no pone inconvenientes: “No hay problema, date una vuelta por la puerta 3”. Allí va, y lo que ve le maravilla. Jolgorio, diversión, la pera. Vuelve a la garita de Pedro Botero y, sin dudar, le dice: “Me quedo. ¿Donde firmo?”. Pedro Botero le presenta el contrato, nuestro antiguo usuario microinformático lo firma sin dudar, y se le indica la puerta principal. Ya dentro, el mismo Pedro Botero lo mete en la caldera 8, coge un tridente, y empieza a pincharle. Nuestro protagonista, asombrado, exclama: “Pero bueno, ¿esto qué es?. ¿Y la marcha de antes?”. Pedro Botero contesta impasible: “Hombre, eso era una demo”<sup>9</sup>.

En consecuencia, la única información potencialmente fiable de que se dispone antes de adquirir un programa comercial se reduce a las pruebas de software que aparecen en las revistas de informática y las opiniones que podamos recabar entre aquellos amigos que lo tengan. De ambas, suele ser mejor la segunda. Salvo que nos beneficiemos de alguna circunstancia particular, por ejemplo que podamos usar el programa cuya adquisición estamos estudiando en nuestro lugar de trabajo, o consigamos una demo totalmente operativa aun-

---

argumento y otros similares. Como veremos enseguida, hay otras formas de hacer las cosas que no suponen automáticamente que todo cliente sea un estafador en potencia. El único argumento irrefutable es que es casi imposible especificar con detalle lo que un programa debe hacer. Es muy fácil saber si un equipo de televisión va bien o mal, pero no tanto si un programa hace lo que debe. El cliente de un producto informático se sentirá satisfecho sólo si percibe claramente que ha recibido algo de valor a cambio de su dinero. Y la evaluación será siempre subjetiva.

- 9 El chiste me lo contó, hace cosa de dos años, un amigo que trabajaba en una empresa de servicios informáticos. Lo he visto reproducido recientemente en un artículo de “Science et Vio Micro” que trataba sobre humor en Internet.

## ***CAPÍTULO 8. PeCés: Programas***

que limitada en su uso a unos cuantos días, la compra se realizará muy probablemente “a tuertas”, como ya dijimos. Y aunque lo de las demos nos parezca más que correcto (ya dije hace un rato que cada cual debe formarse su propia opinión, basada en sus propios juicios) todavía nos quedaría por saber como conseguirlas.

Más o menos lo anterior sería lo que iríamos largando con nuestro hipotético acompañante mientras salimos de los grandes almacenes. Su pregunta inicial, y las que han ido surgiendo con el cuento de las demos, pueden contestarse encaminando nuestros pasos hacia un destino muy sencillo de localizar en cualquier ciudad española: un quiosco de esos en los que se venden revistas de todas clases, hasta de informática. Es obviamente el lugar más adecuado para conseguir información relativa a como es cada programa comercial. Pero no sólo eso. Ahora mismo es también el sitio en el que conseguir, a precio muchas veces irrisorio, una cantidad ingente de programas, demos incluidas.

Para entender el porqué podemos encontrar la solución a nuestros problemas en un lugar tan anodino<sup>10</sup> es preciso que volvamos a tratar el tema de las formas de distribución del software. En el apartado anterior hemos presentado a una serie de compañías grandes. Su tamaño les permite contar con mecanismos propios de distribución. Suelen tener concesionarios en cada país e insertan publicidad en revistas y periódicos. Pero también hemos dicho que los francotiradores podían hacer programas muy interesantes, y está claro que una empresa embrionaria, quizá incluso todavía no formalmente constituida y formada por una única persona, no puede aspirar a vender sus productos de esta forma. Hacen falta políticas de ventas y canales de distribución alternativos.

La informática dispone de un par de figuras que resultan inicialmente sorprendentes. Tienen su propio nombre, y es preciso conocerlas. Vamos a hablar pues de “shareware” y de “freeware”. Si se desea la traducción de tan populares términos, podríamos hablar de “software compartido” en el caso del shareware (aunque en este caso la traducción no se emplea jamás), y de “software de dominio público” en el caso del freeware (y es en este caso la denominación inglesa la que casi nunca se usa).

La idea del shareware es en realidad sumamente tradicional. No deja de ser la aplicación al negocio del software de la clásica venta directa. Algo así como el artesano que hace collares e instala su puestecillo en una calle cualquiera. Se deja que el programa que se desea vender se copie y use libremente. Todo el que quiera puede instalarlo y trabajar con él. Si el producto le gusta, puede registrarse como usuario por una cantidad módica y recibir una serie de beneficios adicionales. El concepto es “pruebe Vd. mi programa gratis, convénzase de que realmente es maravilloso y le sirve para lo que quiere hacer, y luego pague”.

Los programas distribuidos mediante shareware no son programas de uso libre. El fabricante se reserva todos los derechos sobre ellos, pero da la oportunidad inicial de utilizarlos por un período de prueba más o menos indefinido y hasta de distribuir copias a los amigos y

---

10 No es que crea que los quioscos sean algo despreciable. Al contrario. A mí me encantan. Hace muchos años que no consigo resistir la tentación de pasar un rato delante de montones de titulares diferentes y portadas de revistas multicolores, por lo que mis paseos por la ciudad suelen tener numerosas pausas.

### ***Otras formas de ganar pasta, aunque menos***

conocidos. De hecho, suele animar a que se haga así, ya que le interesa que su producto llegue al mayor número de clientes potenciales

¿Qué se gana al registrarse, es decir al pagar la copia de un programa que ya tenemos, hemos estado usando y pretendemos seguir empleando en el futuro?. El shareware se basa ante todo en la buena intención del comprador. Se supone que vamos a pagar los programas que realmente nos sean de utilidad. Si no queremos dar pruebas evidentes de lo ratas que somos y pretendemos acallar los reproches que nuestra conciencia pueda hacernos, hay que acabar soltando unas cuantas pesetas. E incluso si los aspectos morales no nos preocupan demasiado, no está de más pagar. Contribuir a que la compañía que lo ha desarrollado sobreviva nos asegura en lo posible el futuro perfeccionamiento y soporte del producto.

Tal vez Vd. se sorprenda de que tal sistema pueda funcionar, pero el hecho es que funciona. Y lo hace tan bien que permite que el programa desarrollado por una sola persona se venda en todo el planeta. Existen compañías que lo han usado durante mucho tiempo y aún lo emplean normalmente. Su éxito se basa en algo más que la buena voluntad de sus clientes, que no siempre existe. En primer lugar, un mecanismo de distribución tan directo y el hecho de ser compañías muy pequeñas permite que los precios sean siempre muy accesibles. Registrar un programa de shareware puede costar unas 5.000 pesetas. Y para que la gente se anime, pagarlas suele dar derecho a recibir los manuales impresos, o la siguiente versión que aparezca, o versiones aún más completas que la distribuida libremente y que ya ha demostrado merecer nuestro interés.

En la práctica cada compañía adapta a su idiosincrasia particular la distribución mediante shareware. Las menos confiadas sólo distribuyen de forma gratuita versiones con límite en el tiempo de libre utilización o que no incluyen todas las funciones del producto registrado. Lo más frecuente de todos modos, y diríamos que el mínimo para poder considerar un programa como de shareware y no como una demo, es que la versión que se distribuye de forma libre sea totalmente operativa sin límite de tiempo de uso. Es normal que la única restricción a que debamos enfrentarnos sea una pantalla que, al lanzar o salir del programa, nos informe de que la copia no está registrada y no estaría de más que enviáramos unos cuartos si vamos a seguir trabajando con él asiduamente.

La idea detrás del software de dominio público es aún más radical. Ya no se trata de vender nada, sino que simplemente un programa determinado se pone a disposición de todo el que quiera usarlo. Se puede copiar. Se puede instalar en quince ordenadores a la vez si es que los tenemos a nuestra disposición. No se nos pide ninguna cantidad a cambio. Aunque parezca imposible que exista gente capaz de dedicar un montón de esfuerzo al desarrollo de un programa interesante para luego regalarlo, el hecho es que la hay, y conviene saberlo y aprovecharlo en lo posible.

De todos modos, no es totalmente cierto que podamos hacer lo que nos dé la gana con el software de dominio público. Suelen imponerse algunas restricciones, que espero parezcan razonables. Consisten en cosas tales como la obligación de citar al autor y mantener en dominio público las nuevas versiones o desarrollos que se basen directamente en su trabajo. Es decir que el autor no renuncia a sus derechos si algún listo pretende pasarse y vender básicamente lo mismo como si fuera suyo. Sí que suele admitirse en cambio que se cobre una cantidad moderada de dinero por la distribución. En resumen, lo que se busca es mantener el programa y los desarrollados a partir de él dentro del dominio público e incentivar a aquellas compañías que lo deseen a distribuir tales productos. Compañías que

## ***CAPÍTULO 8. PeCés: Programas***

normalmente suelen añadir algunas cosas de su propia cosecha, aumentando su utilidad y justificando aún más el poco dinero que piden.

¿Qué podemos encontrar en el mundo del shareware y el software de dominio público?. De todo, e incluso muchas veces programas únicos, que no pueden adquirirse de otra forma. No es extraño, sin embargo, que cubran de forma preferente aquellos huecos que dejan las aplicaciones comerciales. Más que hojas de cálculo o procesadores de textos, que también los hay, encontraremos editores de texto (ya veremos cual es la diferencia entre un procesador y un editor), calendarios avanzados, pequeños programas que hacen cosas prácticas (por ejemplo imprimir carátulas de casetes musicales), lenguajes de programación poco conocidos, y cualquier otra de una infinidad de posibilidades.

En cuanto a su calidad, no debe caerse en el error de pensar que sólo pueden encontrarse programas defectuosos, limitados, o particularmente incómodos de usar. Hay en el shareware y dominio público programas casi tan buenos como los comerciales, y quizá sobre el casi. Es más, si Vd. busca programas pequeños, rápidos, y ante todo prácticos, es muy probable que sólo los encuentre ahí. Se pueden poner muchos ejemplos para apoyar semejantes afirmaciones, pero debería bastar con uno para que se hiciera una idea bastante correcta. ¿Se acuerda de 4DOS, un sustituto de COMMAND.COM que hemos citado en alguna ocasión y del que sólo puede decirse que es con certeza el mejor intérprete de órdenes disponible para DOS?. Pues es un producto de shareware, en concreto de una compañía llamada JP Software, que por si fuera poco cuenta con una gama bastante más extensa de productos distribuidos del mismo modo, entre ellos un administrador alternativo de programas para Windows llamado "TakeCommand" y versiones de 4DOS para OS/2 y Windows NT. Son los especialistas indiscutidos del sector. Y no es más que uno entre muchos casos análogos.

Y si aún no le basta porque Vd. cree que bueno, el shareware quizá, pero el dominio público es imposible que tenga nada interesante, aquí va otro ejemplo demoledor. Quizá lo más difícil de desarrollar y de mayor importancia para un ordenador sea un tipo de programa que deberíamos conocer bien, el sistema operativo. Ahora mismo, es probable que el sistema operativo de más rápido crecimiento para PeCés, especialmente en el sector de los servidores para Internet (los ordenadores que proporcionan la información a la que acceden los usuarios, como veremos más adelante) sea justamente un Unix de dominio público llamado Linux. No es ninguna broma que un producto de dominio público pueda llegar a competir en un sector en el que hace falta software particularmente potente, sólido, y completo. Si le interesa el tema, sepa que, una vez más, volveremos sobre Linux en su momento, en concreto en el capítulo 10.

Hay buenas razones detrás de la calidad del software de dominio público y del distribuido mediante shareware. Los programas de tamaño moderado, hechos por poca gente, son más sencillos de construir y depurar que los enormes en los que hace falta un equipo numeroso. Su estructura lógica es más sencilla, la gente implicada en su desarrollo los conoce mejor y se coordina más fácilmente, y al final el producto tiene menos errores y puede optimizarse con poca dificultad. Por si fuera poco, el diseño general debe atender tan sólo a la idoneidad del programa. Aspectos que una gran compañía no puede ignorar al plantear un nuevo producto, tales como la forma en que puede afectar a la venta del resto de su gama, y que pueden alterar el diseño y complicarlo sobremedida o incluso aconsejar que no se desarrolle, están fuera de lugar. El único criterio a respetar para hacer un programa de dominio público o shareware es que haga algo útil y lo haga bien, para que él solo sea capaz de salir adelante

### *Otras formas de ganar pasta, aunque menos*

sin más publicidad que la que puedan hacerle sus usuarios. Y en cuanto a posibles diferencias de competencia entre los programadores que desarrollan el software comercial y el de shareware o dominio público, debe partirse de la base de que la misma persona que hace programas 8 horas al día para una gran editora puede desarrollar en sus ratos libres productos de shareware, y tenerse en cuenta que muchos productos de dominio público han surgido en las universidades y son obra de aquellos que enseñan a programar a los demás.

El software de dominio público puede beneficiarse de un fenómeno adicional. Si se pone a disposición de todo el mundo todo el material relacionado con el desarrollo del programa, es posible conseguir que muchos programadores trabajen de forma espontánea en su perfeccionamiento<sup>11</sup>. El producto en cuestión se convierte en un desarrollo colectivo que todos sienten en alguna medida como propio y al que cada uno aporta aquello en lo que es más competente. Con un mínimo de coordinación general puede conseguirse un producto que evolucione rápidamente y que termine superando en mucho las expectativas iniciales. Linux es por el momento el ejemplo más claro de un desarrollo en común de este tipo.

Como siempre, algún problema debe haber en lo que parece una ganga manifiesta. ¿Cuales son los inconvenientes a soportar si optamos por el shareware y el dominio público como fuentes principales para construir nuestra biblioteca de software?. En tiempos podía decirse sin dudar un segundo que los principales problemas eran la falta de soporte al usuario y la irregular calidad de los programas. Un programa comercial era siempre sólido y razonablemente completo, e incluía un manual impreso de calidad realmente ejemplar y magníficas ayudas en línea, mientras que su equivalente en shareware podía ser maravilloso o realmente espantoso y nos daba como único manual un fichero tipo texto que el usuario podía, si quería, imprimir por su cuenta. Por si fuera poco, sus ayudas en línea no eran como para tirar cohetes. Las ventajas de ser un usuario registrado de un programa comercial eran palpables también en otros aspectos, ya que se podían consultar las dudas a un servicio gratuito de asistencia telefónica y las ofertas de actualización eran sustanciosas. El shareware y el freeware quedaban casi exclusivamente al alcance de gente lo suficientemente fogueada como para salir adelante por sí solos.

Aunque puede seguirse manteniendo que, en general y por principio, los programas de shareware y especialmente de dominio público exigen de sus usuarios una mayor competencia técnica y dan un menor soporte, la situación actual debe matizarse mucho más. Por un lado los programas comerciales, como veremos en el capítulo siguiente, han bajado precios pero han eliminado manuales y reducido mucho el nivel de asistencia postventa. Una competencia desenfrenada ha llevado muchas veces al lanzamiento prematuro de pro-

---

11 Los programas se construyen empleando un lenguaje de programación. Inicialmente consisten en un montón de ficheros de texto con órdenes que el ordenador no puede ejecutar directamente. Para convertirlo en un programa de verdad, hace falta otro programa. Ese otro programa, al que en general llamaremos compilador, es el encargado de convertir las órdenes en texto accesibles a los humanos en ristas de bits que la CPU es capaz de ejecutar, es decir en código máquina. La versión original, el texto legible, se llama código fuente, y la versión ejecutable por el ordenador, código objeto. Los programas comerciales suelen ser copias del código objeto, con lo que se dificulta su modificación por personal no autorizado. No es extraño que un programa de dominio público incluya también el código fuente y hasta las herramientas de desarrollo necesarias para su manipulación eficaz, en un esfuerzo dirigido justamente a que cualquiera lo suficientemente capacitado pueda modificarlo a su conveniencia.

## ***CAPÍTULO 8. PeCés: Programas***

gramas poco probados, inestables o incompletos. Por otro lado los programas de shareware y dominio público se han hecho cada vez más “profesionales” y, al menos en las versiones registradas, cuentan ya con manuales impresos de buena calidad y ayudas en línea que no tienen que envidiar nada a nadie. Quizá sólo queden como puntos claramente a favor del software comercial la existencia de versiones traducidas a distintos idiomas y su mayor fiabilidad en términos generales. El shareware y el dominio público suele estar en inglés o en el lenguaje nativo de su autor, y sigue reuniendo lo excelso y lo casi inutilizable. Pero también hay que aclarar que no todo el software comercial, ni mucho menos, se traduce a varios idiomas y que una versión de desarrollo en shareware suele distribuirse claramente como tal y siempre con precios bajos, por lo que no engaña a nadie y es más difícil sentirse estafado.

La idea del dominio público, y en menor grado el shareware, no es en absoluto banal o quijotesca. Bastaría para demostrar lo anterior con recordar que una de las principales bazas con que contó el PeCé para conquistar el mundo informático consistió precisamente en declarar su diseño de dominio público. Y los resultados están a la vista de cualquiera. En cuanto a que haga falta ser algo más que un usuario sin pretensiones para beneficiarse de él, ya hemos sugerido varias veces, y volveremos sobre ello, que la legítima aspiración de ser “sólo un usuario” es probablemente irrealizable en el contexto de la informática personal.

Conseguir productos de shareware y dominio público es bastante simple. Si Vd. tiene posibilidad de acceder a Internet basta con entrar en uno cualquiera de los cientos de servidores que los almacenan y copiar los que más le apetezca a su ordenador. Internet es ahora mismo el centro primario de distribución, el almacén general de uso público. Pero tal vez no pueda conectarse a Internet, al fin y al cabo no es tan raro. O quizá aunque pueda hacerlo no le seduzca la posibilidad de traerse algunas decenas de megabytes a través de la línea telefónica. Pueden ser varias horas de conexión con resultado final más bien incierto. Ya veremos, como siempre en otro capítulo, que también Internet tiene sus puntos negros. En ambos casos existen soluciones y se encuentran, ya hemos llegado, en los quioscos. Muchas revistas incluyen ahora mismo en el precio uno o varios CD-ROM con colecciones de shareware y dominio público, o están dedicadas de forma exclusiva a su distribución. Son la forma más conveniente de hacerse con montones de programas a bajo coste y justifican por sí solos, aunque existen más motivos, la adquisición de un lector de CD-ROM.

Y recuerde que existe una tercera vía. El shareware y el dominio público puede copiarse y distribuirse con libertad, y seguro que Vd. tiene amigos metidos en esto. Comience a usar el boca a boca, algo fundamental y a lo que no siempre se le da la importancia que merece. Como verá si sigue con los micros el tiempo suficiente, hace falta relacionarse e intercambiar información. Sus conocidos acabarán siendo su principal soporte, los que le digan qué programa merece la pena y cual no, o como salir de un problema en un momento determinado. Acostúmbrese a intercambiar con ellos opiniones, programas, y soluciones a problemas. La microinformática es cada vez más un universo caótico en el que resulta difícil salir adelante en solitario. Es preciso poder recurrir a los amigos cuando algo vaya mal, y quizá por eso los usuarios de microinformática muestran espontáneamente cierta tendencia a la agrupación. Une bastante como afición común. O tal vez sea que largas horas ante una pantalla en solitario despiertan automáticamente un notable deseo de compensar el aislamiento charrando con alguien. Puede que la idea de software de dominio público surgiera de ahí, como un intento de independizar en lo posible al colectivo de las acechanzas del exterior. Hace falta un cierto espíritu solidario, por el cual se ayuda hoy al que muy probablemente nos ayudará mañana. No se retrase y entre en el círculo lo antes posible. A ver



### ***Otras formas de ganar pasta, aunque menos***

porque cree que hemos elegido para el capítulo una estructura de conversación con un amigo hipotético.

Serán justamente los amigos los que nos ayuden a paliar los problemas residuales ya citados del shareware y el dominio público. Conseguir una colección de programas es fácil, pero hace falta separar los buenos del resto. A no ser que alguien nos diga qué programas merecen la pena (y volvemos al boca a boca), hay tajo para coger un CD-ROM y echar un vistazo a los programas que contiene. En la práctica sólo puede probarse un programa cuando hace falta ponerlo en marcha. Se tiene una colección de CD-ROM y cuando aparece la necesidad concreta de trabajar en algo nuevo se estudia para ver qué podemos sacar de allí que merezca la pena.

Otra posible fuente para conseguir programas de shareware y dominio público es la venta por correo. Existen empresas que se dedican a recopilarlo y empaquetarlo de forma que resulte fácil de usar y que hacen de paso esa tan necesaria selección de productos de calidad. Sus direcciones pueden encontrarse una vez más en las revistas de informática o mediante el boca a boca. En tiempos distribuían sus productos en disquetes, pero en la actualidad usan de forma casi única el CD-ROM.

Acabaremos con el asunto de los distintos mecanismos de distribución con una última nota. No tiene demasiado sentido diferenciar estrictamente los programas según como se distribuyen. En la práctica muchas empresas que se dedican principalmente a la venta de software comercial usan también procedimientos alternativos de distribución para algunos productos. Por poner sólo un par de ejemplos particularmente contundentes, IBM ha distribuido su OS/2 en shareware en los últimos tiempos a través de revistas, y Microsoft puso desde un principio en dominio público componentes tan importantes como las extensiones de 32 bits para Windows 3.1. Además de que, por supuesto, toda editora comercial que se precie suele usar las revistas y el dominio público para realizar la distribución de las demos de sus programas comerciales. Los mecanismos de distribución son sólo eso, están a disposición de todo el mundo y todos los usan de la manera que consideran más conveniente para sus intereses.

En resumen, mediante cualquiera de los métodos citados es posible conseguir por unos 1.000 duros una cantidad tal de programas que podemos pasar varios meses simplemente probándolos uno tras otro. Para aquel que realmente carezca de experiencia, la solución más eficaz sin duda consiste en buscar el apoyo de alguien que no sólo le pase copias de programas de interés, sino que también se tome la molestia de instalarlos en su ordenador y le enseñe como desenvolverse por sí mismo en lo sucesivo. Mil pesetas para una revista con CD-ROM y cuatro mil pesetas en cervezas (o buen vino) y bocadillos, puede ser un reparto adecuado del presupuesto anterior en este caso.

Supongamos que nos hemos decidido por esta última posibilidad. Acompañados de nuestro mentor, hemos comprado en un quiosco la revista o recopilación que parecía más adecuada y nos hemos ido con él a su casa para que recogiera las copias de los programas adicionales que considerara convenientes. Una visita al supermercado nos ha permitido terminar la faena llenando el frigorífico con provisiones suficientes para cubrir con holgura el tiempo que cueste instalar el software y completar las explicaciones delante del ordenador.

En el próximo apartado veremos qué nos contará más o menos nuestro benefactor mientras montaba los programas que nos permitieran introducirnos en el mundo de los PeCés con

## ***CAPÍTULO 8. PeCés: Programas***

DOS y Windows con un mínimo de comodidad, pudiendo hacer un montón de cosas útiles, y a precio razonable.

### **Hit parade**

### **Montando el piso de LR-PeCé**

Tras acompañar a nuestro nuevo usuario hipotético y a su no menos hipotético amigo a casa del primero y ver como se sientan ante la pantalla del ordenador en que van a trabajar, no nos queda más que ceder la palabra al experto que va a realizar la selección e instalación del software necesario. Se recomienda, antes de empezar, que se lean las notas al pie<sup>12</sup>. Abramos comillas.

“Bueno, lo primero de todo es ver como está esto. El ordenador vino con MS-DOS 6.22 y Windows 3.11 ya instalados, según me dijiste, y lo podemos dejar así. Sigue siendo una combinación bastante normal, aunque tampoco es raro que se ofrezca la posibilidad de sustituirlos por Windows 95. En fin, sobre Windows 95 hablaremos otro día, si te interesa. Ahora a ver que tal tienes configurado el equipo y cuanto sitio te queda en el disco duro para instalar programas. Humm... . Bien, la configuración no está nada mal, y en cuanto a sitio, tienes más que suficiente. Le pusiste un disco duro francamente gordo. 1’2 Gigabytes da para bastante, y ahora mismo no tiene demasiado sentido poner menos. Al fin y al cabo te ahorras cuatro duros y el espacio adicional siempre es de agradecer. Por cierto, veo que los has dividido en dos unidades lógicas<sup>13</sup>. Te alabo el gusto, porque puede venir francamente bien a la hora de hacer según que cosas. ¿Que no es cosa tuya y fue otro amigo el que te hizo toda la configuración e instaló Windows?. Debía haberlo imaginado al ver CONFIG.SYS y AUTOEXEC.BAT. No los ha hecho un novato, desde luego. Le dices cuando lo veas que hizo bien. No creas que es difícil. Si le metes un poco de tiempo, pronto te encargarás tú de todo. En fin, dejémoslo estar porque ya no hay mucho que rascar en ese tema y pasemos a instalar los programas de verdad.

Antes de meternos con Windows, conviene que completemos DOS. Hay que solucionar algunos problemillas. En primer lugar, pronto verás que la mayor parte de los programas de shareware y dominio público que se distribuyen vienen comprimidos, y nos hacen falta programas adecuados para poderlos manejar. No, no es lo mismo que lo de los discos comprimidos. Se trata ahora de crear un fichero en el que se almacenen en un formato más compacto aquellos ficheros que nosotros elijamos. Así podemos meter más datos en cualquier medio de almacenamiento. Al contrario que con un disco comprimido, en el que tú no ves más que una unidad lógica adicional y ficheros normales, ahora vamos a ver los ficheros comprimidos, y somos nosotros los que deberemos crearlos o restaurar a partir de ellos los ficheros originales sin comprimir, dejándolos en el directorio que nos apetezca. Este tipo de operaciones las hacen varios programas de shareware. Se llaman en general

---

12 El conjunto de programas que vamos a citar es, por supuesto, orientativo, y ha sido elegido según el criterio particular del amigo experto que nos ayuda a preparar este capítulo. Criterio que no era otro que proporcionar una visión lo suficientemente completa de las aplicaciones más populares. Aunque resulta asombroso el grado de coincidencia entre su selección y la que haría el autor (¿porqué será?), es obvio que no es la única posible. Las necesidades de cada usuario, y sus preferencias, deben ser las que guíen la elección de un software determinado.

13 Aparecerían como unidades C: y D:, como ya debería imaginarse.

compresores de ficheros. Quizá el más popular sea PKZIP, aunque también es muy frecuente otro que se llama ARJ. Y hay más. A veces, por ejemplo, topará con ficheros comprimidos con LHA, y en otros sistemas operativos tienen también los suyos. Los programas de Windows, sin ir más lejos, suelen distribuirse en un formato comprimido propio. Todos los ficheros de los discos de instalación de programas Windows que veas que terminan la extensión con un carácter de subrayado, están comprimidos, aunque en ese caso se comprime cada fichero por separado. PKZIP y ARJ, que son los que vamos a instalar ahora, crean un único fichero comprimido en el que meten todos los ficheros normales que quieras, ya sea uno o doscientos<sup>14</sup>, y suelen guardar de paso el nombre del directorio en el que se encontraban inicialmente, para que pueda volverse con facilidad a la situación de partida. Como verás, se han convertido en partes esenciales del DOS, con bastantes más posibilidades que el mero transporte de datos.

En fin, empezaremos con PKZIP. No hay problema en conseguirlo en CD-ROM, y seguro que si miramos lo encontraremos en alguno de los que he traído de casa o en el que hemos comprado. Pero seguro que ganamos tiempo si en lugar de ponernos a mirar directorios o las revistas que vienen con los CD-ROM, que incluyen una descripción de sus programas, instalamos el que llevo siempre en los discos de emergencia de mi ordenador. La versión no registrada es totalmente funcional. Aquí está, en este disquete. ¿Ves el directorio de A:?. Pues es el fichero PKZ204G.EXE. Aunque la fecha es 1 de febrero de 1.993, casi no ha cambiado desde entonces.

Para instalarlo, como es un programa DOS, no tenemos más que crear un directorio y copiar allí los ficheros. Vamos a ver, creamos el directorio \PKZ, por ejemplo, y copiamos allí A:\PKZ204G.EXE. Ya está. Te dejo una copia del fichero original por si necesitas pasárselo a alguien, o quieres copiarlo a tu vez a un disquete de emergencia. En realidad, ese no es el programa a usar. Vas a ver una cosa interesante. Fíjate, me meto en el directorio que acabo de crear y lanzo el programa. No se si te ha dado tiempo de leer los mensajes. La cuestión es que el tal programa no es más que un fichero ejecutable, por eso tiene la extensión .EXE, que contiene a la vez el programita de descompresión y los ficheros comprimidos que componen en realidad PKZIP. Como puedes ver ahora en el directorio \PKZ, tenemos ya varios ficheros ejecutables, aparte del PKZ204G.EXE anterior. Los principales son PKZIP.EXE, que se emplea para crear los ficheros comprimidos, y PKUNZIP.EXE, que los descomprime. Pero hay más. ¿Ves ese README.DOC?. Contiene las instrucciones de instalación y algunas notas. Muchos programas tienen ficheros con nombres y contenidos parecidos, y conviene leerlos antes de instalar. Hay dos más que merece la pena comentar. Uno es MANUAL.DOC. Como su nombre indica, es el manual del programa. Lo puedes imprimir si quieres, pero son bastantes páginas. También lo puedes leer directamente en la pantalla. Y el otro es ZIP2EXE.EXE. Cuando crees un fichero comprimido con PKZIP.EXE, el programa le pondrá la extensión .ZIP si no le dices otra cosa. Con eso sabrás que es un fichero que contiene datos comprimidos. Pues bien, ZIP2EXE.EXE nos deja convertir un fichero comprimido .ZIP, que deberíamos descomprimir cuando hiciera falta con PKUNZIP.EXE, en un autoejecutable que se descomprima solo. PKZ204G.EXE se creó así. El nombre de ZIP2EXE viene de una especie de abreviatura que les gusta mucho a los

---

14 Por si se está Vd. distrayendo, debemos hacer notar que lo anterior es una expresión coloquial. También los compresores tienen límites, y no pueden incorporar infinitos ficheros a un fichero comprimido. Algunos miles, sí.

## **CAPÍTULO 8. PeCés: Programas**

americanos. Hay que leerlo 'ZIP TO EXE', es decir 'ZIP A EXE' y como lo del 'TO' son dos letras, lo sustituyen por un '2', ni más ni menos que 'two' en inglés, que suena parecido. Como ves, PKZIP lo forman en realidad varios programas sueltos que se encargan de todo lo necesario. Para aclararte con su uso tienes el manual o los parámetros de ayuda, el famoso 'h', que también funciona con ellos. De hecho, si ejecutas PKZIP sin ningún parámetro te saldrán las ayudas directamente. Lo dejaremos por ahora así, ya que aunque no es muy fácil de usar, después lo arreglaremos un poco en Windows. Para acabar la instalación no hace falta más que añadir al PATH de DOS el directorio de PKZIP, para que se pueda encontrar desde cualquier sitio. Conviene hacerlo porque se emplea bastante. 'EDIT AUTOEXEC.BAT', modificamos el PATH, y la próxima vez que arranques el ordenador podrás lanzar PKZIP sin preocuparte de estar en \PKZ.

Por si te gusta más, y porque también hace falta a veces, vamos a montar también el otro compresor de ficheros importante. Se llama ARJ y funciona de manera algo diferente. Éste no lo llevo encima porque empleo más PKZIP, o sea que habrá que buscarlo en los CD-ROM. Aquí está la versión 2.50<sup>15</sup>. Vamos a instalarla...". Cerremos temporalmente las comillas por no repetir una descripción que resultaría básicamente idéntica a la realizada para PKZIP, aunque ahora con programas y directorios con nombres diferentes. Retomemos el monólogo de nuestro experto una vez terminada la instalación. Volvamos a abrir comillas.

"Pues ya está. Como ves, una vez que se ha instalado un programa, se ha aprendido a instalarlos todos. Los del mismo tipo, al menos. ¡Ja, ja!. Ya ves que volvemos a tener un fichero README.DOC. El manual ahora se llama ARJ.DOC, y podemos aplicarle lo que dijimos para el de PKZIP. Visto PKZIP hay poco que comentar sobre ARJ. La única diferencia es que mientras PKZIP emplea varios programas para las distintas cosas que hay que hacer con los ficheros comprimidos, ARJ es más bien un único programa que las hace todas según le indiques con los parámetros correspondientes. En fin, échale un ojo al manual cuando te haga falta. Es importante que te familiarices con los compresores de ficheros, al menos con uno de ellos, porque se emplean para muchas cosas. Un problema grave del DOS era la realización de copias de seguridad, algo que de verdad hace falta y que tendríamos que comentar con más detalle. Cuando quieras almacenar una copia de todo tu disco C: con todos sus ficheros, o tal vez del directorio de Windows, tendrás que manejar, aunque los comprimas, varias decenas de megabytes. Y eso, en principio, no cabe más que en un disco duro. Si tuvieras una unidad de cinta podrías hacer la copia de seguridad allí, pero no la tienes. Si lo quieres dejar en disquetes, ahora mismo lo mejor es usar PKZIP o ARJ. Son capaces de guardar un fichero de muchos megabytes en varios disquetes, almacenándolo por trozos que luego ellos se encargan de volver a montar. Quédate con la copia porque muchas veces viene muy bien.

---

15 Vamos a poner la referencia de algún lugar en que se puedan encontrar los programas que se van a ir instalando hipotéticamente. Es más porque se vea que no se está inventando nada que por proporcionar una forma de conseguirlos. Las revistas dedicadas a la distribución de shareware, que son las que citaremos principalmente, suelen incluirlos y renovarlos con regularidad. No son referencias únicas, por supuesto, y es posible conseguir la mayor parte de los programas que vamos a manejar en otras publicaciones o a través de las otras fuentes de shareware que ya indicamos. Empecemos. El ARJ 2.50 apareció en "Hot Shareware", número 6, marzo de 1.996.

Vamos a instalar otras herramientas para DOS que también hacen falta. En primer lugar algún duplicador de disquetes. El DISKCOPY de DOS funciona, pero los programas específicos para duplicar disquetes son mucho más rápidos y reconocen más formatos, además de dejarte hacer imágenes de disquetes a ficheros y muchas cosas más. ¡Hombre!, aquí tenemos un par bastante decente. Disk Copy Fast versión 4.9 y Qcopy v5.1<sup>16</sup>. Vamos a instalar el segundo, y si quieres, puedes ir entrenándote por tu cuenta montando el primero cuando yo me vaya”. Cerremos comillas temporalmente para evitar otra tediosa descripción, y volvámoslas a abrir cuando haya algo interesante que oír.

“También te harán falta algunos programas de diagnóstico, para ver que tal tienes la memoria, los discos, o los IRQs y demás. Busquemos alguno en los CD-ROM. Este no está mal. Es el System Info Checker v2.42<sup>17</sup>. Vamos a ver si encontramos algún otro, porque no todos suelen detectar las mismas cosas. Bien, aquí tenemos algo que puede valer, el PC Config v8.02<sup>18</sup>. Montaremos los dos y, si te hace falta, aún tienes otro que seguro que está en tu ordenador. En el directorio de Windows hay un MSD.EXE, abreviatura de Microsoft Diagnostics, que siempre puedes usar. No es muy frecuente que se sepa, pero es parte de Windows. Son programas DOS, sin embargo, y conviene ejecutarlos desde allí. Vamos con ello”. Esperemos un poquito de nuevo, y volvamos a continuación con nuestro experto.

“Los antivirus<sup>19</sup> son otra de esas cosas que conviene tener a mano. Hay muchos en los CD-ROM y se actualizan con frecuencia. Al igual que con los programas de diagnóstico, no todos detectan los mismos y conviene montar más de uno. Quizá el más clásico es el de Mcafee, que se llama SCAN, o Viruscan, como prefieras. Hay versiones para DOS y Windows, pero por ahora montaremos la de DOS. Luego nos ocuparemos de Windows. A ver si lo encontramos por algún sitio. Aquí está. Tenemos el Scan v2.27 en castellano y el Virus Shield v2.27<sup>20</sup> correspondiente. El primero se usa para buscar virus y si es posible eliminarlos, aunque sobre eso habría bastante que decir. El segundo, en general cualquier ‘shield’, que es ‘escudo’ en inglés, sirve para proteger el ordenador de posibles infecciones. Te los voy a dejar instalados y si quieres activas el shield. Otro que resulta bastante popular es el Thunderbyte. Aquí lo tenemos, versión 6.51<sup>21</sup>. Vamos a montarlo también”. Introduzcamos una vez más una pequeña pausa hasta que se cambie de tema.

“Bueno, bueno, esto marcha. Casi hemos acabado con DOS. Te voy a montar un par de programas adicionales, en plan lujo, por si te quieres meter de verdad en cosas serias. Mira, a veces hace falta enredar con las configuraciones y el mecanismo de arranque del ordenador. Hay programas que permiten añadirle flexibilidad al proceso. Pondremos uno. Y otra cosa

---

16 Disk Copy Fast v4.9 y Qcopy v5.1 aparecieron en “Hot Shareware”, 3, diciembre 1.995.

17 “Hot Shareware”, 6, marzo 1.996.

18 “Hot Shareware”, 3, diciembre 1.995.

19 Dejaremos el tema de los virus para el próximo capítulo.

20 Mcafee Virus Scan 2.27 y Virus Shield 2.27 en “Informática Fácil Multimedia”, 34, febrero 1.996.

21 “Hot Shareware”, 4, enero 1.996.

## **CAPÍTULO 8. PeCés: Programas**

es el COMMAND.COM, que tal vez te hayas enterado ya de que resulta un poco limitadito el pobre. Montaremos también un intérprete de órdenes mejorado. He visto por aquí, en las vueltas que hemos dado por los CD-ROM, los programas correspondientes. Para lo primero puede valer éste, el BootIt<sup>22</sup>, y para lo segundo, la mejor alternativa es desde luego 4DOS. Te los voy a instalar. Aquí tenemos el 4DOS v5.5<sup>23</sup>. Por cierto, programas como estos suelen tener un programa de instalación propio, no hace falta montarlos a mano. ¿Lo ves?. Simplemente ejecutamos INSTALL.EXE en este caso, aunque a veces puede ser un .BAT con un nombre parecido, vamos contestando a las preguntas, y ya está. Hay muchas cosas interesantes que decir sobre 4DOS, pero quizá ahora mismo lo que más te haga falta es una orden que incluye y que viene de maravilla para leer manuales en forma de ficheros tipo texto, como los de ARJ, PKZIP o el del mismo 4DOS. Si intentas leerlos desde DOS con un 'TYPE el\_fichero\_que\_sea | MORE' acaba siendo una lata porque no puedes ir nada más que hacia delante. Y para imprimir hay que usar 'PRINT', otra orden. 4DOS incluye una orden nueva, que se llama 'LIST', y que te permite ir adelante, atrás, imprimir, buscar palabras, y otro montón de cosas. Es lo mejor para apañarse con los manuales. Aunque si no te quieres meter en 4DOS también puedes trampear un poco usando 'EDIT' para leer los ficheros. No los modifiques y ya está. No vas a modificarlos, pero 'EDIT' te deja moverte con más comodidad que el 'TYPE | MORE'. ¡Hay que echarle imaginación al asunto, caramba!.

Ya vale de DOS. Ha llegado la hora de meternos con Windows, pues supongo que querrás trabajar ante todo desde allí. Tienes un ordenador que no está nada mal y te lo puedes permitir. Entremos en Windows. 'WIN' y retorno de carro, allá vamos. Por no perder el hilo, quizá lo mejor sería que empezáramos completando los compresores de ficheros y los antivirus con las versiones para Windows. Primero los compresores. Ya te he dicho antes que PKZIP o ARJ no eran particularmente sencillos de manejar. Tampoco es que sean complicados, pero tienen muchos parámetros y al principio te puedes liar. La cosa se puede arreglar desde Windows, incorporándoles las ventajas de una interfase de usuario más sofisticada. Pero hace falta que sepas lo que hacen, eh, porque si no tampoco te vas a aclarar. Vamos a instalar Winzip v6.0a<sup>24</sup>. Más que un compresor en sí mismo, aunque sepa él solito como apañárselas con ficheros tipo .ZIP, es un interfase de usuario en Windows para manejar desde allí aquellos compresores que se hayan instalado en DOS. Una vez configurado, podrás usarlo en lugar de PKZIP, o ARJ, o hasta LHA o ARC si los tuvieras, que no es tu caso. Puede incluso trabajar con formatos comprimidos mucho menos frecuentes, como los de instalación de programas Windows o algunos del mundo Unix. ¿Que para qué tanto?. Internet tiene la culpa. Si te metes en ella, te encontraras con ficheros comprimidos desde Unix, seguro, y hay que poder sacarles partido. En resumen, que hay que montar Winzip 6.0a.

Instalar programas en Windows es tan fácil como hacerlo en DOS o quizá más, aunque suelen pasar más cosas. ¿Ves?. Buscamos en el conjunto de ficheros que constituyen el programa a instalar alguno que se llame INSTALL.EXE, o INSTALAR.EXE o SETUP.EXE

---

22 "Hot Shareware", 3, diciembre 1.995.

23 "Hot Shareware", 1, octubre 1.995.

24 "Hot Shareware", 7, abril 1.996.

y lo ejecutamos desde el 'Administrador de programas'. A partir de ahí basta con seguir las instrucciones y contestar a las preguntas que vaya haciendo. Sólo es necesario saber qué diablos nos pregunta y como queremos contestarlo. No siempre es fácil para uno que acaba de aterrizar, lo reconozco. Hay que saber de qué se trata. Te preguntará en qué directorio quieres dejarlo, si quieres que cree su propio grupo de programas, y cosas así<sup>25</sup>. Winzip crea un grupo propio pero luego, si te apetece, lo puedes cambiar de sitio y pasarlo a 'Aplicaciones', por decir algo. Bueno, ya está. Ahora hay que acabar el trabajo. Nunca basta con instalar. A continuación hay que configurar, y para este programa es fundamental ya que al menos hay que decirle donde tiene PKZIP, ARJ, y compañía. Veamos, le damos al icono un par de clics y aquí lo tenemos. Elegimos 'Options' en el menú principal, y luego 'Program locations', o sea 'Opciones' y 'Ubicación de programas'. Sí, está en inglés, pero no hay nada similar en castellano. Le decimos donde puede encontrar ARJ y hasta donde tiene un antivirus por si queremos comprobar los ficheros que vamos a descomprimir, no sea que estén infectados. Eso es lo principal que hay que configurar en este caso. Como ves, también puedes cambiar los tipos de letra que emplea y otro montón de cosas. Lo pones a tu gusto cuando puedas.

Sigamos. Si te da un poco de miedo lo de las instalaciones recuerda que muy probablemente vas a tener siempre algún fichero con instrucciones. Léelo y ya está. Además, muchos de los CD-ROM de las revistas vienen con un programa general que te permite ver o probar los programas que hay en él y hasta instalarlos directamente, si quieres. De todos modos no siempre funcionan, y no está de más que seas capaz de hacerlo por tu cuenta. En cuanto al antivirus para Windows, aquí tenemos la versión correspondiente del Viruscan 2.27. Es el Winscan v2.27<sup>26</sup>. No hay mucho que comentar sobre él que no hayamos dicho antes. Pero sí que deberías saber que tanto a los antivirus como a los compresores, entre otros, a veces se les atraganta Windows. Si tienes problemas con ellos, inténtalo desde DOS". Tomémonos un pequeño respiro mientras se completa la instalación de Winscan v2.27 y volvamos a abrir comillas.

"Un poco aburrido te veo con lo de las utilidades. No creas que no sirven para nada. Son, sin exagerar lo más mínimo, fundamentales. Sin ellas, tanto Windows como DOS quedan ahora mismo bastante cojos. Pero en fin, aunque tenemos todavía algunas más que montar, las dejaremos para luego. Como parece que te veo un poco ansioso, vamos a pasar a los programas de aplicación, esos que se emplean para hacer cosas de verdad, y luego volveremos a pulir Windows con alguna utilidad más. Lo primero y principal son los editores y los procesadores de texto. No son lo mismo. Un editor sirve para crear o modificar un fichero de texto de esos que ante todo se usan en el mismo ordenador. Es decir para modificar un CONFIG.SYS o un WIN.INI. Más que buscar que queden bien en papel al imprimirlos y se puedan usar miles de tipos de letras con dignidad o se puedan escribir libros con sus índices y todo, se trata de manipular muchos y quizá muy grandes ficheros de texto sin ninguna estructura adicional. No hay capítulos, ni encabezamientos, ni notas a pie de página. Texto en ASCII o en ANSI y poco más. Son una herramienta fundamental pues sirven para meterles

---

25 Sí, se están repitiendo cosas que hemos dicho antes. Pero no está mal que se recuerden, y además estamos transcribiendo una conversación en la que no deberíamos intervenir en la medida de lo posible.

26 Mcafee Winscan v2.27 en "Informática Fácil Multimedia", 34, febrero 1.996.

## **CAPÍTULO 8. PeCés: Programas**

mano a esos ficheros que aparecen en cualquier sitio, esos con extensión .TXT (de texto) o .ASC (abreviatura de ASCII) o todos los .BAT, .SYS, .INI y demás que contienen texto y punto. Con ellos estamos a medio camino entre la utilidad y el programa de aplicación<sup>27</sup>. Windows incluye uno, el 'Bloc de notas'. Si lo prefieres en inglés, 'Notepad', pero es bastante limitado lo llames como lo llames. No es capaz de abrir ficheros de más de 64 Kilobytes, entre otras cosas. Por suerte hay miles de buena calidad. Vamos a instalar por ejemplo éste, el UltraEdit v3<sup>28</sup>. Una cosa que conviene saber sobre ellos es que en bastantes casos te dejan abrir no sólo ficheros de texto, sino cualquier fichero, hasta un ejecutable .EXE. Por supuesto que entonces no verás nada con sentido, ya que el editor siempre considera que es un texto y como tal intenta mostrártelo, eres tú el que debería saber en qué idioma puede estar. Aún así, a veces viene muy bien poder echar un vistazo dentro de cualquier fichero. Quizá no sepas con qué programa se ha creado un documento y no puedas trabajar con él. En casos como esos, un buen editor te va a dejar al menos que veas si el texto está dentro en forma legible y, si es así, vas a poder rescatar los datos.

Por contra, un procesador de textos no tiene como objetivo principal el manipular ficheros de ese tipo, sino el de crear y manipular documentos. Es decir textos complicados, con un formato, secciones, capítulos, y cosas parecidas. Su propósito es ante todo acabar con una copia en papel casi perfecta, y por eso dejan trabajar con distintos tipos de letra, fijar márgenes, poner encabezamientos, numerar páginas, crear tablas, o incorporar gráficos. En Windows suelen ser WYSIWYG, de modo que puedas ver la cosa casi como va a quedar sobre papel, y los buenos de verdad llegan a ser capaces de revisar la ortografía o incluso la gramática del documento en cuestión. Luego nos meteremos con ellos, pero por ahora vamos a instalar uno de shareware. Cualquiera servirá, por ejemplo Breeze<sup>29</sup>. Si usas un procesador de textos, ten en cuenta que debe ser él el que se encargue de montar el documento. No empieces a separar cosas con espacios y a pulsar retorno de carro al final de cada línea. No es una máquina de escribir. Usa tabuladores, o cambia los márgenes, o inserta sangrías, y pulsa retorno de carro sólo cuando quieras de verdad que pase a una nueva línea o un nuevo párrafo. Así el programa será capaz de digerir los cambios de tipos de letra o de márgenes que hagas luego sin descomponer el documento. Los párrafos, por cierto, suelen separarse con dos retornos de carro. En fin, la práctica hace maestros, que se dice. Léete las ayudas. Los procesadores de textos son programas mucho más complejos que los editores y hay que tener en cuenta como trabajan para poderles sacar partido.

Con un buen editor y un buen procesador, ya tienes programas para trabajar en el ordenador de verdad, pues se emplea muchísimo como sustituto de las máquinas de escribir. Casi seguro que será lo primero que hagas de provecho.

Con una hoja de cálculo para poder hacer cuentas, una base de datos, y algún programa para gráficos, tendremos casi todo lo necesario. Las hojas de cálculo las dejaremos por ahora.

---

27 En ocasiones se llaman "editores de programas", simplemente porque pueden servir para crear la versión en "código fuente", la que luego hay que compilar, de un programa.

28 "Hot Shareware", 3, diciembre 1.995. Otras alternativas pueden ser MEDIT v1.70, "Hot Shareware", 7, abril 1.996, o GMUtan v1.03, "Hot Shareware", 4, enero 1.996, entre muchos otros.

29 "Hot Shareware", 3, diciembre 1.995.



Desgraciadamente no hay demasiadas de shareware para Windows. Para DOS había una especie de clónico de Lotus 123 que se llamaba ‘As Easy As’, pero prefiero resolver el problema de otra forma que veremos luego. Vamos con las bases de datos. Para alguien que empiece, en Windows, no hay como FileMaker Pro v2.1<sup>30</sup>. Es un programa que se desarrolló para el Apple Mac por una compañía, Claris, que en tiempos estuvo directamente vinculada a Apple. Quizá aún lo esté. Pero bueno, la cuestión es que la versión Windows es idéntica y hace con notable sencillez casi todo lo que puede hacerse con una buena base de datos, que viene a ser más o menos manipular ficheros de datos estructurados en fichas, ya sean recetas de cocina, un inventario de libros o discos, o cualquier cosa que se te ocurra. De todos los programas de aplicación, las bases de datos son quizá los más abiertos, es decir los que hacen algo menos concreto pero con más posibilidades de servir para un montón de trabajos distintos.

Si quieres meterte en FileMaker Pro hay libros que explican su uso<sup>31</sup>. Son para la versión Mac pero, como te he dicho, ambas son iguales. Una vez que tienes un GUI, podemos decir que los programas no se diferencian en casi nada. No te engañes, de todas formas. El tema de las bases de datos da para mucho más que lo dicho. Antes de que pienses en informatizar con FileMaker Pro todos los videos que tienes en las estanterías, un montón a lo que parece, ten en cuenta que es una demo que no te va a dejar trabajar con más de 50 fichas. Cosas de las demos. Si te gusta, como siempre, te lo puedes comprar en su momento. No es demasiado caro y hay una versión en castellano<sup>32</sup>. Si te interesa de verdad el tema de las bases de datos, me lo dices. Hay muchas más, y en casa debo tener demos o versiones completas de shareware de algunas de ellas. Te he puesto ésta porque creo que es la más fácil de usar y tiene algunas ventajas muy interesantes para los novatos. Y también para los expertos, porque no decirlo.

Vamos con los gráficos. A veces querrás que el ordenador los cree solito, por ejemplo para obtener una representación gráfica de una serie de números. Esto se le da bastante bien. En inglés, este tipo de gráficos se llaman ‘charts’. Cuando veas un programa que se llame ‘chart lo que sea’ ya sabes para qué sirve. Hubo un tiempo en que eran bastante populares, pero las hojas de cálculo han ido mejorando tanto en este aspecto que puede decirse que los han sustituido. Como hemos dicho, las dejaremos estar por ahora.

Cuando llega el momento de pintar, puede hacerse de muchas formas. De pintar, lo que se dice pintar, es decir algo parecido a lo que hacemos en un papel con lápices de colores, pinceles, y aerógrafos, se encargan unos programas que en general se llaman ‘Paint’. Con ellos tenemos un área de trabajo dividida en puntos y podemos colorearlos como mejor nos parezca. Aunque podemos trazar líneas, una vez hechas no existen como tales. Pasan a ser puntos, como todo lo demás. Se puede mover el ratón para intentar dibujar a mano alzada, pero no creas que resulta fácil. Y más si eres torpe dibujando, como yo<sup>33</sup>.

---

30 “Hot Shareware”, 3, diciembre 1.995.

31 “Bienvenido a FileMaker Pro 2.0”, de Pablo Pérez de Ayala. Editorial Anaya Multimedia, 1.993, ISBN 84-7614-529-2, dentro de la colección “Guías Macintosh”.

32 En los últimos tiempos venía a costar alrededor de 15.000 pesetas. Hay una versión posterior, la 3.0, para Windows 95 y puede que también para Windows 3.1 + Win32s.

33 Una vez más viene a cuento recordar que debe elegirse el periférico de entrada adecuado para

## **CAPÍTULO 8. PeCés: Programas**

Otra forma de crear gráficos es lo que podríamos llamar dibujar, y lo hacen otros programas diferentes que en general se llaman 'Draw'. En ellos ya no tenemos puntos sueltos, sino una imagen de un papel en el que definimos líneas, círculos, polígonos, áreas, y demás. En este tipo de programas, los distintos elementos que componen el gráfico siguen existiendo como tales una vez creados, y se pueden seleccionar y modificar con facilidad, por ejemplo para cambiarlos de tamaño, o rotarlos, o un montón de posibilidades más. El ordenador se apaña mejor con este tipo de gráficos, que se denominan en general 'vectoriales' que con los que se crean con programas 'Paint', es decir con gráficos tipo 'bitmap'<sup>34</sup>. No es hora de meternos con formatos de gráficos, o sea que lo vamos a dejar estar.

Y la última gran categoría de programas de dibujo se dedica a algo ligeramente diferente: el dibujo técnico. Si lo que te gusta es hacer planos, con acotaciones y una precisión a prueba de bomba, lo tuyo son los programas de CAD. La palabreja es la abreviatura de 'Computer Aided Design', que podríamos traducir por 'Diseño asistido por ordenador', o si lo quieres de forma más simple, programas de delineación. Los programas de CAD son la base sobre la que se montan sistemas de fabricación computerizados que en tiempos se denominaban CAM o CIM, pero no creo que vayas a montarte uno en casa, ¡ja, ja!, o sea que nos quedaremos en el CAD<sup>35</sup>. Por si te apetece meterlos en alguno de los tipos anteriores, baste saber que crean gráficos de tipo 'vectorial'.

No es difícil encontrar programas de shareware para dibujar de cualquiera de las formas que hemos citado. Un muy buen programa tipo 'Paint' es Paint Shop Pro v3.0<sup>36</sup>. Con los gráficos hay un problema bastante gordo con los formatos de los ficheros. Simplemente existen demasiados, y no todos son equivalentes. Una diferencia a considerar es si conservan los elementos del dibujo, es decir si incluyen algún rasgo de eso que hemos llamado 'dibujo vectorial', o si se limitan a guardar los puntos sin más. En el primer caso se habla a veces de 'metaficheros' de gráficos, pero la cosa resulta un pelín pedante. Un programa 'Paint' competente debe ser capaz de desenvolverse al menos con los diez o doce formatos más corrientes, y Paint Shop Pro cumple sobradamente en este y otros aspectos. En resumen, con él vas a poder crear ficheros con gráficos tipo 'bitmap' perfectamente utilizables en cualquier sitio.

---

cada trabajo. Si quiere dibujar a mano alzada del mismo modo que lo haría con un lápiz y un papel, es preciso usar su equivalente informático. Tal cosa existe y atiende al nombre de "tableta gráfica". En el glosario encontrará una descripción algo más completa.

34 Hace algunos capítulos, en concreto al hablar de modo texto y modo gráfico y de las tarjetas gráficas para PeCés, explicamos un poco que eran eso de los 'bitmaps'. En consecuencia, puede que desee darse una vuelta de nuevo por los capítulos 4 y 6. También encontrará una explicación en el glosario y nos volveremos a meter con ellos dentro de poco en este mismo capítulo.

35 CAM significa "Computer Aided Manufacturing" y CIM "Computer Integrated Manufacturing". En ambos casos se trata de aproximarse a lo que hacía la "Davidson & Prokofiev No. 6 que ya citamos en otra nota al pie hace muchos capítulos, aunque con menos gracia y resultados algo menos concluyentes.

36 "Hot Shareware", 1, octubre 1.995.

El tema de los 'Draw' es más difícil resolverlo con propiedad, y el motivo no es otro que, de nuevo, los formatos de los ficheros que se crean. El estándar por excelencia para PeCés es un programa que se llama Corel Draw. Por desgracia almacena los gráficos en su propio formato, con la extensión .CDR. No es fácil encontrar un programa de shareware que lea y escriba en ese formato, y, en consecuencia, lo que hagas al dibujar de esta forma con un programa que no sea Corel Draw no será fácil de pasar de un ordenador a otro. Tu trabajo no te servirá más que en casa, o deberás recurrir a exportarlo en alguno de los formatos universales de tipo más o menos 'bitmap' que hemos citado para los programas 'Paint'. Como te interesa en este caso no perder los elementos que hayas definido, es mejor recurrir a exportaciones en formatos de tipo 'metagráficos', y perdón por emplear de nuevo la palabreja. En la ayuda de Paint Shop Pro, magnífica por cierto, encontrarás datos sobre las características de los distintos formatos, y eso debería permitirte elegir el más adecuado en función de lo que quieras hacer.

Por suerte, tenemos una versión enterita de Corel Draw a nuestra disposición. Es la 3.0<sup>37</sup>, quizá un poco vieja, pero fue la primera completa de verdad, o sea que no es para hacerle ascos. Con esta versión, Corel Draw pasó a ser más un conjunto de programas para crear cualquier tipo de gráficos que un único programa de dibujo, aunque lo metió todo bajo el nombre de su programa principal. Con él podrás crear 'charts' con Corel Chart, gráficos 'bitmap' con Corel PhotoPaint, y por supuesto dibujos con Corel Draw. Incluye muchas más cosas. Tipos de letra, ejemplos, y hasta una utilidad que se llama Corel Trace para intentar pasar los 'bitmaps' a dibujos vectoriales, pero no confíes demasiado en que ese tipo de cosas funcionen bien. En resumen, hay que montarlo obligatoriamente. Sólo con él tienes cubierto un 80% de todo lo que hace falta para moverse con soltura dentro de los gráficos del PeCé. Por supuesto que hay versiones posteriores. Tengo por ahí algún demo de la última de ellas para Windows 3.1, la 5.0<sup>38</sup>, que puedes pedirme si te interesa aunque no conozco bien sus límites. Hace más o menos lo mismo que la 3.0 pero corregido y aumentado.

Con los programas de CAD estamos en una situación un poco a caballo entre los 'Paint' y los 'Draw'. Hay un programa por excelencia, el archifamoso Auto CAD, que de nuevo crea ficheros con un formato propio. Tienen la extensión .DWG. Pero bastantes programas de CAD pueden al menos leer y escribir en otro formato, también típico de AutoCAD. Se emplea justamente para intercambiar los dibujos entre programas diferentes, o sea para exportar e importar, y usa la extensión .DXF. Por lo tanto, los programas de shareware que son capaces de exportar e importar ficheros .DXF son lo suficientemente universales como para resultar útiles. Vamos a instalar uno cualquiera de los muchos que hay, pero nos aseguraremos de que pueda trabajar con los formatos adecuados. Éste, que se llama Draft<sup>39</sup>, dice expresamente que soporta PCX, WPG y DXF. PCX es un clásico para ficheros 'bitmap', DXF ya lo hemos presentado, y WPG es el formato que emplean los gráficos de un procesador de textos bastante universal que se llama WordPerfect. ¿Que para qué te vale si no lo tienes instalado?. Pronto nos ocuparemos de eso, impaciente.

---

37 "PC Magazine", 93, junio 1.996.

38 Corel Draw v5.0 Working Model, en "Corel 5", Biblioteca Profesional Windows nº 20, Tower Communications S.R.L., 1.994, ISBN 84-88432-37-2.

39 "Hot Shareware", 7, abril 1.996.

## **CAPÍTULO 8. PeCés: Programas**

Tal vez te sorprenda ver que un programa de gráficos vectoriales sea capaz de exportar ficheros 'bitmap'. No tiene nada de raro. De hecho, casi todos los programas de dibujo vectorial lo hacen, igual que pueden incorporar en los gráficos que con ellos se construyen imágenes en formatos 'bitmap'. Estrictamente hablando, los 'metaficheros' de gráficos son aquellos que pueden incluir tanto imágenes 'bitmap' como elementos de tipo vectorial. El propio Windows emplea para intercambiar gráficos entre programas un formato de ese tipo, que usa la extensión .WMF. La abreviatura no es otra cosa que 'Windows Meta File'. Si no te gusta el Draft, hay más programas de CAD en los CD-ROM. Por ejemplo, por ahí he visto a Delta CAD v1.1 y CAD Draw<sup>40</sup>. Puedes acabar usando el que más te guste, como siempre". Hagamos una nueva pausa, ningún motivo particular. Simplemente, tomémonos un respiro tras tanto gráfico.

"Recapitulando, tenemos una colección más que razonable de utilidades para DOS, los equivalentes Windows de las más importantes, y editores, procesadores de textos, una base de datos, y programas para hacer gráficos de casi cualquier tipo desde Windows. Por si te parece que aún no tienes bastante, vamos a completar la faena con un auténtico peso pesado y los complementos de Windows que prometimos hace rato.

¿Te acuerdas de que uno de los programas de dibujo tipo CAD podía exportar en formatos propios de WordPerfect?. Pues, para que te sirva de algo el tal formato, vamos a instalarlo. No sólo WordPerfect, sino también la hoja de cálculo para Windows que nos falta para tener realmente de todo y algunas cosas más. Aquí tengo un CD-ROM con la versión de demostración de un programa que se llama PerfectOffice v3.0<sup>41</sup> y que incluye WordPerfect 6.1; una hoja de cálculo más que buena llamada Quattro Pro v6.0; otro programa para hacer gráficos, en este caso orientado a la realización de ilustraciones para resúmenes o presentaciones en público y que en justa consecuencia se llama Presentations, en concreto la versión 3.0; una agenda que no está tampoco nada mal; y alguna otra cosilla también bastante interesante que nos llevará como quien no quiere la cosa de nuevo al terreno de esas utilidades que dejamos de lado hace un rato.

Lo que vamos a instalar es en realidad una demo de lo que ha terminado siendo el equivalente para Windows de los antiguos programas integrados: los 'paquetes Office'. En realidad se trata de recopilaciones de programas que cubren los campos de trabajo más frecuentes y que se venden juntos en un paquete único y a precio normalmente ventajoso. No entraremos a analizar el tema, que quizá daría bastante de sí, pero la cuestión es que la demo de PerfectOffice es una versión temporal muy completa que al menos te va a dejar hacerte una idea exacta de como son los programas comerciales para Windows. Caduca a los treinta días. Si de verdad te interesa, deberías comprarla.

Como ves, cuanto más gordo es el programa, y los paquetes Office son realmente descomunales, más tiempo cuesta hacer la instalación. Cincuenta o sesenta Megabytes no se pasan

---

40 Delta CAD v1.1, en "Hot Shareware", 6, marzo 1.996, y CAD Draw en "Hot Shareware", 1, octubre 1.995.

41 La demo en cuestión se llamaba exactamente "PerfectOffice 3.0 True Trial" y se distribuyó con un ejemplar del "PC Magazine". No recuerdo con exactitud la fecha a la que correspondía. Desde luego era en 1.995, quizá en abril o mayo.

al disco duro así como así. Y eso que los CD-ROM son una auténtica bendición. Instalar PerfectOffice o un programa equivalente, desde disquetes, puede significar andar a vueltas con veinte o treinta disquetes y perder cosa de una hora o quizá más. Eso si todo va bien. ¡Venga!, a instalarlo”. La advertencia de nuestro experto particular nos da pie para realizar una nueva pausa mientras se lleva a cabo tan titánica labor de instalación. Disfrutémosla pues será la última.

“Uno de los programas incluidos en PerfectOffice 3.0 se llama Envoy. En concreto, la versión que hemos instalado es la 1.0. Puede que te resulte un poco chocante si lo intentas usar sin saber para qué sirve. Con él, estamos de nuevo en el campo de las herramientas o utilidades, en este caso para Windows. Te lo explico. Tal vez sepas ya que Windows permite utilizar multitud de tipos de letra diferentes. De hecho, en muchos de los CD-ROM que estamos manejando hay colecciones de lo más variado. Eso, que a primera vista puede parecer una bendición y probablemente lo es, no deja de crear algunos problemas interesantes. Quizá lo entiendas con un ejemplo. Supongamos que creas, con algún procesador de textos, un documento que emplea un tipo Garamond. Tú lo tienes instalado en tu ordenador y puedes visualizarlo e imprimirlo con facilidad. ¿Pero qué pasa si le dejas el fichero correspondiente a alguien que no dispone de ese tipo en su Windows particular?. Lo normal es que el programa intente sustituirlo por otro que se le parezca en lo posible. Pero no siempre las sustituciones de tipos de letra tienen resultados aceptables. No es la primera vez que un texto escrito con un tipo no estándar aparece en cirílico, o en símbolos griegos o matemáticos, cuando se pasa a un ordenador con un Windows normalito. La forma en la que los tipos de letra acaban sobre papel depende incluso de la impresora que emplees. No es inmediato, en consecuencia, conseguir que un documento Windows en un tipo de letra concreto se visualice y se imprima correctamente en cualquier ordenador. A Dios gracias, se han desarrollado herramientas que permiten hacerlo. Empleando, eso sí, formatos particulares para almacenar los ficheros. Envoy es una de ellas. Tal como viene te permite crear, visualizar, e imprimir, ese tipo de ficheros, que se distinguen mediante la extensión .EVY. Para crearlos no hace falta nada más que imprimirlos a través del controlador correspondiente de impresora, que se llama, como no podía ser de otra forma ‘Envoy Driver’. Digamos que tienes una impresora virtual que crea ficheros .EVY cuando imprimes un documento Windows con ella.

No es el único programa que hace eso. Otro muy frecuente, tal vez el más difundido, es Acrobat, de una casa que se llama Adobe y que es la que creó PostScript, uno de los lenguajes de descripción de documentos más importantes. Vamos a instalarlo también, aunque en este caso se trata solo de un visualizador de documentos en su formato particular, que emplea la extensión .PDF. Algo así como ‘Portable Document File’, aunque no estoy muy seguro de sea exactamente eso. Si quieres crear tus ficheros .PDF, hay que comprar el producto completo. El visualizador lo tenemos aquí, como Acrobat 2.0<sup>42</sup>. Es importante tener tanto Envoy como Acrobat ya que hay cada vez más documentación para programas Windows que viene en esos formatos.

Casi hemos terminado. Puesto que estamos de vuelta en las utilidades, montaremos dos o tres también fundamentales y me iré con la música a otra parte, que ya está bien. Como has podido observar, y eso que aún no hemos acabado, instalar todo el software necesario

---

42 “Hot Shareware”, 4, enero 1.996.

## ***CAPÍTULO 8. PeCés: Programas***

para hacer lo más habitual con un ordenador lleva su tiempo. Cualquier Windows mínimamente serio debe poder ejecutar programas de los llamados '32 bits'. No entraremos en disquisiciones técnicas. Debería bastarte con saber que ese tipo de programas suele trabajar más deprisa y mejor que los equivalentes en '16 bits'. En realidad, todo se reduce a conseguir que Windows, que nació como un GUI para PeCés equipados por lo menos con una CPU 80286, aproveche en lo posible algunas de las características del 80386. Hace ya tiempo que el 80286 pasó a mejor vida. Microsoft tiene en dominio público las extensiones correspondientes y suelen salir en los CD-ROM o pueden traerse a través de Internet. Aquí tenemos una versión reciente que me pasaron tras haberla sacado de Internet. Viene en un fichero comprimido autoexpandible que se llama PW118.EXE, es la versión 1.30c, y vamos a instalarla.

Tras haber puesto a punto la maquinaria de Windows, ocupémonos de la carrocería. Los dos elementos que más se emplean son el 'Administrador de programas' y el 'Administrador de ficheros'. Aunque los originales de Windows pueden valer, es mejor que un GUI resulte tan cómodo como sea posible. Yo prefiero sustituirlos por otros, y de hecho Microsoft los ha modificado muchísimo en Windows 95. Dashboard es una alternativa razonable. Algo después de salir la última versión, la 3.0, la anterior salió en un CD-ROM<sup>43</sup>. Aquí la tenemos, y en castellano por si fuera poco. Vamos a instalarla lanzando, como siempre, el programa correspondiente desde el CD-ROM. ¿Qué es esto?. No va. Ya te dije antes que a veces las cosas no funcionan como se supone que deben hacerlo. En fin, no parece que haya ningún fichero estropeado. Probaremos a copiarlo al disco duro en un directorio propio y a instalarlo desde allí. Ahora sí. Como ves, hay que tener recursos para salir de lo inesperado.

En cuanto al 'Administrador de archivos', voy a dejarte instalado el que más me gusta. Es el de las PC Tools para Windows, un conjunto de utilidades bastante popular. Lo vi en una revista y me la compré solo por eso. Vamos con el FileManager de las PC Tools<sup>44</sup>. Por cierto, ya que estamos con administradores de archivos, has de saber que muchos de los más recientes permiten manejar con facilidad ficheros comprimidos. Te los muestran como un directorio más, quizá un poco particular, eso sí, pero te dejan meterles mano sin demasiados problemas. Lo hacen incluso los de DOS, tales como el archiconocido Norton Commander, que no se por que versión va en la actualidad. De todas maneras, es mi opinión personal, conviene ser capaz de usar los compresores y descompresores directamente, a ser posible desde DOS. Manías, quizá.

Pues ya está. Si quieres juegos, o lenguajes de programación, o cualquiera de las otras mil cosas que se pueden montar, tales como reproductores de CD de audio, programas para imprimir carátulas de cassetes, otros sistemas operativos, desinstaladores automáticos de programas en Windows, tablas de períodos, conversores de unidades, calculadoras, diseñadores de circuitos eléctricos<sup>45</sup>, y vamos a dejarlo ya, seguro que los encontrarás en las revistas

---

43 "PC Magazine", probablemente de febrero o marzo de 1.996.

44 "PC Today", diciembre 1.994.

45 Sería posible cubrir dignamente muchos de los temas citados sin excesivo esfuerzo. Pongamos algunos ejemplos. Juegos había un montón en el monográfico de "Informática Fácil Multimedia", 32, diciembre 1.995. Lenguajes de programación, otro montón, en el monográfico de "Informática Fácil Multimedia", 29, septiembre 1.995. No hay problema tampoco para conseguir otros sistemas operativos. OS/2 ha aparecido varias veces, siendo quizá la versión más

con CD-ROM que vas a ir comprando a partir de ahora. O me los pides, pues tal vez los tenga por casa.

¿Que podría dejar instaladas un par de cosas más?. Desde luego, hay que ver como es la gente. Les das el brazo y te cogen una pierna y si te descuidas también la otra. En fin, quizá te conformes con un reproductor de CD de audio y una utilidad para gestionar los tipos de letra en Windows. Palabra que es lo último que monto. A partir de aquí, tu ordenador es cosa tuya. Vayamos con el Bridge Timber CD Player v5.0<sup>46</sup> y el Font Spec Pro<sup>47</sup>.

Bueno, vale, yo me voy. Que lo disfrutes con salud. Una última advertencia. Recuerda que los programas de shareware debes registrarlos si de verdad los empleas. Los de dominio público son gratis. Y si alguna de las demos hace que te interese de verdad un programa, lo mejor es que lo compres. Muchas veces, sólo el coste de los manuales justifica el precio de adquisición. Llámame si tienes problemas. Si puedo te ayudaré, aunque no garantizo nada. Y si tienes algo que te parezca interesante, dímelo. Conviene estar enterado. ¡Hasta la próxima!”.

Acabaría aquí el hipotético paseo que comenzamos a describir hace rato. Nuestro experto se iría a su casa convencido de que, antes de que terminara la semana, escucharía al otro lado del teléfono la voz de su amigo preguntando por algún problema de los muchos que pueden surgir al intentar sacar partido a semejante montón de software. Lo inevitable suele tener la costumbre de ocurrir, y él había pasado por esto varias veces. Por nuestra parte, vamos a volver a ocuparnos enseguida de cosas algo más serias que la mera acumulación de software.

Terminaremos el apartado, una vez más, con algunas observaciones, en esta ocasión sobre el hipotético episodio que acabamos de narrar. En primer lugar hay que reiterar que el catálogo de software enumerado, aunque bastaría, eso sí, para completar un PeCé bastante operativo, no es exhaustivo. No puede serlo. Debería quedar claro que el número de programas de que se dispone para PeCés es gigantesco. Y en segundo lugar debemos asimismo reiterar que la selección del software de nuestro experto no refleja la realidad de los programas más populares. Se trataba de montar un PeCé con programas de shareware, demos, y programas de dominio público. Por ejemplo, a la hora de instalar un “paquete Office”, el de Microsoft es sin duda alguna el más difundido. Suele decirse que acapara del orden del 80% del mercado. Incluye programas tan populares como el procesador de textos Word, o la hoja de cálculo Excel. Que no se haya instalado una demo de “Microsoft Office” no quiere

---

completa hasta el momento una 2.1, con compilador de C++ y todo, que se incluyó en “Revista Profesional de Programadores”, 1, noviembre 1.994. Linux es algo cada vez más frecuente. Hubo un monográfico en “Informática Fácil Multimedia”, 31, noviembre 1.995, y más recientemente se incluyó la distribución “Slackware 3”, con kernels 1.2.13 y 1.3.20 (tal vez otro año entremos a ver que quiere decir esto exactamente) en el CD-Actual 2 del “PC Actual”, 73, marzo de 1.996. En cuanto a los demás, nos limitaremos a citar el que puede resultar más extraño. Podía encontrarse un programa para ayudar en el diseño de circuitos eléctricos, en concreto Win Circuit v4.0, en “Hot Shareware 7”, abril 1.996.

46 “Hot Shareware”, 3, diciembre 1.995.

47 “Informática Fácil Multimedia”, 34, febrero 1.996.

## ***CAPÍTULO 8. PeCés: Programas***

decir que no exista, o que “PerfectOffice” sea mejor. Quiere decir simplemente que nuestro experto no disponía de ella.

En cuanto a la legalidad del proceder de nuestro hipotético experto, tal vez alguna operación de las descritas no respete estrictamente los derechos de autor. Habría que leer todos los “LICENCE.TXT” o “README.TXT” incluidos con los programas. Pero, seamos sinceros, la realidad real es bastante menos escrupulosa. Noventa (o quizá más) de cada cien “expertos” amigos de alguien, hubieran copiado e instalado programas comerciales, de shareware, y de dominio público, sin realizar ningún tipo de distinción entre ellos. El mundo de los PeCés está abarrotado de “expertos” con no demasiados años, y los legalismos suelen tomarse en consideración tan sólo a edades tan avanzadas como la del autor.

Ya que estamos en este punto, aprovechemos para aclarar algunas cosas sobre la utilización del software que conviene retener. Usar programas implica un esfuerccillo adicional. Hay que instalarlos, aprender a usarlos, resolver problemas cuando se presenten, adaptarlos a nuevas necesidades, configurarlos... . El ejemplo debería permitirnos apreciar que, sólo la instalación, ya lleva un rato, y requiere ciertos conocimientos en principio no obvios. Y eso que, desde luego, los programas para PeCé en Windows se instalan ahora mismo con pasmosa facilidad. No siempre ha sido así, y por supuesto no es así todavía en otros sistemas operativos.

¿Quién se encarga del trabajo de instalar, configurar, y mantener en funcionamiento los programas?. En el software horizontal, el caso habitual entre usuarios particulares del PeCé, Vd. sabrá. Y en el vertical, típico de empresas o usuarios profesionales, Vd. pagará. Las empresas de servicios informáticos se encargan de poner su personal y sus recursos de programación a su disposición, y por supuesto de cobrar por ello. En principio Vd. pone las especificaciones y ellos presupuestan y facturan, tanto más cuanto mayores sean sus exigencias. Seguir por este camino nos llevaría a temas económicos de enjundia, y no conviene olvidar que estamos en un capítulo que debe contribuir a reconciliarnos con la informática y a descansar nuestra ya maltrecha mente, por lo que lo dejaremos para otro momento. En el siguiente apartado intentaremos racionalizar un poco el asunto de la selección del software para PeCés, aunque es bastante probable que lo que digamos resulte válido, como en muchas otras ocasiones, para cualquier ordenador.

### **Principios y criterios**

### ***De como elegir los muebles de LR-PeCé***

Si uno de los LAOs que todavía perseveren a estas alturas en la lectura de este libro intenta resolver su problema particular de selección de software para PeCé recurriendo al sencillo expediente de aprenderse de memoria la retahíla anterior de nombres y versiones para repetirla a la menor ocasión como si fuera el catecismo, apañado va. Lo que hace falta si se quiere uno desenvolverse en el proceloso mundo de un software siempre cambiante no es un catálogo de productos, por bueno que sea. Lo necesario son ideas, criterios, y principios de selección que podamos usar en cualquier momento presente o futuro para elegir el software adecuado. Recurriendo una vez más a la metáfora automovilística, algo parecido a lo que hacemos con los coches. Puestos a elegir entre un Citroën Xantia, un Opel Vectra, un Honda Accord y un Renault Laguna, casi todo el mundo sabe lo que tiene que mirar. El precio, por supuesto, pero también la velocidad, terminación, fiabilidad, red de servicio, imagen de marca, consumo, costes de reparación, y otros mil aspectos deben ser tenidos en cuenta y lo son de forma más o menos inconsciente. Algunos de estos parámetros son directamente



aplicables a los ordenadores y a su software, pero en el campo de la informática de consumo, e incluso en el de la supuestamente profesional, no siempre están claras las ideas más fundamentales.

Y, de nuevo, no es que haya que dedicarse a hacer integrales múltiples para intentar cuadrar el círculo. La respuesta a esta y otras cuestiones se halla una vez más en esos dos elementos que explican todo el funcionamiento y uso práctico del ordenador. Hay que poner orden en el catálogo de los programas posibles, reduciendo su número a unas pocas categorías en principio equivalentes. Hay que establecer algunos criterios para, dentro de ellas, elegir programas concretos. Y poco más. El análisis de algunos aspectos particulares del sistema de ficheros del PeCé y de su interfase de usuario debería ayudarnos también a conseguirlo. Intentaremos en este apartado progresar en esa dirección. Avanzando un rápido resumen de lo que viene a continuación, no para que no lo lea, sino para ayudarle a que no se pierda, digamos que el sistema de ficheros nos ayudará a clasificar el conjunto de los programas en unas pocas categorías, y la interfase de usuario y esas otras consideraciones menores tales como el precio, disponibilidad, y cualquier cosa que se le ocurra, a decidir una vez dentro de ellas.

De hecho, lo poco que podamos aprender en lo que resta de capítulo, pues poco será lo que expongamos, debería servirnos también para afrontar otros problemas aparentemente distintos del que nos ocupa de forma inmediata, que no es otro que el de la selección del software. Por ejemplo, lo de aprender a usar los programas y a sacarles partido, combinándolos cuando sea necesario a base de ir pasando ficheros de uno a otro para acabar con un trabajo completo y a nuestro gusto, se resuelve teniendo en cuenta las mismas ideas.

Ese otro problema de la utilización, que puede resultar en principio más familiar, viene a consistir más o menos en lo siguiente. Normalmente, Don Fulanito de Tal se compra su ordenador, desea hacer cosas con él, y consigue equiparlo con una hoja de cálculo, una base de datos y hasta un procesador de textos. Sin tener ni idea de nada, decide con toda su buena voluntad aprender a usarlos.

Procede con lógica y corrección irreprochables. Lee los manuales. Se compra, si es necesario y tras cuidadosa inspección de los distintos textos disponibles para ver cuales se adaptan mejor a sus gustos de lectura y estructura mental, libros adicionales que explican el manejo de sus programas. Todos ellos escritos por gente que entiende razonablemente de ordenadores y que cuenta con amplia experiencia en el uso de los programas que se describen.

Tras algunos días, o quizá meses, de lectura atenta y observadora y práctica simultánea con el ordenador, puede que nuestro desdichado LAO, no particularmente obtuso sino todo lo contrario, decida que la informática no es para él y proceda, si no a quemar, sí a llevar los libros al cuarto oscuro. Para rescatar algunas migajas de la inversión hecha en el ordenador, es probable que decida asimismo conseguir una colección de juegos y se dedique en lo sucesivo a matar marcianos, una actividad bastante adecuada para desahogar, al menos parcialmente, la frustración producida por este episodio, no tan improbable como pudiera parecer.

¿Cuales son las razones del chandrío?. ¿Porqué, incluso procediendo cuidadosamente, es difícil aprender a usar los programas?. Bien, muchos de los manuales, textos, y hasta cursillos que se dedican a explicar como hacerlo, presentan un grave defecto para la gente que realmente empieza con el ordenador. Lo de siempre. Dan muchas cosas por sabidas. No

## ***CAPÍTULO 8. PeCés: Programas***

busque explicaciones sobre lo que es un sistema de ficheros, una interfase de usuario, un disco duro. Todo eso se supone que ya se sabe. Hay que hacerlo así si no se quiere que no haya libro que baje de las 5.000 páginas. Es disculpable que estas cosas no se expliquen allí, y por eso nos hemos tomado el trabajo de explicarlas aquí. Y Vd., al parecer, se ha tomado también el trabajo de acudir aquí para leerlas.

Es ya menos disculpable que no expliquen otras cosas, mucho más directamente relacionadas con los programas que tratan. En concreto hay muy pocos, pero muy pocos textos que se molesten en explicar cual es la filosofía del programa que se quiere aprender, como se organizan sus datos y las consecuencias, positivas y negativas, de ello derivadas, sus limitaciones, sus puntos fuertes, lo que es fácil y lo que no. Su historia incluso, y hasta una comparación con otros programas similares disponibles para el mismo ordenador. Esto lleva a que falte una información fundamental para que los novatos no se pierdan. Muchas veces, los libros que tratan un programa se limitan a recorrer los menús y a dar un listado de lo que hace cada opción. O sea lo mismo que el usuario del programa puede hacer solo. No es eso, no es eso... .

En la tarea de apagar fuegos de emergencia que este texto parece tener, nos corresponde apechugar un poco con estos fallos ajenos e intentar exponer esos conocimientos generales tan necesarios a la hora de la verdad cuando se trata de sacarle partido a un programa. Una tarea sencilla, en verdad, pues de nuevo todo el misterio consiste en tener en cuenta cinco o seis tontadas.

La respuesta a ambos problemas se encuentra en esas cosas que los que empezamos hace tiempo conocemos necesariamente, no porque seamos más listos, sino porque entonces los ordenadores eran algo absolutamente embrionario. Los programas de aplicación prácticamente no existían. Tenemos que saber qué es un sistema operativo y un lenguaje de programación porque en aquellos tiempos remotos no había otra cosa. Se hablaba también de los distintos formatos de los datos, simplemente porque no había casi nada más que tratar y con algo había que entretener al que había cometido la temeridad de meter algunas decenas de miles de pesetas en un chisme casi inútil. Y a partir de ahí, hemos controlado el desarrollo de los programas de aplicación y conocemos su historia. Tenemos la ventaja de haber visto crecer a los microordenadores y de que se nos explicaran, cuando aún se hacía, esas cosas que ya no se describen y se dan por supuestas.

A modo de resumen absolutamente total y, sin que sirva de precedente, antes incluso de empezar en realidad con ello, lo que hace falta conocer para moverse con soltura a la hora de usar un programa no es otra cosa que el formato de los datos que maneja. Ni más ni menos que algunos detalles adicionales de lo que en general hemos venido llamando el sistema de ficheros del ordenador.

La historieta de la instalación del apartado anterior debería haber permitido deducir con claridad meridiana a todo LAO que se precie que el tema, aparentemente lejano, de los formatos de los ficheros, está también implícito en la selección del software. ¿No nos hemos visto acaso en la obligación de citar decenas de formatos concretos con su extensión y todo?.

Debemos pues volver a ocuparnos de los datos. No tiene nada de extraño que, ocupándonos de conocer más a fondo aquello que el ordenador procesa y produce, podamos conseguir esa base teórica que tan fundamental resulta a la hora de desenvolverse correctamente ante él.

Es la estructura de los datos que un programa manipula lo que define en mayor medida su finalidad. Toda base de datos maneja ficheros estructurados básicamente del mismo modo, por lo que todas ellas son fundamentalmente equivalentes. Todo editor de textos mira cualquier fichero de la misma forma, lo que se traduce en que todos son equivalentes. Cualquier hoja de cálculo trabaja con datos almacenados de manera similar, por lo que todas resultan sumamente parecidas.

Si conocemos las estructuras generales más frecuentes de los datos que se procesan con el ordenador seremos automáticamente capaces de clasificar la multitud de programas existentes en unos pocos grupos de contenido homogéneo. Ya no habrá dos mil bases de datos diferentes, sino que podremos mirarlás a todas de una vez viendo una única imagen básica susceptible de pequeñas variaciones al pasar de un programa a otro. El primer problema de la selección del software, su agrupación en categorías en función de su finalidad práctica, queda automáticamente resuelto. Y de paso también resolvemos el asunto del uso práctico de los programas. Sabiendo qué tipo de datos maneja una base de datos será posible deducir qué podemos hacer con ellos y porqué hay que hacerlo de una determinada manera. En consecuencia, la estructura de los datos determina también automáticamente el funcionamiento y el manejo de cualquier programa.

Dijimos tiempo ha, probablemente en una nota al pie, que Niklaus Wirth, uno de los grandes patriarcas de la informática académica, escribió un libro titulado “Estructuras de datos+Algoritmos=Programas”. Las estructuras de datos son al menos un 50%, probablemente más, de cualquier programa, y no debe resultar sorprendente en consecuencia que podamos sacarle tanto jugo a un conocimiento, siquiera somero, de ellas. Desarrollando un poco el título, podemos decir que la estructura de los datos define lo que hace un programa, su finalidad, para qué sirve, en tanto el algoritmo determina como manipula dichas estructuras, su forma concreta de operar. Un algoritmo particularmente elegante y eficaz podrá conseguir que un programa procese sus datos mucho más rápida, y quizá más sólidamente, que otro, aunque al compartir la estructura básica de los datos que procesan, ambos programas sirvan exactamente para lo mismo<sup>48</sup>. Los algoritmos influirán por tanto en el proceso de selección de un programa concreto dentro de una categoría dada. Aquel que use el algoritmo más eficaz será más rápido y seguro, y en consecuencia preferible. Al contrario que con las estructuras de datos, un usuario no necesita normalmente ocuparse de los algoritmos. Un programador, en cambio, sí.

No se trata de enmendarle la plana a nadie, y menos a Niklaus Wirth, pero podríamos modificar un poquito la ecuación anterior y escribir “Estructuras de datos+Algoritmos+Interfase de usuario=Programas de aplicación”, y con ello tendríamos un resumen prácticamente completo de los aspectos fundamentales a considerar en la selección y el manejo de un programa desde el punto de vista del usuario. A igualdad en el tipo de datos, es decir en la finalidad, y en el algoritmo, es decir en la velocidad y solidez del procesado, los programas podrán diferenciarse también de forma muy importante por su facilidad de manejo. Me-

---

48 Una vez más es preciso matizar bastante afirmaciones tan generales. No hay una separación absoluta entre estructuras de datos y algoritmos. En realidad, ambos están casi indisolublemente unidos. El deseo de usar un determinado algoritmo puede llevar a una estructura de datos dada, o la elección de una estructura adecuada hacer posible la utilización de un algoritmo inservible de otro modo.

## ***CAPÍTULO 8. PeCés: Programas***

tiendo a la interfase de usuario en medio volvemos a tener una visión prácticamente completa del asunto. Y a partir de aquí, todas las demás consideraciones son circunstanciales. ¿Qué es el precio, sino un mero accidente?. Un mero accidente que cada cual debe valorar de acuerdo a su capacidad adquisitiva, eso sí, y que merece ser tenido en cuenta. Pero los asuntos de dinero deberán esperar a otro capítulo.

La teoría parece clara. La práctica suele tener la costumbre de serlo un poquito menos y hace necesarias algunas precisiones. No debemos confundir la estructura de los datos con el formato de los ficheros que los almacenan. Para nosotros, una estructura general va a ser algo ideal. Algo que, al concretarse en la práctica, podrá distorsionarse más o menos levemente, manteniendo sus características básicas pero convirtiéndose en una cosa concreta no estrictamente igual a otra en principio equivalente<sup>49</sup>. Dos ficheros que sirvan para almacenar datos de tipo hoja de cálculo pueden guardar las mismas cosas en diferente orden. Una misma estructura puede concretarse en formatos diferentes. Y son ficheros, con su formato particular, lo que nosotros vamos a ver y manipular en la práctica. La literalidad del funcionamiento de Luis Ricardo interviene inmediatamente para hacer imposible de usar desde un programa un fichero creado con otro, aunque ambos ficheros contengan los mismos datos, compartan la misma estructura básica, y los dos programas sirvan para lo mismo.

¿Cuales son las consecuencias?. Los programas de mayor éxito imponen sus propios formatos y crean un estándar de hecho. Un programa nuevo sólo podrá competir con el estándar si es capaz de usar lo más directamente posible los datos de otro, y eso suele implicar poder leerlos para acceder a ellos, y escribir los resultados de sus operaciones empleando el formato de la mayoría. O al menos, si el uso de un formato particular y no estándar permite recurrir a algoritmos especialmente ventajosos y eso aconseja no mantener una compatibilidad estricta a nivel de ficheros, si puede importar y exportar los datos con facilidad en el formato mayoritario.

El mismo campo de aplicación y el inevitable respeto a los estándares establecidos por el mercado hace que las mayores diferencias entre programas del mismo tipo suelen consistir en aspectos puntuales de sus respectivas interfases de usuario. Y si ambos trabajan en Windows, o en cualquier otro GUI, incluso ese pequeño resquicio de diferenciación quedará muy matizado, ya que uno de los objetivos de un GUI es justamente unificar el aspecto y manejo de los diferentes programas que corren sobre él. Este escenario lleva a que cualquier “experto”, necesariamente entre comillas, en el uso de los microordenadores, pueda decir sin mentir en absoluto algo como lo siguiente: “Hombre, como hoja de cálculo yo uso Excel (por ejemplo), pero no tengo ningún problema en ponerme con Quattro Pro o Lotus 123 (dos productos de la competencia) aunque no las haya utilizado jamás”. Cualquier “experto”, una vez más forzosamente entre comillas, sabe lo anterior, y es por eso por lo que su habilidad consiste en “usar programas tipo hoja de cálculo” y no en “dominar Excel”. Esa habilidad general no es una rara flor surgida de un azar improbable, sino una consecuencia directa de la manera en que el ordenador debe programarse y almacenar los datos que ha de procesar.

---

49 Como decían “Martes y Trece” en un famoso número sobre detergentes: “Es lo mismo, pero no es igual”.

Dándonos una pequeña vuelta por los aspectos más generales del sistema de ficheros, hemos podido deducir la existencia de categorías de programas de aplicación equivalentes, los aspectos fundamentales que pueden diferenciarlos entre sí, e incluso casi deducir la necesaria equivalencia entre ellos si admitimos como inevitable el desarrollo de estándares en mercados lo suficientemente amplios, algo que sabemos ocurre en los microordenadores. Puede que nos hayamos puesto un poco pesados y quizá demasiado teóricos, pero casi debería reconocer que ahí es nada lo que hemos conseguido a cambio.

### **Principios...**

### ***Luis Ricardo se mete a leguleyo***

A partir de aquí, bastará con dar un repaso a las estructuras de datos más generales y a los formatos de ficheros estándar que de ellas se deriven en el mundo de los PeCés, añadiendo algunos brochazos de historia por ponerle un poco de color, para que cubramos con bastante amplitud todos los temas que nos quedan por tratar en este capítulo. Como ve, el consejo de que si quiere enterarse de lo que pasa en un ordenador debe seguirle la pista a los datos, o si lo quiere de forma más concreta a los ficheros, no era ninguna exageración.

A la hora de meternos en harina y describir globalmente las estructuras de datos más importantes, es aconsejable empezar por lo más elemental. A partir de ahí, ya lo iremos complicando. Las estructuras de datos más sofisticadas son simplemente montajes a partir de otras más sencillas, como veremos a no mucho tardar. Si el ordenador era ante todo una máquina de calcular alfanumérica, habrá que empezar por ver qué es para él un número y qué es una cadena alfabética, los dos tipos de datos básicos que son los ladrillos con que vamos a trabajar.

### **Editores de texto y calculadoras**

Las cadenas de caracteres son lo primero de todo. Y ya las conocemos. Hace bastantes capítulos explicamos como colocar un texto en el ordenador. Empleando un código determinado, por ejemplo el ASCII si estamos en DOS o el ANSI si estamos en Windows, metíamos en la memoria del ordenador como una ristra de bytes cualquier cosa que pudiera escribirse. Por poner un ejemplo, la hipotética frase:

Creo sinceramente que la informática es muy sencilla.

quedaría, usando el mapa extendido de caracteres ASCII de un PeCé, más o menos así:

67 114 101 111 32 115 105 110 99 101 114 97 109 101 110 116 101 32 ....

al menos la parte de “Creo sinceramente ...”, pues es una lata ir buscando equivalencias en la tabla ASCII. Completar la totalidad de la frase se deja como ejercicio para el lector<sup>50</sup>.

---

50 Es probable que aquel lector que se atreva a completar a pedal la codificación ASCII de la idiotez anterior, acabe en algún momento visualizando en la pantalla de su ordenador el “volcado numérico” de otras cadenas de caracteres o de cualquier otra cosa. Tal vez se sorprenda entonces al comprobar que lo que se le presenta no es la serie anterior, sino una tal que así: “43 72 65 6F 20 73 69 6E 63 65 72 61 6D 65 6E 74 65 20...”. La explicación es sencilla. Los volcados numéricos suelen presentarse con los números en hexadecimal. ¿Que porqué en hexadecimal?. Usando esa base numérica, los números del 0 al 255 no ocupan más que dos columnas como

## **CAPÍTULO 8. *PeCés: Programas***

Las cadenas de caracteres, en una u otra codificación<sup>51</sup>, son la estructura básica para almacenar de todo en el ordenador. No debe sorprender. Al fin y al cabo su memoria, por grande que sea, puede representarse con exactitud como una serie de bytes ordenados sucesivamente. El parecido es innegable<sup>52</sup>. Un texto, como el anterior, será una cadena de caracteres codificada en ASCII o un código equivalente. Un programa de ordenador será una cadena de caracteres codificada mediante código máquina. El “67” ya no querrá decir “C”, ni el “114” significará “r”, o el “99” será la “c” (fíjese que no son lo mismo las mayúsculas que las minúsculas), sino que podrá indicar a la CPU que lea un determinado byte en una posición concreta de memoria.

Si la teoría que hemos expuesto al principio es mínimamente verosímil, a la primera estructura de datos importante debería corresponderle un tipo de programas. ¿Qué programas sirven para manipular cadenas de caracteres?. Los editores de texto. Para ellos, todo lo que el ordenador contiene son cadenas de caracteres. En principio, aunque no necesariamente, en ASCII, y cualquier fichero que abramos con ellos nos será presentado como un texto en dicho código. Si el fichero que abrimos corresponde a un programa (un fichero .EXE o .COM, ya dijimos que los .BAT eran órdenes como las que nosotros escribimos desde el teclado dejadas en un fichero para que pudieran ejecutarse en serie cuando hiciera falta, y esos sí podrán leerse perfectamente) el editor nos lo mostrará como una cadena imposible de leer, formada por caracteres de lo más variopinto. Como siempre, debe recordarse que somos nosotros los que tenemos que saber lo que estamos haciendo y poner el significado a lo que el ordenador nos muestra<sup>53</sup>.

---

máximo (de 00 a FF), lo que hace las tablas más compactas y con mejor aspecto.

- 51 Todo lector atento habrá observado de inmediato que estamos ya ante un caso concreto de diferencias entre “estructuras de datos” y “formatos” en el sentido que nosotros estamos empleando. Una cadena de caracteres es una estructura ideal. Al concretarse como una serie de bytes en memoria, empleando un código determinado, le damos un formato concreto. Una cadena de caracteres codificada en ASCII no será posible leerla mediante el código EBDIC. La estructura y los datos son los mismos, pero los formatos resultan incompatibles. Si queremos que un programa la lea, deberá manejar ambos códigos y efectuar las conversiones necesarias.
- 52 En bastantes estudios teóricos se considera de hecho que el ordenador es poco más que eso, no crea que está leyendo a alguien capaz de deducir él solito cosas tan raras. Tal vez fuera algo más exacto decir que la memoria del ordenador puede considerarse como una ristra de bits, en lugar de bytes, pero hace tiempo que advertimos que el rigor no nos iba a preocupar en exceso. Por si no bastara, aunque fuera más exacta y potente desde el punto de vista conceptual, lo de la ristra de bits es con seguridad mucho menos sencillo de digerir para el que acaba de aterrizar en la informática. Además, con esta nota debería valer, pues pronto va a volver a salir el dichoso bit. También dijimos hace tiempo que tenía por costumbre aparecer cuando menos se esperaba.
- 53 Es casi seguro que el LAO de una nota anterior se estará preguntando en este momento como se puede conseguir que el ordenador nos muestre las cadenas de caracteres que correspondan a programas en código máquina de forma inteligible, es decir como listados de instrucciones para una CPU. El cambio en la estructura de datos, ahora no tan claro, que pasa a ser un “programa en código objeto de la correspondiente CPU” es decir cadenas de caracteres codificadas con un código especial, implica un cambio en el tipo de programas que hacen ese tipo de cosas. Lo suyo es usar un desensamblador, así llamado porque muestra el contenido de la memoria como instrucciones para el ordenador en “ensamblador”, el lenguaje más cercano a la máquina que los humanos somos capaces de leer. También los “debuggers” o depuradores de programas, son capaces de listar una ristra de bytes de esta forma. Aunque hacen bastantes más cosas, pudiendo

Debería estar claro porqué los editores de texto no conocen márgenes, tipos de letra, encabezamientos, gráficos, ni elemento alguno de esos que normalmente es preciso añadir a un texto para acabar teniendo un documento a todo lujo. Nada similar aparece en las estructuras de datos para las que los editores han sido creados.

Debería estar claro porqué los editores de texto no conocen márgenes, tipos de letra, encabezamientos, gráficos, ni elemento alguno de esos que normalmente es preciso añadir a un texto para acabar teniendo un documento a todo lujo. Nada similar aparece en las estructuras de datos para las que los editores han sido creados.

Sus habilidades se reducen a reconocer espacios, tabuladores, líneas de texto, o signos de puntuación. ¿Porqué?. El código ASCII cuenta con los caracteres necesarios. Un “9” es un tabulador, y podemos meterlo en una cadena en ASCII. Un retorno de carro es un “13” y podemos interpretarlo como un fin de línea<sup>54</sup>. Y así sucesivamente.

Como a toda estructura importante y que tiene su tipo de programas debe asociarse un formato de ficheros en el que almacenar los datos, podemos preguntarnos cual es el tipo de fichero correspondiente a las cadenas de caracteres. En sentido laxo, ya hemos visto que un buen editor de textos es capaz de incarle el diente a cualquier fichero, tenga o no sentido como texto. Pero los textos de verdad, los legibles, suelen almacenarse en DOS empleando las extensiones .ASC (abreviatura de ASCII) o .TXT (abreviatura de “texto”). Todos los que se relacionan con el funcionamiento del ordenador emplean extensiones especiales tales como .BAT, .SYS o .INI, aunque estén en texto ASCII o ANSI, resaltando en este caso más su papel a la hora de que el ordenador funcione que la forma en que se codifica su contenido. También se emplea más raramente, de forma bastante incorrecta, la extensión .DOC (abreviatura de “documento”), que es más típica de ficheros propios de procesadores de textos. Pero en fin, siempre puede abrirse el fichero con un editor y ver si lo que hay dentro está o no en ASCII.

En cuanto a qué programas se han quedado con el mercado y pueden considerarse los estándares del ramo, el caso de los editores de texto es un pelín peculiar. Hay tantos, y los formatos están tan estrictamente definidos o son tan generales, lo mismo da, que no existe un estándar real y todos los buenos pueden considerarse equivalentes. Por si fuera poco, todo DOS incluye uno, el universalmente denominado “Edit”, y todo Windows tiene un “Notepad” (o “Bloc de notas” en traducción castellana). Ya se advirtió, sin embargo, que su potencia es más bien escasa. Editores de los de verdad, de los que abren treinta ficheros a la vez con cualquier tamaño y contenido, hay menos, y no está de más conocer nombres como “Emacs”<sup>55</sup>, “XyWrite”, “Brief” o “MultiEdit”. Todos los anteriores son programas

---

presentarla también como un mero volcado numérico o como si estuvieran, que no es el caso, en ASCII.

54 De nuevo aparecen diferencias de formato incluso en casos tan sencillos como el que estamos viendo. Algunos sistemas operativos, por ejemplo DOS, indican el final de línea con el par “13 10”, es decir “retorno de carro” y “avance de línea”. Otros, como Unix, el Finder del Mac, o el de mi antiguo Sinclair QL, usan el “10” sin más.

55 Emacs es una especie de editor universal surgido en el mundo de Unix. La versión para PeCé se llama MicroEmacs y no estoy seguro si es de shareware o de dominio público. Debe advertirse que, aunque sumamente potente y con un montón de ventajas, resulta más bien duro de abordar.

## **CAPÍTULO 8. PeCés: Programas**

para DOS, aunque algunos tienen también las correspondientes versiones Windows. Por ejemplo MultiEdit. Estos editores de gran capacidad disponen en muchas ocasiones de lenguajes internos que los hacen programables, y están bastante orientados a la creación del código fuente (la versión en texto legible) de programas, una tarea para la que son absolutamente necesarios. Es por esto que muchos lenguajes de programación incluyen el suyo, normalmente bastante capaz. De todos modos, su utilidad general los hace muy interesantes incluso para aquellos que no tienen intención alguna de meterse en programación. Con no usar un 80% de sus posibilidades, asunto arreglado. Suelen ser siempre programas rápidos, más bien pequeños, y de calidad general más que correcta. La elección de uno u otro queda pues a gusto estricto del consumidor. Nuestro experto incluyó uno de shareware en su relación de programas que se llamaba “UltraEdit”, en concreto en su versión 3.0. Es perfectamente válido. Acabaremos con ellos volviendo a insistir en que, se elija el que se elija, no es posible desenvolverse en informática a nivel mínimamente serio sin utilizar editores de texto.

La siguiente estructura de datos a conocer es aquella que nos permita manejar números en el ordenador. En principio, puede pensarse que, puesto que podemos meter en la memoria cualquier cadena de caracteres, los números podrían manejarse como un tipo particular de cadenas de caracteres. Aquellas que constaran sólo de caracteres numéricos, es decir valores ASCII entre 48 y 57, serían números. Hasta podríamos añadir puntos para los millares, o comas para la parte decimal<sup>56</sup>. No es una idea descabellada, pero desgraciadamente no es la que suele usarse en los microordenadores. Por motivos de eficacia, se recurre a métodos de codificación especiales para los números. Se crea otra estructura de datos<sup>57</sup>.

Los números suelen almacenarse como ristras de bits (no de bytes como las cadenas de caracteres), que representan un determinado valor numérico en código binario, es decir en base 2 (ver glosario)<sup>58</sup>. La cosa se traduce a nivel práctico en que sólo es posible manipularlos empleando programas especiales. Ningún editor de textos los va a visualizar correctamente. Y tiene otras consecuencias. Al utilizar base 2 para operar con los números, aparecen imprecisiones en los cálculos, entre otros motivos porque el espacio que se dedica a su almacenamiento tiene límites muy concretos y no pueden realizarse los cálculos intermedios con un número ilimitado de cifras. Aunque conviene quedarse un poco con la copla por aquello de que estamos describiendo algunas mínimas interioridades de una estructura de datos absolutamente básica que es, junto con las cadenas de caracteres, el otro ladrillo

---

No está de más, de todos modos, que le suene el nombre.

- 56 Una vez más repetimos que eso sería así tan solo si no pertenecemos al mundo anglosajón, que emplea dichos separadores justo al revés. Más adelante veremos el porqué de tanta insistencia en un punto aparentemente irrelevante.
- 57 Sistemas operativos como Unix parecen trabajar de forma diferente, e insisten algo más en que los valores numéricos se almacenen en ASCII. Cada programa que vaya a operar con ellos debe realizar las necesarias conversiones por su cuenta. Aunque los programas tengan algo más de trabajo, esto facilita en buena medida el intercambio de datos entre programas, algo que Unix necesita absolutamente para funcionar.
- 58 ¿Lo ve?. ¿Ve como ya salió el dichoso bit?.



fundamental para montar las demás, lo dejaremos estar tras insertar una primera nota monstruosa por si quiere detalles adicionales<sup>59</sup>.

Tan sólo programas tan elementales como las calculadoras, rayanos en la herramienta, permiten manipular directamente números y operar con ellos, y no suelen plantearse su almacenamiento, por lo que no hay formatos de ficheros estándar. También Windows tiene la suya, y dentro del shareware encontrará varios miles más<sup>60</sup>.

---

59 Para que vea de qué estamos hablando exactamente, vamos a describir brevemente algunos formatos numéricos muy sencillos. Si guardamos los datos numéricos en 2 bytes, es decir 16 bits, tenemos un total de 65.536 posibilidades que permitirán almacenar cualquier número del 0 al 65.535. El formato así definido es un entero sin signo en 16 bits. Si queremos números negativos, por ejemplo para restar de vez en cuando, es necesario reservar un bit para indicar el signo. Es habitual representar los números negativos en “complemento a 2”. Eso nos da un rango de valores que va desde el 32.767 hasta el -32768, incluyendo, por supuesto, el cero. La precisión de los cálculos con rangos tan exigüos exige algoritmos muy cuidadosos para poder hacer algo útil. Es posible, obviamente, aumentar el número de bits dedicados a cada número. Muchas veces se dedican 32 bits, 4 bytes, a cada entero. Eso amplía el rango hasta abarcar valores entre +2.147.1483.648 y -2.147.1483.647 si seguimos codificando los números negativos en complemento a 2. Manejar números en punto flotante, es decir con sus decimales y todo, introduce nuevas sofisticaciones. Debe reservarse un espacio para la parte entera (la mantisa) y otro para el exponente, cada uno de ellos con su signo para poder meter también mantisas y exponentes negativos. Esto hace que crezca el número de bits necesarios para codificar cada número y que los formatos se compliquen aún más. No es extraño llegar a 8 bytes (64 bits) para cada número. Se distingue también entre valores de simple o doble precisión según la cantidad de cifras significativas que cada formato permite manipular. Quizá el formato más simple de describir, siquiera sucintamente, es el denominado BCD, abreviatura de “Binary Coded Decimal”. En él se reservan cuatro bits (medio byte) para cada cifra decimal del 0 al 9. Sobran posibilidades, ya que como todo el mundo sabe cuatro bits dan para 16 valores, pero no pasa nada. Más vale que sobre que no que falte. Cada número se mete ahora en memoria como algo muy parecido al ASCII que describimos como hipótesis anteriormente, pero empleando sólo medio byte en lugar de un byte completo. Por supuesto, tampoco ahora es posible visualizarlo con un editor. Muchos microprocesadores son capaces de operar directamente con números codificados en BCD, y esto aporta notables ventajas en cuanto a la precisión de los cálculos a costa de una mayor lentitud. Este tipo de codificación suele emplearse por lo tanto raramente, tan sólo en aquellos casos en los que los errores de redondeo no son admisibles. Por ejemplo en cuestiones relacionadas con la pasta. A mí no me gustaría nada que me desaparecieran 5.000 pelas de mis cuentas del banco por culpa de la manera en que éste guarda los números en sus ordenadores y mucho menos si me están cobrando una comisión de mantenimiento de mi cuenta. Si no le basta con lo anterior y quiere todavía más, ha llegado el momento de recomendarle que se meta a programar. Un buen compilador de C o C++, un par de lenguajes de programación, incluirá sin duda manuales en los que se describirán con todo detalle estos y otros muchos formatos de almacenamiento numérico. Si sigue por ahí, hasta podrá, en un par de años, hacerse sus propios programas para calcular lo que quiera con la precisión que desee.

60 Como habrá deducido cualquiera que haya leído en el glosario la entrada correspondiente a “Autor”, soy un forofo de las calculadoras de Hewlett-Packard. Hace mucho tiempo que adquirí un programa de shareware que emula la serie 1xC de HP, en concreto las HP-11C (cálculo científico), 12C (cálculo financiero) y 16C (cálculo para programación). Se llama “Xact” y es un programa DOS que puede también usarse desde Windows. Creo que ha desaparecido, pero no llore mucho. Seguro que si busca encuentra varios cientos de programas similares. De todos modos, yo no he encontrado todavía ninguno que me guste más. Si Vd. conoce alguno que

## **CAPÍTULO 8. PeCés: Programas**

En el PeCé, son las hojas de cálculo, basadas en estructuras de datos mucho más sofisticadas, los programas que suelen usarse para manejar datos numéricos. Si lo anterior no le basta, puede leer la siguiente nota monstruosa<sup>61</sup>.

Podemos dar las estructuras de datos numéricos por terminadas con sólo hacer una pequeña precisión adicional. Algunas cosas más o menos indeterminadas, tales como las fechas, suelen almacenarse también en el ordenador como datos numéricos. Para ello se recurre a algún tipo de estructura numérica particular. Por ejemplo, el “5 de marzo de 1.995” puede guardarse como el número “050395” empleando cualquier formato numérico de capacidad suficiente. O podemos numerar los días sucesivamente a partir del inicio del calendario gregoriano actualmente en vigor y guardar en los bytes necesarios el número entero correspondiente. O, si estamos interesados en fechas anteriores, usar el mismo esquema con el antiguo calendario juliano<sup>62</sup>. Algo parecido puede decirse sobre las horas. En lo sucesivo, para nosotros una fecha o una hora no será más que un caso particular de una estructura de datos numérica general cuyas particularidades no deben preocuparnos. Ya sabrá el programa que las use como apañarse con ellas.

---

emule decentemente la serie HP 48, hágamelo saber.

61 No son los únicos, de todos modos. Si Vd. pertenece al no muy numeroso colectivo de usuarios de PeCés que tienen algún interés en emplearlo para realizar cálculos de tipo científico, sepa que, por supuesto, se dispone de programas adecuados. El mero cálculo matemático está cubierto en el mundo del DOS por programas como “Derive”, y en el mundo Windows por “Mathematica” y “MathCAD”, entre otros. Tampoco es difícil encontrar shareware o dominio público. Por ejemplo, hace tiempo que tengo una copia de “Mercury”, el nombre que se le dio a un antiguo programa de Borland, llamado originalmente “Eureka”, cuando se dejó de vender y se pasó a dominio público. La estadística está también sobradamente servida con monstruos tales como “Statgraphics”, “Statistica” o “MiniTab”. Otros programas cubren campos aún más especializados. Citemos de pasada a “PSI-Plot”, “SigmaPlot”, “Origin”, “SpectraCalc”, o “Grams”, entre otros muchos. No le sorprenderá en absoluto, si es Vd. del gremio matemático, que la mayor parte de ellos suelen recurrir a manipular sus datos, al menos internamente, en forma de vectores o matrices. Aparte de otro montón de diferencias importantes (operaciones que realizan, formatos de almacenamiento, tamaño admisible en los vectores o matrices a procesar, etc., etc.) se distinguen ante todo de las hojas de cálculo por dedicar una atención expresa a la precisión de los cálculos. No es admisible en este tipo de productos que la validez de la quinta cifra significativa de un resultado pueda resultar siquiera mínimamente sospechosa. Los formatos de almacenamiento estándar que necesariamente existen tienen un ámbito de aplicación tan restringido que no merece demasiado la pena preocuparse por ellos. El intercambio de datos suele resolverse muchas veces mediante ficheros ASCII en forma tabular, algo tan universal como para poderse preparar con cualquier hoja de cálculo. De hecho, en muchas ocasiones los datos se recogen y preparan con ellas y se exportan en ASCII para su procesamiento final en programas de cálculo matemático. Si se mete por aquí, sabiendo lo que tiene que buscar, se enterará enseguida de todo lo que hay.

62 Es sorprendente la cantidad de conocimientos diversos que hacen falta para poderse dedicar con dignidad a la programación de ordenadores. No siempre se han establecido las fechas de igual manera. Un par de papas, de nombres Julio y Gregorio (no me pregunte los números, por favor), estuvieron activamente implicados en la definición de un sistema de calendario lo suficientemente preciso y estable para la cristiandad. De ahí los nombres de los calendarios.

### Bases de datos

Ha llegado el momento de empezar a montar estructuras de datos complejas. La primera a conocer, por general, es la propia de una base de datos. Partamos de la base de que lo que se intenta en este caso es reproducir un archivo con sus fichas. Todo el mundo sabe que un archivo es una especie de cajón más o menos gordo que contiene en su interior un montón de pequeñas tarjetitas de cartón, una para cada uno de los elementos que nos interesa tener controlados, dentro de cada una de las cuales hay distintas zonas en las que apuntamos las características propias de cada elemento, sean las que sean.

Un archivo con recetas de cocina podría tener una ficha para la “Sopa bullabesa”, otra para la “Ensaladilla rusa”, y así sucesivamente, y en cada una podríamos poner una entrada para los ingredientes, otra para la dificultad de cocinarla, otra más para el valor calórico por persona (si estamos interesados en mantener la línea), y cualquier cosa más que se nos ocurriera y consideráramos necesaria. Un archivo para controlar nuestra biblioteca, o la totalidad de nuestros discos, usaría el mismo esquema de fichas pero incluiría características diferentes. El título, autor, fecha de compra, quizá el precio de adquisición, el ISBN y la editorial si se trata de libros, y tal vez hasta si lo hemos prestado, y en caso de que así sea, cuando y a quién.

Por si no se hace una idea muy exacta, supongamos que hemos creado en efecto un archivo para seguirles la pista a nuestros libros y que el formato de las fichas es una versión simplificada de lo que acabamos de decir. La ficha con los datos del primer libro de nuestra biblioteca podría tener el siguiente aspecto:

Título: La informática está chupada  
Autor: Ben Yveras  
Editorial: Libros Optimistas, S.L.  
Precio: 7.000  
ISBN: 111-111-111-X  
Año: 1.996

No es que la tal ficha sea un prodigio de imaginación, pero puede valer para un ejemplo. Nuestro segundo libro no sería menos que el primero y tendría también su pedazo de cartón, en el que quizá pusiera algo así como:

Título: La informática no está tan chupada  
Autor: Ken O'Tefies  
Editorial: Libros Pesimistas, S.L.  
Precio: 9.000  
ISBN: 222-222-222-X  
Año: 1.996

Si queremos traducir rápidamente lo anterior a términos informáticos, para acercarnos al momento en que seamos capaces de hacer exactamente lo mismo con un ordenador, casi es suficiente con decir que las estructuras que se crean para las bases de datos se limitan a reproducir literalmente lo anterior, bien sea en la memoria del ordenador, o bien, en caso de que los datos se guarden en un periférico de almacenamiento, en un fichero. Los conceptos son idénticos y sólo hay que aprender tres o cuatro palabrejas para poder llamar a las cosas por su nombre.

El equivalente a una de nuestras fichas de cartón, la estructura de datos compleja que almacena una serie de características dispares de un elemento cualquiera, se denomina un

## **CAPÍTULO 8. PeCés: Programas**

“registro”. Cada una de las características que contiene se llama un “campo”. Y por supuesto, nuestro archivo de cartón será un fichero “tipo base de datos” cuando se almacene doquiera que sea en el ordenador<sup>63</sup>

Sabiendo esto, podemos pasar a precisar algunas obviedades. Un registro estará formado por una serie de campos en un orden determinado y con un contenido definido. En el caso del ejemplo, cada registro constaría de cuatro campos “tipo cadena de caracteres”, como corresponde a zonas del registro que deben albergar un título de un libro y los nombres del autor y la editorial o el ISBN, y dos campos “de tipo numérico” que servirían para guardar el precio y el año de publicación. Lo de que el ISBN se guarde en una cadena de caracteres es un poco menos obvio pero parece claro que no es un número exactamente, y las cadenas de caracteres tienen la gran ventaja de poder guardar cualquier cosa que se pueda escribir, o sea que podría ser una buena manera de no complicarse la vida. Además, como casi seguro que vamos a querer tener la posibilidad de acceder a los diferentes valores de los campos de forma individual, no está de más que cada campo tenga un nombre que nos ayude a saber qué es lo que contiene. Es, una vez más, el viejo cuento de ponerles nombres a los datos que el ordenador maneja para poderlos usar más fácilmente. A los registros no hará falta llamarlos de ninguna manera y podremos dejar que el ordenador simplemente los numere, por ejemplo. Al fin y al cabo, una vez definida la estructura y pudiendo acceder a cada uno de los campos, no nos hace falta más.

Podríamos incluso intentar echar una miradita dentro del fichero en disco en que quizá guardáramos nuestra base de datos. Suponiendo que optáramos por un formato totalmente legible con cualquier editor de textos, el aspecto concreto que podría adoptar tal vez fuera:

*"La informática está chupada", "Ben Yveras", "Libros Optimistas, S.L.", "7.000", "111-111-111-X", "1.996"*

*"La informática no está tan chupada", "Ken O'Tefies", "Libros Pesimistas, S.L.", "9.000", "222-222-222-X", "1.996"*

Como puede ver, el formato concreto elegido para nuestro supuesto fichero de base de datos usa algunas convenciones simples para distinguir campos y registros. El contenido de cada campo aparece entre comillas y con una coma adicional para separarlo del resto, mientras cada registro se distingue del siguiente por un final de línea. Algún LAO quizá se esté preguntando porqué los nombres de los campos no aparecen por ninguna parte. No es estrictamente necesario, pero es cierto que puede ser interesante guardarlos también. No hay problema en ponerlos como un primer registro un poco especial, con lo que el fichero quedaría:

*"Título", "Autor", "Editorial", "Precio", "ISBN", "Año"*

---

63 A la hora de procesar los datos en él contenidos, cada registro, con sus campos correspondientes, se copiará en memoria. Recuerde que la CPU sólo trabaja con datos en memoria. Podríamos intentar completar un poco la exposición distinguiendo entre variables (datos en memoria) y ficheros (datos en un periférico de almacenamiento), pero se sospecha que, más que aclarar, lo que se lograría sería el efecto contrario. Por tanto, mantendremos un discreto silencio en lo sucesivo sobre si los datos están en RAM, en ROM, en disquete, o en un disco duro, y hablaremos siempre de los ficheros. Baste saber que un registro, o cualquier otra estructura de datos sofisticada, tendrá una imagen en RAM que será la que la CPU use para trabajar.

### ***Principios...***

"La informática está chupada", "Ben Yveras", "Libros Optimistas, S.L.", "7.000", "111-111-111-X", "1.996"

"La informática no está tan chupada", "Ken O'Tefies", "Libros Pesimistas, S.L.", "9.000", "222-222-222-X", "1.996"

Otra cuestión quizá sorprendente es que guardemos el contenido de campos numéricos como cadenas de caracteres ASCII a la hora de dejar los datos en un fichero. En principio tampoco plantea mayor problema. Cualquier programa capaz de manejar datos numéricos, hasta la más humilde calculadora, debe ser capaz de saber convertir números en ASCII al formato necesario para que la CPU los digiera y viceversa. Al fin y al cabo tiene que ser capaz de leer los números que nosotros le pasemos a través del teclado y de presentárnoslos en forma legible para que podamos saber el resultado de una operación<sup>64</sup>. La base de datos, que sabría que "Precio" y "Año" son campos de tipo numérico, se encargaría de efectuar las conversiones necesarias.

Debería ser evidente a estas alturas que estructuras de datos tan complejas deben dar lugar a toda una serie de formatos de almacenamiento distintos y más o menos estandarizados. Así es. De hecho, el que hemos descrito resulta muy apropiado cuando se trata de pasar datos de una base de datos a otra, es decir de realizar operaciones de "exportación" e "importación" de datos, y es uno de tales estándares. Se distingue en DOS con la extensión .CSV, abreviatura de "Comma Separated Values", es decir "Valores separados por comas". Por supuesto hay más, no tan legibles, que al menos citaremos dentro de un instante, en cuanto acabemos de dar una vuelta por los conceptos más generales de las bases de datos.

La estructura de datos que emplean las bases de datos no es más que eso. Un conjunto de campos que podrán ser de tipo numérico o de tipo cadenas de caracteres en un orden y con una capacidad determinada, cada uno de los cuales será accesible individualmente como un campo particular de un registro determinado y globalmente como un registro completo. El tipo concreto de cadena de caracteres o de dato numérico que metamos en el registro es cosa nuestra. Podremos poner, es muy frecuente, una fecha o una hora, por ejemplo. Los programas de gestión de bases de datos ofrecen multitud de tipos de campos para poderse adaptar a casi infinitos problemas.

Y al decir lo anterior, casi podríamos exclamar aquello de "con la Iglesia hemos topado", pues el principal problema que encuentra el usuario no advertido al intentar usar este tipo de programas es justamente su intrínseca indefinición. Un programa tipo base de datos debe servir para cualquier cosa, y sólo ofrece en consecuencia herramientas básicas para que el usuario construya lo que desee. Emplearlos exige seleccionar los distintos campos que vamos a controlar, montar en nuestro caso concreto la estructura de registros necesaria, y llenar los registros con los valores correspondientes a cada campo. Y a partir de ahí, decirle qué queremos que haga. Por ejemplo, "muéstrame un listado de todos mis libros en orden alfabético por autor, y si tienen el mismo autor, por año de publicación". O "preséntame, a partir de todos los apuntes de operaciones comerciales realizadas en el último semestre, un resumen de ingresos y costes e imprime las correspondientes declaraciones del I.V.A."

---

64 Esto hace todavía más razonable la insistencia Unix en que los números se almacenen en ASCII.

## ***CAPÍTULO 8. PeCés: Programas***

Una base de datos se limita a proporcionar las herramientas para poder crear los registros con la estructura concreta que se adapte al problema que queramos resolver, introducir los datos necesarios a manipular, y presentar los datos en la forma que nosotros elijamos. Aparece así el concepto de “máscara”. Esta denominación general sirve para designar la apariencia particular que deseamos adopte un registro cuando nos sea presentado, bien para introducir en él nuevos datos (se habla entonces de “máscaras de entrada”), o bien para sacarlos, por ejemplo como un informe semestral (se habla en consecuencia de “máscaras de salida”). Por supuesto, ambos tipos de máscara son definibles por el usuario, que una vez querrá presentar un listado de libros y otras una declaración de I.V.A.. Se incluyen en consecuencia las opciones necesarias para definir, modificar, y almacenar las máscaras correspondientes a cada fichero tipo base de datos que podamos haber definido. Es obvio que este tipo de manipulación requiere muchas veces que el contenido de algunos campos se obtenga mediante operaciones matemáticas a partir del contenido de otros. No hay problema. Pueden definirse campos que realicen tales operaciones.

Las operaciones básicas que se realizan con una base de datos son también muy sencillas. Suelen consistir tan sólo en la selección de grupos de registros que cumplan el criterio que sea, y en la ordenación de la forma que nos de la gana en función del criterio que elijamos (suele hablarse entonces de una “clave”). A nadie debe extrañar que se pregunte a una base de datos cosas como “Muéstrame todas las recetas de cocina que tengo con valor calórico inferior a 3.000 calorías y en las que se emplee la patata como ingrediente principal. ¡Ah!. Ordénalas por valor calórico decreciente”. Pues bien, en tal pregunta tenemos los dos tipos principales de operación antes citados, la selección y la ordenación. Nosotros sabremos el formato en que nos deben ser presentados, para eso habremos definido la máscara de salida correspondiente, y lo que queremos hacer con ellos. Quizá borrarlos, o copiarlos a otra base de datos, quizá una nueva en la que guardemos recetas para adelgazar económicas. Los programas tipo base de datos deberán proporcionar también las herramientas necesarias para ello.

No hemos hecho más que arañar un poquito la superficie de las bases de datos. Es muy frecuente en la actualidad que los tipos de campos que pueden definirse incluyan también gráficos o incluso sonidos, animaciones, y hasta secuencias de video digitalizadas. Es perfectamente posible crear una base de datos de los empleados de una fábrica en la que cada registro tenga una foto del mismo, por poner un ejemplo<sup>65</sup>.

En resumen, lo auténticamente característico de una base de datos consiste tan sólo en que permite crear y manipular conjuntos de datos de estructura compleja. La potencia implícita en una estructura de datos tan sofisticada hace que tales programas sean la base sobre la que se construyen muchas de las aplicaciones de tipo vertical. Lo que el usuario final ve en estos casos no es más que el resultado final una vez definidos campos, registros, máscaras de

---

65 Todos estos campos un tanto especiales presentan ciertas limitaciones de las que conviene estar enterado. No pueden usarse para realizar ordenaciones o búsquedas en función de su contenido. Son campos, en consecuencia, más decorativos que otra cosa, aunque es innegable que incluso así pueden tener mucho interés. En resumen, si Vd. está interesado en poderle preguntar a su base de datos cosas como “dime que compañeros de trabajo son rubios y de ojos azules” o “ordénalos por el color del pelo, de claro a oscuro”, no lo dude, deberá definir un campo de tipo cadena de caracteres (por ejemplo) correspondiente a “Color de ojos”. Aunque sus registros incluyan maravillosas fotos a todo color.

entrada y salida, y hasta lenguajes simples de selección y ordenación, tal vez directamente accesibles a través de menús que permitan explotar con facilidad el cúmulo de datos que se acaban almacenando en los correspondientes ficheros<sup>66</sup>. Un programa de base de datos sin más puede considerarse un caso particular de esos lenguajes de programación con los que se construyen los programas de aplicación.

Como hasta los programadores más avezados prefieren no partir absolutamente de cero cuando llega la hora de trabajar, todas las bases de datos son capaces de presentar los datos que van almacenando, eso sí una vez definida la correspondiente estructura de los registros, con un par de máscaras de entrada y salida bastante universales. La primera de ellas es la que podríamos llamar “tipo ficha”, que hemos usado implícitamente al presentar los registros de nuestro ejemplo. La segunda es un “tipo tabla” en la que normalmente las filas representan registros y las columnas campos. Si la usáramos para visualizar los registros de nuestro ejemplo, la cosa podría quedar tal que así:

Título	Autor	Editorial	Precio	ISBN	Año
La inf... está chupada	Ben Yveras	Libros Optimistas,S.L.	7.000	111-111-111-X	1.996
La inf... no tan chupada	Ken O'Tefies	Libros Pesimistas,S.L.	9.000	222-222-222-X	1.996

Los usuarios menos especializados son todavía más exigentes, y para ellos existen multitud de bases de datos que vienen lo suficientemente completas como para resultar inmediatamente utilizables. Por ejemplo, los programas de gestión de agendas, tales como Lotus Organizer o WordPerfect InfoCentral (la agenda incluida en el PerfectOffice instalado por nuestro experto particular), no son sino casos particulares de bases de datos definidas y preparadas desde el principio para trabajar sobre un tema concreto.

Veamos algunos de los formatos de ficheros a que han dado lugar las bases de datos en el PeCé. La historia, que ya conocemos, hizo que Ashton-Tate impusiera como estándar a su dBase II en los albores de la microinformática. Tal estándar ha sido continuado por sus sucesores, y en consecuencia los formatos por ellos empleados son la auténtica norma universal del ramo. Casi no debería ser necesario decir que por desgracia no usan un formato ASCII como el .CSV que antes vimos, sino formatos parcialmente binarios y mucho más difíciles de desentrañar. Ostentan en general la extensión .DBF aunque muchas veces es necesario distinguir entre versiones concretas del programa. Si continuamos hablando en general, siempre que Vd. vea un fichero con una extensión en la que aparezcan juntas las letras DB, piense que pertenece a una base de datos. Lo de DB es la abreviatura de “data-base”, es decir base de datos en inglés. Programas importantes en el gremio hay muchos, aparte de la ya mencionada dBase. Baste saber por ejemplo que la tal dBase, que ahora

---

66 Los lenguajes de consulta de bases de datos pueden no ser tan simples. Piense que podemos estar hablando de sacarle partido a bases de datos enormes y muy complejas. Un estándar de este estilo es el llamado “SQL”, abreviatura de “Structured Query Language”, que podemos traducir por “Lenguaje de Consulta Estructurado”. Los programas de bases de datos más potentes son capaces de generar automáticamente el correspondiente código SQL a partir de las opciones que nosotros seleccionemos en un menú, pongamos por caso. A continuación pueden pasar tal orden a otra base de datos remota, tal vez la base de datos central de nuestra compañía que quizá es un programa diferente al que nosotros estamos usando, y recoger los datos que nos envíe para acabar mostrándolos en nuestra pantalla o incorporándolos a una nueva base de datos en nuestro disco duro. Si lo anterior le parece complicado, no lo dude. Lo es.

## **CAPÍTULO 8. PeCés: Programas**

pertenece a Borland, ha llegado a ser mediante sucesivas versiones “dBase 5 para Windows” y cuenta con clónicos más o menos mejorados tales como Clipper o Foxpro<sup>67</sup>, y toda una serie de competidores, entre los que conviene conocer a Paradox, también de Borland, a Access, en este caso de Microsoft, y por citar también al resto de la competencia, a Lotus Approach. Si quiere alguna recomendación sobre cual es mejor, no me atrevo a darla ni siquiera implícitamente, ya que no soy un usuario habitual de este tipo de programas. Volviendo a lo de las extensiones, Access guarda sus datos en ficheros .MDB, es de suponer que queriendo indicar algo así como “Microsoft Data Base”. Y por completar un poco el tema, debe tener en cuenta que en ocasiones las máscaras pueden guardarse aparte de los datos, por ejemplo, y un mismo programa puede crear varios tipos de ficheros con distintas extensiones.

Tener presente que hay una estructura de datos determinada en alguna parte sirve para muchas cosas, aparte de para darse el farde tecnológico con los amigos. ¿Se ha preguntado alguna vez, por ejemplo, a qué se debe que en ocasiones su nombre o domicilio aparezcan mutilados en documentos creados evidentemente mediante el uso del ordenador y una base de datos?. La respuesta es simplemente que los campos de un registro suelen definirse con un tamaño fijo. Si Vd. se llama Txomin Cengotitabarreitiabarría (un nombre precioso del que se siente legítimamente orgulloso) y mantiene relaciones comerciales con una compañía que decidió que era sumamente improbable un primer apellido de más de quince caracteres a la hora de montar su base de datos de clientes, para ellos Vd. va a seguir siendo siempre Txomin Cengotitabarrei a secas. Viene a ser como esos impresos en los que no hay sitio para escribir todo lo que hace falta.

Por desgracia, en el PeCé se consideró durante mucho tiempo que no tenían sentido cadenas de caracteres con longitud superior a 256, y muchas bases de datos limitan sus campos de ese tipo a un máximo de 256 caracteres. Se han inventado artificios para soslayar el problema, por supuesto<sup>68</sup>, pero la situación sigue siendo cuando menos incómoda, especialmente si Vd. desea poder meter en dos o tres campos de un registro cosas tales como resúmenes de un libro, textos de canciones, o cosas parecidas.

Es de suponer que uno de los criterios que nuestro experto usó para elegir FileMaker Pro como base de datos adecuada para un principiante consistió en considerar este punto. Al contrario que otros muchos programas similares para PeCé, FileMaker Pro no asigna un tamaño predefinido a los campos de tipo cadena de caracteres y permite que metamos en ellos cualquier texto que no supere, me parece recordar, los 64 Kilobytes. Esta característica, una interfase de usuario notablemente sencilla de usar ya que por algo se desarrolló para el Macintosh<sup>69</sup>, y un soporte suficiente de diferentes formatos de importación y ex-

---

67 Los clónicos intentan hacer las cosas algo mejor. En este caso concreto, lo que intentaban ante todo Clipper y Foxpro cuando salieron era conseguir una mayor velocidad de operación. No me atrevo a certificar que en la actualidad existan diferencias concretas ya que no he usado jamás ni dBase ni ninguno de sus clónicos.

68 Un tipo especial de campos, los llamados “campos memo”.

69 Como siempre decimos, hay que dar al Cesar lo que es suyo. El Macintosh podrá tener defectos, pero entre ellos no está el que sus programas sean particularmente difíciles de usar. Si dejamos en paz al Macintosh y volvemos a lo nuestro, el ejemplo pone claramente de manifiesto que también las particularidades de las estructuras de datos que un programa define intervienen



portación, lo hacen recomendable en este tipo de uso. Y por cierto, FileMaker Pro crea ficheros con la extensión .FM, lo que resulta obvio pero no se ajusta en absoluto a lo que antes dijimos.

Aprovecharemos el que haya salido de nuevo el tal FileMaker para aclarar que muchas veces, especialmente en aplicaciones profesionales, hace falta que una base de datos pueda coger una serie de datos de aquí y otra de allá. Por ejemplo, podemos tener un fichero que contenga los clientes y otro los pedidos, y puede interesarnos que se relacionen entre sí. No todos programas pueden hacerlo, y se distingue en ocasiones entre “auténticas bases de datos”, que son aquellas que lo permiten, y “gestores de ficheros” que se limitan a manejar datos contenidos en un único fichero. FileMaker Pro es de estos últimos, o al menos lo era hasta que apareció la versión más reciente, la 3.0. Otra cosa absolutamente necesaria es que la información de la base de datos se pueda recuperar de forma lo más rápida posible, motivo que justifica la creación de índices que ayudan a buscar con celeridad cualquier dato. También existen en consecuencia estándares para “ficheros indexados”. Seguir por estos derroteros nos llevaría con seguridad a introducir términos tales como “bases de datos relacionales” y otros similares que es mejor no tocar en un libro que se pretende ante todo de introducción.

Terminemos con las bases de datos llamando de nuevo la atención sobre las estructuras de datos. Los diferentes tamaños de los campos hacen que formatos de ficheros tales como el .CSV que vimos y que son capaces de manejar cualquier cosa que no sea tan rara como un gráfico o algo similar resulten tan útiles y universales. Y el hecho de que sea posible representar una base de datos como una tabla debe permitirnos entender fácilmente sus conexiones con otro de los grandes tipos de programas para PeCé, las hojas de cálculo, de las que pasamos a ocuparnos de inmediato.

### **Hojas de cálculo**

La estructura de datos que emplean las hojas de cálculo es justamente una tabla, en principio más o menos como la que vimos al visualizar por última vez nuestros registros del ejemplo. Aunque parecida a la estructura general de una base de datos, es más rígida, y podemos decir sin temor a equivocarnos mucho que es un caso particular de aquella.

También ahora aparecen algunos vocablos de necesario conocimiento. Cada elemento de la tabla es una “celda”, y, como no podía ser menos, se distinguen también “filas” y “columnas”. Las dimensiones de la hoja, aunque generosas, están limitadas. Es frecuente que podamos meter hasta 8.192 filas y 256 columnas. Las filas se designan mediante números, pero para las columnas se usan universalmente letras. La primera es la A, la segunda la B, y cuando se nos acaban se forman pares y se pasa a AA, AB, y así sucesivamente. 256 columnas dan para llegar hasta IV.

¿Qué puede almacenarse en una celda?. Lo de siempre. Más o menos lo mismo que podíamos meter en un campo de una base de datos. Cadenas de caracteres, números de todo tipo y condición<sup>70</sup>, operaciones a realizar con unos y con otros, por supuesto gráficos, y quizá

---

decisivamente a la hora de hacer un programa agradable. Tal tarea no es competencia exclusiva de la interfase de usuario. Recuerde lo de “Estructuras de datos + Algoritmos = Programas”.

## ***CAPÍTULO 8. PeCés: Programas***

hasta alguna cosa más<sup>71</sup>. Y no debería ser necesario decir que no hay problema en la actualidad para asignar tipos y tamaños de letra a una celda para que nos presente su contenido con la apariencia que creamos más conveniente. Algo bastante parecido a las máscaras de que hablamos antes.

Ya que hemos comenzado a hablar de hojas de cálculo llamando la atención sobre el parecido existente entre su estructura de datos y la propia de una base de datos, podemos insistir en esa línea resaltando algunas otras semejanzas. También las hojas de cálculo son programas tan generales que casi no hacen nada por sí solas. También pueden emplearse para construir programas de aplicación que hagan de manera sencilla algo concreto a base de definir su contenido. También se dejan automatizar con no excesiva dificultad a base de recurrir a lo que en general suelen denominarse “macros”<sup>72</sup>. Si en una base de datos debíamos seleccionar tipos de campos y llenar registros con datos concretos, lo mismo deberemos hacer ahora con las celdas. Al menos hay que llenar la tabla con valores y operaciones de manera que termine sirviendo para algo determinado. Una vez más y de forma más o menos sencilla e inadvertida deberemos programar nuestra hoja de cálculo para que lleve a cabo la tarea que deseemos.

La idea detrás de una hoja de cálculo consiste en montar una especie de formulario capaz de recalcular automáticamente los resultados cuando se cambie algún dato. No estará de más poner un ejemplo sencillo. Imaginemos que nos interesa controlar nuestros ingresos y gastos para estar siempre al corriente de cuanto dinero nos queda. Para hacerlo con una hoja de cálculo podríamos definir un formulario tal que así:

---

70 Números normales, fechas, horas, y demás.

71 He comprobado personalmente, como cualquier usuario de hojas de cálculo, la posibilidad de incluir todos los tipos de datos citados hasta llegar a los gráficos. Desgraciadamente, mi interés en las animaciones, sonidos y secuencias de video es casi inexistente, y jamás he intentado manejarlos con uno de estos programas, por lo que no puedo aportar ningún tipo de precisión sobre este punto.

72 La palabra no se emplea con un significado muy preciso, y acaba siendo una denominación general para cualquier posibilidad de automatización. Abarca desde el almacenamiento y ejecución cuando se desee de series de pulsaciones de teclas hasta auténticos lenguajes de programación “internos”, es decir que sólo existen y funcionan dentro de una aplicación. En ocasiones se intenta precisar y a lo primero se le llama estrictamente “macro” y a lo segundo “lenguaje de macros”. No está muy claro que se gane mucho.

	A	B	C	D...
1	Control de saldo			
2				
3	Ingresos	Gastos	Saldo	
4	5000		5000	
5		3000	2000	
6		10000	-8000	
7	4500		-3500	
8	3500		0	
9	9000	6000	3000	
10	_____			
...				

Aparte de aclarar que los puntos suspensivos son una manera tan rústica como el resto de indicar que la tabla continuaría tanto con filas como con columnas adicionales, veamos con un mínimo detalle qué tenemos en tan troglodítico ejemplo.

La celda A1, ya que conviene saber que cada celda se designa de igual modo que si estuviéramos echando una partida al popular juego de los barcos<sup>73</sup>, ha sido definida como una cadena de caracteres que contiene una mera identificación de nuestro formulario. Como se trata de controlar el saldo, pues lo ponemos, para que se vea. No es ningún detalle de narcisismo, que resultaría impropio al tratarse de algo tan burdo, sino que siempre conviene tomarse la molestia de aclarar qué estamos haciendo con el ordenador. Las cosas se olvidan y a todos nos ha pasado abrir un fichero y perder algún cuarto de hora que otro intentando desentrañar para qué diablos servía cuando lo montamos hace unos meses. Hay que curarse en salud a base de dejarnos notas a nosotros mismos. Cual modernos Pulgarcitos, debemos ir dejando piedrecitas para poder encontrar fácilmente el camino dentro del ordenador en el futuro.

En resumen, A1 contendría la cadena de caracteres “Control de saldo”. Cadenas de caracteres adicionales en A3, C3, y E3 identificarían mediante rótulos el significado de dichas columnas hasta que no indicáramos lo contrario. En consecuencia, las celdas A4..A10 se destinarían a contener datos numéricos correspondientes a ingresos (siempre exiguos, como puede verse) que podemos suponer semanales, mientras C4..C10 se dedicarían a albergar los gastos. No hace falta llenar todas las celdas. Podemos dejar vacías las que no necesitamos. De hecho suele hacerse así para separar las cosas y facilitar la lectura. El truco viene ahora.

---

73 Desconozco las costumbres de los niños modernos en horas de escuela, pero en nuestros tiempos nos dedicábamos con frecuencia a simular batallas navales equipados tan sólo con una hoja de papel cuadriculado y el correspondiente lápiz. Era necesaria además cierta habilidad para eludir la vigilancia del maestro y alguna perspicacia para establecer la psicología del adversario. El juego solía desarrollarse mediante expresiones propias de una comunicación con radioteléfono. “A3” decíamos. “Agua” nos contestaban. “B7”, decía el enemigo cuando era llegado su turno. “Tocado”, no nos quedaba más remedio que contestar al tiempo que proferíamos mentalmente un “¡Maldición!”. Y así hasta que el número de “Hundidos” nos llevaba definitivamente a pique. Tal ocupación contribuyó indudablemente a bajar algún punto nuestras calificaciones medias, pero puede que en aquellos tiempos no importara tanto. Luego ha resultado que además hizo que las hojas de cálculo nos resultaran inmediatamente simpáticas a unos cuantos. Alguien debía saberlo y hasta es posible que se tratara de nuevo de algún complot dirigido a facilitar la precoz informatización de la tropa, pues en los últimos tiempos y con las máscaras quitadas han aparecido modernas versiones comerciales que se llaman justamente “Hundir la flota por ordenador”.

## CAPÍTULO 8. PeCés: Programas

La columna de “Saldo” debe totalizar los valores ella solita. Si tenemos que hacer las cuentas a mano nos bastaría con un editor de textos para ir escribiendo un balance y no se trata de eso. En consecuencia, E4 se definiría como una fórmula. Podría ser perfectamente algo así como “Visualiza el resultado de A4 menos C4”. E5 también tiene truco. A la hora de obtener el saldo debemos tener en cuenta no sólo los apuntes posteriores de ingresos y gastos sino también el saldo anterior. La definiríamos por lo tanto de forma parecida a “Visualiza el resultado de sumar al valor en E4 el contenido de A5 y restarle el de C5”. Las demás celdas de dicha columna podrían seguir explotando tal definición sin más que corregir los índices. Si optamos por una notación más adecuada tendríamos  $E6 = E5 + A6 - C6$ ,  $E7 = E6 + A7 - C7$  y así sucesivamente. Tras definir los campos, meteríamos las cantidades de ingresos y gastos en sus correspondientes celdas y el programa calcularía automáticamente los saldos. Satisfechos con el fruto de nuestro esfuerzo, quizá imprimiríamos lo anterior en papel (tal vez sin las referencias de posición de las celdas para que quedara más bonito) a base de seleccionar la correspondiente opción en el menú con que sin duda contaría cualquier hoja de cálculo con un mínimo de dignidad.

Puede que estemos pensando que no hacía falta tanta complicación para semejante idiotez y que una calculadora y un editor de texto nos hubieran permitido llegar a lo mismo. La auténtica ventaja de las hojas de cálculo consiste en que una vez definida la estructura particular que resuelve nuestro problema, la simulación de supuestos se convierte en un juego de niños. Para responder a la pregunta “¿Cuál sería nuestro saldo si, en lugar de nuestros irregulares ingresos, cada semana nos encontráramos con 10.000 pelas en el bolsillo?” bastaría con sustituir los datos en A4..A9 por los nuevos y automáticamente se nos mostraría el resultado, con resultados parciales y todo. Vendría a ser, por si la curiosidad lo corroe:

	A	B	C	D...
1	Control de saldo			
2				
3	Ingresos	Gastos	Saldo	
4	10000		10000	
5	10000	3000	17000	
6	10000	10000	17000	
7	10000		27000	
8	10000		37000	
9	10000	6000	41000	
10	—			
...				

La imprimiríamos también, sin duda. Y al hacerlo debería quedarnos claro que vamos ganando tiempo si lo comparamos con el trabajo que hubiera sido necesario de tener que volver a usar la calculadora y un editor. No sólo se gana tiempo, sino que los errores son mucho menos probables, y las correcciones más fáciles. Puede que el gasto de la tercera semana, desorbitado, no sea real, sino un error al introducir en la celda correspondiente las mil pesetas que de verdad nos gastamos. Bastaría corregir ese valor para que de nuevo todas las cantidades se ajustaran a la realidad. Veríamos automáticamente algo mucho más tranquilizador, algo que sin duda también enviaríamos a la impresora mientras rompíamos sin piedad las dos hojas anteriores.

	A	B	C	D...
1	Control de	saldo		
2				
3	Ingresos	Gastos	Saldo	
4	5000		5000	
5		3000	2000	
6		1000	1000	
7	4500		5500	
8	3500		9000	
9	9000	6000	12000	
10	_____			
...				

En resumen, una hoja de cálculo es un ejemplo palmario de la puesta en práctica de esas capacidades de simulación del ordenador que tanto ensalzamos en su momento, hace muchos capítulos.

Visto lo anterior, podemos completar la descripción con algunos detalles concretos de manejo. Ese subrayado que aparece en la celda A10 no es otra cosa que nuestro clásico cursor. Como ahora operamos sobre celdas, el cursor correspondiente, movido por las teclas de siempre o mediante el ratón si estamos usando una hoja de cálculo basada en Windows, se situará sobre una de ellas. Será en aquella en que se encuentre en un momento dado en la que se almacenen los datos, ya sean una cadena de caracteres, un número o una fórmula. Como la estructura de datos está mucho más definida que en el caso de una base de datos, la mayoría de las hojas de cálculo proporcionan una serie de trucos para facilitar la introducción de los datos que manejan. Bastará que pulsemos una tecla alfabética, quizá la “T” de “Total” que podríamos poner como rótulo en A10, para que el programa suponga que va a ser una cadena de caracteres lo que queremos meter y abra automáticamente una ventanita con el editor de líneas correspondiente (en las hojas de cálculo más modernas se escribe en la propia celda) que nos permita completarla. Como siempre, la pulsación de retorno de carro confirmará el valor y lo dejará almacenado en su sitio. Un valor numérico se introduce igual, solo que en este caso, como es natural, se pulsa primero una tecla que corresponde a una cifra. Y como solemos desplazarnos en vertical u horizontal, no es necesario siquiera el retorno de carro. Pulsar la tecla de cursor correspondiente al desplazamiento que vamos a hacer basta para validar el dato que estamos introduciendo. Si estamos en A7 escribiendo “4500”, pulsar “cursor abajo” dejará el número en A7 y nos colocará en A8 para que sigamos. Las fórmulas tienen también su identificación. Si empezamos a teclear con un signo igual, un más, o un menos, el programa entiende que vamos a introducir una fórmula. Y como es una lata andar mirando las referencias, podemos mover el cursor a la celda a cuyo contenido queremos referirnos en la operación para que el ordenador la traduzca en la sigla correspondiente. Otro signo de operación nos devuelve a la celda en que estamos trabajando, a partir de la cual podemos reiterar el proceso con nuevas referencias hasta que pulsemos retorno de carro.

Podríamos seguir con multitud de detalles adicionales, pero lo anterior nos debe bastar para ser capaces de extraer las consecuencias necesarias al tema que estamos tratando, que no es otro que la importancia de la estructura de los datos en el manejo de los programas. Mientras que el problema de una base de datos era su intrínseca indefinición, debida a una estructura de datos muy general y por lo mismo muy potente y universal, que hacía preciso un trabajo considerable para ponerlas en marcha y hacer con ellas algo concreto, las hojas de cálculo se benefician de todas las ventajas asociadas a una estructura de datos más rígida. La facilidad de manejo aumenta porque el programa puede suponer muchas más cosas al estar el con-

## **CAPÍTULO 8. PeCés: Programas**

texto de operación mejor delimitado. Las hojas de cálculo siguen siendo programas razonablemente universales, pero su indefinición ya no es tan grave. Son programas intrínsecamente más útiles, aunque deban todavía programarse de uno u otro modo.

Podemos decir que son bases de datos con una estructura de datos predefinida pensando en la realización de cálculos. En consecuencia, pueden incluir un montón de operadores matemáticos de todo tipo (operaciones con vectores y matrices, cálculos de medias, desviaciones típicas, regresiones lineales, sumatorios, cálculos basados en fechas, operadores financieros, y otros muchos) que resultan de innegable interés en el tipo de uso al que se destinan, y realizar suposiciones correctas para facilitar la introducción de los datos.

Por supuesto, y como siempre, deberán incluir la capacidad de gestionar correctamente sus datos. Podrán copiar, borrar y pegar columnas, filas, y celdas individuales, bien vacías o bien conteniendo datos; almacenar los resultados en sus correspondientes ficheros; importar y exportar su contenido para que podamos emplear los resultados obtenidos con ellas en otros programas; y todas esas cosas absolutamente necesarias si se quiere hacer algo con un ordenador.

Para que se vea un poco a qué nos referimos, no sería necesario definir individualmente cada celda de la columna de saldos. A partir de E5 ya hemos dicho que la única diferencia estriba en las posiciones concretas de las celdas que intervienen en la fórmula, y todas comparten la misma relación. Podemos describir su contenido como “Suma al valor de la celda de arriba el contenido de la celda cuatro lugares a la derecha y réstale el contenido de la celda dos lugares a la derecha”. Pues bien, una hoja de cálculo sabe estas cosas y basta con copiar el contenido de E5 y pegarlo en el bloque E6..E10. Ella sola se encarga de hacer que  $E6 = E5 + A6 - C6$  y  $E7 = E6 + A7 - C7$ <sup>74</sup>.

Las hojas de calculo pueden en resumen considerarse auténticos programas de aplicación. Aunque exijan cierto esfuerzo de “programación” para ponerlas en marcha en algo concreto, puede considerarse que es el mínimo preciso para darles la necesaria flexibilidad. Al fin y al cabo, también un editor de textos nos muestra un fichero nuevo en blanco para que nosotros lo rellenemos con lo que queramos.

Si queremos estar al día de estándares y demás particularidades, procede un sucinto resumen histórico. Ya dijimos que los microordenadores tuvieron precisamente en las hojas de cálculo uno de sus grandes valedores a la hora de justificarse como algo útil. VisiCalc fue la primera importante, antes incluso de que los PeCés conquistaran el universo mundo. Su supremacía solo fue contestada por unos pocos programas alternativos. Por ejemplo Multiplan, de Microsoft, fue algo posterior y también gozó de cierta popularidad. De todos modos, el auténtico estándar para PeCé en los tiempos gloriosos del DOS fue Lotus 123, al que ya

---

74 ¿Y si no queremos que las referencias se cambien?. Tal vez E5 contenga en otro de nuestros formularios un valor fijo al que queremos poder referirnos directamente. Las hojas de cálculo trabajan por defecto con un “modo de direccionamiento relativo” que funciona como hemos descrito en el texto, pero es posible introducir direcciones absolutas, que no se actualicen al copiar, cuando se construye una fórmula. No debería sorprender a estas alturas que todo lo que haga falta sea emplear una determinada convención tipográfica. Basta poner A\$4 para que la referencia a la fila 4 se torne permanente, o \$A\$4 para que sea la referencia a esa celda concreta la que no varíe la copiemos doquiera que sea.

presentamos. Conviene recordar que incorporó gráficos y la posibilidad de realizar operaciones elementales de base de datos. Esto llevó a que a partir de entonces cualquier competidor debiera ser capaz de hacer al menos lo mismo. Un competidor interesante de conocer en el mundo DOS, quizá el más importante, es Quattro Pro, de Borland, al que también hemos citado con anterioridad. Aunque el dominio de Lotus 123 era aplastante en el mundo DOS, Microsoft guardaba una más que cierta capacidad de reacción. Fue durante mucho tiempo el proveedor principal de software de aplicación serio para el Apple Macintosh, el primer GUI importante de la microinformática, mucho antes de que se impusiera Windows. Si Lotus 123 reinaba en los PeCés con DOS, Excel, su hoja de cálculo para el Macintosh, dominaba el Mac. Cuando Windows gozó de la popularidad suficiente, Microsoft ya disponía de una hoja de cálculo para GUIs más que válida. Recuerde también que todos los GUIs se parecen mucho. En consecuencia, Microsoft sacó una versión de Excel para Windows y se convirtió también en el estándar de las hojas de cálculo para el GUI de los PeCés. Lotus no fue capaz de reaccionar a tiempo y perdió su liderazgo, aunque no se rindió y ha continuado desarrollando su producto que en la actualidad está probablemente a la altura de cualquiera. También Quattro Pro pasó con presteza al mundo de los GUIs para PeCé, y Quattro Pro 1.0 para Windows resultó un producto notable, con novedades tan importantes entre otras como la introducción del uso generalizado del botón derecho del ratón (los “menús objeto” de que hablamos en su momento) y unas capacidades de generación de gráficos bastante más que correctas. Esos tres siguen siendo los principales actores del mundo de los PeCés en la actualidad.

Hay algunas curiosidades de esta pequeña historia que conviene conocer, bien por graciosas o bien por interesantes en sí mismas. En primer lugar no me atrevo a citar versiones porque creo que fue en este ramo, sujeto a una competencia feroz, donde primero se puso en práctica un truco comercial que dificultó de manera notable su seguimiento racional. Como el número de versión de Excel y Lotus 123 era superior al de Quattro Pro 1.0 y eso les daba el aire de productos más depurados (creo que las versiones de la época eran Excel 4.0 y Lotus 123 para Windows 3.0), Borland decidió pasar de golpe de la versión 1.0 a la 5.0, amparándose en la similitud de características entre la versión DOS entonces en producción y la que hubiera debido ser Quattro Pro para Windows 2.0. Por supuesto fue imitada de inmediato y el resultado fue un cierto caos más bien simpático. Hablando algo más en serio, la competencia en este campo concreto llevó con rapidez a que la similitud entre los correspondientes productos de Microsoft, Lotus, y Borland, fuera algo más que notable. Cualquier innovación interesante de uno de ellos fue incorporada rápidamente a los demás. Por ejemplo, Excel incluyó la posibilidad de definir sumatorios con rapidez, Borland cambió la estructura de datos de forma trascendental pasando a usar “cuadernos” en lugar de hojas únicas<sup>75</sup>, y Lotus, me parece recordar, permitió que los datos se introdujeran directamente en las celdas. Todas esas nuevas características están presentes en las versiones actuales de Lotus 123, Excel, y Quattro Pro, ahora en poder de Corel.

En cuanto a formatos de ficheros propios de las hojas de cálculo, conociendo los sucesivos estándares deberíamos ser capaces de deducir automáticamente que existen dos fundamentales. Los ficheros en formato Lotus 123, y los ficheros en formato Excel. Son siempre

---

75 Borland definió un cuaderno como 256 de las hojas de antes. Se dispone en consecuencia de 8.192 x 256 x 256 celdas. Las hojas se designan mediante letras, de forma similar a las columnas, aunque si así se desea se les puede asignar un nombre algo más explicativo.

## ***CAPÍTULO 8. PeCés: Programas***

formatos binarios, no visualizables con un editor, y en principio más que notablemente complejos. Los ficheros de Lotus 123 ostentan las extensiones .WKS, .WK1, o .WK3, según la versión concreta de que procedan. Todas ellas surgen de un intento de abreviar “worksheet”, que es la denominación que se da en inglés a las hojas de cálculo. Los de Excel usan la extensión .XLS. La competencia ha hecho también que todas sean capaces de leer y escribir en los formatos estándar, por lo que lo anterior no deja de ser una curiosidad poco importante en la práctica. La compatibilidad reina en este sector de forma casi absoluta.

Aparte de los formatos de fichero propios, conviene también conocer los formatos de exportación que se emplean. Es muy frecuente que los datos y resultados puedan exportarse en ASCII mediante formatos análogos al .CSV que ya vimos. Por supuesto, se pierden las fórmulas, tipos y tamaños de letra, gráficos, y demás en el camino, pero si queremos conservarlos siempre se dispone de los formatos propios de las hojas de cálculo, ya suficientemente universales.

Y esto puede llevarnos de nuevo a las bases de datos. Aunque más rígidas, las estructuras de datos de las hojas de cálculo con básicamente tablas de una base de datos. Como hemos dicho hace poco, a partir de Lotus 123 todas permiten de hecho hacer una gestión limitada de bases de datos simples. Los formatos “hoja de cálculo”, tales como .WKS o .XLS se emplean en consecuencia muchas veces para intercambiar datos entre bases de datos, o entre ellas y las hojas de cálculo, y estas últimas son también capaces de leer ficheros tipo .DB. Nada tiene de sorprendente. Una vez más, las estructuras de datos subyacentes definen las posibilidades y el modo particular de operar de las aplicaciones.

Las hojas de cálculo resultan por todo lo anterior programas sumamente útiles. Son formidables para analizar supuestos o preparar informes que incluyan gráficos y necesiten ante todo la realización de cálculos de tipo financiero. Disponen para hacerlo de magníficas herramientas, entre ellas la generación automática de gráficos a partir de datos numéricos. Suelen encontrar sus límites ante problemas con fuerte contenido matemático que requieran solidez en los cálculos o frente a conjuntos de datos realmente grandes. Aunque ocho mil y pico filas parezcan un montón, a veces hacen falta más, y debe recurrirse entonces a los programas especializados en cálculo matemático que ya citamos. Esto no impide que sean con seguridad el segundo tipo en importancia de los programas de aplicación para PeCé si se tiene en cuenta su frecuencia de utilización. El primero, sin discusión posible, son los procesadores de texto. Pasemos a hablar de ellos sin tardanza.

### **Procesadores de textos**

Hablar de principios universales tiene la enorme ventaja de que, una vez que se les coge el aire, los nuevos temas se tratan con profundidad en un voleo. Es sólo un caso particular más en la línea de todo lo anterior. Mucho de lo que se ha explicado sirve para lo que se tiene que introducir.

Al hablar de los editores de textos ya nos vimos en la necesidad de introducir, siquiera indirectamente, a sus hermanos mayores. Los procesadores de texto comparten con sus familiares más modestos la estructura básica de los datos que manejan. Cadenas de caracteres, textos en resumen, por lo que no es necesario volver a reiterar lo ya conocido. La diferencia entre unos y otros consiste ante todo en el objetivo final que se persigue. Un editor debe crear un fichero de texto que muy probablemente no va a salir del ordenador en el que nació, y si lo hace será ante todo para acabar dentro de otro. Un procesador en cambio debe producir un texto que termine primero en papel y luego frente a los ojos de



alguien. Sirve para producir documentos terminados, que tal vez precisen incorporar diferentes tipos de letra, gráficos, ecuaciones, tablas, notas al pie, índices de materias, y todo ese montón de cosas que tan familiares nos resultan por haberlas visto y usado miles de veces en los libros. Un buen procesador de textos deberá ayudarnos a producir documentos presentables, y eso exige que sea capaz de entender qué es un margen, de generar automáticamente todo tipo de índices, de ayudarnos a evitar faltas de ortografía, y de aprovechar al máximo las características de nuestra impresora para producir una copia de aspecto realmente profesional. Si además nos va dejando ver nuestro documento tal como quedaría si lo imprimiéramos en cualquier momento mejor que mejor, pues no sólo será más agradable de usar sino que además nos ahorrará un montón de papel que acabaría inevitablemente en la basura. Harán falta muchas menos pruebas de impresión para lograr una copia final. No sólo eso, sino que además nos permitirá terminar antes el trabajo, pues imprimir cincuenta hojas lleva un rato. En resumen, mejor que sea WYSIWYG.

Estos requisitos obligan a realizar modificaciones en la estructura de datos a manejar. Ya no basta con una cadena de caracteres sin más. Hay que añadirle todo lo anterior. En qué lugar comienza un margen, qué encabezamiento y qué pie de página vamos a usar en un capítulo determinado, donde insertar un gráfico y cual va a ser su contenido, qué palabras deben aparecer en los índices, en qué tamaño de papel lo vamos a imprimir, con qué impresora y con qué tipo de letra en cada momento.

La forma en que se añaden todas estas cosas hace que aparezcan dos tipos diferentes de programas capaces de producir documentos. Los primeros, quizá los más generales pero más complejos de usar, suelen denominarse “formateadores de textos”. En ellos se guarda por una parte el texto en sí, el contenido del libro podríamos decir, y por otra se definen los elementos adicionales que se deben incorporar<sup>76</sup>. En el mundo de los PeCés, los programas más conocidos que recurren a este enfoque son los llamados “programas de autoedición” o “Desktop Publishers”. Si quiere algún ejemplo, citemos a voleo dos de ellos, Aldus PageMaker y Corel Ventura Publisher. Los dejaremos estar, ya que son programas que podemos considerar demasiado especializados<sup>77</sup>.

---

76 No se tome lo de “separados” estrictamente. Pueden estar en el mismo fichero. Estamos más bien hablando de que se distinga explícitamente entre el lenguaje de descripción del documento y su contenido. O, si lo prefiere, de que exista un lenguaje de descripción como tal.

77 De nuevo estamos trampeando un tanto. Cabe distinguir perfectamente entre “TEX”, un buen ejemplo de lo que acabamos de llamar “formateador”, un programa de autoedición, y un auténtico procesador de textos. Es convencional colocarlos en una escala de potencia y sofisticación en el orden formateador-procesador-autoeditor, por lo que hemos tenido la desfachatez de hacer equivalentes los dos extremos. Aunque no se si estoy muy de acuerdo con la clasificación y además es un poco idiota resaltar diferencias entre grupos que son más que parecidos. En fin, pelillos a la mar. Añadir los enriquecimientos al texto “desde fuera” tiene sus ventajas. Garantiza una mayor homogeneidad en la apariencia del documento final y permite crear los contenidos con otros programas tal vez más adecuados o simplemente más accesibles en cualquier condición. Se puede por ejemplo trabajar sobre un libro de 5.000 páginas en un ordenador de bolsillo con sólo 2 Megabytes de RAM y sin pantalla gráfica, usando para ello un editor de textos elemental. En algunos sistemas operativos, como por ejemplo Unix, los formateadores son lo más habitual.

## ***CAPÍTULO 8. PeCés: Programas***

Y los segundos, los auténticos procesadores de textos, simplifican en cierta medida la obtención de un documento a base de incorporar todos los elementos adicionales dentro de una única estructura de datos y de incluir en un único programa la casi totalidad de las herramientas necesarias para crearlos. La idea general es muy simple. Se intercalan en el texto, en el lugar en que deben actuar, las órdenes precisas. Y a partir de aquí los detalles se concretan como más convenga a cada cual.

Por ejemplo, las dos preguntas que surgen automáticamente sobre qué forma adoptan tales órdenes y como se distinguen del resto del texto, admiten diversas respuestas. Es posible definir las órdenes con caracteres ASCII no válidos como parte de un texto normal, pero es una solución más bien limitada. Se prefiere normalmente recurrir a lenguajes más complejos, que tal vez incluyan órdenes en ASCII, y simplemente se marcan para distinguirlas del resto en la forma habitual. Mediante alguna convención tipográfica. Por ejemplo, un carácter como el ASCII 27, nuestro amigo el código “ESCAPE”, no tiene demasiado sentido dentro de un texto, y es un magnífico “marcador de comienzo de código”. ¿Y si, por una remota casualidad, queremos que aparezca?. Tampoco hay problema. Se recurre a un truco también universal y simplemente se define una orden de formateo que lo inserte como tal carácter en forma de texto legible. Por ejemplo, la secuencia “ESCAPE ESCAPE” (dos ASCII 27) podría querer decir justamente eso.

A estas alturas, es probable que se esté pidiendo desesperadamente un ejemplo. Vayamos con él. Como a partir de una cierta edad resulta muy difícil inventar constantemente nuevos ejemplos, especialmente si, como es el caso del autor, siempre se ha carecido de imaginación en términos absolutos, retomaremos uno ya usado y lo adaptaremos a nuestras necesidades. Volvamos pues a Candanchú y veamos qué podríamos hacer para pasar de la versión sólo texto propia de un editor a la típica de un procesador.

La frase, versión texto estricto, sería:

Qué bonito es Candanchú con nieve, y qué bonitas vistas tiene.

Si nos ajustamos un poco al método de descripción propuesto en el apartado anterior, podemos suponer que “ESCAPE” marca el principio y el final del código y que las órdenes están en ASCII. Indicaremos el código de “ESCAPE”, el ASCII 27, como “Esc”. Pongamos algunos tipos de letra:

“Esc”Tipo:Garamond,Tamaño:10“Esc”Qué bonito es Candanchú con nieve,  
y que bonitas vistas tiene.

Con un editor la veríamos tal cual (con los “Esc” como un carácter incomprensible), pero un procesador que entendiera este código nos la presentaría como:

Qué bonito es Candanchú con nieve, y qué bonitas vistas tiene.

Añadamos más cosas. La versión:

“Esc”Tipo:Garamond,Tamaño:12“Esc”Qué bonito es “Esc”Negrita“Esc”Can-  
danchú“Esc”FinNegrita“Esc” con nieve, y qué bonitas vistas tiene.

daría lugar a:

Qué bonito es **Candanchú** con nieve, y qué bonitas vistas tiene.

y a partir de aquí debería estar claro como ir complicando la cosa. Una versión mucho más completa podría consistir en:

```
"Esc"Papel: A4, Orientación: Vertical, Impresora: Computex Laser
2000, Título: Grande"Esc"Oda a Candanchú"Esc"Linea, Tipo:Garamond,
Tamaño:12"Esc"Qué bonito es "Esc"Negrita"Esc"Candanchú"Esc"FinNegri-
ta"Esc" con nieve,"Esc"Linea"Esc"y qué bonitas vistas tiene."Esc"Lí-
nea,Linea,Gráfico:Vista Tobazo"Esc"
```

que nuestro procesador nos presentaría como:

### Oda a Candanchú

Qué bonito es **Candanchú** con nieve,  
y qué bonitas vistas tiene.  
(Y aquí vendría una hermosa vista)

En resumen, un procesador de textos es un programa que define una serie de códigos internos de enriquecimiento del texto para añadirle un formato adecuado para la producción de documentos de calidad y se encarga de interpretarlos y presentarnos el documento sobre el que estamos trabajando de forma más sencilla de modo que sea posible realizar las necesarias modificaciones con comodidad. Muchas veces se recurre a una visualización WYSIWYG para esto.

El código que hemos empleado en el ejemplo anterior no resultaría muy eficaz a la hora de almacenarlo y visualizarlo en plan WYSIWYG con agilidad ya que es excesivamente voluminoso. Los procesadores de textos prefieren en la práctica recurrir a códigos mucho más compactos y menos legibles pero que pueden manipularse de manera más rápida. El usuario no va a perdonar fácilmente que el documento no le sea mostrado con prontitud o que ocupe demasiado sitio al guardarlo en disco. En consecuencia, los códigos de formato no suelen ser fácilmente legibles. O si lo prefiere de otra forma, se guardan empleando códigos más o menos binarios. Y, como tal vez estemos ya sospechando, diferentes de un procesador de textos a otro, lo que hace que el fichero que hayamos creado con "The Final Word 4.0" no sea directamente legible con "Breeze", por citar un par de nombres que deberían sonarnos.

El problema no es tan grave, de todos modos. En primer lugar porque al ser como es la estructura de datos que maneja un procesador, es bastante probable que el texto esté dentro del fichero como tal, es decir legible con un editor, aunque rodeado de un montón de cosas raras. En caso de apuro, siempre se puede recurrir a nuestro editor favorito y recuperar al menos el contenido del fichero, que, no debe olvidarse, suele ser lo más importante. Conviene advertir, sin embargo, que puede no ser una tarea fácil. Y en segundo lugar porque también en este campo ha tenido lugar una notable estandarización alrededor de los programas más populares, con lo que un buen procesador no se limita a ser capaz de explotar su propio formato. También sabe manejar los diez o doce más importantes de la competencia.

Conocer los nombres y las extensiones de los formatos más populares implica, una vez más, estar un poquito al tanto de la historia. El primer editor de textos popular para Pecés fue uno que se llamaba WordStar. Lo editaba MicroPro, y fue durante mucho tiempo la referencia obligada del ramo. Hablamos en pasado porque nos hemos ido a los primeros tiempos del DOS, e incluso antes, a los tiempos del CP/M, y por supuesto semejante producto puede

## **CAPÍTULO 8. PeCés: Programas**

darse por desaparecido. Fue tan popular, de todos modos, que no debería sorprender encontrar algún fichero creado con él. Los procesadores de textos actuales, conocedores de la historia, suelen por lo tanto ser capaces de leer cualquier documento elaborado con alguna de sus múltiples versiones.

¿Qué extensión tienen los tales ficheros?. Sinceramente, no me acuerdo. Y esto nos da pie para traer a colación una cuestión que introdujimos subrepticamente hace muchas páginas y que conviene recordar. Las extensiones se empleaban para indicar el formato de un fichero. Podían permitir identificar el programa con que se había creado. Pero no eran el único modo de hacerlo. Otro sistema era emplear el principio del propio fichero para identificar de forma mucho más completa su contenido. Se hablaba entonces de “headers”, o si lo preferimos, “cabeceras”. De esta manera no se estaba obligado a respetar ningún tipo de convención impuesta por un sistema operativo determinado. Aunque DOS no nos deje poner extensiones de más de tres letras, podemos emplear un “header” de dos mil caracteres para identificar inequívocamente el contenido de un fichero. Pues bien, los procesadores de texto no son particularmente estrictos en el respeto a una serie de estándares sobre extensiones y recurren con bastante frecuencia a identificar los documentos mediante “headers”. Traducido, el procesador de textos deberá saber a qué tipo de fichero se enfrenta en un momento determinado, para lo cual tendrá que conocer la lista de los “headers” más frecuentes.

Para el usuario, es un punto en el que no conviene intervenir. No resulta nada fácil ni siquiera encontrar una descripción de los mismos. Confiados en sus “headers”, muchos procesadores de textos ignoran por completo las extensiones de los ficheros, o se conforman con usar un genérico “.DOC”, abreviatura de documento, que puede querer decir cualquier cosa y que sólo permite suponer que el contenido del tal fichero es probable que esté en algo más complicado que un código ASCII o ANSI. Traducido de nuevo, si vemos un fichero .DOC, cabe la posibilidad de que esté en formato “WordStar 2.000 3.0”, y debería ser el procesador de textos con que queremos leerlo el que supiera como aclararse con él<sup>78</sup>. A pesar de este tipo de prácticas aberrantes, nadie le impide a Vd. usar las extensiones con un poco de coherencia. No sería mala idea guardar los ficheros que produjéramos con WordStar usando la extensión .WS, o los de WordPerfect, por citar otro que ya ha salido, con la extensión .WP. Incluso podríamos distinguir entre versiones recurriendo a extensiones más expresivas, del tipo .WP4, .WP5 o .WP6. A nuestro procesador de textos le daría igual y seguiría con sus “headers”, pero nosotros podríamos saber de inmediato cual fue el programa que usamos para crear un fichero determinado.

Existen algunos formatos que conviene conocer por distintos motivos y con los que también se usan las extensiones de manera inequívoca. Uno de ellos es el denominado “Rich Text Format”, que emplea la extensión .RTF, una especie de formato universal para “texto enriquecido”, que es lo que quiere decir la traducción. En principio puede considerarse una alternativa razonable para pasar datos de un procesador a otro sin perder todas las caracte-

---

78 Al menos en el caso del WordStar normal (la versión 2.000 fue de las últimas y desconozco sus características) el recurso a un editor de textos seguiría siendo una posibilidad a considerar. Los códigos de formato se indicaban en el antepasado de los actuales procesadores mediante secuencias bastante legibles que empezaban con un punto. Se hablaba de “dot commands”, es decir “órdenes punto”. Si nos viéramos en la necesidad de entrar a la brava en uno de sus ficheros, aún tendríamos alguna posibilidad.

rísticas del documento, pero desgraciadamente no se recurre a él con asiduidad. Suele ser más frecuente intercambiar información usando el formato nativo de cada procesador.

Otro que está cada vez más de moda es el formato HTML, cuyos ficheros lucen de forma universal la extensión .HTM. El nombrecito es la abreviatura de “Hypertext Markup Language” y ha ganado en popularidad con la generalización de Internet. Existen varios niveles, pero lo dejaremos estar. Un documento de esos que nos encontramos en Internet (y ya hablaremos de Internet más adelante, no se apure) consiste tan sólo en un fichero para procesadores de texto capaces de entender ese formato. Lo de “Lenguaje de formato hipertextual” que es lo que vendría a querer decir la tal sigla, le viene de la capacidad de incluir enlaces entre distintas partes del texto o entre porciones de documento que se encuentran en distintos ficheros e incluso en diferentes ordenadores. Eso lo hace ideal para el fin que se persigue en Internet, la navegación por un cúmulo de información en principio dispersa en muchos lugares.

Ahora mismo, es casi obligatorio que un procesador de textos que se precie sea capaz de usarlo con mayor o menor soltura. Al contrario que los formatos con códigos ilegibles, HTML es un auténtico lenguaje de estructuración del texto, completamente en ASCII. En realidad es uno de esos lenguajes más propios de los “formateadores de textos” que de los procesadores de que nos estamos ocupando<sup>79</sup>. ¿Que le gustaría saber qué pinta tiene un documento listo para entrar en el hiperespacio informático, uno de esos que deben llenar de contenido las autopistas de la información?. Ya lo sabía, querido LAO, y por lo tanto aquí está la versión HTML de nuestra “Oda a Candanchú”. La cosa luciría tal que así:

```
<html>
<head>
<title>Oda a Candanch&uacute;</title>
<meta name="Author" content="Ken O'Tefies">
<meta name="CreatedBy" content="WP Internet Publisher 6.1">
</head>
<body>
<h1>Oda a Candanch&uacute;</h1>
<p>

Que bonito es <b>Candanch&uacute;</b> con nieve,<p>
y que bonitas vistas tiene.<p>

<p>
<p>

<p>

</body>
```

---

79 Ya dijimos que la diferencia era sutil. Lo auténticamente característico de un formateador de textos no es que exista un lenguaje de descripción del documento, pues también existe en los procesadores, sino que resulte accesible mediante un editor de textos. Es decir, debe estar en ASCII para poder trabajar sobre él con facilidad. Pero los procesadores actuales, y los ordenadores de hoy en día, tienen potencia suficiente para emplearlos también. Al fin y al cabo se trata tan sólo de prepararlos para que usen un formato más.

## **CAPÍTULO 8. PeCés: Programas**

</html>

Sabiendo lo que ya debería saber, tendría que poder entenderlo en buena medida. Por si necesita pistas, los códigos se delimitan ahora con los caracteres “<” y “>” e incluyen cosas tales como el título de la obra (“title” en inglés), su autor, el procesador con el que se creó, y pueden incluirse gráficos con diversas opciones de alineación, entre otras muchas cosas.

A la hora de plantearse el usar uno u otro formato con su procesador de textos debe tener en cuenta que la importación y exportación de datos entre ellos no resulta una cosa fácil si lo que se pretende es mantener intacto el aspecto final del documento. Los lenguajes internos de descripción son demasiado diferentes entre sí. Algo tan banal como pasar un documento de un ordenador a otro, usando el mismo procesador de textos en idéntica versión en ambos equipos, puede tener sorprendentes resultados simplemente porque la impresora disponible sea diferente. Recuerde que no en vano se habían desarrollado cosas como Acro bat o Envoy, que nuestro experto se entretuvo en instalar. No es tan descabellado como pueda parecer el intercambiar el texto en ASCII o en ANSI, los gráficos en un formato estándar de los muchos que existen, y el resto de elementos (tablas por ejemplo) en algún otro formato adecuado a su contenido.

Un catálogo de procesadores de textos famosos para DOS y Windows sería bastante extenso, pues no en vano se trata con seguridad del tipo de aplicación más popular. En lugar de dedicarnos a citar un listado que necesariamente incluiría un alto número de cadáveres<sup>80</sup>, nos limitaremos a dar a conocer los más populares en la actualidad.

Ahora mismo, los procesadores de textos suelen correr bajo Windows, y los tres más importantes sin discusión posible son los de Microsoft, Lotus y Corel (que recientemente compró WordPerfect a Novell, su anterior propietario). Se llaman respectivamente Word, WordPro, y WordPerfect, y no se pretende citarlos en orden de importancia, calidad, o preferencia de ningún tipo. Al igual que hicimos con las hojas de cálculo, correremos un estúpido velo sobre el número de versión de cada cual. No merece la pena seguirles la pista.

Su historia es bastante dispar. Word comenzó hace mucho como un procesador de textos para DOS, y ya gozó en aquellos tiempos de una notable popularidad. Se adaptó con rapidez a Windows, no en vano es un producto de Microsoft, y logró hace ya tiempo una notable parte del mercado. En su momento, hubo usuarios del Pecé que hablaban maravillas de Word 2.0 para Windows. Creo que la versión DOS dejó de existir hace bastante, aunque no me atrevo a afirmarlo con seguridad. WordPro, el producto de Lotus, nació en cambio con Windows. Se llamó inicialmente Ami y luego AmiPro, y lo desarrolló una pequeña productora de software que fue comprada por Lotus, que a su vez, como ya se dijo, resultó absorbida más tarde por IBM. Recientemente, Lotus decidió cambiarle el nombre, y de ahí salió el actual WordPro. WordPerfect, por su parte, ha sido durante mucho tiempo el dueño y señor del proceso de textos para PeCés, con más o menos permiso de sus otros dos competidores. Como también explicamos en su momento, comenzó su historia al conseguir dominar el proceso de textos en DOS, y al contrario de lo ocurrido con Lotus 123 en las hojas de cálculo, fue capaz de sobrevivir a la irrupción de Windows. WordPerfect para Windows 5.2 estuvo listo a tiempo para mantener una notable fracción del mercado frente

---

80 ¿Se acuerda alguien de Lotus Manuscript o de Borland Sprint, por citar tan sólo un par de ejemplos?. Y no es que fueran malos. Simplemente, no sobrevivieron.

a la competencia de Word 2.0 para Windows y AmiPro. WordPerfect fue además de las pocas productoras de software que continuó desarrollando sus productos para DOS después de que Windows se hubiera convertido en hegemónico. Es probable, aunque tampoco me atrevo a decirlo con seguridad, que todavía se puedan conseguir versiones recientes para DOS. Aunque en ningún momento ha podido considerarse clínicamente muerto, WordPerfect ha tenido que soportar un excesivo número de compraventas que han afectado en cierta medida su evolución. Ahora mismo, en junio de 1.996, el producto pertenece a Corel y ha vuelto a ser distribuido con normalidad.

A la hora de decidir entre uno u otro de los grandes, podemos aplicar a los procesadores de texto lo mismo que dijimos para las hojas de cálculo sin excesivo temor de incurrir en disparates. Todos ellos son productos excelentes que se han imitado tanto unos a otros que resultan bastante equivalentes. Todos soportan un enorme número de formatos de documentos, de gráficos, y hasta de hojas de cálculo, con el fin de facilitar el intercambio de datos. La decisión termina por tomarse muchas veces en base a consideraciones accesorias tales como la imagen de marca, el precio, la existencia de ofertas particularmente ventajosas en el momento de la compra, o la experiencia particular de cada cliente con una marca determinada.

No es que no existan diferencias. Pero no son sencillas de localizar para un recién llegado. Por ejemplo, Word emplea una estructura de datos más propia de un programa de autoedición que de un procesador. Se ha basado siempre en “plantillas” que definen el formato del documento y que se superponen al texto como algo en principio ajeno a él. WordPerfect usa un sistema de códigos en línea básicamente idéntico al que nosotros estamos suponiendo y permite visualizarlos en su totalidad en cualquier momento. Pero también ha añadido la posibilidad de emplear plantillas, como su competidor. Por el momento es el único que incluye un corrector gramatical, incluso en castellano, además de los ya clásicos corrector ortográfico y diccionario de sinónimos<sup>81</sup>. En resumen, allá cada cual.

Terminaremos con ellos con una advertencia. El grado de sofisticación de los procesadores de textos para PeCé es más que notable. Todos tienen WYSIWYG, corrector ortográfico, algunos hasta gramatical, generan tablas, incluso calculan, y tienen de todo. Hasta demasiado. Como les ha ocurrido a las hojas de cálculo y las bases de datos, su éxito en cuanto al tamaño, a la ligereza del producto, ha sido lamentable. Son sistemas pesadísimos. Olvídense de emplear con dignidad un procesador de textos para Windows de los más recientes con un PeCé del que sólo pueda decirse que “no está mal”. Tal vez pueda hacer algo con 8 Megabytes de RAM y una CPU 80486 DX2, pero le aseguro que notará enseguida la diferencia si tiene la suerte de pasar a 16 Megabytes de RAM al menos y un procesador más potente, que los hay, como veremos en el último capítulo. Sólo la instalación completa de uno de estos monstruos puede consumir 30 o 40 Megabytes de disco duro, o quizá más.

---

81 Si quiere estar al tanto de los nombres en inglés, el corrector ortográfico suele designarse “spelling checker”, y el diccionario de sinónimos “Thesaurus”. Volviendo un poco a los correctores gramaticales, el de WordPerfect estaba inicialmente en inglés. La adaptación al castellano no parecía fácil, pero resultó bastante más que correcta. Se incluye a partir de la versión 6.1, me parece recordar. Por cierto y ya que estamos con adaptaciones a diferentes lenguajes, los procesadores de textos deben necesariamente disponer de herramientas adaptadas al lenguaje en que Vd. desee escribir. De hecho, algunos procesadores se distinguen del resto por su orientación “multilingüe”. Uno de ellos, por si le interesa, se llama Accent.

## ***CAPÍTULO 8. PeCés: Programas***

Ya casi hemos acabado. De las estructuras de datos que es obligatorio conocer sólo nos quedan los gráficos. Intentaremos acabar con ellos rápidamente para poder olvidarnos definitivamente de los aspectos más arduos del PeCé. Empecemos cuanto antes.

### **Gráficos**

La primera estructura de datos, de la que se derivan multitud de formatos estándar, quizá la más intuitiva, surge de la manifiesta necesidad de meter en el ordenador las imágenes que normalmente hallamos en la vida real. Fotografías, textos impresos, imágenes de televisión. Todas ellas suelen construirse mediante el mismo procedimiento. Se pintan puntos homogéneamente distribuidos en una superficie en los colores correspondientes. Es decir tenemos una matriz de puntos individualmente manipulables. Así, mas o menos, se genera una imagen de televisión, o se toma una fotografía, o se imprime un texto. Como el ordenador también necesita generar imágenes para sus propios fines (visualizar, imprimir), y este método de la matriz de puntos es muy flexible ya que permite sacar tanto gráficos como letras (pintables a base de puntos al fin y al cabo) puede considerársele la forma más inmediata de manipular imágenes de cualquier tipo en el ordenador.

En principio no parece mala idea aplicar tal método al ordenador, pues almacenar la imagen en una estructura que refleje su estado final sobre el papel o la pantalla nos va a ahorrar trabajo a la hora de presentarla al usuario. Va a permitir trabajar con mayor rapidez.

La estructura de datos que se define tiene una estructura bastante simple. Si necesitamos una matriz de puntos, definimos una matriz de valores en la memoria. Existe una correspondencia directa. Digamos que se hace un mapa en memoria de la imagen a manipular, en el que se le asigna a cada punto un valor, manteniendo de alguna forma el orden en que aparecen. Surgen así los ya conocidos gráficos “bitmap” o de tipo “mapa de bits”, aunque suele preferirse el inglés. Sólo habrá que considerar dos aspectos de la imagen para tenerla totalmente caracterizada. Su definición y el número de colores que soporta. En función de uno y otro hará falta guardar en la memoria más o menos valores de pixels y cada uno de ellos requerirá más o menos bits. Si vamos a almacenar y procesar una imagen de 10x10 centímetros de tamaño con una resolución de 10 puntos en vertical y 10 en horizontal por centímetro en blanco y negro, necesitaremos  $10 \times 10 \times 10 \times 10 = 10.000$  valores en memoria cada uno de los cuales tiene 2 posibilidades. Un bit bastará para almacenar el color de cada punto, y con diez mil bits, algo más de mil bytes, saldremos del paso. Por supuesto, la cantidad de memoria necesaria para almacenar el mapa de bits de imágenes más definidas y con mayor colorido será mayor. Si tenemos una buena impresora en color, que nos deje imprimir aunque sea en 256 colores con una resolución de 120 puntos por centímetro (más o menos 300 puntos por pulgada) cada punto necesitará un byte (256 valores) y harán falta  $20 \times 20 \times 120 \times 120 = 5.760.000$  bytes para guardar una imagen de 20x20 cm. Ya lo vimos al hablar del modo gráfico y no debería hacer falta insistir.

Pongamos un ejemplo. Una línea sería, en uno de los cuadros con resolución 10x10 de antes:

```
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0
```



```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

¿Qué pasa si queremos tener más colores que el mero blanco y negro?. Basta con aumentar el número de bits asignados al almacenamiento de cada pixel. Un byte por pixel nos dejará trabajar con 256 niveles de gris o 256 colores, lo mismo da. Si el rojo es el color 130, y el blanco el 255, la línea anterior pero en rojo sobre fondo blanco, quedaría:

```

255 255 255 255 255 255 255 255 255 255
255 130 255 255 255 255 255 255 255 255
255 255 130 255 255 255 255 255 255 255
255 255 255 130 255 255 255 255 255 255
255 255 255 255 130 255 255 255 255 255
255 255 255 255 255 130 255 255 255 255
255 255 255 255 255 255 130 255 255 255
255 255 255 255 255 255 255 130 255 255
255 255 255 255 255 255 255 255 130 255
255 255 255 255 255 255 255 255 255 130

```

Puede que lo anterior esté claro, pero nos hemos saltado un paso sin explicar. ¿En base a qué el rojo es el 130 y el blanco el 255?. La introducción de colores se hace en el ordenador mediante lo que se denomina una “paleta”, que consiste simplemente en la asignación de valores numéricos a los colores disponibles. Se crea un código numérico de colores en el que, en el caso anterior, el 130 es rojo (un matiz determinado de rojo) y el blanco es el 255. Es posible modificar el código de colores, la paleta, para cambiar por completo la visualización de una imagen. Y es también posible que cuando leamos un gráfico no se nos presente con los colores correctos porque la paleta que se esté usando no sea la apropiada.

Aunque pueda parecer una buena idea y tenga una utilidad innegable por adaptarse perfectamente a un montón de casos reales, este tipo de almacenamiento de los gráficos, esta estructura de datos, no deja de tener inconvenientes claros. No hace falta más que analizar un poco lo que pasa al generar la línea del ejemplo anterior. El programa de manipulación de gráficos bitmap con el que la he podido pintar me ha dejado trazarla, con total seguridad, mediante una orden análoga a “pinta una recta desde aquí hasta allá, en este color y con este grosor”, donde las posiciones se han indicado con toda comodidad mediante el puntero correspondiente, probablemente un ratón. Sin embargo, una vez completada la orden, el programa ha generado el correspondiente mapa de puntos y lo ha almacenado en memoria en la matriz de bits necesaria. A partir de aquí la línea a desaparecido.

Dentro de el ordenador ya no tenemos una línea sino un conjunto de puntos que podemos manipular individualmente. La línea ya no está aunque nosotros la veamos. En el cuadro anterior yo veo la línea, Vd. también, pero el ordenador no. No puedo decirle que quite la línea anterior y la pinte en otro color o con una anchura distinta, o entre otros dos puntos<sup>82</sup>“Undo”, es a. En resumen, la estructura de gráficos bitmap es eficaz pero no conserva bien la información. Por si fuera poco, es enormemente voluminosa en cuanto queremos trabajar con un número de colores y una resolución digna, como debería haber quedado claro hace ya muchos capítulos. Este último punto hace que se pierda la ventaja de velocidad

---

82 Es difícil deshacer los cambios en un gráfico bitmap. Muchos programas de manipulación de este tipo de gráficos lo saben y ofrecen la posibilidad de volver atrás varios niveles. Esta capacidad de “deshacer” lo hecho,

## CAPÍTULO 8. PeCés: Programas

que argüimos inicialmente. Es más rápido generar la imagen a visualizar sobre la marcha, en la resolución adecuada a cada momento, que manipular una estructura de datos de 20 Megabytes. Y es sólo una primera consecuencia desagradable. El espacio de almacenamiento necesario para guardar 30 fotos a todo color y con detalle resulta monstruoso con los bitmaps.

Quizá Vd. esté pensando algo así como “Hombre, no puede ser tan difícil hacer que el programa sea capaz de aislar una serie de puntos contiguos tan bien definidos como los de la matriz anterior, o sea que el primer problema no parece tan grave.”. Ciertamente, pero sólo en el caso anterior, en el que tenemos una línea continua, sin interrupciones y en blanco y negro. Un caso real suele ser mucho más complejo. El ojo ve una línea aunque tenga pequeñas discontinuidades y esté sepultada entre un montón de puntos adicionales de diversos colores. Por ejemplo, una línea en color gris oscuro, con grises más próximos al negro cuanto más alto sea su valor numérico, podría almacenarse en la matriz bitmap anterior, más o menos de la siguiente forma:

```
1 2 2 2 2 2 2 2 2 2
1 7 5 3 4 7 2 2 2 2
1 3 7 6 2 2 2 2 2 2
1 3 3 7 1 2 2 2 2 2
1 1 4 3 5 4 2 1 2 1
1 2 1 1 1 7 1 2 1 1
2 1 1 1 1 1 7 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 1 0 1 1 1 0 1 1
```

Si se entretiene un poco verá que la recta sigue ahí, exactamente en el mismo lugar y sólo con una pequeña irregularidad, un punto algo menos oscuro que los demás. Pero ahora es mucho menos claramente aislable. Por supuesto, la cosa sería mucho más compleja de haber usado un número no muy elevado de colores pero más real, por ejemplo 256 niveles de gris. Si se ha revisado la vista recientemente, podemos decir que la situación se parece bastante a esos tests para detectar el daltonismo en los que se muestran cifras formadas por puntos coloreados en medio de una serie de puntos de otro color no muy diferente. Si Vd. no es daltónico, seguro que los vió. Podemos decir que el ordenador es absolutamente daltónico y no tiene el nivel de agudeza visual necesario para extraer la información existente en bitmaps mínimamente complejos, como los que se encuentran en la realidad. Es por esto que nuestro experto recomendó a su amigo que no confiara demasiado en la habilidad de Corel Trace para convertir gráficos bitmap en gráficos vectoriales. Y ya hemos llegado al otro tipo de estructura de datos gráficos de obligado conocimiento. Nos ocuparemos de ella enseguida, justo en cuanto digamos que, a pesar de sus defectos, las estructuras bitmap son necesarias porque de hecho se adaptan perfectamente al procesado de cosas tan comunes como las imágenes fotográficas y similares, a las que no se puede renunciar bajo ningún concepto.

Es necesario, ante este panorama, crear una estructura de almacenamiento de datos gráficos que conserve mejor la información con que se generó el gráfico para poderlo modificar con mayor facilidad y de una forma más natural. Hacen falta estructuras para almacenar gráficos “vectoriales”, para decirle al ordenador como debe trazar el gráfico de forma similar a lo que haría si tuviera una mano, un lápiz y un papel. También esta es una forma potente de definir los gráficos, si bien menos cercana a su visualización final. Podemos imaginarlas en principio como una especie de lenguaje de descripción gráfica que podemos suponer en

ASCII, y que incorporaría todas las instrucciones necesarias para guiar nuestra mano a la hora de dibujar. La línea del ejemplo anterior quedaría, si la almacenáramos en una estructura de datos de tipo gráfico vectorial, y aisláramos las órdenes en plan HTML, más o menos así:

```
<Tamaño de papel: 10x10 cm><Resolución: 10x10><Paleta: Rojo 130,  
Blanco 255,...><Fondo: Blanco><Trazo: Rojo><Grosor: 0'1 mm><Línea:  
1,1 a 6,6>
```

Resulta evidente que con este tipo de estructura es mucho más sencillo modificar un gráfico. Basta cambiar el valor asignado al grosor de la línea, o a su color, o a los puntos que definen sus extremos, para que obtengamos resultados diferentes. Los gráficos de tipo vectorial, aquellos que usan esta estructura de datos, se adaptan mucho mejor a las habilidades del ordenador. Por si fuera poco, es perfectamente posible incluir dentro de ella partes bitmap si es necesario, y es una forma de almacenamiento mucho más compacta, con lo que los gráficos así definidos se pueden almacenar más fácilmente. Por si no bastara, tampoco sufre algunos otros problemas derivados de la manipulación de gráficos bitmap. Uno de los más desagradables es la aparición de líneas escalonadas cuando se aumenta su tamaño. Hacer más grande una imagen de este tipo implica aumentar el grosor del punto, y el resultado final, a partir de determinado nivel, son líneas formadas por cuadrados tan evidentes como los unos y los ceros de nuestro primer ejemplo. Al ampliar un gráfico vectorial podemos mantener constante la resolución de la visualización o la impresión. Si trabajamos con una impresora de 300 puntos por pulgada, los bordes de nuestras líneas se imprimirán siempre con esa finura, independientemente de su grosor. Su nitidez será siempre perfecta.

La diferencia entre un programa “Paint” y un programa “Draw” no es, una vez más, otra cosa que un reflejo de las diferentes estructuras de datos que ambos emplean. También como siempre, las operaciones que ambos permiten realizar serán distintas y otra consecuencia de lo anterior. Los programas “Paint” podrán realizar cualquier operación que pueda definirse en términos de los puntos que constituyen la imagen. Será posible cambiar el color de un único punto, o hacer que cambie su valor en función de los valores de los que le rodeen. Es posible cambiar el contraste y brillo de una fotografía, o hacer más nítidos o más difusos los límites de las imágenes que en ella aparecen. Basta para eso con coger cada uno de los puntos que la forman y cambiar su valor mediante una función matemática que tome en cuenta los valores de los puntos contiguos. Se definen así multitud de filtros diferentes con distintos fines. Pero no será posible seleccionar y manipular elementos que nosotros vemos con claridad y que no aparecen por ninguna parte en el bitmap que el ordenador manipula. No podremos mover una casa de sitio manteniendo el fondo inalterado.

Por contra, un programa “Draw” ofrecerá con prodigalidad herramientas para seleccionar diferentes elementos de un dibujo, modificarlos de casi cualquier modo concebible, o generarlos mediante órdenes descriptivas. No será posible en cambio pintar con ellos a mano alzada usando algo parecido a un aérografo, o aumentar el contraste del dibujo si no es cambiando directamente sus colores por otros.

Lo de “metagráficos” o “metaficheros” que nuestro experto citó en su momento no es sino una precisión que en ocasiones se hace para indicar que una determinada estructura gráfica puede almacenar a la vez partes bitmap y vectoriales, algo no tan extraño, por lo que muchas veces se ignora y se habla simplemente de formatos vectoriales sin más. No todos lo permiten, de todas maneras.

## **CAPÍTULO 8. *PeCés: Programas***

Si intentamos extraer alguna consecuencia adicional de lo que acabamos de exponer, debería ser evidente que aquellos programas basados en gráficos que requieran habilidades especiales recurrirán a estructuras de datos más complicadas y adaptadas a sus necesidades. Por ejemplo, en el caso de un programa de CAM o CIM, de esos que permiten planear un proceso de producción asistido por ordenador, será necesario no sólo conocer la forma de una pieza, sino también el material con que debe fabricarse, sus tolerancias, y multitud de detalles adicionales. La estructura de datos que manipulen será más o menos gráfica, pero tendrá una parte notablemente similar a una base de datos. O, por poner un ejemplo más sencillo y probablemente más a nuestro alcance, será posible mezclar en cierta medida los gráficos bitmap y vectoriales para crear programas que ofrezcan las posibilidades más interesantes de ambos mundos mitigando en lo posible sus defectos. Igual que se hacen programas a la medida de un determinado problema, también las estructuras de datos pueden modificarse o crearse según sea necesario.

Con las estructuras vistas, ha llegado el momento de hablar de formatos de ficheros. Si partimos de la base de que deben reflejar la estructura con la que trabajan, y que por tanto habrá formatos bitmap, vectoriales, y metagráficos, resultará difícil perderse. Algo que de ignorar este hecho resulta sumamente probable, pues son numerosos como las arenas del desierto. Sólo por abrir boca, veamos una lista de extensiones propias de ficheros con gráficos tipo bitmap o “raster”, como también se los llama en ocasiones<sup>83</sup>. Agárrese: .PCX, .TIF, .BMP, .CLP, .CUT, .DIB, .GIF, .IFF, .IMG, .JIF, .JPG, .LBM, .MAC, .MSP, .PBM, .PCD, .PCX, .PGM, .PIC, .PNG, .PPM, .PSD, .RAS, .RAW, .RLE, .TGA, .TIF. No crea que hemos rebuscado demasiado. Todas son bastante comunes, aunque la lista, desde luego, es bastante completilla.

Pero en realidad existen muchos más. Las extensiones indican más un tipo genérico de formato que otra cosa. Como los gráficos bitmap son muy voluminosos, en muchas ocasiones se han ido desarrollando con el tiempo versiones avanzadas de los formatos que permiten guardarlos comprimidos. Si cogemos por ejemplo el formato TIFF (abreviatura de “Tagged Image File Format”, que no traduciremos porque ya está bien), correspondiente a la extensión .TIF en ficheros DOS, existen dentro de él las siguientes versiones: TIFF sin comprimir, TIFF con compresión RLE, TIFF con compresión Huffman, TIFF con compresión LZW, TIFF con compresión Fax Grupo 3 y TIFF con compresión Fax Grupo 4. Puesto que la extensión es única para todas ellas, se distinguen entre sí mediante los correspondientes “headers”, algo que resulta natural pues es preciso montar unas cabeceras bastante considerables para precisar como debe generarse un gráfico por más bitmap que sea. Como siempre, los programas se encargarán de distinguirlos.

---

83 Es posible traducir “raster” por “barrido”. Tal denominación quiere indicar que ese formato es adecuado, evidentemente, para su empleo en dispositivos de visualización “de barrido”. Una impresora o una pantalla son de ese estilo, pues en ellos un haz trazador (que puede ser un haz electrónico o el cabezal de la impresora) recorre homogéneamente la superficie de visualización generando la imagen a su paso. En cambio, los trazadores gráficos o “plotters” son periféricos en los que la imagen se genera con un mecanismo de tipo vectorial, y necesitan que los gráficos a imprimir se les envíen codificados en un lenguaje de ese tipo, normalmente HP-GL. ¿Que a qué se parece?. Como se trata de manejar una especie de mano robotizada, consiste en cosas tales como “Coge el color rojo, vete al punto (5,2), baja la punta, traza una línea hasta el punto (15,12)...” y así sucesivamente.

Pero, una vez más, conviene saber estas cosas<sup>84</sup>. Si Vd. emplea un programa de manipulación de gráficos “bitmap” para convertir una imagen de un formato a otro, algo sumamente frecuente, es probable que al intentar grabarla como .TIF le aparezca por algún lado una opción para seleccionar alguno de sus subformatos<sup>85</sup>.

En cuanto a formatos tipo vectorial o metagráfico, pues no vamos a hacer distinción entre ellos, una lista análoga se vería tal que así: .CDR, .CGM, .DRW, .DXF, .GEM, .HGL, .PIC, .WMF, .WPG. Es bastante más corta, pero podemos decir de ella casi lo mismo que antes. Por razones diferentes a las de los ficheros “bitmap” ya que ahora no es necesario comprimir<sup>86</sup>, existen numerosos subformatos. Por ejemplo, los ficheros de Corel Draw, manteniendo la extensión .CDR, han cambiado de una versión a otra y no son iguales los .CDR de Corel Draw 3 que los de Corel Draw 5.

Al igual que ha sucedido en otras ocasiones en que la cantidad de detalle parecía excesiva, es posible simplificar el asunto de los formatos gráficos sin perder demasiadas posibilidades en el camino. Aunque conviene que le suenen las extensiones y por eso las hemos puesto, para que pueda consultarlas cuando haga falta, es perfectamente posible limitarse a dos o tres formatos de cada tipo en el uso cotidiano del ordenador. No se pierde mucho usando preferentemente los formatos .TIF y .PCX e ignorando los demás hasta que, por el motivo que sea, nos tengamos que enfrentar a ellos. Tampoco está de más conocer los .PCD (“Kodak Photo CD”, el que se emplea para guardar en CD-ROM las fotografías digitalizadas) y los .GIF (“Graphics Interchange Format”, otro formato bastante universal). En cuanto a los vectoriales, son obligatorios los formatos .CDR, especialmente si dispone de Corel Draw en alguna de sus versiones, .CGM (“Computer Graphics Metafile”, otro de los universales, en este caso metagráfico, y tal vez llamado a grandes cosas en Internet), y el clásico .DXF (formato de intercambio de AutoCAD). Si está Vd. metido en un mundo con impresoras que usan el lenguaje PostScript, le interesarán los formatos .EPS (“Encapsulated PostScript”), y si usa WordPerfect es más que probable que acabe manejando con soltura el

---

84 Y quizá algunas más. Por ejemplo, muchas veces los ficheros “bitmap” se estructuran en distintos “planos”, se distinguen gráficos “entrelazados”..., hay casi infinitos detalles adicionales. Calma ante la adversidad, de todos modos. No se desespere prematuramente querido LAO. Como verá enseguida si sigue leyendo, no debe preocuparle demasiado.

85 Tantos tipos generan un alto número de potenciales incompatibilidades. Esa imagen que queremos pegar en nuestro texto usando nuestro procesador favorito puede estar en un formato que éste no sea capaz de leer. Quizá en un subformato particular de uno que sí reconoce en general. Es fundamental que los programas para manipular gráficos reconozcan un altísimo número de formatos diferentes y sean capaces de convertirlos entre sí. Y a veces se quiere cambiar el formato no de una, sino de quince imágenes. Viene de maravilla entonces el poder mandárselo con facilidad al programa para que lo haga automáticamente. El ya citado “Paint Shop Pro” se desenvuelve dignamente ante ambas necesidades. Pero es tan sólo un programa tipo “Paint”, por lo que también tiene sus limitaciones.

86 O sí que lo es. Puesto que un tipo metagráfico puede almacenar partes bitmap, no estará de más comprimir. Y a veces los ficheros gráficos siguen siendo demasiado voluminosos incluso empleando gráficos vectoriales. Corel usa habitualmente un formato .CDX para intercambiar datos entre sus distintos módulos y hacer el almacenamiento aún más compacto. Por cierto, tal vez haya ido notando que una “X” en una extensión suele indicar un formato de intercambio (“exchange” en inglés).

## **CAPÍTULO 8. *PeCés: Programas***

formato .WPG (“WordPerfect Graphics”). En resumen, como siempre decimos, simplifique al máximo. Ya lo irá complicando poco a poco.

Y si lo que desea es simplificar al máximo de verdad, recuerde que hay un formato .WMF (“Windows Meta File”) propio de Windows, a través del cual pueden pasarse entre programas casi cualquier tipo de gráficos. Y que, por si fuera poco, el portapapeles sigue estando ahí, que también admite gráficos, y que el “cortar y pegar” funciona en ocasiones sorprendentemente bien. Por no hablar de opciones aún más sofisticadas que Windows proporciona con cierta generosidad, tales como el intercambio de objetos, en el que de nuevo no entraremos. No confíe en exceso, de todos modos, en que tan maravillosos mecanismos simplificados sean la solución universal. A veces no marchan como debieran y aparecen efectos indeseados que no pueden resolverse sino intercambiando los gráficos a través de ficheros en un formato adecuado.

En resumen, no hace falta tanto detalle. Basta con los conceptos generales. Hay que saber que los formatos existen y conocer sus nombres y alguna característica sumaria (si son o no vectoriales, por ejemplo), para poderlos usar con provecho. Para buscarlos cuando queramos exportar algo, por ejemplo, y no asustarnos al ver los quince formatos distintos que nos muestre el programa. Pero nada más, a no ser que Vd. quiera programar<sup>87</sup>. Se trata de aprender lo necesario, los conceptos fundamentales, y de manejarlos. No hay que dejarse deslumbrar por los detalles. En informática son imposibles de manejar, y además no duran. Con las ideas generales y sabiendo donde buscar información adicional, vamos más que servidos. Más o menos lo que hicimos en DOS con los parámetros de ayuda o la orden “HELP”. Por cierto, las ayudas de “Paint Shop Pro” son también magníficas en lo que a formatos gráficos se refiere.

Pongamos un ejemplo a modo de resumen global de uso de programas e intercambio de datos mediante ficheros. Imaginemos que queremos crear un informe con WordPerfect en el que debemos incluir un gráfico de los ingresos que antes controlamos mediante nuestra hoja de cálculo. Imaginemos también que tenemos una impresora de inyección de tinta con resolución de 300 puntos por pulgada en blanco y negro para imprimir el resultado. ¿Cómo lo haríamos?.

Partiendo de que tenemos ya en la hoja de cálculo los datos numéricos, pues por algo los introducimos allí, lo más razonable sería emplear la propia hoja de cálculo para generar el gráfico, pues por algo nos ofrece generosas opciones de creación de gráficos tipo “chart”, justo lo que procede en este caso.

Usando la correspondiente opción del menú, pasaríamos de:

---

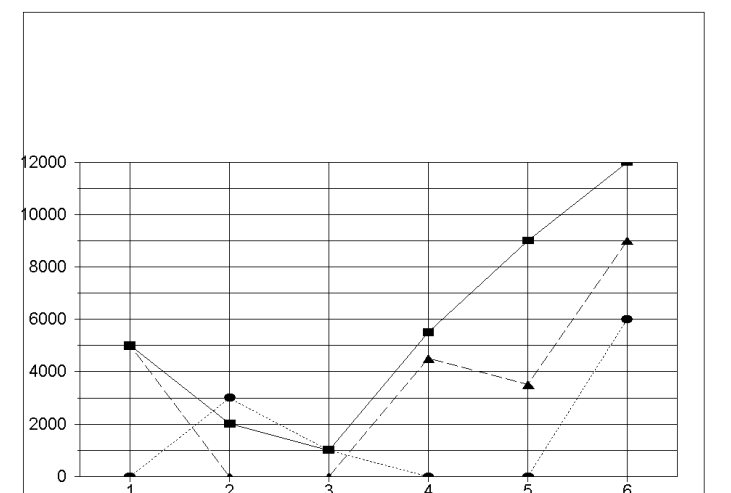
87 Y quizá ni siquiera entonces, pues tal vez su lenguaje de programación cuente con alguna biblioteca ya preparada para manejar diferentes formatos gráficos.

	A	B	C	D...
1	Control de saldo			
2				
3	Ingresos	Gastos	Saldo	
4	5000		5000	
5		3000	2000	
6		1000	1000	
7	4500		5500	
8	3500		9000	
9	9000	6000	12000	
10	—			
...				

a algo más o menos similar al gráfico siguiente que, evidentemente, pegaríamos a continuación en el documento que prepararíamos con el correspondiente procesador de textos, tras realizar su exportación en algún formato bitmap (más que suficiente para añadir una imagen a imprimir) o a través del portapapeles. Ni más ni menos que lo que se ha hecho para incluirla en el libro que está leyendo.

La imagen de marras no está mal pero tampoco está bien. Muestra algunas pequeñas aberraciones que están fuera de lugar en lo que debería ser un informe a todo lujo. Le sobra esa espantosa zona en blanco en la parte de arriba y, por si fuera poco, algunas de las cifras del eje vertical intentan salirse del cuadro. Habría que retocarla para obtener algo presentable. Podríamos pensar en darle un paseito por Corel Draw para al menos corregir los dos defectos citados.

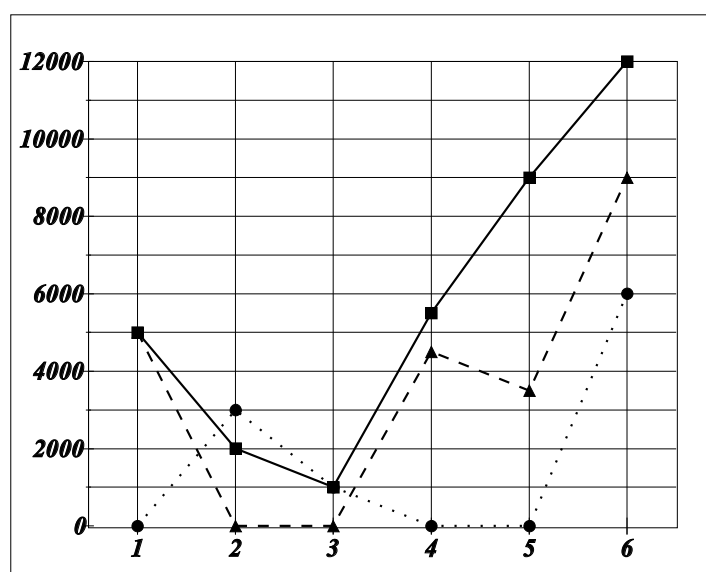
No es mayor problema, siempre que tengamos en cuenta lo que queremos hacer. Una vez en Corel Draw habría que seleccionar los valores en el eje vertical y desplazarlos un poquito. Sería absolutamente imposible si hubiéramos exportado el gráfico en un formato bitmap. Para poder manipular el “12000” hace falta que el programa pueda verlo porque esté en una estructura de datos que contenga algo así como “Rótulo: 12000”, y tal cosa no existe



## CAPÍTULO 8. PeCés: Programas

ni por asomo en un formato .PCX. Recuerde que sólo hay puntos individuales, en este caso en blanco y negro, en cualquier formato bitmap.

Podríamos pasar el gráfico a Corel Draw a través del portapapeles, o exportarlo en algo tal como .WMF o .CGM, que no están mal para estas cosas. De una u otra forma y tras unos toquesitos en Corel Draw y remitirlo de nuevo a WordPerfect mediante el mecanismo (adecuado) que mejor nos pareciera para que nuestro no tan hermoso gráfico acabara encontrando su lugar en el mundo, la cosa podría quedar tal como se muestra de nuevo en el gráfico siguiente que, deberá Vd. reconocer, querido LAO, “es lo mismo, pero no es igual”, como decían Martes y Trece.



Vaya por delante que no he hecho ningún tipo de trampa en el ejemplo anterior. El gráfico de partida no es un montaje deliberadamente incorrecto, sino el resultado que una hoja de cálculo bastante reconocida produjo de forma automática. Tal vez no me esmeré demasiado con él, pero desde luego no lo monté mal aposta. El ordenador parece adorar este tipo de resultados “casi correctos”, y es así como en la práctica funcionan muchas de las opciones automáticas que teóricamente están ahí para que el ordenador lo haga todo por nosotros.

Es probable que si Vd. intenta en algún momento repetir una manipulación similar encuentre multitud de vías alternativas a la utilización de la importación y exportación de archivos. Tal vez acabe inmerso en una u otra forma de “incrustación de objetos”. Pinchar dos veces el gráfico anterior proveniente de Corel Draw 5 a través del portapapeles haría que se lanzara una versión reducida de éste para facilitar su modificación. Puede que la cosa le encante y le parezca absolutamente preferible a las rudimentarias manipulaciones que acabamos de medio describir, pero no se entusiasme demasiado. Le advierto que, al menos en mi caso, para obtener una calidad de impresión como la anterior era absolutamente necesario exportar el gráfico desde Corel Draw en formato .PCX con 300 puntos por pulgada de resolución e importarlo a WordPerfect en ese formato. No debe ser la única vía para



conseguirlo, por supuesto, pero pasar el gráfico a través del portapapeles (una vez más algo mucho más cómodo y automático) conducía inexorablemente a uno de esos resultados “casi correctos”. En este caso concreto la calidad de impresión era lamentable. ¿Y para qué sirve un gráfico que se imprime mal?.

Acabemos de una vez con los gráficos pasando una acelerada revista a los programas más o menos estándar que conviene conocer. En cuanto a los de tipo “chart”, que hemos dejado un poco de lado y que, la verdad sea dicha, lo tienen un poquito difícil en los últimos tiempos por la competencia de las hojas de cálculo, decir que pueden ser una magnífica alternativa para la creación de gráficos a partir de tablas numéricas. Los mejores incluyen bastantes de las capacidades de los programas de dibujo vectorial y en ocasiones hasta tipos de gráficos predefinidos que no se encuentran fácilmente en las hojas de cálculo. Citemos dos, por abrir boca. Tal vez el más clásico sea Harvard Graphics, que comenzó su dilatada carrera en los tiempos del DOS y que ha seguido evolucionando bajo Windows. Si busca algo un poco más sofisticado puede intentarlo con Standford Graphics. En cuanto a programas tipo “Paint”, es obligatorio conocer a Photoshop, de Adobe. Puede considerarse el modelo en que se miran los demás. Pero ya dijimos que en este campo concreto existen muy buenas alternativas, incluso de shareware, como el tantas veces citado Paint Shop Pro. En cuanto al dibujo, o si lo preferimos los gráficos vectoriales, el estándar indiscutible del PeCé es Corel Draw. Y tenga en cuenta, repetimos, que no es en realidad un programa de dibujo vectorial, sino un conjunto de programas que cubre la práctica totalidad de los gráficos a excepción del CAD<sup>88</sup>. Y en este ramo es el archiconocido AutoCAD, de la compañía Autodesk, el que impera. Existen por supuesto productos alternativos tales como Microstation. Y si desea introducirse en el tema con un coste razonable, Autodesk tiene una versión “ligera”, el AutoCAD LT, y en tiempos existió un producto distribuido por Anaya llamado Drafix CAD que no estaba pero que nada mal. No se si se sigue vendiendo.

Si lo que busca es algo más aplicado a campos concretos, siguen existiendo un montón de posibilidades. Hay legión de programas dirigidos a la elaboración de presentaciones gráficas. Baste citar a Lotus Freelance, Microsoft PowerPoint o WordPerfect Presentations. Si desea elaborar imágenes tridimensionales, incluso en movimiento, lo suyo es 3D Studio, también de Autodesk, que cubre asimismo las animaciones más sencillas con su Animator. Y si busca algo que mezcle en sabia proporción lo vectorial con lo bitmap, quizá lo suyo sean productos tales como Aldus Freehand o Corel Xara.

Una vez más, la relación no pretende ser exhaustiva. Hay muchos más programas, tal vez mejores, que no se han citado. Sabiendo qué tiene que buscar, tiene un enorme terreno para explorar a su disposición.

Sólo nos restan las supongo que ya familiares notas para terminar con los gráficos. En este caso son bastante variopintas. Un término que quizá convenga que le suene es “render”, ya

---

88 Corel Draw 5 incluye hasta un programa bastante competente de “Desktop Publishing”, el Corel Ventura. Anteriormente se le conocía por Xerox Ventura, pero Xerox se retiró vendiendo su producto. Esto puso en su momento a Corel al límite de la irrupción en el proceso de textos. Un par de años después adquirió WordPerfect a Novell. Por el momento, las gamas de gráficos (Corel Draw) y de proceso de textos, hoja de cálculo, base de datos y demás, lo que en general se denomina “ofimática”, (Corel WordPerfect Suite) permanecen separadas.

## ***CAPÍTULO 8. PeCés: Programas***

que puede aparecerle en algún menú, sobre todo si usa programas tipo CAD o aplicaciones gráficas especializadas. Podríamos traducirlo en este caso concreto por “visualización”, o más correctamente por “generación de imagen”. Se trata simplemente de generar y normalmente presentar una visualización detallada del gráfico complejo que se está creando, a partir de los datos almacenados normalmente en una estructura de datos de tipo gráfico vectorial. Suele aplicarse casi exclusivamente a gráficos tridimensionales, y existen incluso algunos estándares específicamente diseñados para este tipo de operaciones<sup>89</sup>.

En segundo lugar, no estará de más subrayar de nuevo la tendencia a una descarada obesidad que presentan los gráficos de tipo “bitmap”. Quizá en el futuro, y tras el desarrollo de algún nuevo método de compresión, se conviertan en algo agradable de manipular y almacenar. Por el momento, el problema es de tal magnitud que llegan a usarse con ellos métodos de compresión que no permiten conservar la totalidad de los datos presentes en la imagen original. Se renuncia a parte de su calidad a cambio de reducir su tamaño a límites tolerables. El formato .JPG, que sigue la norma JPEG (ver glosario) es de ese estilo. A pesar de todo tipo de artimañas, los gráficos “bitmap” siguen siendo una notable fuente de problemas y son sólo tolerados debido a la absoluta necesidad de darles cabida en el ordenador. Si Vd. quiere encontrar los límites de su PeCé con DOS, Windows, y el programa de proceso de gráficos que prefiera, no tiene más que intentar meterle mano a un gráfico “bitmap” de diez o más megabytes, algo no tan difícil de obtener como pudiera pensarse. Aunque tenga 32 Megabytes de RAM y un ordenador en consonancia en cuanto a CPU, capacidad de almacenamiento, y velocidad de visualización, le garantizo que podrá apreciar en toda su extensión lo que significa querer realizar con el ordenador un trabajo que le viene manifiestamente grande. Y si es Vd. de esos que piensa en adquirir un PeCé para realizar ante todo ese tipo de trabajo, tenga en cuenta lo anterior y algunas cosas más, entre las que no es la menor la necesidad manifiesta de dotar a su PeCé de algún sistema de almacenamiento con la suficiente capacidad. Yo me inclinaría por incluir al menos un lector magnetoóptico de 230 Megabytes. Y es sólo el principio de lo que debería ser una larga relación<sup>90</sup>. Advertido queda.

En tercer lugar, y sobre todo por si no le gusta leer las notas al pie, ya que se trató levemente en alguna de ellas, recalcar que la multiplicidad de formatos y subformatos gráficos obliga a disponer de algún programa capaz de realizar con la mayor facilidad posible conversiones entre ellos. Es de agradecer asimismo que sea posible encargarle la conversión automática de varios ficheros a la vez, sin necesidad de ir de uno en uno. Tampoco está de más que el tal programa ofrezca con generosidad opciones de visualización y catalogación de los gráficos que vayamos acumulando con el tiempo. Como el contenido de un fichero gráfico debe verse para saber con un poco de exactitud a qué corresponde, este tipo de programas suele recurrir a la creación de imágenes reducidas de su contenido (“thumbs”, que no

---

89 Además de que sea una opción en programas especializados, algo que resultaría más bien irrelevante, existe una clara tendencia a incluir partes dedicadas a gestionar el “renderizado tridimensional” (perdón por la expresión) en las versiones más recientes o en preparación de los modernos GUIs.

90 Hace bastantes capítulos se citó la leve tentación que siente el autor a montar una especie de asesoría microinformática. Configurar un Pecé para un campo de aplicación concreto, en este caso el trabajo intensivo con gráficos, es un buen ejemplo del tipo de ocupación al que debería dedicarse tal empresa. Lo dejaremos estar, por si alguna vez me veo obligado a ganarme los garbanzos con eso.

traduciremos, es el nombre que se les da) que facilitan en buena medida la localización de un gráfico concreto. Por suerte, no es problema encontrar buenos programas que hagan todo lo anterior. En el campo del shareware podemos citar a Paint Shop Pro, a Graphics WorkShop, y a ThumbsPlus, y en cuanto a programas comerciales, Corel Draw incluye desde su versión 3.0 un módulo llamado Corel Mosaic más o menos con el mismo fin.

Programas como los anteriores se encargan más de facilitar el trabajo con el ordenador que de producir nuevas cosas. Este tipo de “operaciones de mantenimiento” son tan necesarias que de hecho dan lugar a un grupo de programas particular absolutamente necesario y que ya citamos. Las herramientas.

### **Herramientas**

En el caso anterior podríamos hablar de herramientas gráficas, pero tal denominación no suele emplearse. Herramientas mucho más familiares son los administradores de ficheros, o los programas dedicados a recomponer discos duros o disquetes con defectos surgidos por la causa que sea, o a realizar copias de seguridad, o a ayudarnos a configurar el ordenador. Todo ello tareas imprescindibles en el trabajo diario. Es normal que los programas dedicados a estas cosas se vendan agrupados para cubrir un número de necesidades suficientemente amplio. No podemos acabar un recorrido por el mundillo del software sin citar algunos de tales programas. En su momento presentamos a las Norton Utilities y las PC Tools, e indicamos que ambos productos habían sido absorbidos por una única compañía, llamada Symantec<sup>91</sup>. También hablamos de un administrador de ficheros para DOS, el llamado Norton Commander, que es bastante popular. Con eso debería bastar, si añadimos que por supuesto hay muchas más y que es totalmente imprescindible tener algún conjunto de herramientas a mano. Lo inevitable ocurre más pronto o más tarde, y es seguro que va a tener que recurrir a ellas en algún momento de su vida al lado de Luis Ricardo.

¿Cómo encajan los paquetes de herramientas en nuestro esquema organizado alrededor de las estructuras de datos?. Es posible definir una herramienta como un programa que, antes que definir una nueva estructura de datos y de realizar un trabajo basándose en ella, se encarga de usar las estructuras ya existentes, definidas con anterioridad por otros programas, con el fin de mejorar el funcionamiento general del ordenador o de aportar pequeñas utilidades que no requieren un almacenamiento de datos. Un programa de aplicación se emplea para crear ficheros en un formato determinado, adecuado a la tarea que debe realizar. Una herramienta sirve para mantener el ordenador en funcionamiento pero no crea nuevas estructuras ni, en consecuencia, define nuevos formatos.

Un paquete de herramientas usará las estructuras definidas de antemano por el sistema de ficheros del sistema operativo para reparar un disco duro, realizar una copia de seguridad,

---

91 Elegir entre uno u otro producto resultó desde el principio bastante difícil. Al igual que sucedió tiempo ha en el mundo de las calculadoras científicas, escindido en dos bandos irreconciliables entre los amantes de las calculadoras RPN de Hewlett-Packard y las de notación algebraica de Texas Instruments (y lo dejaremos estar porque no viene demasiado a cuento), una parte de usuarios del PeCé encontró algo parecido al paraíso en las Norton Utilities y un grupo no menos numeroso se decantó claramente por las PC Tools. Ambos son productos excelentes siendo claramente distintos y dedicándose a lo mismo, y la elección debe dejarse, necesariamente, a gusto del consumidor.

## ***CAPÍTULO 8. PeCés: Programas***

o permitirnos copiar con facilidad ficheros entre diferentes unidades. Deberá ser capaz, eso sí, de extraerles el máximo rendimiento. Incluso más que el propio sistema operativo. Si DOS borra un fichero, un paquete de herramientas deberá poder hacer no sólo eso, sino también recuperarlo si es necesario o destruirlo absolutamente para que no quede ni rastro de él y además sea irrecuperable. Deben poder operar en condiciones límite, justo allí donde terminan las capacidades incluidas en el sistema operativo. Son programas en los que un altísimo nivel de calidad es absolutamente necesario.

La definición anterior permite considerar como herramientas a un montón de esos pequeños programas más o menos inclasificables que siempre andan por ahí. Una calculadora, por ejemplo, puede ser considerada parte de nuestro conjunto de herramientas, o un calendario sofisticado, por citar tan sólo dos ejemplos habituales. La denominación suele aplicarse sin embargo, y de forma más estricta, a todo lo relacionado con el mantenimiento del sistema de ficheros del ordenador.

¿Qué puede aportar una herramienta tal como un administrador de ficheros perfeccionado que no esté ya disponible en el sistema operativo?. Un montón de cosas. Por ejemplo, la posibilidad de visualizar el contenido de cualquier fichero de forma adecuada a su contenido. Si el fichero corresponde a un gráfico nos dejará ver la imagen, si a un texto o un documento realizado con un procesador, nos permitirá leerlo correctamente, si almacena un programa, nos los presentará como tal<sup>92</sup>. O la de realizar un índice de nuestros ficheros que contengan textos de tal modo que luego sea posible localizar automáticamente cual es el que contiene la frase “hasta aquí hemos llegado” dentro de nuestro disco duro con 8.000 ficheros y decenas de directorios. En resumen, un montón de cosas sin las que se corre el riesgo de estar perdido en algún momento.

De todas maneras, no conviene ser demasiado riguroso con las clasificaciones y las teorías. Es preferible concentrarse más en el aspecto práctico. Por ejemplo, es posible discutir si los compresores de ficheros, es decir los famosos y sumamente necesarios PKZip, ARJ, o cualquier producto análogo, definen o no sus propias estructuras de datos. Al fin y al cabo crean ficheros con formato propio y hasta con extensiones sumamente estandarizadas, en concreto y respectivamente .ZIP y .ARJ. Y desde luego son claramente clasificables como herramientas. Pero puede argüirse que tales ficheros no corresponden sino a nuevos formatos, mejorados en cuanto a requerir menos espacio de almacenamiento, de estructuras ya existentes, en concreto en este caso las del sistema de ficheros. Aunque nos sintamos tentados de entrar al trapo y dedicar toda nuestra energía intelectual, siquiera por unos minutos, a apoyar sin reservas uno u otro punto de vista, conviene tener muy presente que lo realmente importante, más que la disquisición filosófica de si son churras o merinas, es que los programas encargados de realizar copias de seguridad o de comprimir datos crearán sus propios formatos y que pueden resultar incompatibles entre sí. PKZip no será capaz de reconstruir un disco duro a partir de una copia de seguridad creada con las PC Tools. Aunque las PC Tools si que pueden, en sus versiones más recientes, manejar ficheros comprimidos con PKZip. Es uno más de los múltiples formatos que reconocen.

---

92 Es bastante probable que lo que veamos en este caso sea un volcado hexadecimal, quizá acompañado de algún dato complementario. Para expertos.

Las herramientas no son los únicos programas que no crean datos. Otro grupo con parecido comportamiento y que probablemente le resultará más familiar son los juegos<sup>93</sup>. Y su número es absolutamente enorme dentro del PeCé. La popularización de estos equipos ha llevado a que lo que se propuso inicialmente como un ordenador con vocación seria y profesional, sea en la actualidad el soporte universalmente preferido para perder el tiempo. De todos modos, conviene no perder la perspectiva. Muchas veces es preciso relajarse ante los desmanes a que nos somete el PeCé, o cualquier otro ordenador si hemos de ser exactos, y puede que los juegos sean parte de esas herramientas tan imprescindibles. Devolver la cordura al humano que se sienta ante la pantalla es también una operación de mantenimiento absolutamente necesaria.

### **Juegos**

Un catálogo de juegos para PeCé nos ocuparía varias páginas. Intentaremos dar una rápida visión del tema citando algunos clásicos y clasificándolos de acuerdo a varios tipos generales. El primer grupo, casi aquel en el que todo el mundo piensa automáticamente ante la palabra juegos asociada a ordenador, es el que se ocupa de matar marcianos. Hay incontables ejemplos con diverso grado de sofisticación. Se ha llegado incluso a incorporarlos a algunas versiones del DOS<sup>94</sup>. Por supuesto pueden encontrarse múltiples versiones de Startrek, la popular serie de televisión y cine. O sea que si Vd. desea ponerse en la piel del Capitán Kirk o ser Mister Spock por algunas horas, su PeCé puede, una vez más, ayudarle en su propósito. A este tipo general suele conocerse como “juegos de arcade” en denominación más o menos técnica.

Bastante clásicos son también los juegos de aventura, normalmente ambientados en mazmorras, laberintos, mundos alienígenas, y demás lugares hipotéticos o remotos. Suele tratarse con ellos de buscar a una persona, llegar a recorrer un laberinto sin ser devorado en el empeño, o algo similar. Aunque últimamente se manejan a base de ratón, desplazando el correspondiente monigote por la pantalla, hubo un tiempo en que se trataba de establecer un diálogo a través del teclado, empleando para ello un lenguaje al uso. El ordenador decía “Estas en una habitación en la que hay una llave”, y mostraba una imagen estática del recinto. Se tecleaba “Coger”, y el ordenador respondía “Tienes una llave”. Y así sucesivamente. Estas eran las “aventuras conversacionales”, de las que todavía quedan furibundos partidarios.

Lo de matar marcianos requiere ante todo un mínimo de habilidad y un grado notable de reflejos felinos, pero no depende en demasía del planteamiento intelectual que se realice. Aunque parezca mentira, la experiencia parece demostrar que con eso no basta, y pronto los juegos de ordenador, más o menos de aventura o arcade, incorporaron notables sofisticaciones que exigen un mayor grado de reflexión. Aparecieron así los juegos genéricamente denominados “de estrategia”. Quizá el primero, o al menos uno de los más famosos y antiguos, fue el denominado Elite. Desarrollado originalmente para el Commodore 64, uno

---

93 No lea lo anterior como “jamás un programa de juegos va a crear un fichero”. Por supuesto que va a escribir en disco siempre que lo necesite. Guardará allí el estado de una partida que queramos interrumpir temporalmente y continuar en otro momento, o esas tablas de puntuación que nos empeñaremos inevitablemente en superar.

94 Novell DOS 7 incluía uno bastante respetable denominado Netwars. Era posible hasta jugarlo en red con varios jugadores al mismo tiempo.

## ***CAPÍTULO 8. PeCés: Programas***

de los micros de 8 bits ya desaparecidos, una breve descripción de su argumento puede darnos una idea bastante exacta de este tipo de juegos. El jugador se encontraba al mando de una nave espacial, con recursos limitados, en un cosmos ni hostil ni favorable en principio. A partir de aquí su oxígeno, su dinero, su combustible, su comida, y sus armas, se iban agotando en función del consumo que de ellas se realizara. El azar hacía que existiera la posibilidad de comprar combustible, comida, agua, cohetes, o reparar los daños o mejorar el navío interestelar que se estaba usando en alguna estación espacial. Pero por el camino podían encontrarse piratas que quizá intentaran saquearnos, y por supuesto todo lo anterior exigía gastar una pasta que quizá no tuviéramos. El protagonista podía dedicarse a la piratería, atacando a su vez navíos de carga y robándoles dinero y cualquier cosa de valor que encontrase, pero entonces la policía espacial lo perseguía como al fuera de la ley en que se había convertido. O quizá prefiriera convertirse en un honrado comerciante, o.... Por supuesto este tipo de juegos era mucho más completo que un simple aniquilador de marcianos, y tuvieron un notable éxito que aún sigue. Los aspectos de estrategia se han incorporado en la actualidad a muchos juegos.

Otro tipo bastante popular son los juegos que podríamos denominar de “habilidad no agresivos”. ¿Quién no ha oído hablar o no ha jugado alguna vez una partidita de Tetris?. Piezas formadas por cuatro cuadrados en distinta disposición caen del cielo y deben colocarse sin dejar huecos para ir las eliminando. Resulta francamente adictivo y pueden perderse horas intentando superar la máxima puntuación existente hasta el momento. Hay también multitud de ese estilo. Por ejemplo, todo aquel que posea Windows dispone de un Buscaminas incluido en el precio. Y otro gran número de PeCé adictos han oído al menos hablar de los Lemmings, una multitud de pequeños moñacos a los que hay que intentar salvar de la extinción sacándolos de un laberinto al uso. En fin, que también puede Vd. jugar a placer con un PeCé si le repugna lo de dar rienda suelta a su violenta naturaleza aunque sea de forma figurada.

Y no podían faltar a la cita las versiones de ordenador de los juegos habituales. Damas, ajedrez, solitarios de todos los tipos y colores, incluso uno incluido también en Windows, backgammon, y otros muchos, nos esperan para ayudarnos en el empeño de dejarnos los ojos en los monitores sitios en nuestras mesas. Nada que decir sobre ellos.

Y, por supuesto, simuladores de vuelo. ¿Le encantaría pilotar un avión, quizá un sofisticado caza a reacción?. Es posible hacerlo, con distintos grados de precisión en la simulación. Hasta puede intentarlo con la lanzadera espacial, el famoso “Shuttle” de la NASA. Los simuladores son otra gran familia, cuyo más famoso representante quizá sea el Microsoft Flight Simulator<sup>95</sup>.

Ya que estamos con simulaciones y aprovechando que al ordenador se le dan francamente bien, especialmente si, como es el caso de los juegos, no es necesario que sean muy precisas, otra gran familia se basa en la simulación de situaciones más o menos familiares. Intentar

---

95 El Microsoft Flight Simulator (“simulador de vuelo de Microsoft” por si a estas alturas todavía no se ha soltado con el inglés) ha estado disponible para los PeCés desde la noche de los tiempos. Hubo una época, incluso, en la que se distinguía si un ordenador era o no “compatible PeCé” realizando una sencilla prueba: si se podía ejecutar el Flight Simulator era compatible, y si no, no. La vida era entonces, por supuesto, mucho más sencilla.

controlar el funcionamiento de una ciudad, de La Tierra como ecosistema, de un imperio financiero basado en el ferrocarril, o de la misma civilización como aventura humana, pueden ser una maravillosa forma de jugar. Quizá hasta resulten ligeramente instructivos y todo<sup>96</sup>. Algunos de los mejores juegos para ordenador se basan en esta idea. Es obligatorio citar entre otros a Sim City, Sim Earth, Railroad Tycoon, y Civilization. Todos ellos pueden considerarse clásicos y bastantes han conocido, como Elite, versiones sucesivas que los han ido mejorando. Por ejemplo, el original Sim City hace tiempo que dio paso a Sim City 2000 (una simulación de una ciudad como espero que haya quedado claro hace rato, aunque seguro que no es tan evidente que "Railroad Tycoon" sea traducible por "El magnate del ferrocarril").

Los tiempos (mediados del 90), no parecen de todas maneras particularmente dados a nivel oficial al pasatiempo intelectual con intención siquiera lejanamente instructiva. No es extraño que una de las más recientes adiciones al campo del juego con ordenador sea lo que podríamos llamar, a imagen del cine, "género gore", o si se prefiere una traducción algo más castiza "juegos de destripe". Retomando la idea del juego de aventura y añadiéndole algún toque avanzado a nivel de simulación gráfica, por ejemplo un laberinto tridimensional en el que es posible desplazarse con soltura, el objetivo consiste ahora en aniquilar a los monstruos que se oponen a nuestro avance provocando en el empeño la mayor profusión posible de líquidos (o sólidos) corporales. Es posible ayudarse para ello de herramientas tales como una sierra mecánica, pistolas, fusiles de asalto, lanzamisiles, y desintegradores diversos, entre otras armas de distinto nivel de sofisticación. Quizá el clásico universal de este tipo sea el llamado Doom, que cuenta con un razonable número de imitadores. Como lo de destripar bichos puede tener su atractivo pero no acaba de convencer demasiado si somos nosotros los aniquilados a la primera de cambio, el complemento fundamental de este (y algún otro) tipo de juegos es un conjunto de "cheats". La palabreja es el equivalente inglés a "trampas" o "engaños", y corresponde simplemente a una serie de claves que nos hacen invulnerables a los ataques del enemigo, o nos dan infinitas municiones para nuestras armas, o nos permiten cualquier otro tipo de descarado ventajismo. Tales conocimientos suelen adquirirse a través del boca a boca o encontrarse en las publicaciones especializadas<sup>97</sup>. Casi podríamos decir que son una parte integral del juego<sup>98</sup>.

---

96 Siempre y cuando quede clara la diferencia entre la simulación realizada por el ordenador en el contexto de un juego y la realidad, necesariamente mucho más compleja y tal vez ni siquiera simulable si lo que se pretende es una descripción mínimamente seria. Algunos llegan a hablar de programas de "edutainment", curiosa fusión de los vocablos ingleses "education" y "entertainment". Habría mucho que discutir al respecto, pero una nota al pie no parece el lugar más apropiado.

97 Hay varias revistas dedicadas casi en exclusiva a los juegos para Pecé.

98 Quizá a Vd. le parezca que este tipo de juegos y prácticas muestran claras tendencias antisociales y deberían ser prohibidos terminantemente. No voy a discutir tal opinión, pero sí que me atrevería a rogarle que no sea excesivamente tajante al considerar este u otros temas. Supongo que habrá quien se tome en serio estas cosas, pero es bastante probable que alrededor de tales barbaridades flote una notable cantidad de humor negro, quizá de gusto bastante discutible, eso sí. O sea, que la gente que le da al Doom también pueden ser bellísimas personas. Para muchos de ellos, el monstruo al que se destripa representa ante todo al propio ordenador, que es quien más dolores de cabeza nos suele proporcionar.

## **CAPÍTULO 8. PeCés: Programas**

Si quiere alguna recomendación sobre juegos de indudable calidad y de los que no deba avergonzarse ante ninguno de sus amigos, ahí va. Si le gusta volar, déle al Flight Simulator de Microsoft. Si le encantan los pasatiempos con orientación más o menos matemática, los programas de generación de fractales producen imágenes de sorprendente belleza y aspecto caleidoscópico<sup>99</sup>. El arquetipo de estos programas es el llamado Fractint, que va por el momento por su versión 18 y pico. Si le fascinan las estrellas pero no le llega para comprarse un telescopio, echarse al monte, y empollar varios tomos de astronomía, un programa como Dance of the Planets puede ser aquello que siempre deseó aunque no lo supiera<sup>100</sup>. Y si le maravillaban en su niñez las máquinas del profesor Franz de Copenhague, o en su adolescencia le fascinaron los forgendros<sup>101</sup> y los inverosímiles dispositivos de las películas del cine mudo, The Incredible Machine está hecho para usted. Añádale los Lemmings, el Tetris, y algún otro pasatiempo similar, y esté seguro de que tiene tajo para quemar horas y más horas ante el ordenador sin hacer casi nada de provecho. ¡Ah!, y no se olvide de Civilization o de su sucesor, Civilization II.

Los juegos, abstracción hecha de su nula utilidad, son de los programas más sofisticados y exigentes de los que se dispone. Un buen juego es un programa de excelente calidad que estruja el ordenador, especialmente en cuanto a velocidad de cálculo y visualización, al límite de sus posibilidades. En consecuencia, tienden a ser más bien egoístas. Quieren todo el ordenador para ellos solos, y pueden presentar notables interferencias con el funcionamiento del resto de programas que tengamos instalados. Quizá no los pueda lanzar desde Windows, o deba modificar su CONFIG.SYS y AUTOEXEC.BAT para ejecutarlos desde DOS. Esté preparado para este tipo de cosas si quiere dedicarse a ellos. Y no es la única dificultad que deberá vencer. Por ejemplo, algo que tal vez no le haga demasiada gracia es que no es nada frecuente que se traduzcan al castellano<sup>102</sup>.

---

99 Me permito recomendarle el conjunto de Mandelbrot, siempre sorprendente, o los fractales basados en las fórmulas de Lyapunov. Pero hay muchísimos más, también de innegable belleza.

100 Si le gustan las estrellas, no basta con comprarse un telescopio. Hay que saber muchísimas cosas más. Y por si fuera poco, nuestras contaminadas ciudades a nivel del mar no son lo más adecuado para ver esa estrella de pequeña magnitud en la constelación de Orion que andamos persiguiendo. Por algo los observatorios se instalan en lugares remotos. Dance of the Planets, por citar sólo uno de los programas de “simulación de la cúpula celeste” aunque, eso sí, quizá el más completo, es un auténtico telescopio informático. Varios meses antes de que los periódicos siguieran la colisión del cometa Shoemaker-Levi con Júpiter, los poseedores de Dance of the Planets podían ver una simulación sumamente precisa del suceso en sus monitores. Incluso si Vd. tiene un telescopio y un buen lugar de observación, le interesa el programa para calcular efemérides, entre otras muchas cosas.

101 Forges, el famoso humorista, dedicó sus esfuerzos durante una temporada a pintar este tipo de cosas. El resultado fue que, también con esto, nos alegró la vida a unos cuantos.

102 Civilization II, al menos para mí, ha supuesto una agradable sorpresa en cuanto a la seriedad de las empresas dedicadas a entretener al público con el ordenador. Es el primer juego con que me he encontrado que tiene un nivel de producción realmente superior al de algunos programas serios de marcas bastante conocidas. Sirva como muestra el hecho de haber sido traducido completamente al castellano, incluyendo los manuales, con bastante cuidado, notable acierto, y admirable respeto por el sentido y espíritu de la versión original inglesa. Y a un precio más que razonable. No me atrevo a decir que sea revolucionario en este aspecto, ya que no conozco demasiado el sector y cabe la posibilidad de que dicho nivel de perfección sea habitual hace



Modificar una configuración, aunque sea para poder castigar aún más a una especie tan maltratada como la marciana, nos coloca al borde del aprendizaje serio de las interioridades más recónditas de nuestro PeCé. Si le encantan los mecanos, los rompecabezas, y las máquinas normales tienen pocos secretos para usted, pronto o tarde sentirá la fascinación de programar. Querrá hacer que su PeCé haga lo que usted le diga, no lo que un programa ya enlatado le deje hacer. Además, lo de programar viste mucho, y podrá proclamar a los cuatro vientos, para pasmo de sus amistades, que ese programita que tan bien le viene para hacer alguna tontería lo hizo usted.

Podemos atrevernos a decir que el último tipo de juegos de ordenador al que podemos jugar es justamente ese. Ir a lo esencial y hacer del propio PeCé el objeto del juego. No necesitamos matar marcianos cuando podemos dedicarnos a controlar, por si fuera poco en la realidad, a una máquina compleja que en cierta medida sustituye a todas las demás.

Le ruego que no se escandalice si es Vd. de los que considera que la informática y la programación son algo tan serio y trascendental en las modernas sociedades tecnológicas que roza lo blasfemo el hacerlo equivalente, aunque sólo sea medio en broma, a un juego de matar marcianos. Por supuesto que si algún adolescente que lea esto siente la llamada de la RAM y la ROM y desea dedicar su vida a la Informática (ahora necesariamente con mayúscula), hará bien en localizar los lugares en los que poder cursar con rigor tales estudios. Pero también, si lo que nos gusta son las moléculas y el material de vidrio, es posible comprarse un juego de Química y enredar con él y nadie se escandaliza por eso. Es probable que bastantes químicos competentes hayan empezado por ahí<sup>103</sup>.

Con la informática (ahora necesariamente con minúscula, es decir con el cúmulo de conocimientos sobre ordenadores que todo el mundo debería tener para no quedar más o menos fuera de juego en nuestra sociedad) ocurre algo bastante normal en el campo del aprendizaje científico o técnico. Programar, dedicarse a ella profesionalmente, es una tarea que exige una dedicación exclusiva. O quizá si no exclusiva, si al menos excesiva. O tal vez exclusiva

---

tiempo, pero es todo un detalle.

- 103 Canonizar de entrada cualquier ciencia hasta el extremo de que haya que acercarse obligatoriamente a ella entre reverencias y genuflexiones es una pretensión aberrante. Hay un lado “amateur” y de juego en el conocimiento humano que tiende a ignorarse o despreciarse en estos tiempos tan inclinados a venerar el profesionalismo en todos los órdenes. Se ha llegado a tal extremo que basta que fulanito diga “soy un profesional”, así, sin más calificativo, para que se le otorgue una especie de patente de corso para perpetrar desmanes. Un profesional competente merece todos mis respetos y lo sitúo en lo más alto de la escala, pero por debajo de él no estaría de más colocar al amateur igualmente competente en lugar de al profesional menos capaz. Auténticos genios han empezado enredando con las materias a las que luego han dedicado su vida. Me parece haber leído que Juan Oró, insigne químico, comenzó montándose un laboratorio de aficionado en el desván de la panadería de sus padres. Yo mismo, también químico profesional si bien mucho menos insigne, creo deberle bastante de mi vocación a un juguete llamado “Quimicefa”. Siempre lo deseé en mi niñez, aunque nunca llegara a poseerlo y jugar con él. Puede que por eso me decidiera a estudiar Química cuando llegué a la Universidad. Quizá, llevando la analogía hasta sus extremos, la diferente escala a la que Juan Oró y yo jugamos con la Química cuando éramos críos pueda explicar la escala asimismo enormemente diferente de la competencia profesional a que hemos llegado, aunque sospecho que, para mi desgracia, sería necesario tener en cuenta muchos más factores.

## ***CAPÍTULO 8. PeCés: Programas***

y excesiva a la vez. En resumen, hay que tener vocación y entregarse. Como la gente no es tonta y lo sabe, el usuario medio que no quiere ser otra cosa que eso huye de la programación como de la peste. Tal enfoque es el normal ante la Química, la Física, las Matemáticas, la Economía o la Filosofía, y suele funcionar aceptablemente con estas materias porque la gente de la calle no se encuentra todos los días ante un rotavapor, un detector de rayos cósmicos, o cosas similares. Es posible en consecuencia comprar de vez en cuando una revista de divulgación y limitar nuestros conocimientos sobre esas materias al estricto mínimo necesario para mantener una conversación inteligente con el barbero, que los hay sorprendentemente letrados por cierto. Por contra, las probabilidades de tener que sentarnos alguna que otra vez por semana ante un PeCé son desgraciadamente muy superiores y tienden a aumentar, por lo que hace falta saber más.

Esa dedicación amateur y de juego ante el ordenador viene pero que muy bien. Hace posible dedicar a familiarizarse con él un montón de horas que de otro modo sería imposible obtener. No creo equivocarme mucho si digo que el 90% de la gente no profesional, es decir sin un título que acredite sus conocimientos, que está involucrada a distintos niveles en el desarrollo de la informática en nuestro país, ha salido de ahí. Muchos expertos entre comillas han aprendido por libre, ante todo porque el ordenador les resultaba divertido, y han acabado ganándose mejor o peor la vida con esto como consecuencia de alguna de esas jugarretas que el destino prodiga.

### **Lenguajes de programación**

En fin, cortemos el rollo. De una u otra manera, quizá quiera aprender a programar. ¿Cómo hacerlo?. Si lo que desea es irse introduciendo poco a poco, le vendrá bien saber que los ordenadores se programan en algún lenguaje particular de los muchos que existen, y que lo primero de todo es elegir aquel que vamos a emplear para comunicarnos con Luis Ricardo. La elección no es sencilla, pues son numerosos como las arenas del desierto. Un lenguaje no es más que un conjunto de palabras, de órdenes que el ordenador entiende si lo prefiere, susceptibles de ser agrupadas en secuencias con la complejidad suficiente como para hacer algo útil. Al igual que ocurre con los lenguajes humanos, se diferencian entre sí de múltiples maneras. Unos son particularmente sencillos de leer, otros son sumamente potentes pero enormemente crípticos, los hay más rápidos y más lentos, éste es especialmente eficaz cuando se trata de definir y gestionar nuevas estructuras de datos y aquel es incapaz de añadir una sola a las que ya entiende de partida, el que yo conozco puede ir de perlas a la hora de hacer cuentas y ser casi inservible para manipular una base de datos. No existe algo como el lenguaje de programación universal, ya que muchas veces es el problema que se quiere resolver el que impone la necesidad de emplear uno u otro<sup>104</sup>.

Sin embargo, y ya que no se trata de convertirse en un programador profesional, sino de meternos en programación lo mínimo para acabar de entender a Luis Ricardo, hay una receta sumamente simple si tenemos a nuestra disposición un PeCé. Empiece aprendiendo

---

104 En la práctica uno se enfrenta a la necesidad de aplicar un ordenador concreto a la resolución de un problema particular. La potencia de ese ordenador, el conjunto de lenguajes de que se disponga para él, el tiempo disponible para desarrollar el programa, los conocimientos previos de la gente implicada en el proyecto, el presupuesto que tengamos, la misma naturaleza, por supuesto, del problema que queramos resolver, y podríamos seguir muchas más líneas, van a determinar la elección de un lenguaje.

el DOS. Tampoco hace falta que llegue a dominar todas sus órdenes. Cójale el aire. A partir de ahí, siga con programas “batch”, esos montones de órdenes que se agrupan en un fichero .BAT. Cuando lo anterior se le quede pequeño, piense en aprender un lenguaje de programación de verdad. Mucha gente ha empezado con uno que se llama BA SIC. Aunque tiende a considerársele más bien limitado, la experiencia parece demostrar que es de los más sencillos de dominar. Cuando BASIC no le convenza, piense en algo más sólido. Aquí ya la cosa se complica. Si lo que busca es ante todo más velocidad, quizá se incline por meterse en ensamblador. Si desea dedicarse a hacer cosas más complejas, tal vez opte por PASCAL. O quizá prefiera saltar directamente a lo que parece el final inevitable y aprender C. Si, lo de la “C”, o hasta “C++”, no es sino el nombre de un lenguaje.

En cualquier caso y para todo lo anterior en cualquiera de sus etapas, me permito ofrecerle un consejo que Vd. puede, pero no debería, ignorar. Para programar, lo primero es decidir qué vamos a hacer que el ordenador haga. Elija, siempre, algo que le interese de verdad. Si lo que intenta mandarle a Luis Ricardo es algo que a Vd. ni le va ni le viene, el final inevitable es que lo dejará estar al segundo día.

Aprender a programar requiere tanta dedicación no porque los lenguajes sean complejos de manejar, sino ante todo porque es necesario acumular mucha información sobre el funcionamiento interno de Luis Ricardo<sup>105</sup>. Es también por esto por lo que resulta tan interesante y útil en términos generales. Puede estar seguro de que va a comprar y leer un montón de libros con títulos en principio casi surrealistas, y a acumular un montón de papelotes con información de lo más variopinta. Pero a cambio, y aunque no se convierta en un programador profesional competente, su nivel de recursos ante cualquier ordenador crecerá de manera muy notable. Los lenguajes de macros, esos que permiten sacarle el jugo de verdad a muchos programas de aplicación, no le espantarán demasiado, de inmediato entenderá un montón de la jerga que aparece en las ayudas o en los manuales de sus programas, y será capaz de salir del paso ante esos problemas que dejan tirados a sus amigos<sup>106</sup>.

En fin, en cualquier caso es probable que nos estemos saliendo de los estrechos márgenes en que debe moverse este libro al citar siquiera un tema como la programación. Si lo he hecho es ante todo porque creo que resulta inevitable a la hora de completar esos conocimientos de informática práctica necesarios para moverse con soltura ante un PeCé, incluso si queremos limitarnos a usar un 99% del tiempo programas comerciales de aplicación. Y además, no está exento de utilidad.

Por aquello de que al menos le suenen una serie de nombres, tenemos que citar unos cuantos lenguajes. Sólo clasificarlos resulta ya difícil. Una forma de hacerlo de las menos problemáticas se fija en lo próximos que se encuentran al ordenador. Lo único que éste entiende es el código máquina, siempre particular de un tipo de CPU determinado. Lo componen ristras de bits, de ceros y unos, imposibles de entender por cualquier humano en su sano juicio. La

---

105 Tanta más cuanto menor sea el nivel del lenguaje en que programemos. Muchísima en ensamblador, bastante poca en Cobol.

106 Si le interesan los lenguajes de programación, es de nuevo posible introducirse en ellos a bajo coste recurriendo a esos CD-ROM que tantas veces hemos nombrado. “Informática Fácil Multimedia”, por ejemplo, les dedicó su número 29, correspondiente a septiembre de 1.995.

## **CAPÍTULO 8. *PeCés: Programas***

versión del código máquina digerible por nuestros congéneres es lo que se denomina “lenguaje ensamblador”. Potente, rápido, pero todavía excesivamente espartano para determinadas cosas. Vienen luego lo que se ha dado en llamar “lenguajes de nivel intermedio”, entre los que podemos citar a C y a Forth. El primero de ellos, C, es quizá la niña bonita del momento. Surgido alrededor del sistema operativo Unix, acumula la suficiente potencia y versatilidad como para haberse convertido en el estándar de la programación de sistemas informáticos, y tiene una enorme ventaja en la facilidad con que un programa construido en él se puede llevar de un ordenador a otro. Están luego los “lenguajes de alto nivel”, mucho más cercanos a las instrucciones que los humanos gustamos de manejar que a lo que el ordenador prefiere, y ahí podemos meter a lenguajes tan antiguos como FORTRAN o COBOL y a muchos otros entre los que podemos citar a BASIC, PASCAL, ADA, APL, y algunos cientos más. Dentro de ellos podemos encontrar lenguajes con campos de aplicación bastante delimitados, tales como LISP o PROLOG en inteligencia artificial, FORTRAN en cálculo matemático, COBOL en las aplicaciones de gestión bancaria, etc.

No lo dijimos en su momento pero lo decimos ahora. Cuando quiera empezar a programar trabaje en DOS y use el ordenador en modo texto. Luego, cuando eso no tenga secretos para Vd., métase con los gráficos. Y mucho más adelante, tal vez acabe estando listo para abordar la programación en un entorno gráfico tal como Windows. El nivel de complejidad de los programas que deben trabajar con un GUI es tal, que se han desarrollado nuevas versiones de lenguajes de programación dedicadas a ellos. No es estrictamente cierto que ese sea el motivo, pero existe también en la actualidad cierta tendencia a la “programación orientada a objetos”<sup>107</sup>, con el fin de poder desenvolverse con corrección en este terreno. C++ es la versión orientada a objetos de C, y creo que Delphi es más o menos lo mismo con Pascal como base. El lenguaje clásico del ramo se llamaba Smalltalk, pero no es muy fácil que lo encuentre por ahí en este momento. De hecho, en la actualidad se proponen más “entornos de programación” que auténticos lenguajes. En ellos, y sobre un lenguaje que se toma como base, se monta un entramado de utilidades que ayudan a desarrollar con mayor comodidad los programas. No debe sorprenderle en consecuencia encontrar en las estanterías cosas que se llaman “Visual Basic”, o “Visual C”, y otro montón de desarrollos de este tipo.

Al hilo de lo anterior, debe tener en cuenta también que el lenguaje de programación es sólo la herramienta fundamental para desarrollar programas, pero que son necesarias otras muchas para llevar a buen puerto tan compleja tarea. Hacen falta utilidades que ayuden a eliminar los errores que puedan contener (los llamados “debuggers” o depuradores), editores de texto para poder escribirlos de forma adecuada, y algunas cuantas cosas más.

Tal vez la cantidad de lenguajes de programación le asuste en cierta medida, y debo reconocer que no le falta razón. Pero consuélase. Todos sirven estrictamente para lo mismo, y al igual que el usuario de una hoja de cálculo no tiene demasiados problemas en colocarse ante otra de la competencia ya que existe una fundamental similitud entre ellas, lo mismo ocurre con los lenguajes. Una vez que se sabe programar en un par, los demás se pueden

---

107 Si, ya se que hemos nombrado a los objetos varias veces sin extendernos nunca demasiado en explicar lo que son. Tampoco lo haremos ahora. En todo caso, les dedicaremos un poco de atención en alguno de los capítulos que quedan.

### ***Y criterios...***

aprender, si es necesario, en un abrir y cerrar de ojos. Si Vd. sabe montar en bicicleta, la marca del chisme es lo de menos<sup>108</sup>.

Y con lo anterior, podemos dar por terminado nuestro recorrido por las distintas estructuras de datos que el ordenador maneja y su conexión con los programas para PeCé. ¿Que no hemos hablado para nada de ellas en el caso de los lenguajes de programación?. Cierto. Pero en algún momento hemos sacado lo del código fuente y el código objeto por ahí y, si Vd. va a dedicarse a programar, esté seguro de que la falta de este pequeño detalle es lo que menos debe preocuparle. Además, podemos decir que son sólo cadenas de caracteres.

Al menos ahora debería ser capaz de entender porqué cuando se trata de hacer un publi-postage, es decir de conseguir que el ordenador genere automáticamente esas cartas con las que los grandes almacenes nos felicitan nuestro cumpleaños, va a ser preciso recurrir a una base de datos. Todo el truco consiste en definir una máscara de salida adecuada. O de adaptar dicho conocimiento a un procesador de textos que ofrezca tal opción y entender porqué entonces es preciso enfrentarse a entradas de menú parecidas a “definición de campos y registros”. Nuestro procesador favorito deberá incluir el mínimo de gestión de bases de datos necesario para poder abordar una tarea más propia de ellas.

### **Y criterios...**

### ***Luis Ricardo se mete a juez***

Bueno, tras haber profundizado un poquito en las estructuras de datos, los programas deberían haber quedado clasificados en grandes grupos bastante homogéneos en cuanto a su utilidad y manejo. Ni más ni menos que eso es lo que dijimos cuando empezamos el apartado y a estas alturas debería estar claro el porqué. Pero también dijimos que una vez logrado tan importante objetivo, iba a ser necesario establecer una serie de criterios que nos permitieran seleccionar un único programa entre todos los que se encuentran disponibles dentro de una categoría determinada. En resumen, ha llegado la hora de que intentemos aclarar qué debe mirar para elegir Microsoft Word en lugar de Corel WordPerfect. O viceversa.

Los criterios de selección de programas son, vamos a ser muy pesados una vez más, bastante subjetivos. Pero a pesar de todo es posible establecer una serie de pautas generales a las que es conveniente que se ajuste cualquier programa que merezca que nos rasquemos el bolsillo.

Aparte de algunas obviedades tales como que esté disponible para el ordenador y sistema operativo sobre el que pretendamos ejecutarlo, y que su precio no nos obligue a recurrir a una entidad financiera cuando llegue el momento de desembolsar su importe<sup>109</sup>, no estará

---

108 Las diferencias entre lenguajes están de todos modos ahí, y la afirmación anterior debería tomarse como cierta en general pero no estrictamente correcta en la práctica. La forma en que se programa en BASIC es muy diferente de la que se usa en Forth, en C, en ensamblador, o no digamos en un entorno de programación para Windows. Y la diferencia va mucho más allá de que para hacer que el ordenador saque en pantalla “Hola”, haya que escribir PRINT “Hola” o .(Hola) CR.

109 “¡Que exageración!”, quizá piense más de uno. Quizá. Pero no estará de más que sepa que, sin salirnos del PeCé y al lado de esas maravillosas hojas de cálculo y procesadores de texto con precios de 25.000 pesetas para abajo, existen otros programas, tal vez no mejores pero sí propios

## ***CAPÍTULO 8. PeCés: Programas***

de más que posea en el mayor grado posible las siguientes características, que paso a citar en orden prácticamente aleatorio: apertura, sencillez de la estructura de datos, rapidez, pequeño tamaño, estandareidad, gran difusión, potencia, programabilidad, no protegido contra copia, y una interfase de usuario cómoda. Por supuesto que la lista podría alargarse. De hecho le recomiendo que lo haga con aquellos factores de su elección que considere se echan a faltar palmariamente.

Mientras tanto, intentaré justificar un poco la importancia de cada una de las características que yo creo conveniente considerar. En cuanto a la apertura, es decir a la capacidad de manejar un número tan elevado como sea posible de formatos diversos, debería tenerse siempre en cuenta que lo importante de un programa son los datos que con él se van a producir. Y como nunca se sabe de donde van a salir o donde deberemos hacerlos llegar, es cruelmente necesario que el programa que elijamos sea capaz de leer y escribir, importar y exportar, tantos formatos como se pueda. No basta que Word lea su propio formato. Quizá alguien me pase un fichero creado con WordPerfect y querré poder abrirlo aunque sólo sea para leerlo. Y querré luego, probablemente, grabar el resultado en el formato de la competencia. A la hora de incorporar un gráfico, un tipo de datos pródigo donde los haya en formatos absolutamente incompatibles entre sí, no es posible estar seguro de antemano de en cual de ellos me va a llegar. Una vez más mi procesador de textos deberá ser capaz de leer cuantos más mejor.

Que el programa se base en una estructura de datos sencilla, siempre dentro de lo posible ya que por ejemplo una base de datos es una base de datos y por tanto no suele haber mucho que rascar en ese caso, es también muy importante. Vimos al hablar de FileMaker Pro que tal vez su gran ventaja era la no imposición de límites innecesarios al tamaño de los campos tipo texto que podía incluir, lo que contribuye a simplificar la estructura de sus registros. Si hablamos de procesadores de texto, es sumamente probable que uno que emplee códigos de formato en línea sea más sencillo de entender en su funcionamiento que otro que recurra al también clásico esquema del formato externo, especialmente si el primero de ellos nos muestra siempre que lo deseemos tales códigos y el segundo los oculta como si fueran algo de lo que avergonzarse. Es este otro caso en el que deberíamos distinguir entre sencillez aparente y real. Recuerde que la información debe ocultarse solo lo necesario para no crear más problemas de los que evita. No pasa nada porque un programa pueda mostrarnos sus interioridades. Al contrario, eso quizá nos permita resolver problemas de otro modo insolubles. Es por esto por lo que la estructura de datos debe ser sencilla en lo posible, para poder entenderla más fácilmente y ser capaz de aprovechar al máximo el programa. Tal vez deberíamos hablar con mayor propiedad de la adecuación de la estructura de datos al problema que vamos a manejar y de la necesaria capacidad de visualizarla y modificarla fácilmente, pero quizá sea un poco excesivo meternos por ahí.

---

de campos más especializados y en los que se supone un mayor poder adquisitivo, que pueden tener precios de partida 10 veces superiores. Es decir, de 250.000 pesetas para arriba. Y que puede ocurrir que sea preciso añadirles algún módulo por aquí y alguno por allá para completarlos hasta que puedan hacer lo que necesitemos que hagan, y que eso puede llevar el precio hasta 4 veces lo anterior. Es decir, hasta 1.000.000 de pesetas. O más. Y no invento nada. He visto listas de precios de software para PeCé con cantidades de ese orden para una única licencia de uso.

Lo de que el programa sea rápido en su funcionamiento no debería requerir gran justificación. La diferencia entre que una tarea se realice de forma prácticamente instantánea y que le cueste diez minutos es, simplemente, la que separa en informática lo utilizable de lo inútil, y el margen tolerable es mucho más escaso de lo que se pueda pensar. Un programa lento es un programa inservible. Dicho esto, también hay que precisar que no debe confundirse un programa lento con uno sobrecargado. Cualquier programa, por bueno que sea, tendrá dificultades más que notables para manejar con fluidez un gráfico bitmap de 30 Megabytes. Y tal vez los problemas no sean culpa suya, sino del sistema operativo que dice soportarlo. Ahora bien, todo editor de gráficos decente deberá, por decir algo, funcionar correctamente al manipular un fichero de 5 Megabytes.

El pequeño tamaño puede que no sea imprescindible, pero resulta bastante conveniente. De él se derivan varios beneficios indirectos sumamente interesantes. El programa pequeño ocupará poco sitio en nuestro disco duro, dejando más espacio para los datos, y de nuevo conviene insistir en que es lo único realmente de valor que tenemos en el ordenador. Se lanzará más rápidamente ya que le costará menos cargarse en memoria, y por si fuera poco son los programas pequeños los que suelen funcionar con mayor rapidez<sup>110</sup>. Además, muchas de esas tareas de mantenimiento que resultan inevitables se verán favorecidas. El programa se instalará más fácilmente, se podrá tener una copia de seguridad en un número limitado de disquetes, podrá usarse en ordenadores con menor capacidad de disco duro o quizá incluso desde un disquete, y así sucesivamente.

Desgraciadamente, el pequeño tamaño es una de esas características casi imposibles de conseguir, y la razón no es otra que su contraposición con una no menos importante: la potencia. Algo parecido ocurre con la mayoría de las demás. La cuestión es que si queremos que un programa sea capaz de hacer casi de todo en un campo de aplicación determinado, es casi imposible que sea pequeño. Máxime si queremos que incluya una colección completa de datos ya preparados para que resulte útil de inmediato, nada más sacarlo del paquete. Un paquete de gráficos capaz de cualquier cosa y que incluya un montón de ficheros con imágenes, dibujos, y tipos de letra para poderlos utilizar a voluntad en nuestros trabajos es lo ideal, pero tiende a llenar un CD-ROM. Por lo tanto, hay que limitarse a buscar programas que lleguen a un razonable compromiso entre tamaño y potencia. Y ya se sabe que los compromisos aceptables dependen muchísimo del parecer de cada uno. O sea que Vd. verá lo que en su caso quiere decir “pequeño” y qué significa “potente”.

---

110 No necesariamente el programa pequeño es siempre el más rápido, pero la experiencia tiende a confirmar tal equivalencia. En realidad, existe alguna base lógica para apoyarla. La elección de los lenguajes de programación más cercanos a la máquina, en concreto ensamblador, no facilita precisamente la tarea del programador, pero a cambio permite obtener programas sin un byte de más y sumamente rápidos. Recurrir a otros lenguajes implica en mayor o menor grado montar el programa a base de módulos ya existentes, y eso hace que parte del código pueda repetirse en diversos lugares y que el programa deba realizar una buena cantidad de trabajo no inmediatamente productivo. Es decir, se cambia facilidad de programación por un programa potencialmente menos optimizado. No conviene olvidar, de todas formas, la importancia de los algoritmos y las estructuras de datos también en este punto. Un programa gordo que use los más adecuados al problema a resolver puede ser mucho más rápido que otro realizado en ensamblador que no cuide tanto su planteamiento.

## ***CAPÍTULO 8. PeCés: Programas***

El excesivo tamaño de los programas ha llegado a acuñar un término claramente despectivo. Se habla de “fatware” (software obeso) cuando es preciso referirse a programas tan gordos, lentos, y exigentes en cuanto a la configuración del ordenador sobre el que pueden ser utilizados, que acaban resultando escasamente operativos. El término, y el problema subyacente, es tan real, que hasta se le han dedicado bastantes páginas en revistas de informática indudablemente serias. Baste citar el ejemplar de Byte correspondiente a abril de 1.993.

Dos características que no sólo no se oponen sino que suelen ir asociadas son la estandareidad y la gran difusión. Nos interesa muchísimo comprar un programa que se haya vendido como rosquillas. Un número de usuarios elevado es la mejor garantía para que dure muchos años y sea sometido a revisiones periódicas que lo mantengan al día. Lo ideal es que el tal programa llegue a crear un estándar, de manera que todos los programas de la competencia acaben pareciéndosele y terminen por hacer posible incluso el pasarnos en su momento a uno de ellos, más barato y mejor, sin perder en el camino ni uno solo de los datos que con tanto esfuerzo hemos ido produciendo a lo largo de los años. Conviene recordar que será lo único realmente de valor que poseamos, repetimos por tercera pero no última vez.

También podemos leer lo de la estandareidad como el respeto a los estándares ya existentes. Un programa deberá respetar, a ser posible por completo, todas las convenciones propias de un ordenador y sistema operativo determinado. No deberá romper nada, no podrá interferir en el funcionamiento de ningún otro programa de la competencia, y tendrá, por volver a un punto que ya hemos citado, hasta que ser capaz de leer y escribir los datos que otros programas ya considerados estándar produzcan.

El respeto estricto a un estándar de software no debe confundirse con la obligatoriedad de comprar el programa que lo ha creado. Suele ser más conveniente observar un respeto a lo que podríamos llamar “estandareidad relajada”. Me explico. Si nuestro ya conocido TLW 4.0 crea un estándar en el campo del proceso de textos, nuestros también ya familiares jefes de línea de productos y directores más o menos ejecutivos lo aprovecharán de inmediato para aumentar su margen de beneficios. No será mala idea entonces comprar un producto de la competencia, igual o más capaz y casi seguramente más barato, si respeta asimismo el estándar creado por TLW 4.0.

La programabilidad es algo que pocas veces se tiene en cuenta. El motivo no es otro que la tendencia a asustar que tiene la programación en general y un cierto desconocimiento de la realidad del trabajo con el ordenador. Por motivos que expondremos con más detalle en el próximo capítulo, es sin embargo muy conveniente que el funcionamiento de un programa se pueda automatizar. Debería bastar con recordar que la capacidad de programar el ordenador es, como vimos bastantes capítulos atrás, una de sus más grandes ventajas. No parece razonable renunciar a ella de entrada. Todo programa potente debería incorporar algún mecanismo, sea un auténtico lenguaje de programación, una herramienta para almacenar acciones de modo que puedan repetirse con rapidez cuando se desee, o cualquier otro tipo de invento, que permita hacer que sea él el que trabaje sin necesidad de que nosotros estemos encima.

Lo que sí que parece evidente para todo poseedor de un PeCé es que la interfase de usuario de un programa que merezca la pena debe ser lo más cómoda posible. Pero no está tan claro lo que tal frase significa. ¿Consiste en que, si se trata de un programa en Windows, tenga casi infinitos botones para hacer algo más rápido el acceso a sus distintas funciones?. ¿Puede tener una buena interfase de usuario un programa en modo texto que se maneje tan sólo



mediante combinaciones de teclas?. En mi opinión, ya que aquí desde luego cada cual es absolutamente soberano de tener sus propias ideas sobre lo que puede querer decir comodidad, la respuesta a la primera pregunta es “no”, y a la segunda “sí”.

A pesar de todo, existe cierto consenso sobre algunos puntos. La interfase de usuario debe ser configurable para que cada cual la adapte a sus gustos personales. Debe ser coherente, utilizando de manera comprensible una serie de convenciones tal vez propias pero necesariamente con sentido y cuanto más lógicas mejor. Debe incluir ayudas lo más completas e inteligibles posible. Y debe respetar que lo más importante, repetimos una vez más, no es el manejo del programa, sino la realización de un trabajo, por lo que deberá dar preferencia a la presentación de los datos que con él se vayan produciendo y a proporcionar información relacionada con su explotación. Lo que quiere decir lo anterior es que no vendrán pero que nada mal unas líneas que nos informen, cuando estamos creando un gráfico, de la posición que ocupa el cursor con el que estamos dibujando, o que se nos muestren reglas y escalas que nos ayuden a colocar los objetos que lo componen, o, si estamos escribiendo un texto, que en algún sitio se nos informe de qué tipo de letra estamos usando y en qué tamaño. La cosa llega a tal extremo que programas tan poco dudosos de culpabilidad en el aparentemente imperdonable pecado de “aspecto espartano” como Word de Microsoft permiten, si el usuario lo desea, quitar del medio, en una pantalla Windows, cualquier cosa que no sea el texto que estamos escribiendo. Por supuesto que también, y he aquí un ejemplo de configurabilidad de la interfase de usuario, es posible recuperar de inmediato barras de botones, menús, y demás parafernalia. Existe asimismo un notable consenso sobre el hecho de que los GUIs, desde luego, aportan más de una ventaja a la hora de dotar de una interfase cómoda a muchos programas.

Un punto tan evidente como la traducción o no al castellano merece al menos ser considerado dentro de los aspectos que influyen en esa comodidad de la interfase de usuario que acabamos de citar. Pero puede llegar a tener implicaciones más importantes aún, afectando incluso a la misma operatividad del producto, si el programa debe ocuparse de realizar tareas relacionadas con el lenguaje. Un procesador de textos, sin ir más lejos, deberá necesariamente incluir un diccionario ortográfico y de sinónimos adecuado para el idioma en que vayamos a escribir.

Y no es la única característica de este tipo. Una opción que muchos programas ofrecen es el denominado “Undo”, es decir “deshacer”. Sirve para anular un cambio que hemos realizado. Si acabamos de suprimir un párrafo de nuestro texto y justo al verlo desaparecer nos damos cuenta de que deberíamos haberlo conservado, o no nos gusta que el fondo de nuestro dibujo sea verde y preferimos el azul cielo anterior, pulsar “Undo” nos permite regresar a la situación de partida. La existencia de tal opción es un auténtico salvavidas. Tanto es así que muchas veces se puede volver atrás muchos pasos y hasta rehacerlos a voluntad, de lo que se ocupa la opción “Redo” habitualmente asociada a la anterior. Disponer de tales funciones no solo mejora la interfase de usuario, sino que aumenta la funcionalidad del producto y hace su empleo mucho más tolerante frente a los inevitables errores de operación. En resumen, lo hace también más sólido y potente. Otros dos ejemplos del mismo estilo pueden ser la posibilidad de abrir varios ficheros simultáneamente desde un único programa y el soporte de alias (nombres a nuestra elección para designar cualquier cosa, desde renombrar una orden DOS hasta poder referirnos a un directorio de la red no como PRINTER\_PC\DRIVE\_C\PRINTERS\USER\TIPO2 sino como “LASER” sin más<sup>111</sup>).

## ***CAPÍTULO 8. PeCés: Programas***

Pasemos a otra cosa. No es momento, ni quizá lugar, de extendernos en alegatos para criticar la política de algunos editores de software que protegen contra copia a sus productos. Nos limitaremos a reconocer su derecho a recurrir a tal práctica para proteger sus intereses y a indicar que, a cambio, parece ser una clara fuente de problemas de gravedad variable para los usuarios que los adquieren legalmente. En el capítulo siguiente veremos algún ejemplo de esta última faceta, no tan conocida, de la protección contra copia. Si desea más explicaciones de forma inmediata, me permito sugerirle que lea las entradas “Piratería” y “Protección contra copia” en el glosario. El asunto es tan grave que algunos famosos periodistas informáticos llegan al extremo de negarse por norma, sólo rota en casos excepcionales, a comentar productos de este tipo<sup>112</sup>.

La relación anterior no es otra cosa que una primera aproximación a lo que debería probablemente ser una larga lista de características a considerar. Por citar algunos elementos más que deben entrar necesariamente en el proceso de evaluación de un programa comercial, citemos que éste debe soportar, y hacerlo bien, los periféricos necesarios para su funcionamiento. Un procesador de textos deberá ser capaz de sacar el máximo partido a cuantas más impresoras mejor, un editor de gráficos tendrá que poder entenderse con nuestro escáner, si tenemos la suerte de tener uno, y así sucesivamente. El nivel de soporte al usuario a que tengamos acceso tras realizar la compra, el prestigio de la marca, y otro montón de cosas que en algún caso hemos citado más o menos de pasada, habrán también de tenerse en cuenta.

Por si no lo ha notado, en realidad estamos siempre dando vueltas a los mismos temas y poco a poco les añadimos más y más detalles en función de los conceptos que vamos introduciendo. Quiérese decir que lo que dijimos en los capítulos 4 y 5 sobre como debe ser el software, o las historias sobre distintos editores al principio de este mismo capítulo, pueden aportar alguna luz adicional sobre los criterios a considerar cuando hay que elegir un único programa entre varias alternativas.

Y también deberíamos decir que muchas de las características citadas como deseables ocultan una potencial fuente de problemas. Por ejemplo, esa tan necesaria configurabilidad, esas opciones tan numerosas, entran claramente en este grupo. Si quiere detalles, vea “Opciones” y “Configuración” en el glosario. Pero no es el momento de hablar de problemas. Lo haremos en el próximo capítulo.

No estaría de más poner algún ejemplo para intentar bajar de las alturas teóricas en que de nuevo parece que nos hemos metido. Hay que ver qué puede querer decir todo lo anterior en la práctica. Pondremos tres. El primero, facilito, será el análisis de un programa que nuestro experto recomendó para ver como se ajusta, o no, a los criterios anteriores. El segundo nos permitirá cambiar notablemente el registro de la discusión. Y el tercero será el análisis de todo un grupo de productos que nos queda por tratar y que ya citamos al

---

111 A la hora de mantener o simplemente usar una red, este tipo de cosas pueden resultar fundamentales. Y ya me acuerdo de que las redes se salen del campo que nos hemos marcado.

112 Por ejemplo, Jerry Pournelle, cuya habitual columna en la revista “Byte” es de lectura poco menos que obligada.

introducir Windows. ¿Se acuerda de los integrados?. Pues volveremos a ocuparnos un poco de ellos para ver como han evolucionado.

### **Caso 1: FileManager**

Ocupémonos de inmediato en ver en qué criterios podía apoyarse nuestro experto cuando realizó una afirmación tan osada como: “En cuanto al ‘Administrador de archivos’, voy a dejarte instalado el que más me gusta. Es el de las PC Tools para Windows, un conjunto de utilidades bastante popular. Lo vi en una revista y me la compré solo por eso”. ¿Qué tiene el FileManager de las PC Tools que no tenga el administrador de archivos que viene con Windows?. ¿Merece la pena preocuparse en añadir algo para sustituir un programa que ya tenemos gratis y que al parecer es usado a plena satisfacción en un 90% de los casos?. Intentaremos responder a la primera pregunta. La segunda deberá respondérsela Vd. mismo.

Muchos de los criterios que expusimos son sólo aplicables de forma marginal a un producto de este tipo. La rapidez en las operaciones que realizan no depende demasiado de ellos mismos, sino del propio sistema operativo. El administrador de ficheros se limita a pasar la orden “Copia el fichero tal a tal otro”, algo poco menos que instantáneo, y a partir de ahí deja que el viejo DOS (aunque estemos en Windows) se ocupe de todo. El pequeño tamaño y la no protección contra copia son también universales. La sencillez en las estructuras de datos que manejan tampoco viene demasiado a cuento. La FAT está ahí, ya definida de antemano, y poco puede hacerse al respecto. Una gran difusión tampoco es especialmente necesaria en este caso.

Eso nos deja sólo con algunos puntos a considerar. El programa empieza a tener alguna ventaja bastante clara si consideramos su estandariedad, necesariamente entendida en este caso como respeto a los estándares existentes, y la apertura, en el sentido de ser capaz de sacar partido a cuantos más tipos de datos mejor. FileManager es perfectamente capaz de entender no sólo la FAT normal en nuestros discos de PeCé, incluyendo por supuesto aquellos a los que podamos acceder a través de una red local, sino también la de discos comprimidos con cualquiera de los compresores más difundidos (DoubleSpace, Stacker o SuperStore) y la de los ficheros comprimidos con PKZip. Aunque no haga extensible su magnanimidad a los ficheros creados con ARJ, no deja de ser una ventaja en los tiempos que corren.

Las ventajas anteriores pueden también considerarse como aspectos relacionados con la potencia del programa, que es capaz de desenvolverse correctamente ante necesidades que, aunque habituales, no son cubiertas por todos los productos de la competencia. Pero hay mucho más que decir sobre este aspecto. No sólo puede listar el contenido de un fichero comprimido, sino que también puede crearlo o expandirlo. Es posible presentar un directorio de múltiples maneras. Así, podemos cambiar la clásica de árbol de directorios por un lado y ficheros del directorio actual por otro, por una en la que se nos presenta la lista completa de ficheros de una unidad en el orden que queramos (alfabético, por tamaño, por fecha de creación) independientemente del directorio al que pertenezcan. Pueden incluirse en el listado los ficheros borrados recientemente, por si queremos intentar su recuperación. Y además, pone a nuestra disposición un cúmulo de utilidades que rápidamente se revelan como absolutamente necesarias. Son ese tipo de cosas que, una vez probadas, resulta difícil creer que hayamos podido vivir tanto tiempo en la ignorancia. Hay un magnífico visualizador de ficheros para poder echar un vistazo dentro del que nos apetezca que soporta la mayoría de los formatos más frecuentes. Hay una herramienta de búsqueda que nos permite localizar con facilidad aquellos ficheros que contengan una frase determinada o cuyo nom-

## ***CAPÍTULO 8. PeCés: Programas***

bre responda al criterio que indiquemos. Tenemos otra que resulta fundamental si hemos de llevar datos de un sitio a otro, o queremos mantener copias de seguridad, que permite sincronizar el contenido de dos directorios, es decir copiar a uno de ellos los ficheros que falten y estén presentes en el otro o que hayan sido modificados más recientemente. Y por supuesto da acceso a otras herramientas de las PC Tools que nos dejan reparar un disco defectuoso, compactarlo, recuperar ficheros borrados...

Por si lo anterior no bastara, la interfase de usuario acaba de completar las ventajas. No es que sea revolucionaria. Al contrario, a primera vista puede parecer francamente anodina. Pero no es una desventaja ya que resulta familiar de inmediato. Mientras que el administrador de archivos de Windows recurre a ventanas superpuestas para dejarnos acceder a diferentes unidades, el FileManager de las PC Tools emplea ventanas en mosaico, repartiendo la pantalla entre ellas. A mí nunca me han gustado este tipo de ventanas, pero resultan innegablemente preferibles en este caso. Una apariencia menos sofisticada demuestra ahora ser mucho más conveniente. No hay que desplazar la ventana con el disco D: para poder copiar un fichero a ese directorio que quedaba oculto por la ventana del disco C:. El programa usa el botón derecho del ratón para dar un acceso rápido a las funciones más frecuentes. Si lo pulsamos sobre un fichero nos aparecen encima las opciones necesarias para copiarlo, borrarlo, visualizar o modificar sus atributos, y demás. Y es ampliamente configurable. Puede elegirse entre dos niveles de menús, uno para usuarios noveles, que da acceso a las funciones menos sofisticadas y más seguras, y otro avanzado que permite acceder prácticamente a todo. Pueden ponerse o quitarse botones en diferentes estilos, con o sin texto explicativo. Puede cambiarse el contenido de los menús, añadiendo o suprimiendo cosas o cambiando el orden en que se nos presentan. Y si Vd. es de los que prefiere comunicarse con su sistema de ficheros a través del teclado, el FileManager puede mostrarle una línea de órdenes para que teclee ese "DIR" tan familiar. Y debería estar claro que guarda un histórico de las órdenes, para poder recuperar aquellas que desee.

Quizá a Vd. la relación anterior le esté dejando bastante frío, o le parezca el resultado de los delirios de un informático chiflado al que se le dio permiso para añadir todo lo que se ocurriera, por ridículo que pudiera resultar, a un administrador de ficheros. Puede que le esté pasando por la cabeza hace un par de párrafos la pregunta de para qué diablos pueden servir tantas cosas. Lo cierto es que todas ellas cubren necesidades reales, hay muy poco celofán en una lista tan larga. El FileManager es ante todo un producto serio y práctico que incluye cosas necesarias. Pero cuya utilidad sólo se aprecia en su justa medida cuando surgen los problemas. Allí donde un administrador de ficheros normal te deja tirado, el FileManager demuestra que esa enigmática opción servía de verdad para algo práctico y nos salva la vida una vez más sin hacer ningún aspaviento.

Por ejemplo, un problema muy frecuente consiste en no encontrar un fichero. Sabemos que ayer estuvimos trabajando en algo, que el fichero que grabamos se llamaba PROBLEMA y podía o no tener una extensión que en cualquier caso desconocemos, y aparte de eso sólo estamos seguros de que no está en la unidad y el directorio en que lo estamos buscando. ¿Dónde lo pudo dejar el maldito programa que lo grabó?. Vaya por delante que muchas veces este problema se debe a una falta de organización del disco duro. Un novato que no sólo desconozca lo que es un directorio sino que tampoco se plantee organizarlo con un mínimo de solidez lo sufrirá frecuentemente. Pero hasta a los usuarios experimentados les toca enfrentarse a él de vez en cuando. Lo que suele ocurrir es que no se presta atención al directorio en que el programa está dejando los datos por defecto (por excesiva confianza, por prisa en terminar, por lo que sea), y el fichero acaba donde no debería o donde nosotros

no suponemos que va a acabar. En resumen, no es una situación rebuscada o particularmente improbable. La solución desde FileManager es sumamente sencilla. Hasta tenemos dos opciones. O bien recurrimos a un listado completo de los ficheros de una unidad, prescindiendo de directorios, donde será sumamente sencillo localizar nuestro PROBLEMA punto lo que sea si lo presentamos en orden alfabético<sup>113</sup>, o nos vamos directamente a la herramienta de buscar ficheros que se llama SmartFind (traducible como “buscador inteligente”, pero no se fíe demasiado de los nombres en estas cosas). Aunque una a una ninguna de las nuevas opciones le impresione particularmente, debería reconocer que todas juntas pueden cambiar el panorama una barbaridad.

¿Es FileManager un programa perfecto, sin defectos?. En absoluto. En el próximo capítulo veremos que, de hecho, tal afirmación tiene siempre visos de absurda. Es un defecto que no soporte ficheros comprimidos creados con ARJ. Y otro que necesite<sup>114</sup> una cierta cantidad de memoria en la fatídica zona de los primeros 640 Kilobytes del PeCé. Le garantizo que resulta muy frustrante, casi parece una broma de muy mal gusto, enfrentarse a un mensaje que dice que no hay memoria para lanzar el programa cuando Windows nos jura y perjura que tenemos 30 Megabytes de RAM real y no se sabe cuantos más de virtual a nuestra disposición. Tampoco es particularmente programable. Y si Vd. es de los que ha optado por la última versión de Windows (por el momento), la conocida como Windows 95 y de la que también nos ocuparemos un poco en el próximo capítulo, sepa que el FileManager de las PC Tools no soporta su nuevo sistema de ficheros<sup>115</sup>. Por si fuera poco, tiene un defecto que puede resultar muy grave. Jamás ha sido traducido al castellano.

---

113 “Pero seguiremos sin saber en qué directorio está”, quizá esté pensando. La lista se presenta agrupando a todos los ficheros, pero el path de cada uno aparece junto a su tamaño, fecha, y demás atributos. O sea que de eso nada.

114 Al igual que otros muchos programas Windows.

115 Al que se conoce como VFAT, me parece. Y ya que estamos, aprovecho para precisar que aunque no hemos entrado en demasiadas profundidades, tanto el administrador de ficheros de Windows como el FileManager de las PC Tools han ido cambiando poco a poco a lo largo de las versiones. No sé si el administrador de ficheros de Windows 3.11 es igual al de la 3.1 o algo mejor, pero desde luego el de Windows 95, el llamado Explorer, ha cambiado enormemente. Tampoco el FileManager de la primera versión de las PC Tools es exactamente igual al de la segunda versión. Un ejemplo. Ambos soportan ficheros .ZIP pero sólo el de la versión 2 es capaz de entender los de la versión 2.0, los más frecuentes. De cualquier modo, se trata de presentar un ejemplo, no de hacer un exhaustivo análisis de la evolución de dos productos paralelos, pues no puede hablarse ni siquiera de que sean competidores. Microsoft vende Windows 3.1, 3.11 o 95 e incluye un administrador de archivos ya que es una pieza absolutamente fundamental incluso para empezar a andar. Intenta hacerlo digno por aquello de que no se pueda decir que Microsoft hace malos productos. Central Point (ahora Symantec), o Norton (ahora también Symantec), proponen productos adicionales para DOS o Windows que deben necesariamente ser mejores que aquellos a los que sustituyen si quieren que alguien los compre. Deben justificar que se gaste un dinero para sustituir un componente que viene gratis. Por cierto, si le interesa cambiar el Explorer de Windows 95, Symantec vende el llamado Norton Navigator, que por supuesto es capaz de entenderse con la VFAT. El nombre induce a confusión ya que lo de “Navigator” se relaciona ante todo con productos destinados a Internet (de la que también nos ocuparemos más adelante), pero no se engañe. Es un administrador de archivos.

## ***CAPÍTULO 8. PeCés: Programas***

Si nada de lo anterior le parece particularmente trascendental, se puede concluir que carece de defectos importantes. Unido a las ventajas que hemos enumerado y a algunas más (por ejemplo, es tan sólido como es de desear en un administrador de ficheros, que no debe colgarse jamás), la calificación final tendría que ser al menos un eufemístico “sumamente interesante”. No es extraño que nuestro experto lo recomendara.

### **Caso 2: PADRE**

El primer caso ha sido realmente facilito. Pero el segundo va a exigir un poco más de esfuerzo. De hecho podríamos decir que va a ser un problema con truco. La cuestión a responder es ahora la siguiente: ¿Es PADRE, el popular “Programa de Ayuda a la Declaración de Renta Española”, un buen programa?

Si somos dados a abordar los problemas sistemáticamente y nos encantan las recetas, procederemos de forma análoga al caso anterior. Cogemos la lista de criterios e iremos cotejándola con las características de PADRE. ¿Potente?. No parece que mucho. ¿Abierto?. ¿A qué?. ¿Programable?. Desde luego que no. ¿Estándar?. Pues hombre, no rompe nada pero desde luego no es un programa Windows...

La realidad es que tal enfoque carece de sentido en este caso. Aunque válidos en general, los criterios que hemos expuesto son más propios para juzgar programas de tipo horizontal que vertical. Un programa horizontal era uno de esos que valen un poco para todo dentro de un campo de acción determinado y que el 90% de los usuarios de PeCés pueden encontrar de utilidad, justo los que hemos estado analizando en general al hablar de estructuras de datos, en tanto que uno vertical era uno de esos que se dirigen a un colectivo muy concreto y proponen una solución muy concreta para resolver un problema también muy concreto.

Tan enorme diferencia de planteamiento hace preciso ajustar los criterios de valoración. PADRE deberá ante todo ser un programa fácil de usar para alguien que quiera hacer su declaración de renta. La idea que tengan del ordenador muchos de sus usuarios puede limitarse a saber que es una cosa con un teclado a través del que escribimos lo que vemos en la pantalla y que al final podemos llevar a papel mediante la impresora. Es también posible suponer que sabrán como es el impreso para realizar la declaración del IRPF y que admitirán la necesidad de proporcionar los datos precisos para completarlo a través del ordenador.

Con tal punto de partida, PADRE es, indiscutiblemente, un excelente programa. No emplea Windows, pero no es un defecto sino una virtud que le permite ser muy poco exigente con el ordenador que lo va a soportar. Basta un PeCé con cualquier CPU, 440 KB de RAM libres, y 2 MB de disco duro disponibles, para poder usarlo. A Dios gracias, al Ministerio de Hacienda no se le ha pasado por alto el hecho de que buena parte de los PeCés que andan por el mundo y por España tienen bastantes años de antigüedad, y no son pocos los que ni siquiera son capaces de lanzar Windows 3.1<sup>116</sup>. ¿Es imposible, pues, que tenga una interfase de usuario digna?. En absoluto. La interfase es la mejor posible, la única que todo el mundo

---

116 Hace como quien dice cuatro días, a menos de cuatro años del 2.000, todavía intentó un amigo reparar un 80286 con ¿1 MB de RAM? de una medio marca. No estaba dispuesto a tirarlo, y me pareció muy bien. El disco duro de ¿40 MB? había muerto. Desgraciadamente, el fabricante le pidió 70.000 (isetenta mil!) pesetas por uno de repuesto, y la BIOS del chisme de marras no admitía más que unos pocos discos duros de características fijas, por lo que no pudimos aprovechar uno de 120 MB que un benefactor anónimo tenía guardado en un cajón y nos cedía

espera. Se simula en pantalla el impreso a rellenar. O sea que no está pero que nada mal por ahí<sup>117</sup>. ¿Que no es configurable?. No. Pero se ve bien, no hace falta configurarla necesariamente para obtener algo funcional, y añadir opciones es también complicar un programa. No tiene sentido si el colectivo a quien va dirigido no las espera ni las desea. Por tanto la falta de configurabilidad es ahora una virtud. Y las ayudas están bastante bien. Y tenemos las herramientas necesarias para explotar los datos. Por ejemplo una calculadora. La potencia del programa y su rapidez son cuando menos suficientes, y hasta cuenta con opciones de agradecer que lo hacen aún más útil. Nos muestra con velocidad como saldría nuestra declaración si la hacemos individual o conjunta, por ejemplo. De su pequeño tamaño ya hablamos antes, y no está protegido contra copia. Además, respeta los estándares lo suficiente como para no romper absolutamente nada y poderse lanzar, si así lo queremos, en una ventana DOS abierta en ese Windows del que tan orgullosos estamos.

Un punto a comentar de su interfase de usuario es que se toma hasta la molestia de ocultar el sistema de ficheros del PeCé. Los datos se graban automáticamente y sólo hace falta dar el nombre del contribuyente. No está mal. Podemos considerarlo positivo en principio ya que de nuevo hay que recordar que la gente que lo use puede no querer saber absolutamente nada de normas 8.3, directorios, o unidades.

En resumen, es un programa excelente, aunque muy cerrado, sin soporte para Windows, y nada programable. Si hemos de mantener un mínimo de análisis tradicional por aquello de que no pierda demasiado el oremus, querido LAO, podríamos por supuesto sacarle innumerables defectos. Por ejemplo, solo sirve para realizar la declaración simplificada o abreviada. Nada de IVAs o impuestos de sociedades. Y puede argumentarse con bastante razón que resulta incluso demasiado cerrado. Tanto que la evolución ha llevado a añadirle en los últimos ejercicios la posibilidad de guardar las declaraciones de distintos contribuyentes. No vale con uno solo ya que al parecer no es nada raro que el chico de la casa haga la declaración de sus padres, que no quieren saber nada de PeCés, y ya de paso la de los abuelos, y quizá hasta la de la tía Enriqueta, que tan orgullosa está de las habilidades informáticas de su sobrino. Pero seguimos sin poder pasarle nuestros datos, almacenados penosamente a lo largo del año en una hoja de cálculo, para que nos realice automáticamente la declaración a partir de ellos.

El motivo por el que hemos puesto a PADRE como ejemplo práctico es ante todo para que se dé cuenta en su justa medida de la relatividad de los criterios en informática. Hay quizá sólo uno universal y absoluto, y es la adecuación del programa al problema que debe resolver en las condiciones de uso en que debe hacerlo. Un programa que cumpla tan general criterio

---

gratis. Hubo que desistir.

117 Si le parece que estamos manipulando y hasta ignorando descaradamente mucho de lo que se ha dicho antes, de eso nada. ¿Se acuerda por ejemplo de que hablamos de una tendencia bastante popular, y hasta moderna, a la hora de diseñar la interfase de usuario, que consistía en imitar la herramienta que el programa iba a sustituir?. Una agenda informática se nos presentaba con su imagen o un reproductor de CD de audio con los mandos normales en este tipo de chismes. Pues el caso de PADRE entra absolutamente dentro de esa familia. Si hay que rellenar un impreso, se presenta una imagen de él. Los entornos gráficos permiten aplicar este enfoque a otros problemas, pero también el modo texto puede emplearlo en bastantes ocasiones. No sea estrecho de miras, hombre.

## **CAPÍTULO 8. PeCés: Programas**

será bueno. Pero, en la práctica, eso puede querer decir cosas muy diferentes. Hay que entender los porqués y analizar por nuestra cuenta. No repita como un lorito lo que nadie le diga. Y mucho menos lo que yo le diga. Déle al coco. Al suyo.

Un ejemplo. Acabamos de decir que está muy bien que PADRE grabe automáticamente. Pues algo parecido hace FileMaker Pro, esa base de datos que tanto le gustaba a nuestro experto. En cuanto abrimos uno de sus ficheros, va grabando automáticamente y de forma inmediata todos los cambios que se hacen. No busque opciones como “Grabar” o “Cerrar” en los menús. No están. A lo más un “Archivar una copia”, que nos da la oportunidad de almacenar con otro nombre el estado particular de nuestra base de datos en un momento determinado. En mi modesta opinión, y Vd. debería elaborar la suya, semejante idea es ahora un inconveniente grave. ¿Porqué?. Una base de datos es un programa abierto por naturaleza, horizontal, que exige necesariamente saber qué es un fichero, un directorio, y demás historias. Queriendo hacernos la vida más fácil, FileMaker Pro nos mete en un berenjenal considerable con su manía de grabarlo todo. La mayor parte de las veces, especialmente al principio, cuando hemos de familiarizándonos con un programa, nos interesa enredar con él, meternos en los ejemplos y probar cosas. No hay mejor forma de entender qué es borrar un registro que hacerlo en vivo y en directo. Pero también queremos poder volver a empezar dentro de dos o tres semanas, cuando tengamos tiempo otra vez, y no nos apetece nada en absoluto que se borren de verdad los registros del fichero original que tenemos en el disco. Y FileMaker Pro los borra también allí, inmediatamente y sin pedirnos permiso. Para evitar que desaparezcan cosas que queremos mantener, es preciso comenzar nuestras sesiones con FileMaker Pro grabando precisamente una copia para poder volver cuando sea a la situación de partida. Intentando evitar un problema que para muchos resulta inexistente (no es tan tremendo decirle al ordenador que grabe el fichero cuando haga falta) nos mete inevitablemente en uno mucho más real y peligroso. Y además inevitable, ya que no parece haber forma de desactivar tal comportamiento. La interfase de usuario no es configurable justo en el punto que más lo necesita<sup>118</sup>. Lo que puede estar muy bien en un programa vertical, puede perfectamente carecer de sentido en uno horizontal, especialmente si es de aplicación tan general y exige necesariamente tantos conocimientos a sus usuarios como una base de datos.

¿Debemos sacar como consecuencia de lo anterior que los programas verticales son mucho mejores que los horizontales?. ¿Podemos llevar el argumento hasta el punto en que pretendamos que el mundo de los PeCés, basado ante todo en el uso de programas abiertos y horizontales de utilidad general, es ridículo?. Antes de meterse por ahí, tenga en cuenta como ha surgido PADRE. Un grupo de programadores expertos lo ha desarrollado sin especial ánimo de lucro y con un grado más que notable de competencia, y lo han puesto a la venta a un precio mínimo, realmente testimonial y que seguramente cubre tan sólo los costes. El beneficio en este caso se obtiene por otro lado. Pero el programa no sirve para nada más. Si Vd. quisiera hacer todo aquello que puede pensar en encargarle a su ordenador

---

118 Yo al menos no he encontrado forma de hacerlo, aunque no digo que no exista. Por lo demás es un magnífico programa, con muchas más virtudes que defectos. Pero hubiera sido tan sencillo hacerlo aún mejor... bastaría con una entrada en “Preferencias” que permitiera activar o desactivar, a gusto del usuario, tan aberrante comportamiento. Alguien, en algún lado, aplicó con demasiada rigidez la idea de que no había que complicarle la vida a la gente, y el resultado fue justo el opuesto al que se pretendía.



mediante programas verticales, debería contratar una empresa de servicios informáticos para que desarrollaran para Vd. esos maravillosos programas verticales que tanto le admiran. Y pagar en consecuencia. Y no son nada baratas. Y hasta puede que no encontrara fácilmente alguien que le preparara un programa para hacer proceso de textos.

La utilización de programas verticales, diseñados para ser usados por personas inexpertas para hacer lo más automáticamente posible una única cosa, tiene un enorme sentido si detrás ponemos, de una u otra manera, a un grupo de personas encargados de mantener, controlar el uso, y desarrollar en el futuro los programas. Todo el misterio de que mucha gente use ordenadores intensivamente en su trabajo sin tener idea de qué son, como funcionan y para qué sirven, se reduce a que se les pone ante programas verticales parecidos a PADRE. Pero el coste es prohibitivo para un usuario particular. A éste le resulta mucho más conveniente invertir una cantidad moderada de dinero en un programa abierto y de uso general de gran difusión que él mismo pueda aprender y mantener en lo sucesivo, sin más coste adicional que el tiempo que deba dedicar a su aprendizaje. Es posible hacer la declaración de renta con una hoja de cálculo. Y no le quepa ninguna duda de que PADRE se ha desarrollado probablemente con una base de datos, un programa absolutamente horizontal, cuyo núcleo sigue estando ahí aunque no se le dé acceso a él<sup>119</sup>. En resumen, Vd. puede acceder mediante su PeCé a programas tanto verticales como horizontales. Ambas posibilidades existen. Pero Vd. deberá juzgar cual prefiere y su bolsillo le permite. La respuesta tradicional es que las horizontales son más apropiadas para un usuario personal, y las verticales mucho más convenientes en un entorno profesional, para confiarlas a un personal asalariado no particularmente formado y siempre que se contraten los servicios de una empresa para que se encargue del mantenimiento del sistema informático.

Más que de criterios de selección de software, nos hemos metido a discutir lo que podríamos llamar principios generales de planteamiento y utilización de un sistema informático, aunque sea personal. La elección vertical-horizontal no es la única a que debemos enfrentarnos. Podríamos meter muchas más, tales como si el ordenador lo vamos a usar nosotros o no, o el presupuesto de que disponemos para sacar nuestra instalación adelante. Sólo dedicaremos un mínimo de atención a una de ellas. Incluso dentro del software horizontal, se plantea muchas veces la disyuntiva entre la elección de programas sencillos y con un campo de aplicación limitado pero que cubra nuestras necesidades actuales, o de programas mucho más completos, que hagan eso y muchas cosas más aunque puedan costar algo más caros y ser más difíciles de usar en un primer momento debido a su mayor complejidad. Es decir, si son más convenientes los programas pequeños o los gordos.

Allá cada cual con su respuesta. Yo siempre he creído que es mejor empezar desde el principio con un programa que nos venga sobrado. Por varios motivos. Los grandes están-

---

119 Los programas verticales se desarrollan mediante lenguajes de programación o algo similar, como por ejemplo un programa abierto, tipo base de datos, que puede verse a su vez como un lenguaje de programación especializado. El programa terminado se “cierra” para que el lenguaje de programación con que se ha desarrollado no sea accesible al usuario más de lo necesario, que muchas veces es nada en absoluto. Como al menos el “motor” encargado de explotar los datos debe estar ahí, eso puede exigir que se incluya un núcleo capaz de hacer funcionar el programa. Se habla de módulos “runtime”, que normalmente pueden distribuirse libremente con los programas desarrollados en un entorno particular. No nos meteremos más con esto. Ya se enterará si alguna vez le da por programar.

## ***CAPÍTULO 8. PeCés: Programas***

dares se articulan con ellos. El precio no es un factor en su contra. Si se tiene en cuenta su capacidad, pueden ser incluso muchísimo más baratos que los más limitados. Puede que hoy solo queramos escribir cartas, pero quizá pasado mañana nos veamos en la obligación de redactar un texto de cien páginas. Entonces descubriremos que nuestro programa de escribir cartas no nos sirve y tendremos que comprar otro que cubra las nuevas necesidades. El esfuerzo, el tiempo, y el dinero invertidos en el primer producto no habrán servido para mucho. Y tendremos que ponernos a aprender el programa gordo justo cuando menos tiempo tenemos a nuestra disposición. Un buen consejo es fijarse en lo que hacen los profesionales, o nuestros amigos más expertos, y casi siempre optan por programas sobrados de potencia. Pero ese es mi punto de vista, y también era partidario del ordenador gordo. Vd. puede no estar de acuerdo con ninguna de las dos cosas.

### **Caso 3: Suites u Offices...**

El tercer caso práctico nos va a permitir completar la exposición de los tipos de programas con uno que pretende englobarlos a todos. ¿Se acuerda de los integrados, aquellos primeros intentos de facilitar el intercambio de datos en los PeCés cuando todavía no había un GUI, un entorno integrador en la jerga que entonces se usaba, que diera una solución mucho más general a tan importante problema?. ¿Se acuerda de que también se pretendió resolver mediante ellos lo de la unificación de la interfase de usuario, algo que de nuevo necesitaba un GUI competente para solucionarse a plena satisfacción?. Pues ahora podemos entender mucho mejor su historia.

En su momento, se pretendía que un integrado sustituyera con un único programa todo terreno a varios programas especializados de alto de gama que se comunicaban bastante mal entre sí y se aprendían mucho peor, ya que cada cual usaba sus propias convenciones. Lotus 123 quería inicialmente ser eso, al igual que Open Access, otro producto de la misma época. La idea era que con cualquiera de ellos pudiera hacerse, y con ventajas, todo lo que podía lograrse con WordStar, VisiCalc o Multiplan, y dBase.

Ya dijimos entonces que no tuvieron un excesivo éxito. Los motivos, ahora que sabemos lo que sabemos, son fáciles de entender. El principal, a mi modo de ver, es que las estructuras de datos a manipular para realizar un proceso de textos, una gestión de base de datos, y no digamos la producción de gráficos, son demasiado diferentes entre sí como para poder integrarse armoniosamente en un único producto. Además de que no sólo eran programas demasiado gordos para los tiempos que corrían, sino que muchas de sus teóricas capacidades eran más bien embrionarias. Lotus 123 hacía un muy buen papel como hoja de cálculo, pero era un más que discreto programa de gestión de bases de datos, y como procesador de textos mejor no hablar. Open Access era una aceptable base de datos, pero sus competencias en los demás campos eran bastante discutibles. A no mucho tardar, Lotus 123 acabó por considerarse lo que su estructura de datos fundamental decía que era, una buena hoja de cálculo. Y Open Access terminó siendo una base de datos digna. La elección de tales fundamentos para crear los correspondientes integrados debería estar también clara. Es precisa una estructura de datos potente y flexible, cuanto más mejor, para crear un programa todo uso. La estructura de un procesador de textos no sirve para preparar una facturación con comodidad, pero la de una base de datos, y en menor grado la de una hoja de cálculo, sí.

La diferente evolución de ambos productos es también sencilla de entender. Lotus 123 acabó siendo el estándar de las hojas de cálculo porque les añadió una flexibilidad que les vino muy bien. Hizo ese tipo de programas más completo y agradable, y además sufría moderadamente un problema que en otros integrados resultaba insoportable: el excesivo tamaño.

Los integrados fueron la primera entrega del luego famoso “fatware”, especialmente en su segunda generación, cuando intentaron hacerse aún más potentes. Productos como Symphony, el hermano mayor de Lotus 123, fueron simplemente desechados de entrada<sup>120</sup>. Por su parte, Open Access tenía que enfrentarse a un estándar quizá más establecido y muy anterior, el conocido dBase, si alguna vez quería ser una base de datos competente, y en este terreno no aportaba mucho que mereciera la pena. En lugar de cambiar el estándar de las bases de datos, desapareció.

Por supuesto, los procesadores de texto ignoraron el fenómeno, y también los programas de producción de gráficos. En lugar de tener un integrado para todo, la gente acabó con un integrado que se usaba como hoja de cálculo y los programas adicionales y especializados necesarios para hacer lo demás. Los mismos o parecidos problemas que antes y menos sitio en el disco duro.

Tras la debacle de los integrados de segunda generación, que se pretendían aún más potentes, y cuando estuvo claro que algún día llegaría a haber un GUI universal para PeCés que resolviera como era menester los dos problemas fundamentales que tantas veces hemos citado, la producción de integrados se reorientó. En lugar de querer ser el programa definitivo y más potente, y en consecuencia más gordo y más caro del mundo de los PeCés, pasaron a ser programas todo uso de introducción al ordenador. O sea algo más pequeño y con pretensiones mucho más modestas. La idea era que uno comenzaba por usar Microsoft Works, o Ability Plus, o Framework, y cuando estos se le quedaban pequeños se pasaba a Lotus 123, a WordPerfect, y a dBase.

Este nuevo enfoque permitía soslayar el problema del excesivo tamaño y el del precio desorbitado, y además colocaba a los integrados fuera de la competición con los programas más serios. En esta nueva orientación tuvieron mucho más éxito. Productos como Microsoft Works crearon una especie de estándar del ramo y siguen en producción, ahora para Windows. Pero tienen todos los problemas del programa pequeño que antes hemos comentado multiplicados por el número de aplicaciones que en ellos se incluyan. No hacen nada bien en realidad, y tras la introducción hay que pasar al uso serio. Mayores necesidades en cualquiera de los campos cubiertos por un integrado modesto obligará a cambiar de programa. Eso crea problemas, por lo que es mejor empezar directamente con algo que nos puedan durar más tiempo.

Los GUIs acabaron por imponerse en los PeCés, y Windows 3.1 llegó a ser más o menos lo que la gente esperaba desde hacía años. El intercambio de datos y la unificación de la interfase de usuario, esos dos tremendos problemas que tanto había costado solucionar, vieron mitigados sus efectos hasta el punto de que muy pocos se acordaban ya de su existencia. Herramientas como el portapapeles, el DDL, el OLE en versiones 1 y 2, y hasta la modesta estandarización impuesta por los programas DOS en unos cuantos formatos universales, deberían haber llevado al olvido la mera idea de integrado. Al fin y al cabo Windows se justificaba ante todo en base a que cualquier programa de cualquier fabricante pudiera

---

120 Recuerde, como dijimos en otra nota, que aquellos monstruos se considerarían ahora un prodigio de esbeltez. Nadie en su sano juicio llamaría hoy gordo a “Symphony”. Si se siente inspirado, puede reflexionar algunas horas sobre este punto.

## ***CAPÍTULO 8. PeCés: Programas***

comunicarse correctamente con los demás, y todos, fuera cual fuera su origen, se parecieran lo suficiente en su manejo como para resultar inmediatamente familiares a cualquier usuario.

Pues de eso nada. Las razones técnicas anteriores palidecen en importancia frente a las leyes del mercado. Para un editor de software no está mal dominar un segmento determinado, pero es mucho mejor si se dominan todos. Tras la idea del integrado ha estado siempre la voluntad de los fabricantes de software de monopolizar todo el software para PeCés con un único producto.

Así, hace no muchos años, algo parecido a los integrados volvió a salir a escena. Una vez más se trataba de sustituir con un único producto a todos los programas de alto de gama. Creo que fue Microsoft quien reeditó la fórmula con su Microsoft Office, y comenzó la actual guerra de los “paquetes office” o de las “suites”, como también se conocen. Como Windows funcionaba razonablemente, ya no era necesario desarrollar nuevos productos. Se trataba claramente de ganar cuota de mercado en todos los sectores a base de aprovechar el tirón del mejor o más famoso producto con que se contara. Si un editor tenía una muy buena hoja de cálculo que vendía a 35.000 pesetas, se ofertaba a 45.000 un paquete en el que se incluían esa misma hoja de cálculo, el procesador de textos de la casa, una agenda, un programa para generar gráficos de negocios, y tres o cuatro cosas más. Era tan sólo un cambio de envoltorio, algo así como el “compre cuatro por el precio de dos”. El éxito de la idea sorprendió, al parecer, a todo el mundo, y eso hizo que, uno tras otro, todos los editores se lanzaran a proponer semejantes paquetes<sup>121</sup>. Hoy en día la competencia más feroz por ver quien domina el mundo de los programas Windows parece plantearse entre Microsoft Office, Lotus SmartSuite y Corel WordPerfect Suite, creo que se llaman. Con amplia ventaja para el primero.

No habría mucho que objetar al fenómeno anterior si no fuera por un par de cosas que no siempre quedan lo suficientemente claras. Al fin y al cabo lo de comprar a precio reducido nunca viene nunca mal.

Lo primero que me hace muchísima gracia son los argumentos usados en la promoción de las suites. Se sigue hablando de facilitar el intercambio de datos y de unificar la interfase de usuario, como en los tiempos de Lotus 123 y Symphony. Deberíamos ser un poquito más serios y partir de la base de que si es preciso recurrir a un conjunto de programas especiales para que funcionen aceptablemente los dos puntos que justifican un GUI, el problema, en caso de existir, se encuentra a nivel de ese GUI que no es tan integrador como debiera, y es ahí donde debería corregirse.

Y lo segundo es que con las suites se reeditan, multiplicados por un factor bastante alto, los problemas que hicieron morir a los integrados. Llevados por el espejismo de una compra

---

121 Muchas veces he leído en las críticas de coches de tipo deportivo o poco prácticos, que su justificación consiste en generar lo que parece llamarse “efecto locomotora”. Una compañía lanza un coche de prestigio con el que seguramente no ganará dinero con el objeto de mejorar su imagen de marca y vender muchas más unidades de sus productos más populares. Los paquetes office, las suites, son un ejemplo bastante claro de este mismo truco comercial aplicado a otro campo. Si mi procesador de textos o mi hoja de cálculo no son tan buenos como los de la competencia, aún puedo aspirar a monopolizar el mercado y arruinar a los demás editores si, aprovechando el tirón de mi paquete de gráficos, consigo quitarles sus clientes.

barata, el usuario suele acabar con un grupo de programas de los que usa un máximo de uno o dos. Los demás suelen dormir el sueño de los justos ocupando una cantidad indecente de espacio en el disco duro. Tal vez comprar los dos mejores programas que realmente empleamos de cualquier fabricante nos hubiera salido mucho mejor de precio que esa suite que tan barata nos pareció. La idea de una interfase de usuario única es ridícula. Aunque Vd. compre la mejor suite del mundo, aún así, va a tener que emplear el administrador de ficheros, el de programas, y el de impresión de Windows, su panel de control, y otras doscientas aplicaciones que no forman parte de la suite. Le conviene aprender a moverse entre aplicaciones que se parezcan lo justo y nada más.

Porque es que además, las opciones de una base de datos, su funcionamiento, jamás pueden ser iguales, como ya deberíamos saber, a las de un procesador de textos o una hoja de cálculo. Se basan sobre estructuras de datos diferentes que determinan operaciones y posibilidades distintas, por lo que el parecido se limita a una serie de detalles cosméticos sobre cuantas barras de botones aparecen o dejan de aparecer y muy poco más<sup>122</sup>.

Por si fuera poco, existe cierta picaresca. El mismo fabricante que vende una suite vende los productos aislados a mayor precio. ¿Cómo es posible?. Si mira atentamente puede ver que las versiones incluidas en la suite no son necesariamente las más recientes. Y si cuenta los precios de actualización quizá el negocio no sea tan redondo.

Como las necesidades de cada grupo de usuarios son diferentes, han llegado a aparecer al menos dos versiones de las suites. La normal incluye necesariamente una hoja de cálculo, un procesador de textos, y un editor de gráficos de negocios (un programa de “presentación”), al que siempre se añaden algunas pequeñas cosas más tales como agendas o algún otro detallito. La profesional cuenta siempre, además de lo anterior, con una base de datos, y quizá con algunas herramientas para redes.

A mí, y a nivel absolutamente personal, no me gustan las suites. Creo que distorsionan el mercado de los programas de aplicación hasta extremos difíciles de soportar, sin aportar nada real a cambio. Lo propio es que si alguien quiere dominar el sector de las hojas de cálculo intente desarrollar una mejor y más barata que las demás, no que siga fabricando procesadores de textos, magníficos eso sí, y aprovechando su tirón para evitar que la mejor hoja de cálculo para Windows, por supuesto de la competencia, se venda como debería. Además, ninguna de las suites cubre realmente mis necesidades. En ninguna hay un grupo de programas para gráficos ni lejanamente similar a Corel Draw, y sin eso, para mí, están absolutamente cojas.

Pues bien, para que se dé cuenta de lo relativos que deben ser los criterios de selección de programas, para que vea porqué hemos metido semejante ejemplo junto a los demás, procedo a hacerle saber, inmediatamente después de decir lo anterior y tras añadir que podría

---

122 Una vez más, hay que dar al Cesar lo suyo. Al menos Microsoft y Corel WordPerfect proponen también lenguajes de programación para automatizar el funcionamiento de los programas que integran una suite. Quizá también Lotus. No me atrevo a decir si funcionan o no. Jamás los he probado, pero disponer de tales herramientas podría ser una ventaja mucho más real que esos cambios de apariencia tan idiotas de que estamos hablando.

## **CAPÍTULO 8. *PeCés: Programas***

seguir desgranando una agria letanía sobre lo que opino de las suites, que hace no demasiado yo mismo adquirí una.

“¡Traición!”. “¡Perjurio!”. “¡Vaya tomadura de pelo!”. Seguramente esos y otros similares son los gritos de los pocos LAOs que todavía perseveran en la lectura de este texto ante tamaña desfachatez. Me explico. Sigo siendo partidario de los productos aislados, comprados de uno en uno. Pero cuando Novell los lanzó en lo que entonces se llamaba PerfectSuite, agrupó espontáneamente mi hoja de cálculo y mi procesador de textos en una suite, y les añadió cosas que me interesaban tanto como el ya citado Envoy. El coste de actualizar mi WordPerfect 5.2 a la versión 6.1 era tan sólo ligeramente inferior al que se proponía para conseguir el suite completo, y allá que me fui. Esa agrupación espontánea de productos aislados que yo ya había elegido individualmente hacía algunos años me permite seguir denigrando las suites y poder comprar una de ellas si el precio me resulta interesante sin incurrir en contradicción. Además, la posterior adquisición de PerfectSuite por parte de Corel ha mejorado aún más el panorama. Pero no me retracto de ninguna de mis opiniones sobre las suites. Simplemente mi caso es un poco particular, y como tal debe ser analizado y valorado. Si le resulta conveniente por algún motivo razonable el comprar una suite, pues no se lo piense dos veces y adelante. Pero sepa en qué terreno se está moviendo y procure que no le den gato por liebre<sup>123</sup>.

Y al finalizar los ejemplos, prácticamente hemos acabado el capítulo. Una vez más ha resultado demasiado largo. Si tuviera un poco de vergüenza debería pedir disculpas inmediatamente, ya que prometí un capitulito desintoxicante y de nuevo no está muy claro que el resultado se ajuste a lo prometido. Pero no debo tener demasiada en los últimos tiempos, ya que no sólo no voy a disculparme sino que creo que ha sido absolutamente imprescindible para acabar unos temas y preparar los que nos quedan por ver. No pude evitarlo<sup>124</sup>. Espero que al menos, y por ya muchos y muy diversos motivos, vaya quedando clara la idea general de que es sumamente útil seguirle la pista a los datos en todo momento.

Algunas cosas más deberían también ir quedando suficientemente patentes. Si tras todo lo anterior le acaba pareciendo que juzgar un programa exige tener la oportunidad de probarlo en el ordenador en que se vaya a usar, durante una cantidad de tiempo razonable, pues la verdad es que parece que va entendiendo lo que estamos queriendo decir de muy diversas

---

123 Por ejemplo, supongamos que Vd. compró el PC Magazine en que venía Corel Draw 3.0 completo y con su licencia de uso y todo. Es decir, el ejemplar de junio de 1.996. Supongamos que se registró. Es probable que pueda comprar el suite de Corel a precio de usuario registrado. Por un precio realmente ridículo acabaría siendo el legal propietario de una colección de software de calidad algo más que apañada y que constaría de un procesador de textos, una hoja de cálculo, un paquete de gráficos de verdad, y algunas otras cositas.

124 Algo así le repetía un estupendo John Malkovich a una no menos estupenda Michelle Pfeiffer en la película “Las amistades peligrosas” de Stephen Frears mientras le destrozaba meticulosamente, y ya de forma definitiva, su maltrecho corazón. Tampoco él salía ileso de la aventura. Aunque yo tampoco he podido evitarlo, espero que las consecuencias no sean tan terribles y puedan solucionarse mediante el procedimiento habitual, es decir recurriendo a un rato de sano esparcimiento y quizá a moderadas cantidades de algún bebedizo estimulante. Nos queda muy poco para acabar y tampoco podré evitar que los dos capítulos que restan sean muchísimo menos técnicos y probablemente bastante más divertidos.

maneras. Por lo menos por lo menos, para poder comprar un programa sin arriesgarnos a tirar ese dinero que tantos esfuerzos nos cuesta directamente a la basura, hay que tener la oportunidad de usar una demo durante una temporada. No se aprecian las incompatibilidades, por poner solo un ejemplo, a partir de un folleto de propaganda del producto. Ni tampoco esas sutiles ventajas, ese cúmulo de detalles, que son los que acaban por hacer que un programa merezca el honor de que lo encarguemos a nuestro proveedor habitual.

¿Va a servir de algo tanto esfuerzo, o el panorama del software para PeCés va a cambiar tan deprisa que cuando acabe de leer la última palabra de este capítulo ya nada de lo anterior se le podrá aplicar?. Volveremos a ello, especialmente en el próximo capítulo y al final del todo, en el epílogo, pero la extraña manera de abordar las cosas que estamos utilizando tiene la gran ventaja de que casi todo lo que aprenda tiene muchas posibilidades de perdurar bastante tiempo. Los nombres de los programas variarán, por supuesto, pero las estructuras de datos seguirán ahí. Hace mucho tiempo que no se introducen nuevas estructuras, nuevos tipos de programas, y algunos de los intentos más legítimamente revolucionarios han sido ignorados. El panorama es asombrosamente estable cuando se mira de la forma que estamos indicando. No está nada claro que la gente quiera cosas diferentes en el mundo del software para PeCé.

Por ejemplo, hace algunos años, y desarrollada inicialmente para un ordenador llamado Next (ver glosario), apareció una especie de hoja de cálculo llamada Improv que modificaba notablemente el funcionamiento habitual de este tipo de programas. Lotus, el padre de la criatura, llegó a sacar una versión Windows. Que fue ignorada de manera casi unánime. La gente siguió comprando Lotus 123, o Excel, o Quattro Pro. Sólo algunas estructuras mixtas que palian los muchos defectos de los gráficos bitmap parecen merecer un poco más de interés, y han hecho que aparezca un tipo nuevo de programas de creación de gráficos, a caballo entre el bitmap y el vectorial, en el que podemos incluir a Corel Xara y Aldus Freehand, que creo ya citamos.

¿Cómo es el software fuera del PeCé?, quizá se pregunte. Con excepción del Mac, probablemente más potente y más hostil. Si se mete por ahí en algún momento, no será extraño que se vea sumergido en un mundo en el que sigue habiendo caracteres de control, líneas de órdenes, y demás inventos antediluvianos. También lo que aprendió en capítulos anteriores debería servirle en una situación tan poco probable. Al fin y al cabo, Vd. es un recién llegado con pocas ganas de complicarse la vida. ¿O no?.

Aunque hemos dedicado mucho espacio a hablar de estructuras de datos y algoritmos, hemos terminado ocupándonos de efectos económicos y argucias de vendedores. Por suerte o por desgracia, también eso forma parte de la informática actual, e ignorarlo es tan o más peligroso que desconocer la norma 8.3. En realidad hemos estado bordeando peligrosamente, o invadiendo claramente, un terreno que pertenece con mayor propiedad al siguiente capítulo. Razón de más para que terminemos éste de inmediato. Y para que le sirva de consuelo, si cree que de una forma u otra he acabado por endilgarle uno de esos palos teórico-trascendentes que la mayor parte de lectores intenta evitar como la peste, sepa que no preguntaré nada de esto en el examen final. No habrá. Daré aprobado general a todo aquel que consiga terminar el libro.