

CAPÍTULO 5

*Software: Manejo de programas*Solo ante el peligroCara a cara con Luis Ricardo

Distribución a particulares Luis Membrado Giner Andador Anayet, 4, 6º C 50015 Zaragoza e-mail: lmg00009@telefonica.es lmg00009@teletelnet.es	Distribución a librerías Mira Editores Concepción Arendal, 22 50005 Zaragoza Tel: 976 354165 Fax: 976 351043 e-mail: lcentral@ctu.es
--	--

Entro en la habitación y me siento ante él. Pulso el botón de encendido. Obediente, se pone en marcha. Poco después, termina de escribir cosas, de hacer parpadear las luces de la unidad central, y algo en la pantalla parece estar invitándome a que le diga qué quiero que haga. Me mantengo a mi vez a la espera. Tamborileo con mis dedos cerca del teclado y reflexiono.

No es la primera vez que nos enfrentamos. Ahora ya sé como ejecutar el programa que siempre empleo. En realidad es muy fácil. Basta con escribir TLW y darle a la tecla gorda esa de la derecha. ¡Que risa!, la primera vez no sabía que había que usar el teclado y estuve un buen rato moviendo el ratón como un loco. Como no pasaba nada, llegué a llamar a la casa que me lo había vendido, creyendo que estaba estropeado. Ellos me dieron la clave de entrada.

Ahora ya no me ocurren esas cosas, pero no termino de verle el sentido, a pesar de que ayer hasta conseguí que lo que había escrito en la pantalla saliera por la impresora. Fue al pulsar por error una de las teclas de la fila de arriba, que yo creía que no valían para nada. Los márgenes no salieron bien y la letra era horrible, pero bueno, algo es algo. Tal vez el problema sea que el cable de la impresora esté en mal estado. Mi amigo Pablo me enseñó una carta suya hecha con un equipo como éste y tenía un aspecto magnífico. Quizá debiera comprar otro cable. Esperemos que hoy salga mejor. Me gustaría entender como funciona, pero debo ser muy torpe. De todas maneras, iparece todo tan raro!. En fin, vamos a empezar...

Lo anterior no es un exagerado cuentecillo de terror informático en estilo más o menos kafkiano. Creo que muchísimos usuarios de equipos informáticos (yo, al menos, lo he hecho) han pasado o siguen pasando por situaciones similares. Es casi inevitable cuando uno es un usuario novel.

El problema no es el cable de la impresora. El problema es el mismo de entonces. Consiste en que la primera vez que nos pusimos ante un ordenador muchos no sabíamos qué recursos teníamos a nuestra disposición, cuales eran las pautas de comportamiento a seguir, por qué razón podíamos esperar que las cosas fueran de una forma y no de otra, qué teníamos que buscar y qué ignorar.

Nos enfrentamos a un ordenador bastante hostil muy a la brava y, tras muchas heridas, desesperos, trabajo, y con algo de suerte, salimos adelante. Parece que se nos haya olvidado a todos, que creamos de verdad que, como los ordenadores actuales son más potentes y sencillos de manejar que los de hace 15 años, no hace falta que le expliquemos a nadie lo que aprendimos.

O quizá estemos pensando algo así como: “Ya hay bastantes torpes estorbando con los ordenadores. Si alguien no es capaz de aprender solo hoy en día, con la cantidad de libros,

© 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, compre un ejemplar en lugar de imprimirlo ya mismo. Le saldrá más económico y el resultado será de mejor calidad.

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

CAPÍTULO 5. Software: Manejo de programas

revistas, y cursillos que hay, y lo fáciles de usar que son los programas, mejor que se compre un juego de marbianos y se pase el día matándolos. Así le parecerá que usa el ordenador para algo y no molestará. Uno menos.”

Yo creo que sí que hay que contarlos, y por eso estoy escribiendo esto. Puedo hacerlo, porque no crea que aprendimos grandes maravillas. Nada de métodos ignotos de resolución de ecuaciones diferenciales de increíble dificultad. Nada de oculta sabiduría del Oriente revelada misteriosamente por un mago que se nos aparecía las noches de Luna llena a través de nuestro monitor. Cuatro estupideces tan gordas que realmente me sorprendería mucho que no las entendiera prácticamente todo el mundo. Es tan fácil...

¿Y cómo, si hay que saber tan poco, hay alguien capaz de dinamitar su ventaja contando estas cosas?. No hay tal. Yo, ya lo sé. Como verá, no se trata sólo de recetas. También hay que recorrer un camino por el que yo ya he pasado. Ya sé qué hay y qué no. Tengo experiencia (15 años por lo menos). Y, en última instancia, casi preferiría que los ordenadores se entendieran mejor. Así podría rentabilizar aún más mi ventaja. Los haría más populares y evitaría que la cosa acabara en un bluff. Algo bastante difícil ya, por cierto. Aunque de este modo sea mucho más fácil para cualquiera, aún así, Vd. tiene un montón de trabajo por delante. Para empezar, acabar de leer el libro. Vamos a facilitarle la tarea todavía más entrando prontamente en materia.

Como dicen de nuevo “Les Luthiers”: “El camino de la sabiduría es laaargoooo...”¹, y si el LAO no desea o no tiene el tiempo necesario para llegar a esta especie de zen informático, existe otra vía. Dejando bien claro que es absolutamente necesario un mínimo de lo anterior y, por tanto, hay que leer lo que sigue, es perfectamente posible reconocer las limitaciones en tiempo, inteligencia, habilidad, o ganas de cada cual, y aspirar tan sólo a convertirse en un digno currito informático. Si Vd. limita el número de programas a usar, sólo quiere hacer cuatro cosas, y se aprende lo mejor que pueda en su totalidad esos cuatro programas, tenga por seguro que puede que le saque bastante más partido a un ordenador que muchos virtuosos.

Sólo hay que saber dos cosas

El equipo para domar a Luis Ricardo

Dijimos al final del capítulo anterior, por abrir boca, que con sólo dos cosas, exactamente dos, iba a ser posible manejar cualquier ordenador. Me reafirmo en ello y para entrar como se ha anunciado en materia, paso a exponerlas.

Para manejar cualquier ordenador con soltura y poder considerarse, tras un período de práctica no muy largo, un virtuoso, tan sólo es necesario entender dos componentes del software: el sistema de ficheros y la interfase de usuario.

Al sistema de ficheros lo presentamos sucintamente en el anterior capítulo. Hay más que decir sobre él y lo iremos diciendo, pero puede que convenga empezar por lo otro, la interfase de usuario. Podemos dar varias razones. Ya sabemos algo de lo primero y tiempo

1 Por terminar la broma, la cosa sigue tal que así: “...encontrarás la destreza en Kyoto, hallarás la fuerza en Ku Wan, pero La Paz... se encuentra en Bolivia”.

habrá, incluso en este capítulo, de profundizar. No se debe aburrir al lector hablando siempre de lo mismo. Por otra parte, es mejor disponer de una visión general cuanto antes, y son dos temas que es preferible tratar simultáneamente. Además, conocer la historia de las interfases de usuario aclara muchísimas cosas, y es por ahí por donde empezaremos, tras dar una definición ahora mismo.

Una interfase es, ya lo vimos, el punto de contacto de dos cosas diferentes. Lo habíamos usado para los enchufes que permiten conectar cosas adicionales a la unidad central. La frontera era el interior del ordenador, y la interfase el punto de contacto para poner el exterior a su alcance. Una de nuestras interfases de expansión era el punto de contacto entre el ordenador y un equipo adicional. Pero hay una frontera mucho más importante. Aquella que separa al ordenador y al usuario. La que hemos tratado de presentar con nuestro cuentecito. La interfase de usuario no es un enchufe, sino el conjunto de métodos que se ponen a nuestra disposición al diseñar un programa para dejarnos acceder al ordenador y manejarlo. Es el punto de contacto entre nosotros y el ordenador, la vía a través de la que vamos a comunicarnos con él.

Paneles de conexión

Luis Ricardo Neanderthal

Vimos, al hablar de formatos de datos, que cada programa lo hacía como mejor le parecía. Vimos, al hablar de la apariencia de Luis Ricardo, que cada programa podrá hacer que se nos presente como a su diseñador le venga en gana. Todo lo relacionado con el manejo del ordenador está marcado por esta profunda indefinición, tan peligrosa². Ya que entender y manejar las distintas interfases de usuario requiere saber como reconocerlas a simple vista, y sentirse cómodo ante ellas parece posible tan sólo tras perderles el respeto que nos puedan infundir, nada mejor que empezar dándonos una vuelta por sus humildes orígenes. Aireemos sus trapos sucios.

Yo pensaba empezar la sección con algo así como “...al principio fue el teclado...”³, pero creo que, para ver lo cutre que puede llegar a ser el manejo de un ordenador, voy a empezar por el principio de verdad.

La interfase de usuario de los primeros ordenadores modernos (nos olvidamos de Babbage y demás caterva) era un panel de conexiones parecido al de las telefonistas de antes. Unos agujeros en un tablero donde se montaban puentes eléctricos. Con este simpático invento se le decía al ordenador lo que queríamos. ¿Suena tétrico?. Lo era. Al menos para el desdichado que tenía que emplearlo. Para el que lo había construido era francamente fácil de hacer, como espero que crea pues no voy a ponerme a explicar porqué. Ya tenemos

2 Nota filosófica de ampliación: es absolutamente inherente a la naturaleza del ordenador como máquina. Una vez más, no podía ser de ninguna otra manera. Como consecuencia, es inevitable, y no busque alternativas. Acéptelo así y asúmalo lo más profundamente posible. Intente meterse en la piel del ordenador. Podríamos bromear diciendo algo así como “¡Oh, extranjero! Esta es la primera regla del zen informático que deberás observar” si no fuera porque tanto el zen como nuestros problemas son cosas bastante serias.

3 Como se avisó, voy a mentir siempre que me parezca, por algo soy yo el que escribe. Enseguida veremos que no fue el teclado lo primero, ni mucho menos.

CAPÍTULO 5. Software: Manejo de programas

bastante tajo entre manos. ¿Qué pasaba si yo era nuevo en el oficio y tenía que ponerme a manejar un ordenador?. Cogía los enormes y probablemente precariamente escritos manuales a mi alcance y me los leía tratando de entenderlos. Dejemos a nuestro desdichado operador prehistórico en tan apasionante tarea e intentemos extraer alguna enseñanza.

Lo que sí que quiero que vea es que, ya con esta primera y rústica interfase de usuario (IU⁴), aparece una dicotomía de lo más previsible entre coste-facilidad de construcción y comodidad de empleo. Es fácil y barato hacer una IU muy difícil de usar, pero exige bastante más tiempo, dinero, y talento, hacer una que sea sencilla y potente de uso. No sólo eso, sino que las IU más simples no van a requerir más que una mínima parte de los propios recursos del ordenador para funcionar (memoria, CPU, almacenamiento en disco), en tanto que con una muy sofisticada puede que el 80% del tiempo de trabajo y de sus recursos en general se los chupe la dichosa IU.

Y también que los manuales están ahí para algo más que para hacer bulto. Que yo sepa, ni siquiera ahora es posible coger un ordenador, enchufarlo, y entenderlo todo maravillosamente por arte de birlibirloque. Hay que leer cosas. Por ejemplo este libro. Y cuando hay problemas, acordarse de que hay unos manuales donde debería haber información adicional.

Extraídas tan sabias enseñanzas, olvidémonos rápidamente de los paneles de conexión, felizmente extintos. No duraron mucho, de todos modos. Se pasó pronto a las famosas tarjetas perforadas, que tuvieron una vida mucho más dilatada. Ya las citamos y no tienen mayor interés. Por suerte pertenecen también al pasado.

Y luego vino el teclado

Luis Ricardo Cromagnon

De las IU no extintas y, en realidad, ni siquiera en peligro de extinción, las primeras en aparecer y más antiguas, por tanto, son las basadas en el teclado. Fueron posibles con un mínimo de comodidad a partir de la adición de teclados y pantallas al ordenador⁵. Como regla general, totalmente aplicable a la actualidad (al fin y al cabo el Cromagnon es de anteaer), yo le diré mis cosas al ordenador con el teclado y él me contestará a través de la pantalla. Consecuencia inmediata: hoy en día, si quiero enterarme de lo que Luis Ricardo tenga que decirme, más valdrá que mire a ver si hay algo en la pantalla.

Con una pantalla⁶ y un teclado es posible poner en funcionamiento la primera de las grandes IU con las que vamos a tener que enfrentarnos: la línea de órdenes. En inglés, “command

4 En adelante, IU, por abreviar. Si lo desea en inglés “User interface” o UI. Como las alternativas parecen equivalentes, usaremos la castellana.

5 Al principio había teclados pero no pantallas. Una impresora se encargaba de mostrarnos lo que el ordenador tuviera que decir. Era maravilloso gastar toneladas y toneladas de papel para intentar decirle a Luis Ricardo que no, que no era eso lo que queríamos que hiciera, sino lo otro. Yo todavía he usado equipos así, y, aunque no soy un niño, espero que sirva para demostrar que no hace tanto. Otra posibilidad era que un teletipo (que ya vimos) hiciera a la vez de teclado e impresora. Esto añadía aún más morbo al asunto.

6 No estamos hablando de una pantalla gráfica de alta resolución y enorme número de colores sino de CUALQUIER pantalla, por ejemplo una que sólo permita visualizar una línea de texto

Y luego vino el teclado

line”. La cosa consiste en que el ordenador me muestra algo en una línea, con lo que me indica que está listo para atenderme y tal vez hacerme caso. Ese algo es lo que se denomina un “prompt”. Con él, el ordenador me dice que es mi turno de escribir. El método habitual consiste en que yo veo en pantalla un símbolo más o menos parpadeante (el cursor)⁷ y quizá un texto adicional para orientarme. Por ejemplo algo así como esto:

DIME QUE HAGO AHORA: _

podría ser un prompt perfectamente válido⁸. El cursor es el punto en que va a ir apareciendo lo que yo escriba. Yo tecleo las órdenes en una línea de texto, el ordenador me muestra lo que voy tecleando, hasta me lo suele dejar corregir ya veremos como y, cuando he terminado, hace lo que le he dicho si lo entiende, me muestra los resultados también normalmente a través de la pantalla, y vuelta a empezar.

Una “línea de órdenes” se llama así porque yo sólo tengo una línea para escribir. Nada más⁹. Tal vez, si el equipo es medio decente, el ordenador emplee más líneas para sacar la respuesta, pero yo sólo tengo una para decirle lo que debe hacer.

Esta IU, es casi el mínimo hoy día. Sus ventajas son la escasa cantidad de recursos necesarios para equipar a un ordenador con ella (implementarla, se llama a eso, y será el vocablo que empleemos en lo sucesivo) y la enorme potencia que pone a nuestra disposición. No de forma cómoda, hay que admitirlo, pero la “línea de órdenes” tiene una relación potencia/coste descomunal.

Vamos a estudiar tan portentoso invento con un poco de detalle porque veremos que se pueden aprender muchas cosas. Por ejemplo, qué diablos son exactamente y para qué se emplean los famosos caracteres de control que nos habían aparecido en el código ASCII. Pero eso vendrá luego. Comencemos con sus ventajas e inconvenientes. Para ello basta que

de 20 caracteres. ¿Que no hay de eso ya?. Mentira y gorda. La próxima vez que pase por unos grandes almacenes, tenga a bien fijarse en los terminales de punto de venta (ver glosario) y tendrá sobrados motivos para retractarse inmediatamente, osado LAO. Ya que estamos, tampoco se trata de un teclado de PeCé de los buenos, sino como veremos a no mucho tardar de CUALQUIER teclado.

- 7 El cursor es algo multiforme, susceptible de adoptar diversos aspectos. A veces es un subrayado parpadeante, otras un bloque fijo, en ocasiones una rayita vertical... Estos caprichosos cambios de forma no lo son tanto en ocasiones, ya que pueden usarse para indicar cosas. Nosotros lo pintaremos como un subrayado. Cuando lea los ejemplos que siguen, el subrayado del final indica simplemente donde se encontrará el cursor. Imagínese lo parpadeante, además. Como habrá adivinado, el que lo sea es tan sólo una forma fácil de llamar su atención y de facilitar su, siempre necesaria, localización.
- 8 Lamentablemente, los prompts son normalmente menos explícitos. Suelen parecerse mucho más, como veremos, a algo así como: “C:> _”. Veremos también, de todas maneras, que es perfectamente posible, e incluso fácil, convertir tan inhóspito paisaje en la versión que aparece en el texto principal.
- 9 La longitud de la línea puede ser superior a la anchura de la pantalla, sin embargo. Por ejemplo, es posible que podamos escribir nuestras órdenes en una línea de hasta 256 caracteres y sólo tengamos líneas de 80 en pantalla.

CAPÍTULO 5. Software: Manejo de programas

consideremos como le pasamos las órdenes a Luis Ricardo. Como un texto. La línea de órdenes implica la existencia de una serie de palabras que Luis Ricardo entiende y que son las que yo empleo para decirle qué tiene que hacer. Tenemos un idioma con el que comunicarnos.

Si este idioma fuese el castellano, con sólo que fuese el inglés, los ordenadores serían maravillosos. Desgraciadamente, nuestro apreciado Luis Ricardo no es muy hábil para los lenguajes humanos. Le resultan tan difíciles que hay que montar idiomas artificiales para hablar con él. Por suerte, aunque tan sólo relativa, los idiomas que alguien tan estúpido es capaz de entender son muy simples. Y los humanos los podemos aprender sin gran esfuerzo. Pero los tenemos que aprender nosotros. Un gran inconveniente.

De todos modos, incluso con esos idiomas tan sencillos, la capacidad expresiva de las palabras es tal que podemos, estrictamente hablando, hacer que sea posible mandarle al ordenador cualquier cosa. Se puede poner la totalidad de los recursos del ordenador a nuestro alcance con una IU tan sencilla como ésta. Una gran ventaja.

Por desgracia, los lenguajes de ordenador tienden a ser muy poco explícitos. ¿A qué se parecen?. Bien, una orden válida, con el prompt anterior, podría ser como sigue:

```
DIME QUE HAGO AHORA: DIR_
```

DIR es la palabra que le indica al ordenador que me muestre un listado de los ficheros que tengo accesibles en el momento de darle la orden. El listado de los ficheros se llama “directorio”, y DIR no es sino una abreviatura fácil de recordar que nos evita escribir toda la palabra¹⁰. Un poco críptico, pero manejable. La cosa se complica cuando quiero pulir algo la cosa, ya que lo anterior, si tengo más de 24 ficheros a mi alcance, implica automáticamente dificultades para ver el listado en la pantalla. Simplemente no caben, y eso hace que la pantalla se desplace hacia arriba, con lo que no veo los primeros que han salido. En un ordenador medianamente rápido es estrictamente imposible leerlo.

Ha llegado la hora de los parámetros. Si añadimos algo así como:

```
DIME QUE HAGO AHORA: DIR /W_
```

es bastante más probable que lo pueda leer, ya que ahora Luis Ricardo va a intentar meter todos los nombres de ficheros que pueda en la misma línea, con lo que podré ver muchos más en una pantalla. Pero si tengo muchos de verdad, aún tendré los mismos problemas de antes. Una cosa parecida a:

```
DIME QUE HAGO AHORA: DIR /W /P_
```

conseguiría no sólo que Luis Ricardo aprovechara las líneas, sino que, en caso de ser necesaria más de una pantalla para listar todos los ficheros, Luis Ricardo esperara a que yo se lo dijera para pasar a la página siguiente.

10 No hay que olvidar que la gente a este lado de la pantalla no tienen porqué ser maravillosos mecanógrafos. Esto, y alguna cosa más, explica la profusión de abreviaturas en los lenguajes informáticos.

Mejoras en la línea

El lenguaje de una línea de órdenes consistirá habitualmente en una serie de pequeñas palabras susceptibles de ver modificado su comportamiento mediante parámetros aún más pequeños. Y normalmente disponibles en buen número.

Como se ha dicho, soy yo el que tengo que aprender el idioma. Soy yo el que tengo que saber lo que hace cada una de, pongamos, 25 palabras, y unos 10 parámetros por palabra. Tengo que memorizar un buen montón de cábala. Nada cómodo. Éste es uno de los problemas que espantan a buen número de usuarios, y uno de los principales inconvenientes de las líneas de órdenes. A cambio de una enorme potencia puesta a nuestra disposición con el uso de muy pocos recursos del ordenador, tengo que aprender y memorizar un lenguaje más bien abstruso. Una lata.

Si le añadimos que las posibilidades de corrección en una implementación no muy sofisticada de una línea de órdenes son más bien escasas, la cosa se hace francamente inhóspita. Por ejemplo, la única posibilidad de corrección puede ser borrar el último carácter que se ha escrito¹¹. Esto hace que para pasarle a Luis Ricardo algo inteligible si he cometido el error, al parecer imperdonable, de escribir:

```
DIME QUE HAGO AHORA: DUR /W /P_
```

que por supuesto Luis Ricardo no va a entender, yo tenga que llevar el cursor hacia atrás hasta el carácter equivocado, viendo con horror como lo que he tecleado previamente desaparece, dejándolo en consecuencia como:

```
DIME QUE HAGO AHORA: D_
```

y encontrándome tristemente obligado a reescribir penosamente el resto. Además, con todo el cuidado del mundo para no meter la pata de nuevo y tener que volver a empezar.

Mejoras en la línea

Luis Ricardo y sus caracteres de control amaestrados

No existen palabras aptas para oídos pudorosos que permitan calificar adecuadamente la comodidad de uso ofrecida por una IU de línea de órdenes tan asilvestrada como la que se termina de describir. Tanto más cuando, como vamos a ver, es posible lograr versiones mucho más civilizadas con muy poco esfuerzo adicional.

Los problemas asociados a una IU de línea de órdenes pueden resolverse en buena medida si nos preocupamos de implementar mecanismos que permitan una corrección fácil de los errores de mecanografiado, dejen recuperar de las órdenes dadas con anterioridad, y que eviten en lo posible la necesidad de memorizar las órdenes y los parámetros. El como se hace esto y, sobre todo, el ver como se hacía, nos va a llevar a otra serie de puntos interesantes.

11 Esto, que es básicamente lo que vimos con el teletipo que repetía caracteres solo que ligeramente mejorado ya que al menos lo veo limpio en la pantalla, ha sido lo habitual en el mundo de los PeCés hasta hace cuatro días. No es de extrañar que hayan adquirido fama de desagradables y feos. La cosa es aún más lamentable cuando se tiene en cuenta que, con un pelín tan sólo de esfuerzo, una línea de órdenes no es algo tan odioso. Por lo que no existe razón admisible que justifique la tortura a que los PeCés han sometido a sus usuarios durante largos años.

CAPÍTULO 5. Software: Manejo de programas

Comencemos con la posibilidad de corregir fácilmente los errores. Para poder corregir¹² un texto, sea cual sea, sólo hace falta poder desplazar el cursor al punto en que se encuentre el error a corregir, poder borrar, y poder insertar caracteres¹³. Si, de algún modo, podemos pasarle al ordenador las órdenes necesarias para hacerlo, nuestra línea de comunicación va a mejorar mucho.

Es preciso que en este momento nos imaginemos que el teclado que tenemos para comunicarnos con el ordenador es de los de antes, de los primeros que se emplearon para hablar con ellos. Hace no tanto tiempo, no era fácil encontrar teclados diseñados específicamente para trabajar con ordenadores. Los teclados eran los de las máquinas de escribir eléctricas, y en ellos, como todo LAO sabe, únicamente podemos encontrar teclas alfabéticas, números, una tecla de retorno de carro, y la tecla de retroceso. También teclas de tabulación y de alzamiento y bloqueo de mayúsculas.

¿Porqué las IU de líneas de órdenes sólo permitían corregir borrando el último carácter escrito?. Simplemente porque existía una tecla para ello. La tecla de retroceso, cuando se pulsaba, le indicaba al programa encargado de manejar la línea de órdenes que borrara el último carácter y me dejara que pusiera otro. O que siguiera borrando con el anterior si volvía a pulsar retroceso. El teclado, poco adecuado, imponía claras restricciones a lo que el programa que implementaba la IU permitía hacer.

Ya que estamos con esto, ¿cómo le indicaba yo al ordenador que había terminado de escribir y era su turno de hacer lo que fuera, si es que lo entendía?. La tecla de retorno de carro servía maravillosamente. Cuando yo la pulsaba, el ordenador leía mi línea y pasaba a intentar digerirla y, caso de conseguirlo, llevarla a la práctica. Esto explica también porqué hablamos de “línea de órdenes”. Sólo tengo una línea. En el momento en que pulse el retorno de carro, la famosa tecla gorda de la derecha de las máquinas de escribir eléctricas, el ordenador pasará a la acción.

Vamos a ver como funcionaba la cosa con un poco de detalle¹⁴. El programa me presentaba el prompt en pantalla y me dejaba escribir, para lo que se encargaba de controlar el teclado, viendo qué tecla pulsaba y actuando en consecuencia. Si se pulsaba una tecla con significado alfanumérico (letras, signos de puntuación, números) la almacenaba en memoria como parte de la orden, aún no completa, que yo estaba escribiendo. Si se pulsaba “retroceso”, tecla no alfanumérica, significaba que el último carácter no era válido. Se borraba de la memoria. Si se pulsaba “retorno de carro”, tecla de nuevo no alfanumérica, significaba que la orden ya estaba terminada y entonces el programa intentaba descifrarla y ejecutarla. Si era una orden correcta, la hacía. Si no lo era, se sacaba un mensaje de error. Y en todos los casos el ordenador, al terminar, me volvía a mostrar el prompt y lo que hubiera escrito hasta entonces y dejaba que yo siguiera escribiendo. Y esto es todo, muy buenas tardes.

12 En inglés “edit” se llama a eso, y en bastantes ocasiones se traduce al castellano como “editar”.

13 No hace falta nada más, aunque es frecuente que podamos borrar a la derecha y a la izquierda del cursor, y que podamos sobrescribir un texto.

14 ¡Vamos a mirar dentro de un programa!. ¡Y de uno que nada más y nada menos implementa toda una IU que se usa con enorme frecuencia en la actualidad!. ¿A que es maravilloso?. ¿A que no lo hubiera creído posible cuando comenzó a leer el libro, querido LAO?.

¿Qué le llegaba en realidad al programa cuando yo pulsaba una tecla?. ¡Hombre, de nuevo el código ASCII!. Busque la hoja correspondiente, ponga un dedo para poderla consultar y... si yo pulso una “a”, el teclado manda un 97. El programa lo almacena en memoria como tal 97 y vuelve a sacar el prompt. Si pulso un retroceso, el teclado manda un 8 (Back Space), uno de nuestros caracteres de control, algo no alfanumérico, y mi programa se encarga de borrar el 97 anterior. Si pulso retorno de carro, el teclado manda un 13 (Carriage Return), y el programa sabe que he terminado con esta línea.

Ya dijimos que los caracteres de control estaban ahí para codificar acciones en un teletipo. El 13 y el 8 tienen significados concretos y útiles, pero de muchos de los demás 31 primeros caracteres ASCII no puede decirse lo mismo. Cuando comenzaron a implementarse líneas de órdenes con teclados y pantallas en lugar de con teletipos, se quedaron sin empleo. Para que se vea de qué hablo, el 7 es “BEL”, abreviatura del inglés “Bell”, que significa campana. Este simpático código era el encargado de hacer sonar la campanita del teletipo. “Plinc...”, el final de la línea está cerca.

A algún avisado o compasivo programador de IU de las de entonces le pasó por la cabeza lo siguiente: “Si fuera posible emplear los caracteres de control que ya no valen para nada para hacer otras cosas, podríamos por ejemplo hacer que se pudiera corregir la entrada de órdenes con mayor facilidad. Un carácter de control podría desplazar el cursor sin borrar, no como con la tecla de retroceso. Otro podría permitir borrar hacia la derecha, y otro indicar que se van a insertar caracteres. Al fin y al cabo con tres o cuatro de los 25 o más que no se usan tendríamos bastante”. Armado con tal genial idea, fue a ver al director de gama de productos de la empresa para la que trabajaba e intentó venderle la moto. “Hombre Javier, si lo hacemos, tendremos una enorme ventaja sobre la competencia. Nuestros programas serán más cómodos de usar, se venderán a toneladas y nos forraremos”. Javier, en cambio, no estaba por la labor de complicarse la vida, al fin y al cabo ya estaban ganando dinero a espuertas, y no se le ocurrió más que decir: “No, si como idea no está mal. Pero no has tenido en cuenta que el problema real son los teclados. Harían falta teclas adicionales y ningún fabricante de teclados se va a poner a fabricar versiones especiales para ordenador. Total, se venden 200 al año. Si encuentras un modo de resolver lo de los teclados, adelante. Pero te encargas tú”. Nuestro programador, inteligente además de avisado y compasivo, reconoció que esta vez el jefe, además de ninguna gana de trabajar, tenía razón, y se puso a pensar en ello. “Bien, tal vez no sea posible hacer teclados especiales completos. Pero las teclas de alzamiento de mayúsculas, al fin y al cabo, duplican la totalidad del teclado con una sola tecla adicional. Yo no tengo una tecla para la “a”, otra para la “A”, otra para la “n” y otra para la “N”. Cuando quiero mayúsculas, pulso alzamiento+lo que sea y ya está. ¿No se podría convencer a ningún fabricante para que añadiera solo una tecla más, una de alzamiento especial que me permitiera generar caracteres de control si los asociamos con una tecla alfanumérica y tendríamos ya el problema resuelto?”.

Así nacieron (es un suponer) los teclados para ordenador. Sólo con una tecla adicional, la tecla de control, que Vd. puede encontrar en su teclado bajo el rótulo de “Control” o “CTRL” probablemente, la cosa volvió a mejorar muchísimo. Esta tecla se encarga simplemente de generar un valor ASCII 64 unidades menor que el carácter alfabético que figura en la tecla junto a la que se pulsa. Si Vd. presiona la tecla “Control” y a la vez la tecla A¹⁵, no obtiene una “a” minúscula ni mayúscula, sino el carácter de control “SQH” o ASCII 1.

15 La tecla A, pulsada sola, genera una “a”. Con el alzamiento de mayúsculas genera una “A”. Con

CAPÍTULO 5. Software: Manejo de programas

Esto hizo posible llevar a la práctica la idea de nuestro avisado programador. Hubo, en algún momento, que pensar en como ponerlo en los manuales del equipo de la empresa, y dado que, aunque no lo parezca, lo de escribir “CTRL+A” resultaba bastante farragoso, se optó por abreviarlo con un acento circunflejo. En la documentación relativa al “Uso de la línea de ordenes mejorada” las cosas aparecían pues como “^A”, por ejemplo. Esto explica algunas de las abreviaturas que aparecen en los caracteres de control de nuestro mapa ASCII¹⁶.

Al disponer de pulsaciones adicionales fue posible emplearlas para permitir la corrección de errores de mecanografiado. No sólo se pensó en permitir desplazar el cursor hacia atrás paso a paso sino que, como sobraban caracteres sin emplear, hasta fue posible desplazarlo más rápidamente, palabra a palabra por ejemplo. Y hacia delante y hacia atrás, incluso. Si queremos ver en concreto qué hacían las nuevas teclas, un buen ejemplo podría ser el siguiente listado¹⁷:

Insertar -Sustituir

Ctrl-V Alterna entre los modos de inserción y sustitución

Borrar

Ctrl-H Borra el carácter anterior al cursor

Ctrl-G Borra el carácter en la posición del cursor

Ctrl-T Borra la “palabra” en la posición del cursor

Desplazamiento del cursor

Ctrl-S Carácter anterior

Ctrl-D Carácter siguiente

Ctrl-A Palabra anterior

Ctrl-F Palabra siguiente

Para que se vea que, a pesar de los rudimentarios medios de que disponían, los antiguos hacían en ocasiones las cosas medio bien, mire un teclado y fíjese en las teclas A, S, D, F. Están juntas y en el orden citado. La S desplaza el cursor a la izquierda, hacia su lado respecto al centro del grupo. La D hacia la derecha. Y la A y la F lo mismo pero más lejos, o sea, palabra a palabra. ¿Se ve la lógica de las asignaciones?.

Bien, hemos dejado a nuestro avisado programador con el asunto de la corrección resuelto. Pero, como hemos dicho, era también inteligente, y pensó: “Hemos resuelto un problema pero ha sido sólo para empeorar otro. Ahora la gente no sólo se tiene que aprender las órdenes sino que, además, debe memorizar unas pulsaciones bastante absurdas. No sirve de nada que ^V pase a modo de inserción si la gente no lo sabe o no se acuerda de cual era

la tecla de “Control” genera el ASCII 1.

- 16 Al respecto de nuestro mapa ASCII, el porqué aparecen unos curiosos símbolos gráficos al lado del número, la abreviatura en siglas, y la pulsación correspondiente para generarlos, es una cuestión interesante y sobre la que se pueden decir muchas cosas. Cosas que, piadosamente, ignoraremos.
- 17 Listado que no se ha inventado ni pertenece a algún antediluviano programa ya no disponible. Es totalmente actual, e incluso en mi opinión de un muy buen programa, como veremos.

la tecla. Podríamos hacer que el ordenador mostrara en pantalla las instrucciones necesarias para su operación, guiando al usuario y evitándole la necesidad de memorizar estupideces. Total con un poco de memoria, algunos Kilobytes para tres o cuatro pantallas de texto, ya sería bastante. Voy a pedirle permiso a Javier”. Y dicho y hecho, fue a hablar de nuevo con su jefe. Este se olió la tostada. “¿Aún más mejoras?. Bueno, has hecho un buen trabajo con lo de los teclados y las ventas van subiendo. Haz lo que te dé la gana siempre que no tengamos que aumentar los costes de producción y déjame en paz. Tienes un mes de tiempo y después me lo cuentas”. Nuestro programador no esperaba otra cosa y se fue, contento como unas castañuelas.

Uno de los caracteres de control que aún no se usaban se asignó simplemente a la acción de mostrar en pantalla una serie de textos de ayuda que explicaran lo que había que hacer. Por si a algún burro se le olvidaba cual era la pulsación de teclas que tenía que usar para salir de apuros, decidió incluirla en el prompt para que la viera a todas horas. El usuario se enfrentaba así a un ordenador que le decía:

```
DIME QUE HAGO AHORA (^J ayuda):_
```

y problema resuelto. Si se pulsaba CTRL+J, el usuario podía leer una explicación de los caracteres de control disponibles, justo el listado de antes.

Contento, pero aún no satisfecho, volvió de nuevo a pensar. Le quedaban tres semanas de las cuatro que le había dado su jefe. “Bien”, pensó, “no hemos complicado la cosa. Pero no hemos resuelto lo de la memorización de las órdenes y los parámetros. Podríamos emplear las pantallas de ayuda para dar acceso a una especie de manual que permitiera al usuario saber qué órdenes tiene a su alcance y qué hace cada una. Metiendo el tal manual en un fichero del disco y haciendo que el ordenador lo liste de forma controlada cuando se pulse otro carácter de control libre, problema resuelto. Si alguien no sabe qué hace una orden, o ni siquiera conoce qué órdenes tiene a su disposición, basta que lo lea. Y para que la gente sepa como llegar, bastará con añadir una línea a la ayuda que aparece con CTRL+J diciendo qué combinación de teclas hay que usar para verlo”. Lo hizo, y henos aquí con las pantallas de ayuda convertidas en un manual en línea (ver glosario).

Esto resolvía muchos problemas. Se le ocurrió otra idea genial con los parámetros. Pensó que muchas veces la gente no quería recorrer todo un manual, sino que sabía qué orden quería usar pero no se acordaba de todos los parámetros disponibles. La solución podía ser un parámetro especial, por ejemplo un “/H” (de “Help”, ayuda en inglés) o “/?”, que simplemente sacara en pantalla un listado de aquellos disponibles para la orden en cuestión. Si el usuario tecleaba:

```
DIME QUE HAGO AHORA (^J ayuda): DIR /?_
```

el ordenador respondía mostrando algo así como:

```
Propósito:  Muestra información sobre ficheros.
Formato:    DIR [/1 /2 /4 /A[:][:]-lrsda] /B /C[HP] /D /E
            /F /H /I"text" /J /K /L /M /O[:][:]-lacsdegrinsu]
            /P /R /S /T /U /V /W] [fichero...]
fichero:    El fichero a mostrar.
            /1 (una columna)           /K (sin cabecera)
```

CAPÍTULO 5. Software: Manejo de programas

/2 (dos columnas)	/L (minúsculas)
/4 (cuatro columnas)	/M (sin pie)
/A (usa atributos)	/N (normal)
/B (simplificado)	/O (ordenado)
/C[HP] (Comprimido)	/P (ausa)
/D (sin colores)	/R (sin retorno)
/E (mayúsculas)	/S (subdirectorios)
/F (camino completo)	/T (aTributo)
/H (oculta puntos)	/U (sUmario)
/I (descripciones)	/V (ordenación Vertical)
/J (ustifica nombres)	/W (ancho)

y con esto, tema resuelto. Contento, y ahora sí, satisfecho, dio el asunto por terminado y se fue de vacaciones las dos semanas que le quedaban.

Con este cuentecito espero que hayamos conocido algunos otros inventos capaces de hacernos la vida mucho más fácil si debemos enfrentarnos a una IU de línea de órdenes. La posibilidad de corregir con comodidad y las pantallas y parámetros de ayuda, aunque haya que usar caracteres de control como método de acceso a ellos, es lo que hace una línea de órdenes digerible. Queda una única cosa que nuestro programador olvidó. La posibilidad de reutilizar cosas escritas anteriormente. Pero esto nos llevará a hablar de la siguiente gran IU, las “pantallas de órdenes” o “editores de pantalla completa¹⁸”.

Por ahora, y antes de pasar al siguiente apartado, un rápido resumen. Dos cosas. Que hasta una línea de órdenes resulta aceptable si se implementa correctamente. Y que lo que sabe un usuario experto es casi lo mismo que uno novel. Estrictamente casi nada, aunque un casi nada muy distinto. Sabe que tiene a su disposición textos de ayuda, manuales en línea, que probablemente tenga mensajes de interés en la pantalla, que debe pasarle las órdenes al ordenador como una serie de palabras con parámetros, y que puede corregir de algún modo. Sabe qué quiere hacer. Equipado con tal conocimiento se sienta ante el ordenador sin temer sorpresas. Ya mirará sus dudas en los textos de ayuda. No se va a molestar en memorizar. Con saber como sacarlos, y el ordenador se lo muestra constantemente, ya vale. Quizá, con el uso habitual del equipo durante dos o tres meses acabe acordándose del formato completo de las órdenes más habituales, pero qué importa.

Sólo una última nota. La utilidad de los textos de ayuda es nula si no se entiende lo que dicen. Tal vez una lectura atenta de la pantalla que describe las opciones de DIR le permita darse cuenta cabal de qué hablamos. Hace falta comprender como va un ordenador, entender una terminología, para sacarles provecho. Es eso lo que estamos intentando hacer, entre otras cosas, con este texto.

18 Por cierto, las líneas de órdenes, o mejor dicho las “interfases de línea”, también se denominan en ocasiones “editores de línea”.

Editores de pantalla completa

Luis Ricardo nos da más espacio

Una línea está bien, si está bien montada. Pero nadie pondrá en duda que en cuanto a comodidad, la gente suele opinar que cuanto más, mejor. Tenemos una pantalla de 25 líneas. ¿Porqué no dejar, ahora que hemos conseguido caracteres de control de sobra, que el usuario se desplace libremente por toda la pantalla?. Al fin y al cabo, podríamos ampliar las pulsaciones asignadas a acciones y permitir que el cursor pase a la línea anterior, o a la siguiente, que se borren líneas completas, que se inserten líneas vacías...

Podríamos mejorar la IU que estamos construyendo con algunas pulsaciones más, como las siguientes por ejemplo, y permitiendo que el usuario se desplazara libremente por la pantalla.

Insertar - Sustituir

Ctrl-V Alterna entre los modos de inserción y sustitución

Ctrl-N Introduce una línea nueva en la posición del cursor

Borrar

Ctrl-H Borra el carácter anterior al cursor

Ctrl-G Borra el carácter en la posición del cursor

Ctrl-T Borra la "palabra" en la posición del cursor

Ctrl-Y Borra la línea en la posición del cursor

Desplazamiento del cursor

Ctrl-S Carácter anterior Ctrl-D Carácter siguiente

Ctrl-A Palabra anterior Ctrl-F Palabra siguiente

Ctrl-E Línea anterior Ctrl-X Línea siguiente

Ctrl-R Página anterior Ctrl-C Página siguiente

Ctrl-W Desplaza la pantalla hacia abajo

Ctrl-Z Desplaza la pantalla hacia arriba

No hay necesidad de memorizar nada, pues seguimos teniendo los mecanismos de ayuda de antes, y esto añade la tan ansiada posibilidad de recuperar ordenes.

La cuestión es que normalmente le decimos a Luis Ricardo más o menos siempre lo mismo. Tecleamos DIR unas 200 veces. Tanto, que mucha gente se pregunta si no sería posible pasarle al chisme el mismo DIR de hace 10 minutos. Los editores de pantalla completa son una solución a este problema. En ellos puedo desplazarme con mi cursor no sólo en horizontal, sino en vertical, y reutilizar cualquier cosa que haya escrito antes. Llevo el cursor hacia arriba, me pongo en la línea en que se encuentre la orden vieja que me interesa, le doy a retorno de carro, el ordenador lee la línea en que está el cursor, lo hace, y aquí paz y después gloria. El problema con este enfoque es que una pantalla de 25 líneas es más bien muy poco. Para que la cosa funcione hace falta que el ordenador considere no que tiene 25 líneas, sino varios cientos de ellas. La pantalla es entonces una ventana que puedo desplazar a voluntad hacia arriba o hacia abajo ¹⁹ a través de la que miro mi texto, mis órdenes, lo que el ordenador dijo hace 10 minutos, o lo que sea. CTRL-W y CTRL-Z del listado anterior se ocupan de ello.

CAPÍTULO 5. Software: Manejo de programas

Desgraciadamente, los editores de pantalla completa requieren más recursos que las líneas de órdenes o editores de línea. Hace falta más memoria del ordenador para guardar mi pantalla virtual²⁰ de 5.000 líneas. Esto ha hecho que, dado que no aportan más potencia que la modesta línea de órdenes por cuanto seguimos comunicándonos con el ordenador a través del mismo lenguaje y los mismos parámetros, no hayan sido muy frecuentes a nivel de manejo del sistema operativo del ordenador²¹.

Hace falta conocerlas, sin embargo, pues son lo habitual en programas como los procesadores de texto. En ellos no pasamos órdenes, sino que simplemente almacenamos lo que estamos escribiendo. Muchas veces es necesario reutilizar un trozo ya escrito, o queremos poder ver lo que tenemos 200 líneas antes. Las IU de pantalla completa lo hacen posible fácilmente.

La reutilización en IU de línea de órdenes se resuelve habitualmente a base de hacer que la misma pulsación que me llevaría a la línea anterior, muestre en la línea en la que me encuentro la última orden dada. Sigo pegado a una única línea pero tengo a mi alcance todas las órdenes anteriores. En algunos casos incluso puedo listarlas separadamente para verlas y seleccionar una. Este enfoque mixto es francamente práctico, y debería ser el mínimo, junto con todo lo anterior, de una buena línea de órdenes. Que las hay, y felizmente en cantidad.

No todo son caracteres de control

Los otros recursos de Luis Ricardo

Hemos visto ya de entrada lo más difícil de las IU. Caracteres de control otra vez, IU muy generales como los editores de línea y de pantalla completa. Era necesario. Nos las encontraremos miles de veces disfrazadas (o no) bajo los más variopintos ropajes. Y son la auténtica base con la que se manejan los programas de ordenador a todos los niveles. Si las hemos entendido y no nos asuntan ya, nada en un chisme de estos será capaz de espantarnos. Habremos vencido a la bestia. Pero no deja de ser cierto que, como único recurso de Luis

19 A este desplazamiento del contenido de la pantalla en dirección vertical se le denomina “scroll”. No se traduce. Denomínase “pan” o “panning”, y tampoco se traduce, al desplazamiento del contenido de la pantalla en dirección horizontal. Conviene saberlo para poder entender las ayudas...

20 Se denomina virtual a cualquier invento que el ordenador considere y pretenda hacerme creer que tiene aunque no sea cierto. En este caso la pantalla de 5.000 líneas. Yo puedo trabajar como si fuera real, pero es un artificio. No veo las 5.000 líneas de vez, pero puedo moverme por ellas con mi ventana de 25 líneas. Las 5.000 líneas existen en memoria, como una estructura de datos que el ordenador visualiza a razón de 25 líneas por vez de acuerdo con mis instrucciones. La cosa funciona, y sistemas similares se emplean para un montón de cosas.

21 En este momento, no puedo sino incluir una mención de honor para mi primer ordenador. El New Brain tenía una interfase de usuario a nivel sistema operativo consistente en un editor de pantalla completa. Esto con 32 K de RAM y 32K de ROM. Nunca entendí porqué el resto de ordenadores no tenían cosas similares, y la racanería de los PeCés en estos temas cuando contaban con más memoria, potencia, y capacidad de almacenamiento, me resultó francamente indigesta. Esta es una de las razones que hizo que me resistiera a usarlos mientras fue posible.

No todo son caracteres de control

Ricardo en su comunicación con los humanos, estas IU tan generales adolecen de un serio defecto. Uno que no hemos nombrado explícitamente hasta el momento.

Los editores de línea y de pantalla completa²² nos dan la posibilidad de hacer cualquier cosa. Todo lo que el chisme es capaz de hacer. Pero nos muestran para empezar una pantalla en blanco. Sé que puedo hacerlo todo, pero no sé qué. Consecuencia, el “¿Y ahora qué?”, me devora. Este problema es lo que podríamos denominar el “síndrome de la hoja en blanco” y tiene efectos devastadores²³.

Cuando se trata de poner todos los recursos del ordenador a disposición del usuario, no queda más remedio que afrontarlo; pero si se trata sólo de conseguir que el ordenador haga una cosa en particular con un número razonablemente reducido de opciones, como es el caso de programas concretos para cosas concretas (y vuelta con contabilidades, proceso de textos y demás etcéteras), es perfectamente posible evitarlo, y así debe hacerse a base de recurrir a IU menos potentes pero más acogedoras.

Estas IU son los menús y las “botoneras”²⁴. Vamos a empezar con los menús, que son una cosa muy familiar. Si Vd. va al restaurante de comida rápida de la esquina tal vez se haya visto enfrentado a algo así como una carta en la que aparece un texto someramente parecido a éste:

- 1.- Tortilla española
- 2.- Pizza
- 3.- Hamburguesa

Tras un atento examen, la consideración de cual de las opciones resulta más de su gusto, ofrece más garantías de buen estado, y es más difícil elaborar con sobras tal vez de varias semanas de antigüedad, Vd. decide y le grita valientemente al camarero el número de su elección: “Un uno”. Eso es todo. Inspirados en este maravilloso mecanismo, los programadores decidieron emplearlo para dejar que el usuario de un programa eligiera lo que deseaba que hiciera el ordenador. Por ejemplo, nuestro desdichado usuario del principio del capítulo podría haber tecleado trabajosamente “TLW” en su línea de órdenes y golpeado la tecla gorda de la derecha, para acabar frente a algo así como²⁵:

-
- 22 Esta denominación es más general que lo de “línea o pantalla de órdenes”, y refleja el hecho de que no necesariamente vamos a encontrar estas IU para pasarle órdenes al equipo, sino que también pueden servir, y las encontraremos en esos menesteres, para otras cosas. La utilizaremos a partir de ahora siempre que no nos refiramos específicamente a una petición de órdenes.
- 23 El nombre viene de que parece ser típico de escritores enfrentados a una idea no muy clara de lo que quieren hacer y como hacerlo. La visión de la hoja en blanco esperando en la máquina de escribir, tiene en este caso un efecto paralizador. Supongo que todo LAO habrá notado ciertas similitudes globales entre éste síndrome y muchos de los problemas que se han citado, en los que todo podía ser pero nada era.
- 24 Denominación que acabo de sacarme de la manga pero que explicaremos en breve. Permanezca atento.
- 25 Esto es sólo un ejemplo hipotético. TLW es un programa de alta tecnología, muy evolucionado, sumamente sofisticado, y ha abandonado esta precaria IU hace tiempo. Lo iremos modificando

CAPÍTULO 5. Software: Manejo de programas

- 1.-Ver qué textos tengo creados
- 2.-Crear un texto nuevo
- 3.-No hacer nada. Fin y a casa

Elige número y pulsa Retorno de Carro: _

Pulsar el 1, el 2, o el 3, seguido de la tecla gorda de la derecha²⁶, le hubiera permitido elegir la opción de su gusto. Esta es la versión más rudimentaria de una IU basada en menús. Todo lo que puede decirse es que funciona. Evita el síndrome, da acceso a las opciones necesarias, y punto.

Desearía, y espero que no sea mucho pedir, que el LAO se diera cuenta de qué es lo que está sucediendo en realidad con una IU de este tipo. Vamos a entrar dentro de esta IU igual que anteriormente entramos dentro de una línea de órdenes. El programador, el que ha hecho el programa, ha decidido que esas eran las opciones que me interesaban y sólo esas. Ha almacenado en el interior del programa las órdenes necesarias para hacerlas. Por ejemplo, la opción 1, si la ejecutáramos desde la línea de órdenes podría ser algo así como: “DIR *.TXT”, que a estas alturas debería resultar vagamente familiar. Cuando yo pulso el 1 y RC (indicaremos así a partir de ahora la pulsación de la tecla gorda de la derecha), el programa le pasa al ordenador el dichoso “DIR *.TXT”, y ya está. Algo similar para las demás.

Lo que está haciendo mi programa al usar esta interfase no es más que otro ejercicio de “ocultamiento de la información” basado en el transformismo de Luis Ricardo. Hace que yo crea que lo que puedo hacer es 1, 2, o 3, me muestra esa apariencia y, en función de lo que yo elija, traduce por su cuenta la orden al lenguaje que el ordenador entiende, el que yo debería usar si empleara la línea de órdenes. No está mal, ya que me evita de nuevo tener que recordar las órdenes e incluso saber que semejante lenguaje existe²⁷. Cualquiera, casi, puede usar con comodidad el invento. Otra gran ventaja, que comparte con todas las IU que veremos en lo sucesivo, es que hace posible usar el ordenador sin teclear casi nada más de lo estrictamente necesario, es decir, lo justo para introducir mis datos.

Hemos visto ventajas. Vayamos con los inconvenientes. En primer lugar, ya se ha dicho que una IU de este tipo es menos potente, por cuanto sólo me deja hacer las cuatro cosas que el programador ha decidido por mí que yo debería querer hacer. ¿Y si quiero hacer algo distinto?. Mejor sales del programa y usas la IU de línea de órdenes. En segundo lugar, al menos en esta versión rudimentaria, los menús tienden a ser más bien farragosos. Ocupan mucho sitio en pantalla con esta presentación de restaurante. Y, además, recuerde que teníamos una cosa que se llamaban parámetros. Es posible que si yo elijo la opción 1, la de que me enseñe un directorio, desee que me lo muestre de una forma concreta. Por ejemplo

a lo largo de los siguientes ejemplos. Se ruega paciencia.

- 26 La tecla gorda de la derecha, nuestro retorno de carro de las máquinas de escribir eléctricas y los caracteres de control, suele recibir muy diversos nombres. También se denomina “Enter”. Otras veces aparece rotulada con un símbolo gráfico. Otras veces no tiene ningún tipo de rotulación.
- 27 En mi opinión, este último efecto ya no es tan beneficioso. Como con el bit, el ASCII, y otro millar de cosas, creo que es mejor saber que existen, que están ahí, y como funcionan, aunque no las manejemos.

No todo son caracteres de control

a lo ancho. Para poder pulir las opciones, los menús suelen estar encadenados, de modo que uno me lleva a otro en el que se termina de completar la orden. 1 y RC me llevaría probablemente a algo como:

- 1.- Ver directorio a lo ancho
- 2.- Ver directorio página a página
- 3.- Volver al menú anterior

Elige número y pulsa Retorno de Carro: _

Seleccionar ahora la opción 1 pasaría al ordenador sin yo saberlo un “DIR *.TXT /W”, en tanto la opción 2 implicaría un “DIR *.TXT /P”. Pero dése cuenta de que yo tenía muchísimas más opciones con mi línea de órdenes.

Un menú, en resumen, está bien para un número reducido de opciones (se dice que mejor no poner más de siete por menú al diseñar un programa si no se quiere atosigar al usuario) pero ocupa mucho sitio en pantalla y acaba siendo pesado de usar al tener que ir recorriendo menú tras menú para acabar haciendo algo en concreto. Uno corre el peligro de recordar que una vez hizo algo con ellos sin poder acordarse de en qué menú estaba. El recorrido por las opciones tiende a desorientar si no están muy bien agrupadas, por lo que un diseño correcto de una IU basada en ellos exige cierta reflexión.

De todos modos, los menús son útiles, y se emplean habitualmente. No es frecuente que nos encontremos con el tipo del ejemplo, de todos modos. Algunos de sus defectos se corrigen fácilmente y llevan a versiones más sofisticadas que son las normales hoy día. En primer lugar lo de que ocupan mucho sitio en pantalla. Es posible ponerlos en una sola línea si reducimos la cantidad de texto que explica cada opción. Los dos menús anteriores se convertirían en:

- 1.-Ver textos 2.-Crear un texto nuevo 3.-Fin y a casa

y

- 1.- A lo ancho 2.-Página a página 3.- Menú anterior

No resultan tan explícitos, pero para un usuario que sepa qué está haciendo son suficiente. También el tener que pulsar el número y RC se convierte rápidamente en una pesada carga. Si tengo que recorrer 5 menús para llegar a lo que quiero hacer, son 5 veces que tengo que pulsar RC. Diez pulsaciones en lugar de cinco. ¿No habría modo de simplificar la cosa?. Ningún problema. En cuanto pulso el 1, el 2, o el 3, el programa pasa la orden, sin esperar el RC. Pierdo la posibilidad de corregir una selección equivocada, pero como siempre voy a poder volver al menú anterior, no hay mayor inconveniente.

Y esto nos lleva directamente al tema de las botoneras. Las IU basadas en ellas intentan hacer el uso del ordenador lo más rápido posible. El hecho es que nada hay más rápido que pulsar un botón para hacer algo. Yo puedo diseñar un programa muy ágil a base de dejar que todas las opciones sean ejecutables pulsando un botón. ¿Qué botón?. El programador decide. Podemos emplear caracteres de control libres. CTRL+lo que sea me permitirá hacer un directorio. Necesitaré memorizar, pero si incluyo ayudas la cosa resultará soportable. O puedo hacer que las teclas alfanuméricas hagan cosas si estoy seleccionando una opción de un menú. En este caso no las empleo, ya volverán a funcionar normalmente cuando termine

CAPÍTULO 5. Software: Manejo de programas

de seleccionar opciones. El programa hace que una misma tecla tenga varias funciones dependiendo del modo en que esté el programa. Si estoy eligiendo opciones, las teclas hacen cosas. Si estoy escribiendo, escriben. Tiende a resultar un poco lioso (en tiempos se hablaba de que los programas debían evitar en lo posible tener distintos modos de funcionamiento para resultar cómodos), pero funciona.

Una forma de implementar las botoneras y los menús se basa en su uso simultáneo, con lo que se evitan algunos de los problemas de unos y otros. Es lo que hemos dicho antes. Muestro un menú con una tecla de acceso y en cuanto la pulso, la opción se ejecuta. El texto del menú muestra la tecla a usar, que ahora no necesito memorizar, y el acceso sigue siendo rápido. Incluso puedo simplificar más los menús de antes, y dejarlos en:

Ver textos Crear un texto nuevo Fin y a casa.

y

A lo ancho Página a página Menú anterior

Ahora indico la tecla activa poniéndola en mayúscula y en cuanto la pulso, ejecuto la opción. V me llevaría al segundo menú y P me mostraría el menú página a página. Si dos opciones coinciden en la inicial, siempre puedo marcar otra. Por ejemplo, supongamos que la tercera opción de mis menús fuera siempre “Anterior”, no “Menú anterior”. El programador siempre podría presentar el segundo menú como:

a lo ancho Página a página Anterior

y la pulsación de V, N, me mostraría el menú a lo ancho, mientras que V, A, me devolvería al primer menú.

Quien dice mayúsculas puede querer decir caracteres en video inverso, subrayados, o como sea, simplemente algo que indique cual es la pulsación a realizar para seleccionar una opción.

Otra versión de los menús, ya no basada en botoneras, recurre a una selección de opciones mediante la posición del cursor. Siempre que se pueda mover, por ejemplo mediante caracteres de control que es lo único que hemos visto por ahora, es posible hacer que el cursor se desplace por las opciones, marcando una. Yo me coloco sobre lo que quiero, normalmente esto se indica con un cambio en la apariencia de la opción por ejemplo subrayándola (o en video inverso), y la ejecuto dándole a RC. Esto es bastante elegante, aunque no tan rápido como las botoneras. Ya no hay que indicar teclas activas y todos los menús funcionan a base de las mismas pulsaciones. Desplazamiento y RC. Por ejemplo, si deseara seleccionar la segunda opción del segundo menú, me colocaría sobre ella:

A lo ancho Página a página Anterior

y pulsaría RC.

Hasta el momento, hemos visto IU que se pueden implementar en un ordenador que trabaje en modo texto, con una pantalla más bien cochambrosa (sin gráficos y quizá de pocas líneas) y con teclados estrictamente de máquina de escribir eléctrica. Todavía el paleolítico²⁸.

28 Paleolítico al parecer no terminado, por cuanto es bastante probable que, si Vd. se fija, encuentre

Y aparecieron los teclados para ordenador

Vamos a suponer que todas ellas fueron un invento de nuestro avisado programador de antes y que éste, aún traumatizado por sus dificultades iniciales con los teclados de máquina de escribir, encontró con ellas el medio de aumentar las ventas de los ordenadores de forma vertiginosa. El número de equipos vendidos ascendió desde las profundas simas en que se encontraba a los cientos de miles de unidades al año. Con esta realidad a sus espaldas, nuestro programador, hace tiempo ascendido a jefe de línea de producto de su compañía (el anterior, Javier, pasó a director ejecutivo) decidió de nuevo encaminar sus pasos hacia la empresa que le había fabricado los primeros teclados con tecla de control. Albergaba aviesas intenciones.

Y aparecieron los teclados para ordenador

Luis Ricardo se viste a medida

¡Ya está bien de teclados modificados!”, espetó al diseñador de teclados, “Hay un mercado de cientos de miles de unidades que justifica perfectamente un diseño a la medida. Si no lo hacéis vosotros, me iré a la competencia.”. Sus deseos fueron, evidentemente, órdenes.

No sólo se empleaban los teclados para escribir, aunque también. “Mantengamos el teclado alfanumérico tal cual está, ya que a la gente le resulta familiar”. Pero había que desplazar el cursor, tanto para corregir como para seleccionar opciones de menú. Hacían falta teclas dedicadas sólo a eso. “Hay que añadir un bloque de teclas para poder mover cómodamente el cursor”. Muchas veces se usaba el ordenador como un instrumento de cálculo, por lo que se pasaba mucho tiempo metiendo números. Las teclas numéricas de la línea de arriba del teclado de máquina de escribir no tenían la mejor distribución para hacerlo. Mejor un teclado como los de una calculadora. “Vamos a conservar las teclas numéricas del bloque principal pero vamos a añadir un bloque numérico como el de una calculadora”. Y el toque maestro de nuestro avisado programador. “Muchas veces se emplean teclas para hacer cosas, como con las pulsaciones de caracteres de control. Sería mejor poner teclas vacías adicionales que cada programador pudiera emplear como gustara. Y ya que estamos, vamos a añadir más teclas de alzamiento, sólo por si acaso. Así nos aseguraremos que siempre vamos a tener botones de sobra.”.

Dicho y hecho. Los teclados para ordenador habían nacido. En ellos encontramos un bloque alfanumérico como de máquina de escribir, un bloque de desplazamiento del cursor y de edición, un bloque numérico, algunas teclas de alzamiento adicionales (por supuesto CTRL y alguna más incluso), y un bloque de teclas vacías en variado número y que reciben el nombre de teclas de función.

Como el objeto de casi todas ellas ya se ha explicado, y simplemente con los nuevos teclados se evitó la confusión de tener que pulsar CTRL+S para desplazar el cursor a la izquierda una posición (ahora había una tecla con una flecha a la izquierda que hacía sólo eso²⁹), nos

múltiples ejemplos de cosas similares en buena parte de los ordenadores de hoy día.

29 No quiere decirse que con los nuevos teclados desaparecieran de la noche a la mañana los programas que usaban teclas de control. En informática se suele funcionar a base de añadir cosas, no de cambiar unas cosas por otras. Por tanto, en muchos programas se podían usar tanto los famosos CTRL+ lo que sea o las teclas de cursor. Ambas hacían lo mismo. Pero como lo que se veía y resultaba más cómodo era usar las teclas de cursor, lo de los caracteres de control, su empleo y su significado, quedó sólo al alcance de unos cuantos dinosaurios que conocían la

CAPÍTULO 5. Software: Manejo de programas

vamos a centrar en las teclas de función que son tan sólo un caso particular de las botoneras ya presentadas.

El motivo para crearlas era el caos que se había producido en el empleo del teclado. Según el programa, pues cada uno elegía el método que le parecía más conveniente, uno podía tener que pulsar una tecla alfabética normal, o una tecla asociada a la tecla CTRL, o algún otro tipo de extravagancia para hacer algo, para ejecutar una función. CTRL+K podía mostrar un directorio en un programa, mientras en otro programa equivalente era necesario pulsar V, N, J, sucesivamente, para conseguir el mismo efecto. Además de que en ese segundo programa CTRL+K no servía para nada.

Si todos los programadores estaban tan encantados con la ejecución de funciones mediante botoneras, no parecía mala idea dejarles unas cuantas teclas vacías, en principio sin cometido alguno, para que ellos las usaran como quisieran. Con esto al menos dejarían en paz el resto del teclado. Los programadores recibieron la cosa con alborozo e hicieron generoso uso del nuevo invento. Pero no solucionó nada. Fue un recurso más, que como siempre se añadió, pero no sustituyó, a los ya existentes. La confusión continuó, ahora aumentada con un elemento más. En un tercer programa no se hacía el directorio con CTRL+K (hacía otra cosa), ni con V, N, J, (servía a su vez para algo absolutamente diferente). Se hacía, lo han adivinado, pulsando “F1”, la primera de las teclas de función.

Las teclas de función, además, hacen que un problema ya existente antes de su invención se manifieste mucho más crudamente. Consiste en lo siguiente. Pongamos que tengo 8 teclas de función en mi teclado de ordenador diseñado a la medida. Un número francamente adecuado para un menú. Podemos hacer que sus opciones sean seleccionables mediante una tecla de función, lo que parece correcto. F1 me mostrará el espero que ya famoso directorio. Esto me lleva al segundo menú, y allí asigno de nuevo las teclas de función. Ahora F1 me saca el directorio a lo ancho. Ante mis ojos, F1 cambia su significado con total desfachatez. Y esto es sólo el principio. Si recorro los menús durante un rato, F1 va a cambiar constantemente, lo que acaba siendo inquietante. Tengo una tecla sin cometido preciso y que cambia de propósito cada vez. ¿Una?. ¡Varias!. Uy, uy, uy...

En realidad esto es lo mismo que pasaba al seleccionar una opción por su número. Pero algún mecanismo psicológico hace que la cosa resulte inquietante con las teclas de función y no con los menús. Tampoco era problema con los caracteres de control o con las teclas alfabéticas cuando se empleaban para seleccionar opciones, porque como no se veían...

El problema está en el “etiquetado” de las teclas de función. Si les asignamos un significado fijo en todos los programas, perdemos su versatilidad, y estarían mejor con un nombre encima que indicara qué es lo que hacen siempre. Y si no, si efectivamente se usan para lo que están, para cambiar de significado según necesidades del guión, hay que montar algún sistema simple y claro que al menos nos ayude a no perdernos.

La IU que emplea de forma más adecuada las teclas de función que yo conozco es el que podíamos denominar “interfase Hewlett-Packard³⁰”, que podemos denominar así por cuan-

historia. Y ahora también a su alcance querido LAO. O al menos eso espero.

30 Hewlett-Packard es una compañía multinacional estadounidense dedicada a la fabricación de

Y aparecieron los teclados para ordenador

to se empleó sistemáticamente (y aún se emplea, aunque ya casi tan sólo en sus calculadoras) en equipos de la marca. Consiste en que las teclas de función se agrupan en el teclado de forma que queden lo más cercanas posible a la pantalla. Por ejemplo en la fila superior del mismo. Los teclados de ordenador de Hewlett-Packard (HP en lo sucesivo) contaban con 8 teclas de función con esta disposición, e incluso he llegado a ver ordenadores de la marca que las incorporaban en la parte baja de la propia caja de la pantalla. Bien, a partir de aquí, con que reservemos por ejemplo las dos últimas líneas de la pantalla para indicar en todo momento qué hacen las teclas de función, y las usemos para acceder a todo menú que le mostremos al usuario, casi vale. No hay más que hacer corresponder la posición de las teclas de función y las de sus etiquetas en pantalla. Cada tecla acaba teniendo una explicación de su función en cualquier momento más o menos encima. Hasta podemos poner las etiquetas en video inverso, para que den la apariencia de teclas aisladas del resto de la pantalla. El esquema es simple y funciona. No quita mucho sitio de trabajo, total un par de líneas, y uno sabe siempre dónde tiene lo que hace cada una de las dichas teclas de función y que todos los programas de la marca marchan (marchaban) igual. Quizá su único defecto, aparte de los propios de una interfase de menús con acceso por botoneras, es que no permite grandes explicaciones en las etiquetas. No hay mucho sitio para escribir. Pero menos hay en una tecla de las de verdad y se emplean para lo mismo, con lo que quizá le estemos sacando pelos a una calavera.

Las teclas de función son en la actualidad moneda corriente. A nada que el programador se tome la molestia de rotularlas, o si queremos decirlo de otro modo, de arbitrar algún mecanismo de ayuda en línea que contribuya a su fácil identificación, van muy bien como método de acceso sistemático a las opciones de los menús y no requieren más que un teclado equipado con ellas. Veremos incluso que, aunque no tengan un sitio preciso en los nuevos diseños de IU, han sido sustituidas en ellos por un invento francamente similar.

En ocasiones, se prefiere usarlas como algo independiente de los menús. Se les asigna un significado fijo dentro de un programa (variable de un programa a otro en principio), y se emplean como un método de acceso rápido a las funciones más frecuentes. Como una botonera estricta, podríamos decir, totalmente separada de los menús que puedan emplearse dentro del mismo programa, a cuyas opciones se accede con cualquiera de los métodos de selección habituales.

En ocasiones, incluso, y el PeCé como veremos es un ejemplo, algunas teclas de función adquieren un significado único, constante e independiente del programa que se esté usando. Sigo creyendo que, en estos casos, estarían mejor con un rótulo que indicara claramente su cometido escrito indeleblemente sobre ellas, y que dejaran de considerarse teclas de función.

equipos de medida para electrónica, aparatos de análisis químico, ordenadores, calculadoras, y algunos otros etcéteras. Siempre bastante conocida en el mundillo científico, en la actualidad es bastante más popular de lo que solía por cuanto domina el mercado de impresoras para ordenadores personales y tiene una presencia creciente en el de PeCés.

CAPÍTULO 5. Software: Manejo de programas

Ventanas y ratones

Luis Ricardo y sus mascotas

Nuestro avisado programador, ya jefe de línea de producto de su compañía, estaba razonablemente contento. El negocio iba bien. Pero su felicidad no duró mucho. Su anterior puesto había sido ocupado por otro programador más joven que un buen día solicitó entrevistarse con él. Lo recibió.

Y tuvo que oír lo siguiente. “No es que yo diga que las IU que implementamos en nuestros programas estén mal. Han funcionado y siguen haciéndolo muy bien en aplicaciones³¹ sencillas”. “¿Sencillas?”, pensó nuestro ya jefe de línea de producto, “¿Se atreve este majadero a llamar sencillos a programas como mi archiconocido TLW v2.0?”. “Por ejemplo, tomemos TLW v2.0”, continuó el osado programador joven, “Funciona bien. Pero muchos usuarios desearían poder manejar varios textos a la vez, y coger algo de aquí y algo de allá. Ahora mismo no hay forma de hacerlo fácilmente. Harían falta varias pantallas, una para cada documento, de modo que pudiéramos verlos todos a la vez”.

“¿Acaso me estás diciendo que vamos a tener que construir e intentar vender ordenadores con 10 pantallas, que saldrán carísimos y nadie va a poder poner encima de su mesa?” bramó nuestro ya jefe. “Nada de eso, por supuesto”, continuó con notable aplomo el joven, “se trata simplemente de que podríamos hacer que la pantalla del ordenador se dividiera en trocitos que se comportaran como pantallas completas, en ventanas lo llamo yo, que, aunque no tan amplias, sí que servirían mediante desplazamiento de su contenido para este propósito”. Aprovechando el aturdimiento del jefe, siguió implacable. “Y por otra parte, la selección de opciones de menú con el cursor... Chist, chist, está, digamos, caduca. Lo de desplazarse sólo en horizontal o en vertical una posición cada vez es francamente poco ágil. Si equipáramos nuestros ordenadores con un chisme que sirviera de puntero, y que permitiera el desplazamiento rápido en cualquier dirección, podríamos rediseñar los menús. Ponemos una barra horizontal fija con el menú principal, seleccionamos con el puntero, y eso nos desplegaría una ventana, por ejemplo en vertical, debajo de la opción seleccionada. Con el puntero de nuevo seleccionamos la segunda opción. Al terminar, borramos la ventana y dejamos la pantalla como estaba antes de abrirse. Bonito, no ocupa espacio apenas, rápido, elegante. Y nos olvidamos de todos los malabarismos con el teclado. El teclado para escribir. Para seleccionar usamos el puntero.”.

Era demasiado. ¿Un imberbe iba a decirle a él, a él, que había dedicado la mayor parte de su carrera a mejorar los teclados y adaptarlos a los ordenadores, que ahora sólo iban a servir para escribir?. Pero no se dejó llevar por el primer impulso. En lugar de echarlo lo pensó mejor.

“No, si como idea no está mal. Pero no has tenido en cuenta que el problema real es el dichoso puntero. Haría falta un periférico adicional y son caros. Ningún fabricante se va a poner a fabricar versiones especiales de bajo coste. Si encuentras un modo de resolverlo, adelante, incorporamos las mejoras a la próxima versión de TLW, la 3.0. Pero te encargas tú”. Nuestro joven programador tan sólo añadió: “Bueno. La misma casa que nos fabrica

31 Llámase “aplicación” a un programa. Es otra forma de nombrarlo que normalmente indica un programa lo suficientemente completo como para permitir hacer algo útil de forma bastante exhaustiva.

los teclados ha sacado hace poco un puntero que no está mal y es barato. Ratón, lo llaman. El nombre, además, tiene gancho para el marketing. Ya se sabe que hay que mirar mucho lo de las ventas. Me ocuparé de los detalles ahora mismo”. Y salió, contento como unas castañuelas.

Nuestro avisado programador, ahora jefe de línea de productos de su compañía, se derrumbó en su butacón sepultado por el peso abrumador de los acontecimientos, que él sabía conducían inexorablemente a su ascenso a director ejecutivo. Pero la historia no había terminado. La dejaremos por ahora, sin embargo.

Nuestro nuevo benefactor, el programador joven, realizó admirablemente su trabajo. Aparecieron así los precursores de los entornos WIMP, y que nosotros designaremos “entornos WMP”. Se trata de IU basados en ventanas, ratones, y menús desplegables (por favor mire la nota al pie)³². Espero que la idea haya quedado más o menos clara en el relato. Se basan en una estilización de los menús, que se emplean como método de acceso general a las posibilidades del programa, pero no es ésta su principal contribución a la comodidad de uso. Con buen criterio, nuestro joven programador ha destacado las ventajas en rapidez de acceso a las opciones de los menús hechas posibles por el uso del ratón (lo que de nuevo es un ejemplo de la necesidad de periféricos adecuados para hacer aquello que queremos con un ordenador), y en comodidad general aportadas por las ventanas, a las que no entiendo como no se las alaba mucho más de lo que se hace. Siguen teniendo, por supuesto, los inconvenientes ya citados de las IU basadas en menús.

El ratón, chisme que probablemente todo LAO haya visto por lo que no parece necesario describirlo con detalle (ver glosario en todo caso), es el periférico tipo puntero por excelencia hoy día. ¿Que cómo marco y elijo las opciones con él?. Desplazo el cursor, me coloco donde quiero, y pulso un botón con el que está equipado. Así de simple. El botón del ratón es el moderno RC de antes. A este procedimiento se le denomina “hacer clic”³³. No añadiré ningún comentario, aunque me cuesta resistir la tentación.

Sólo una nota. Los teclados siguen existiendo. Y siguen teniendo teclas de desplazamiento del cursor. ¿Porqué?. Cuando escribimos, cosa que hacemos muchas veces, siguen siendo lo mejor para desplazarse carácter a carácter o palabra a palabra.

32 Lo de W.I.M.P. es la abreviatura de Windows, Icons, Mouse, Pull-down menus, es decir ventanas, iconos, ratón, y menús desplegables (o menús persiana, si se prefiere). Nos olvidaremos por el momento de los iconos y lo dejaremos en consecuencia en WMP. La denominación completa ha caído en desuso y ahora se habla de otra cosa, que es lo que terminará de ascender a nuestro avisado programador a las cimas de la dirección ejecutiva. Estamos todavía ocupándonos de IU que se pueden montar en ordenadores con pantallas sólo de texto, y por eso no nombramos los iconos. Las IU “casi WIMP” de que hablamos son prácticamente equivalentes desde el punto de vista funcional.

33 Disponer de un ratón pone también a nuestro alcance “hacer doble clic”, “arrastrar”, “soltar”, y algunas otras expresiones mucho más difíciles de explicar y nombrar que de poner en práctica a nada que las veamos hacer.

CAPÍTULO 5. Software: Manejo de programas

Por cierto, que lo de desplazarse palabra a palabra o incluso párrafo a párrafo no se le da nada bien al ratón. Por el contrario, el teclado no se presta a dibujar a mano alzada. Pero, realmente, tampoco el ratón. Para eso están las tabletas gráficas (ver glosario)³⁴.

Interfases gráficos

Luis Ricardo postmoderno

Habíamos dejado a nuestro avisado jefe de línea de productos sepultado bajo un peso abrumador. Disfrutó de una precaria tranquilidad durante un par de años, tiempo durante el cual TLW v3.0 se reveló como un filón casi inagotable. Realmente, había sido revolucionario. Muchos programas de otros fabricantes acabaron adoptando los mismos mecanismos en sus IUs. Aunque la versión 3.2 de TLW se estaba vendiendo aceptablemente, él veía con claridad que habían dejado de tener ventaja sobre la competencia. Incluso, siendo sinceros, había algún programa ligeramente mejor. No se le ocurría nada. Cuando jugaba al tenis con Javier (que ya era director general), éste le preguntaba qué nueva carta tenía guardada para la próxima versión, la 4, la que debía devolverles a una posición cómoda. Y se le notaba impaciente. Malo, malo.

Fue entonces cuando el joven programador que le había sustituido volvió a solicitar una entrevista. Lo recibió con mucha menos reticencia, aunque para no revelar su ansiedad dejó que pasaran algunos días. En esta ocasión, él habló primero, porque se había olido la tostada.

“¿Aún más mejoras?. Bueno, hiciste un buen trabajo con lo de las ventanas y el ratón. TLW 3.x se ha vendido y aún se está vendiendo bien. Haz lo que te dé la gana siempre que no tengamos que aumentar los costes de producción y déjame en paz. Tienes un mes de tiempo, pero antes cuéntame qué diablos quieres hacer ahora”. De este modo podría tranquilizar a Javier si era necesario.

La explicación del programador ya no tan joven fue concluyente. “Los ordenadores que se están vendiendo hoy en día no son como los de hace cinco años. Por el mismo precio, o incluso menor, ahora todos equipan pantallas gráficas, ratones, y cantidades de memoria, capacidad de almacenamiento en disco, y potencias de CPU casi utópicas tan sólo dos años atrás. Tanto, que para hacer un proceso de textos sobra ordenador por todos lados. Estamos desperdiciando un montón de recursos que nunca se emplean. Creo que podríamos pasar todo el diseño básico de nuestra IU a modo gráfico sin que se enlenteciera apreciablemente. Pasaríamos a trabajar en WYSIWYG, con lo que veríamos las cosas tal cual van a quedar, no como ahora, que hay que echarle una buena cantidad de imaginación al asunto. Las pantallas serían mucho más bonitas. Literalmente las podríamos dibujar. Incluso en colores. Basta de limitaciones causadas por el juego de caracteres del ordenador en modo texto, que no nos dejan trazar más que los recuadros de las ventanas. Hasta he pensado que podríamos sustituir las teclas de función por botones dibujados en pantalla. Con un dibujo encima que indique qué hacen, tenemos un número ilimitado de accesos directos a funciones a nuestra disposición. Y ya se sabe que una imagen vale más que mil palabras, con lo que resultarán muchísimo más explícitos que las anticuadas etiquetas de las teclas de función. Jefe, todo eso vende. Es moderno, es seductor, la gente quedará maravillada. Nos forraremos”. Y se fue, de nuevo contento como unas castañuelas.

34 No puedo evitar emitir una risita ante esta situación. ¡Je, je, je!.

Interfases gráficas

Nuestro avisado jefe de línea de producto, antaño programador, se derrumbó una vez más en su butacón, sepultado nuevamente por el peso abrumador de los acontecimientos que él sabía conducían inexorablemente a su ascenso a director ejecutivo. Se lo contó a Javier durante su partida semanal de golf. Tres meses después, con TLW 4.0 ya a la venta, fue ascendido a director ejecutivo. Nuestro programador ya no tan joven pasó a jefe de línea de producto.

Como siempre, procedamos tras el cuentecito a retomar el hilo de la exposición. Digámoslo rápidamente. Es aquí donde estamos ahora mismo en cuanto a IU. Los menús, el ratón, y demás inventos, conducen casi inexorablemente a lo que se denominan actualmente GUI o interfases de usuario gráficas³⁵.

Los entornos WMP, de los que los GUI son un hermano mayor mucho más dotado, añaden de entrada más potencia a los menús, con lo que los hacen casi (pero sólo casi) igual de capaces que nuestras famosas líneas de órdenes. El truco, que detallaremos en el próximo apartado, se basa ante todo en la limitación del número de menús necesarios para completar una orden mediante el empleo de paneles de diálogo, en la rapidez de uso hecha posible por el ratón, y en la utilidad intrínseca de las ventanas. Partimos pues de una IU notablemente potente y razonablemente cómoda.

¿Qué aporta el modo gráfico?. Ante todo hay que reconocer que poder trabajar en modo WYSIWYG es algo bastante conveniente. Aunque no se logra nunca una fidelidad total de la representación en pantalla, este modo de trabajo potencia enormemente una de las grandes ventajas citadas anteriormente para el ordenador: la capacidad de simulación. Viendo en todo momento con un 85% de precisión (algo absolutamente normal hoy en día) como van a quedar las cosas, su comodidad y, por que no decirlo, su utilidad, aumentan en buena medida.

Además, la capacidad de dibujar la apariencia de nuestro programa hace ahora posible llevar la simulación de la herramienta que intentamos sustituir con un programa de ordenador al extremo de que tenga exactamente el mismo aspecto externo. Si mi programa se encarga de gestionar una agenda, puede presentarse en pantalla como un dibujo de la misma, con tapas, páginas que podemos pasar mediante el ratón, solapas de separación de secciones, y demás elementos familiares, con lo que el usuario se encuentra en una situación que le desorienta mucho menos. Un programa que se encargue de hacer sonar discos compactos en nuestro lector del ordenador puede presentarse con los mandos habituales, y así sucesivamente. Nada que objetar por este lado.

El trabajo en modo gráfico facilita, además, algunos intercambios de información entre programas. Cosas absolutamente necesarias, como meter un gráfico dentro de un texto, son

35 GUI es, una vez más, una abreviatura inglesa. De “Grafical User Interface”, en este caso. Y se traduce como aparece en el texto principal. Alguna vez, en alguna revista, Vd. tal vez lo vea como IGU, abreviatura castellana de interfase gráfica de usuario que no suena muy bien, para que engañarnos. Por aclarar posibles dudas, digamos que a este tipo de inventos se los llamó al principio “entornos WIMP”, denominación que introdujimos antes con las IU basadas en ratones y ventanas. Al estar en modo gráfico, se podían presentar en pantalla “iconos”, pequeños dibujitos de variado propósito. El nombre, aunque no está de más conocerlo, ha caído ciertamente en desuso.

CAPÍTULO 5. Software: Manejo de programas

mucho más fáciles, y por si fuera poco se pueden ver. No sólo no hay tampoco nada que objetar, sino que es una ventaja más bien enorme.

Por todo ello, casi podríamos decir que el modo gráfico puede verse como la forma natural en que un ordenador debería trabajar. Las ventajas en comodidad de uso, y hasta potencia por cuanto ésta es más accesible, son claras.

Subrayando que se trata ante todo de una opinión personal, que tal vez pocos suscriban, creo que la principal contribución de los GUI es hacer el ordenador mucho más seductor. Nuestro joven programador no andaba descaminado al hablar en términos de belleza y seducción como argumentos a considerar. Ante todo, los GUI son bonitos, y es un placer enredar con ellos. Dejan jugar con el ordenador, aunque no hagamos nada útil. Un usuario con un GUI es un usuario probablemente feliz, aunque quizá no muy productivo. A base de proporcionar una satisfacción idiota a un usuario inexperto, consiguen que éste no se aburra y, poco a poco, vaya aprendiendo como hacer las cosas. Son la mejor cura por el momento al fenómeno de desesperación ante la propia ignorancia seguida de abandono definitivo. Puede parecer que lo anterior es una crítica, pero en absoluto. Esta es una gran ventaja de los GUI. Quizá un poco cara como veremos, eso sí.

Una consecuencia de lo anterior es que, también en mi opinión, una interfase GUI debe ser todo lo elegante, bonita, y cómoda que pueda diseñarse. No basta con sacar cualquier gráfico, ni con un WYSIWYG mediocre. Tienen que ser tan perfectos como sea posible. El usuario debe quedar impresionado ante la apariencia de la pantalla del ordenador. Las letras deben verse bien dibujadas en todos los tamaños, los colores deben permitir transiciones suaves, las ventanas han de tener buen aspecto, no hay que recargar las pantallas, los elementos gráficos tienen que ser agradables...

Como en cualquier caso, de todos modos, existe un lado oscuro. No creo que a ningún LAO que se acuerde de lo que decíamos hacia el final del anterior capítulo sobre la tendencia a la obesidad de los datos en modo gráfico, le sorprenda escuchar ahora que no todo ordenador está, por mucho que lo desee su propietario, dotado para manejar con un mínimo de dignidad este tipo de IU, sobre todo si se ajusta a los requisitos de exquisitez antes mencionados.

Ante todo, si se desea que un GUI funcione bien, hace falta un ordenador muy potente³⁶. Con grandes recursos gráficos, mucha memoria, gran capacidad de almacenamiento, y una CPU capaz de trabajar muy deprisa. Precisaremos más cuando, en el próximo capítulo, veamos equipos en concreto. Baste por ahora decir que cualquier ordenador no vale para

36 Más que una verdad absoluta, esta afirmación es una realidad lamentable. Un programador puede hacer las cosas de muchas maneras. Es posible programar un GUI bastante digno, tan bueno como cualquiera de los que hoy se encuentran, empleando una cantidad ridícula de recursos. En el mundo de los PCs sería posible citar a GEM y a Geoworks como ejemplos lamentablemente desaparecidos de forma fulminante. Hasta en equipos mucho más birriosos los ha habido que funcionaban correctamente. Por desgracia, y por motivos que tal vez se vayan intuyendo al leer los siguientes capítulos, hoy día se prefiere hacer programas enormes y lentos y, en consecuencia, son precisos magníficos ordenadores para poder ejecutarlos. ¿Se acuerda de lo que dijimos, tiempo ha, sobre Carlos Sainz, los Lancia, los 600, y, en resumen, hardware y software en la velocidad de funcionamiento del chisme?.

trabajar con él mediante un GUI. Intentarlo lleva ante todo a un enlentecimiento tal que cualquier posible ventaja queda ampliamente contrarrestada. Un ordenador debe, en primer y principal lugar, trabajar rápido, el tiempo de respuesta de la IU tiene que hacer que no se note que está, y un GUI puede fácilmente requerir más del 50% del tiempo de trabajo de nuestra CPU, entre otros perniciosos efectos. Por más que nos guste este tipo de IU, puede ser que a nuestro ordenador se le indigeste igual que doscientos huevos comidos de una sentada.

Es probable, por tanto, que bastantes usuarios un poco serios y no tan noveles lamenten, cuando menos en privado, el gran aprecio que parecen sentir los diseñadores de IU modernos por aspectos utilizables como argumentos inmediatos de ventas. Puestos a pedir novedades en una IU, no serían de desdeñar una mayor solidez y coherencia del entorno de ejecución de los programas. Todo usuario un poco experto sabe que el ordenador no debe colgarse jamás, ser rápido, y ofrecer recursos amplios ante todo tipo de desastres que sin duda se acabarán produciendo. Y está dispuesto a sacrificar gustoso algunas sofisticadas apariencias para lograrlo. Desgraciadamente, apreciar estas capacidades no está tan al alcance del que jamás ha usado un ordenador. Vende mucho más para el que compra su primer equipo una apariencia que la solidez de funcionamiento. Ya se irá enterando su propietario de que hay otras cosas tan o más importantes. Pero no se ven, y para entonces habremos realizado una venta más. Quizá con su segundo ordenador sea más cuidadoso, o esto nos permita vender, además, una versión mejorada de nuestro GUI.

Es perfectamente posible montar en un ordenador no muy potente un entorno WMP en modo texto casi tan eficaz y cómodo de uso como un GUI, pero es preciso renunciar a buena parte de los embellecimientos. La potencia que nos sobraba para ejecutar TLW podría haberse empleado en algo diferente a maquillar su aspecto. Quizá en desarrollar nuevos programas que hicieran mejor las cosas o que hicieran cosas diferentes. Pero debe reconocerse que esto no vende tanto como un buen GUI. No está tan bien situado en la postmodernidad actual.

Para matizar cualquier impresión de crítica acerba que pudiera desprenderse de lo anterior, que, recordemos, se da sólo a título de opinión personal, no queda sino terminar reconociendo una vez más los grandes méritos de un GUI bien implementado siempre que se ejecute sobre un ordenador adecuado.

All together now

Luis Ricardo hombre orquesta

Tal vez a estas alturas del texto, bastantes LAOs hayan desarrollado la sospecha de que, a pesar de las apariencias, se procura no contar cosas innecesarias. Al final, hasta lo del código ASCII parece tener más sentido de lo que pudiera parecer. En efecto, el autor está cuando menos intentando que así sea. Algún motivo debe existir por lo tanto para que se haya dedicado tanta atención a interfases aparentemente tan obsoletas como los editores de líneas y demás inventos antediluvianos si nos encontramos ya, como es el caso, cómodamente situados en la era de los GUI.

El motivo no es otro que, en la práctica, todas estas IU se presentan asociadas. Incluso con el GUI más moderno, de vez en cuando toparemos con un editor de línea. Hemos dicho que por otra parte andaban los sistemas de ficheros, y de hecho una IU simplemente es la vía de comunicación con ellos. En algún momento nuestra IU nos pedirá el nombre de un

CAPÍTULO 5. Software: Manejo de programas

fichero, y muy posiblemente nos encontremos entonces ante un editor de líneas. Eso sí a todo color, con ratones, ventanas, y demás adiciones.

En alguna nota al pie se ha dicho ya que no suelen producirse extinciones en masa en las filas de las IU. Una no sustituye a la anterior. Simplemente la absorbe, añadiendo nuevas posibilidades. Las teclas de función, los menús más rudimentarios, incluso los caracteres de control, pueden tener su interés en un momento determinado y los encontraremos dentro de nuestros GUI, dispuestos a prestar el servicio que siempre prestaron, ofreciéndose generosamente a aportar alguna sutil ventaja, a hacer ligeramente más rápido o más familiar a un antiguo usuario el uso de un programa.

Es frecuente por lo tanto que tengamos varias, incluso muchas formas de acceder a la misma función. No es en sí mismo algo malo. Puede crear cierta desorientación a un novato, pero es posible que poco a poco éste vaya pasando de una a otra en cuanto conozca su existencia. Tan sólo un consejo. Si Vd. es efectivamente nuevo en el gremio, sea coherente al principio. Si usa los menús, úselos siempre. Si le gustan las teclas de función, empléelas habitualmente. No mezcle cosas. Es probable que el poco tiempo que pueda perder al seleccionar rutinariamente funciones de una forma ligeramente inadecuada y no tan directa como otra asimismo disponible, se vea ampliamente compensado por su familiarización progresiva y sin despistes con su programa. Tiempo tendrá de liarse con posibilidades adicionales.

Existen algunos otros recursos en las IU que no hemos expuesto con detalle en los anteriores apartados. Por cuanto también los vamos a encontrar dentro del montón, convendría que se conocieran. Vamos a ello.

Una forma en que se intenta facilitar la entrada de datos al usuario, cuando estos son requeridos, consiste en visualizar lo que podemos denominar “campos de entrada de datos” y en desactivar selectivamente algunas teclas. Pongamos un ejemplo. Si el programa necesita en un momento dado un nombre, sin números, de un máximo de 8 letras, el usuario puede encontrarse ante cosas parecidas a:

Nombre? : *****

donde el cursor está al principio de una zona “marcada” de algún modo. Cuando teclea una letra, ésta sustituye al asterisco, y el cursor pasa a la siguiente posición. Si se pulsa un número no pasa nada, quizá como mucho suene un pitido, indicándonos que no, que no es eso lo que el programa espera. Al terminar de teclear el dato pedido, bien porque se pulse RC o quizá automáticamente por haber llenado el espacio disponible, todo vuelve a la normalidad.

Si Vd. se encuentra ante lo anterior, no asustarse. Este método de petición de datos es una herramienta bastante cómoda, y permite que los programas funcionen más sólidamente. Un antiguo adagio informático afirma que “Garbage in, garbage out” (si metemos basura sacamos basura), resaltando el hecho de que una buena parte de los errores de operación de los programas se deben a la introducción de datos incorrectos. El ordenador necesita un número y yo le doy mi nombre de pila, por ejemplo. La entrada controlada de datos es una buena forma de evitar estos problemas.

Respecto a los textos de ayuda de los programas, es también posible que en la actualidad los encontremos ya bastante educados, en la versión que se denomina “ayuda contextual” o “ayuda sensible al contexto”. Anteriormente hablamos de que consistían en una especie de manual con instrucciones dentro del ordenador al que podíamos acceder cuando fuera

preciso. Lo de “sensible al contexto” quiere decir simplemente que el programa considerará donde estoy cuando pido la ayuda para sacarme la página del manual que contenga información de interés en esa situación. Esta forma de funcionar hace a las ayudas mucho más prácticas.

Ya que estamos con textos de ayuda, podemos aprovechar para presentar a otro invento, aquel que da título a este libro: los ficheros “Readme.1st” o “Leeme.ya”, en tan sólo una de sus denominaciones ya que también se suelen llamar “Read.me”, “Lee.me”, “Leeme”, y de otras mil formas. Siempre nombres que se presten a indicar que la información que en ellos se contiene es de interés y preferiblemente debe conocerse antes de empezar a usar el programa a que se refieren. Se emplean para explicar aquellas modificaciones de última hora que no se han podido meter en los manuales impresos, que siempre requieren un mayor tiempo de preparación, o para aquellas cosas que deben necesariamente tenerse en cuenta si se quiere que no haya problemas.

Acceder a ellos, lamentablemente, requiere que se sepan como usar las herramientas que permiten visualizar su contenido, lo que los aleja un tanto del alcance de cualquier mortal. En ocasiones, y mediante métodos diversos, se hace que sean autovisualizables, por llamarlo de algún modo. A pesar de todo, su cruel destino consiste en ser ignorados por casi todo el mundo, que piensa en ellos tan sólo tras haber topado con los problemas que intentaban prevenir. Como consejo, búsquelos siempre que tenga que comenzar a usar un programa nuevo. Y léalos, como su nombre indica, antes de empezar a hacer nada.

Los manuales en línea también hace ya tiempo que se sofisticaron en cierta medida. Poco más difícil que meter un manual dentro del ordenador, y, sin embargo, mucho más vistoso, es meter una serie de ejercicios prácticos que ayuden a un nuevo usuario a aprender las posibilidades de un programa a base de practicar de forma controlada. Este tipo de invento son los “tutoriales en línea”. Semejante anglicismo en la denominación es perdonable, si bien no debe ponerse una confianza desmesurada en la eficacia del invento. Suelen ser útiles tan sólo como introducción y en caso de extremo apuro. No sustituyen en absoluto al manual si éste está bien explicado.

Lo de los paneles de diálogo, que introdujimos como algo que aparecía a partir de los entornos WMP, merece algún comentario adicional. Un problema de los menús normales era que obligaban a un farragoso recorrido cuando había que precisar la función a ejecutar. ¿No sería posible presentar todos los parámetros disponibles a la vista para elegir entre ellos?. Ésta es la idea de los paneles de diálogo. Es una especie de menú con buena apariencia en el que se presentan simultáneamente todas las opciones posibles, de forma que yo pueda marcar, normalmente con la ayuda del ratón, aquellas que me interesen. Si tiene sentido seleccionar varias, puedo marcar varias. Si son mutuamente excluyentes, el programa sólo me deja marcar una. Son una especie de entrada controlada de opciones aplicada a los parámetros de una función (o de operación del ordenador, ya que se emplean también muy frecuentemente para facilitar la configuración del programa).

Los paneles de diálogo no sólo son cómodos y claros sino que, al dar acceso a todos los parámetros, resuelven el problema de la falta de potencia asociada a los menús. Con ellos, si su programador lo decide así, puedo hacer lo mismo que con una línea de órdenes.

Tan sólo quedan un par de puntos que comentar antes de dar el apartado por finalizado. Vimos que existía una cierta tendencia en los modernos GUI a reproducir la apariencia de

CAPÍTULO 5. Software: Manejo de programas

las herramientas que implementan en el ordenador. Esto evita en cierto grado la desorientación del usuario, pero no resuelve el problema de su uso. Ya vimos, y volvemos a insistir en ello, que el problema consiste en que con el ordenador son posibles nuevas opciones, aquellas que dan sentido a su empleo en lugar del de la herramienta original, y estas deberán incluirse a través de nuevos mecanismos o nuevas apariencias. La similitud no podrá ser nunca total, y, en ocasiones, puede quedarse en mero maquillaje. Un posible problema de este enfoque en el diseño de IU es que, a fuerza de desear dar a nuestro programa una apariencia familiar, terminemos por dificultar su uso al adaptarla al ordenador. Precaución con este tema, por tanto.

Y para que se vea la futilidad de algunas ideas aparentemente maravillosas y la eterna presencia en informática de técnicas precipitadamente consideradas obsoletas, nada mejor en mi opinión que un análisis de lo que ha pasado con los botones que pretendían sustituir a las teclas de función en los GUI. Para evitar que vuelva penosamente a la hoja en que nuestro programador joven los proponía, cito literalmente sus palabras: "...hasta he pensado que podríamos sustituir las teclas de función por botones dibujados en pantalla. Con un dibujo encima que indique qué hacen, tenemos un número ilimitado de accesos directos a funciones a nuestra disposición. Y ya se sabe que una imagen vale más que mil palabras, con lo que resultarán muchísimo más explícitos que las anticuadas etiquetas de las teclas de función...".

En los GUI, encontramos con profusión los antedichos botones. ¿Quiérese decir que han sustituido a las teclas de función y han conseguido su propósito?. Sólo en parte. No se emplean tanto las teclas de función, es cierto, y los botones se emplean en su lugar, también es cierto. Pero lo de que la cosa resulte más inmediatamente reconocible que las anticuadas etiquetas es un fiasco total. Cuando uno se enfrenta a 50 botones dentro de un programa, todos ellos más bien pequeños y equipados de gráficos multicolores que pueden querer decir cualquier cosa, pocos son capaces de saber qué va a hacer uno de esos infectos inventos cuando se haga clic sobre él por primera vez. Tanto es así que ha sido necesario rediseñarlos. Lo mejor de todo viene ahora. Para que sean reconocibles se han añadido, mediante uno u otro mecanismo, etiquetas que explican, en correcto texto legible y no mediante gráficos de los que sólo teóricamente valen por mil palabras, para qué diablos sirven. ¡Henos aquí de nuevo con las etiquetas de las teclas de función reinventadas en versión GUI!. Parece mentira lo que avanzan las cosas gracias al progreso³⁷.

Consejos para programadores sádicos

Luis Ricardo, de nuevo intratable

Imaginemos que Vd. sea un perverso de esos que gozan chinchando al prójimo. Imagine-mos también que la desgracia ha querido que sobre sus espaldas haya recaído la responsabilidad de diseñar la IU de un programa. Vd. está seguro de que se va a vender por cientos de miles de unidades, y se le hace la boca agua ante la posibilidad de martirizar a tan notable porción del género humano. ¿Cómo dar rienda suelta a sus bajos instintos mediante el diseño de la IU de la aplicación?.

37 El autor no puede reprimir una carcajada, mucho más rotunda que la anterior risita: ¡Jo, jo, jo!.

Consejos para programadores sádicos

Nada más fácil. Una serie de recetas simples y de efecto fulminante están a su alcance. Veámoslas rápidamente.

El primer recurso con que cuenta, no por conocido menos dañino, consiste en sobredimensionar el programa de tal modo que sea gordo y lento. Que si alguien quiere usarlo tenga que gastarse un buen dinero adicional en mejoras a su ordenador. Ataquémosle el bolsillo, y si no que se enfrente a una lentitud de uso exasperante. Y podemos añadirle algunas gotas de picante haciendo que alguna opción poco habitual pero necesaria cuelgue irremediablemente el equipo.

En segundo lugar, es probable que desee ser incoherente. Que una función se seleccione de una forma y otra similar de otra absolutamente distinta. Esto resulta molesto de entrada, ya que así la gente se agobia al principio cuando no encuentra la segunda función, totalmente necesaria por otra parte, y cree que no está disponible. Y, además, obliga a que haya que buscar en las ayudas.

...Donde le espera una nueva sorpresa. Redáctelas de modo que resulten incomprensibles³⁸. Trúfelas con expresiones pseudo informáticas complicadísimas. Hable de “la configuración más probable por defecto del dispositivo que abre la secuencia de inicialización del bus adicional en caso de estar conectado”. Esto viste muchísimo, aunque no tanto como si inventa su propio lenguaje. Por qué hablar de teclas de función cuando Vd. puede llamarlas “recursos aceleradores de teclado”, para qué llamar párrafo a un párrafo si Vd. está perfectamente capacitado para inventar un nuevo nombre, algo así como “Textrón: el elemento estructural del documento textual”. ¿Que parece el nombre de un medicamento?. Bueno, ¿y qué?. Aplique lo de las denominaciones incomprensibles no sólo a las ayudas, sino también a los menús, y en general a cualquier cosa que aparezca en pantalla.

Después, y al hilo de la invención de un nuevo e incomprensible lenguaje, sea radical y postmoderno en el diseño de su IU. Además, véndalo agresivamente. Todo el mundo ha estado equivocado al usar editores de pantalla completa para los procesadores de texto. Lo que hace falta realmente es un nuevo invento. ¡La edición monocarácter!. Un radical enfoque minimalista y simplificador que hace todo el diseño transparente como el cristal siempre que uno lea detenidamente el manual de 500 páginas que acompaña al programa. ¿Que nadie más lo usa?. Mejor que mejor. Venda su novedad. Y si elige esta vía, entonces sí, entonces sea coherente. Que nadie pueda usar sus viejos hábitos, que no existan tablas de salvación fuera de la ubicuidad del “monocaracterismo”.

38 Mi favorita en este aspecto, aunque por motivos diferentes a los que se citan, era la ayuda de un WordPerfect 4.0, me parece recordar. No sólo no era sensible al contexto (estaba en orden alfabético), sino que además uno podía deducir por los extraños giros del lenguaje y léxico en ella empleados que se trataba de una adaptación al castellano bastante torpe a partir de una apresurada traducción inicial al catalán del original inglés. El resultado podía calificarse cuando menos de sorprendente. Por cuanto, de una u otra forma, voy a poner por las nubes a versiones posteriores de este programa, espero que la compañía no considere ofensivo que se aireen algunos de sus pecados de juventud. Entre otras formas de alabanza, no es la menor el que en su momento adquiriera WordPerfect 5.2 para Windows, con el que se está redactando el tocho que el LAO tiene entre las manos.

CAPÍTULO 5. Software: Manejo de programas

Esto nos lleva a otro recurso de seguro efecto. No respete los estándares que se hayan establecido a lo largo de décadas para el uso de los programas. Si todo el mundo accede a la ayuda pulsando CTRL+H, haga Vd. que el usuario tenga que pulsar la tercera tecla de función (F3).

Y si quiere realmente ganarse el aprecio de los usuarios de las versiones anteriores del programa, haga que la nueva defina un nuevo formato de datos y no sea capaz de manejar directamente los antiguos. Esto los enloquecerá, seguro.

Bien, si Vd. sigue todos estos consejos, esté absolutamente seguro de que va a ir a aumentar el número de parados en breve plazo. Y con razón, ya que Vd. no sólo es un indeseable sino que también es francamente torpe. Su programa es tan espantoso que no va a venderse. Su diversión va a quedar frustrada. No va a tener público.

Por eso un sádico inteligente sabe que debe utilizar esos métodos con moderación. Hay que hacer un programa que funcione lo bastante bien como para que la gente lo compre y lo use habitualmente. Hay que enganchar al público. Y eso sí, una vez enganchado, darle caña. Ahora me cuelgo, ahora saco una ayuda incomprensible, ahora me invento una novedosa terminología, y así sucesivamente. Divertidísimo. No se puede imaginar Vd., querido LAO, lo bien que nos lo pasamos en la “Asociación de programadores incordiantes” cuando nos reunimos a contar maldades. Porque, a juzgar por el número de programas que entran plenamente en la descripción anterior, debe haber bastantes organizaciones de este tipo.

Volviendo temporalmente a la seriedad, hasta hace no mucho hemos estado viendo más o menos como debían ser las cosas en una IU. Lo que se desea dar a entender es que la realidad será ligeramente menos idílica y, por una u otra razón, habrá que ir desarrollando cierta capacidad de aguante e improvisación para resolver problemas de uso de los programas. Problemas que, ciertamente, no deberían estar allí, pero no por eso dejan de estar.

Es ésta también una razón por la que conviene tener una idea general bastante clara. Sin ella, es difícil sentirse a gusto, pero es prácticamente imposible resolver una de estas dificultades cuando se presentan, por cuanto no existen recetas y todo depende de la comprensión que el usuario afectado tenga del entorno informático y de su capacidad de invención. Tal vez el único consejo que se puede dar es que no se ponga nervioso cuando algo raro ocurra, piense, analice que ha podido pasar, y, si puede, intente hacer lo que quiere de otra manera.

Ya que estamos con cosas que no terminan de ser como debieran, aprovechemos para comentar un par de puntos. En primer lugar, si Vd. está trabajando con una IU equipada de ventanas, es bastante probable que se dé cuenta enseguida de que es mucho mejor una ventana grande que una pequeña. La pantalla de texto normal, de 25 líneas de 80 caracteres, o la gráfica normal de 640x480 puntos, están bien como ventanas únicas, pero tienden a producir cierta sensación de claustrofobia cuando hay que dividir las entre tres o cuatro. Piense siempre en que en este tipo de IU va a ser muy conveniente trabajar en resoluciones mayores, quizá 50 líneas de 132 columnas en modo texto, o 1.024x768 puntos en modo gráfico. Muchos ordenadores modernos lo permiten, y es un requisito casi indispensable para hacer un uso adecuado de entornos WMP o GUI.

El segundo punto es algo ligeramente más sutil y en lo que realmente el usuario tiene pocas posibilidades de actuar. No debería ser cada programa el que eligiera su IU, sino que todos

deberían emplear la misma, con los mismos recursos e idéntica apariencia en lo posible. Para poder funcionar así, lo más conveniente es que la IU, sea del tipo que sea, esté incluida como uno de los recursos del sistema operativo. Cuando es así, éste actúa como un elemento de homogeneización en la apariencia de los programas dando a todos el mismo aspecto, con lo que éste se hace rápidamente familiar al usuario. La IU a nivel sistema operativo tiene un claro efecto integrador de las distintas aplicaciones, especialmente si incluye servicios que faciliten el intercambio de información entre programas. Es por esto por lo que en la actualidad, las IU, especialmente los GUI, aparecen más como partes o extensiones de un sistema operativo que como algo propio de un programa aislado. Y así debe ser.

Sistemas de ficheros

Organizando el almacén de Luis Ricardo

Bueno, es de esperar que sepamos ya bastante más sobre IU que al empezar el capítulo. En mi modesta opinión, con lo anterior tenemos ya un buen 50% del funcionamiento del ordenador dominado. No conviene olvidar de todos modos que las IU no son más que la forma de comunicación con el ordenador. No son ellas, sin embargo, las encargadas de almacenar datos, programas, establecer mecanismos de ejecución, ni otras mil cosas igualmente necesarias.

Por ponerlo de otro modo, las IU son una vía de comunicación, una puerta de acceso. ¿Pero de acceso a qué?. Respuesta: al sistema de ficheros. Él se encarga de todo lo demás, él es el núcleo que define como funciona el ordenador.

Vimos ya qué era un fichero, y que la información podía guardarse en ellos con distintos formatos que convenía conocer. Para terminar de entenderlos lo suficiente a nivel general hacen falta tan sólo unas pocas cosas. Dejar constancia, por si no se sospechaba ya, que los programas que vamos a usar cotidianamente no serán sino algunos de los ficheros que tengamos en el ordenador, en algún sitio. Ficheros en los que los datos, en este caso las instrucciones para el funcionamiento del ordenador, se encontrarán como siempre en un formato especial y probablemente bajo nombres que indiquen lo que son.

Y aclarar también cuales son los sitios en que podemos encontrarlos y como estarán organizados. Como vimos, un fichero era un conjunto de datos almacenados de forma permanente en un periférico de almacenamiento. Ha querido el destino que a los periféricos de almacenamiento, cuando se miran de este modo, se los denomine “unidades”³⁹. Una unidad será un lector de disquetes, o nuestro disco duro, o cualquier otro periférico de almacenamiento. Dentro de un sistema operativo dado, cada unidad recibe a su vez un nombre. Por ejemplo, podemos suponer que nuestro sistema operativo llamará a sus lectores de disquetes “flpx_”. Si tenemos dos, el primero será “flp1_” y el segundo será “flp2_”, por decir algo. Los discos duros podrían denominarse “hdsx_”, es decir “hdsk1_”, “hdsk2_”, y así sucesivamente⁴⁰.

39 También “dispositivos”, o de otras mil formas, en realidad. La denominación anterior es quizá la más habitual. Lo de “dispositivo”, en inglés “device”, es también frecuente, pero hay que reconocer que tiene un significado algo más general que no entraremos a detallar.

40 Estas denominaciones, elegidas por cuanto parecen bastante claras, ya que “flp..” es la abreviatura de floppy (disquete en inglés), en tanto “hdsk..” es la de “hard disk” (disco duro en

CAPÍTULO 5. Software: Manejo de programas

Cuando queramos acceder a un fichero habrá que indicar su localización, dar su nombre completo. Supongamos que quiero hacer algo con un fichero al que he llamado “Cuentas”, que lo tengo en un disquete, y que he puesto el tal disquete en el segundo lector de disquetes de mi ordenador. Bien, en este caso, para mi ordenador, el nombre completo del tal fichero, bajo el cual él es capaz de encontrarlo, es “flp2_Cuentas”. No es lo mismo que “flp1_Cuentas”, por cuanto yo podría tener otro “Cuentas” en otro disquete en el primer lector, ni “hdsk2_Cuentas” que estaría en el segundo disco duro, ni “flp2_textos” que estaría en el mismo disquete pero sería otro fichero con otro nombre.

No hay absolutamente nada raro en esto. Es como ir a buscar a una persona y decir “Busco a Juan Pérez”. Juan Pérez (el fichero) no es algo muy definido. Mucho más fácil será encontrar a mi Juan Pérez si digo “Busco a Juan Pérez, que vive en Zaragoza”. Con esto de Zaragoza (la unidad) la cosa está ya algo mejor.

¿Ya basta?. No. El problema surge cuando se trata de manipular 800 ficheros almacenados todos en la misma unidad⁴¹. En cuanto tenemos muchos algos y buscamos uno, no tarda en aparecer el efecto de “la aguja en el pajar”. Ni yo, ni nadie, se acuerda de la lista de 800 ficheros que tiene en un disco duro. Lo que se hace es decirle al ordenador que me saque una lista de los que hay. Pero, aún así, buscar el nombre deseado entre 800 no es fácil⁴². La forma de salir de este familiar atolladero es asimismo familiar. Se hacen grupos. Si quiero poder moverme entre 800 ficheros, es mucho mejor que los agrupe, clasificándolos bajo el criterio que sea en montones más pequeños. Como se ve, la informática raramente es original en sus métodos. Si lo es, en cambio, en sus denominaciones. A aquellos sistemas de ficheros que permiten hacer esto, hoy casi todos ellos, se les denomina “sistemas de ficheros jerárquicos”.

Aparece así la noción de “directorio”. Un directorio no es más que un grupo de ficheros, nuevamente con un nombre. Como puede verse, estamos empeñados en nombrarlo todo. No queda más remedio. Supongamos que hacemos 80 grupos de diez ficheros cada uno en nuestro hipotético disco duro. Viendo la lista de uno sólo de esos grupos, encontraremos con seguridad nuestro “Cuentas” en la primera ojeada.

Lo de “jerárquicos” viene de que no suelen montarse grupos aislados, sino que podemos hacer “montones de montones”, es decir, que dentro de un directorio, puede encontrarse

inglés) son similares a las que se usaban en el Sinclair QL, un ordenador ya extinto. Si el lector prefiere algo más actual, podemos decir asimismo que tienen cierta semejanza a lo que se acostumbra encontrar en UNIX.

41 ¡Vamos a animarnos a usar de inmediato la terminología aprendida!. Por si aún no se ha acostumbrado, lea “en un mismo disquete” o “en un mismo disco duro”, es indiferente. ¿Que 800 son muchos ficheros y nadie tiene tantos?. Vd. es verdaderamente un pardillo. En un disco duro del ordenador en que escribo esto hay, exactamente, 5.513. Y no son muchos para lo que se ve por ahí.

42 Bien, hay formas de manejarse decentemente en esta situación. Por ejemplo, se puede mostrar la lista en orden alfabético. Le estoy haciendo un poco de trampa. Pero no crea que mucha, ya que estoy suponiendo que conozco el nombre del fichero que busco, cosa que no siempre es cierta.

otro o varios directorios. Esto da lugar a una cierta subordinación entre ellos. Los directorios menos cruciales serán aquellos que sólo contengan ficheros. Los que además de ficheros contengan directorios serán más importantes y estarán por encima en orden de importancia. Al final habrá siempre un directorio que no estará dentro de ningún otro. Este será el más importante. Se le denomina “directorio raíz”⁴³. A un directorio que está dentro de otro, por debajo y subordinado a él, se le llama “subdirectorio”.

La nomenclatura, eso espero, es por el momento bastante clara. Tiende ligeramente a la confusión en su uso cotidiano por cuanto el tema de las jerarquías es siempre algo relativo. Si yo empiezo a partir de un punto cualquiera, incluso el penúltimo subdirectorio, o sea, uno muy poco importante en orden absoluto, y no me importa considerar lo que haya por encima de él, me referiré a él como directorio a secas, y hablaré de subdirectorios sólo cuando deba nombrar aquellos que se encuentren por debajo suyo. En cuanto cambie de punto de vista, mi “directorio principal temporalmente” pasará a ser lo que de verdad es, un subdirectorio. No se establecen comparaciones entre grupos separados, es decir, que un subdirectorio no lo es de otro por estar más lejos del raíz que éste. Es necesario que esté dentro de él, o dentro de uno que está dentro de él.

Los nombres de directorio se separan entre sí mediante alguna convención tipográfica. Por poner un ejemplo, vamos a considerar que en nuestro caso el signo de subrayado actúa como separador. Un posible nombre completo de un fichero, con subdirectorios y todo, podría ser algo así como esto “hdsk1_Contabilidad_Año1994_Enero_Cuentas”. Puede verse que el empleo de este tipo de organización permite no sólo localizar las cosas, sino también agruparlas adecuadamente y darles nombres completos bastante descriptivos. El nombre completo me indica que está en el primer disco duro de mi ordenador, que tengo un grupo de ficheros con datos de contabilidades, que dentro de éste tengo grupos organizados por años, que dentro de cada año están agrupados por meses, y que dentro del grupo de enero tengo un fichero que se llama “Cuentas”. Ya que estamos con tipografías, observe que no hay espacios en blanco. El espacio en blanco se usa casi únicamente en informática para separar cosas. Un nombre, sea de lo que sea, jamás puede contener espacios⁴⁴.

Por supuesto, yo podría tener, en el caso del ejemplo anterior, otro fichero llamado “Cuentas” siempre que estuviera en otro directorio. Algo así como “hdsk1_Contabilidad_Año1994_Febrero_Cuentas”. No puede haber jamás dos ficheros con el mismo nombre completo⁴⁵.

43 El nombre viene de que esta organización puede visualizarse fácilmente mediante un diagrama arborescente, que comienza en la “raíz” (el directorio inicial) y se va ramificando hacia las “hojas”. Viene a ser como un árbol boca abajo. No se habla de “directorios hojas” (podríamos considerar que las hojas son los ficheros, siempre el final de la arborescencia) aunque sí se habla en ocasiones de “ramas” y “ramificaciones”.

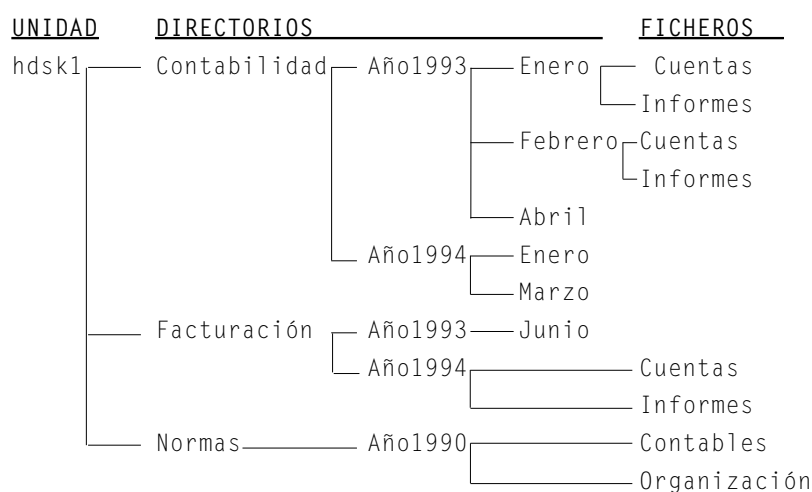
44 Bueno, bueno. Hay trucos que permiten, con oscuros propósitos, meter espacios en blanco dentro de los nombres de algo. Son trucos. Sucios. No lo intente. Si alguna vez topa con ellos, que Dios le ampare. En ocasiones, por ejemplo, se emplean en algunos esquemas de protección contra la copia de programas (ver glosario).

45 Bueno. Si Vd. tiene dos disquetes diferentes y los lee, cada uno cuando le corresponda, en el mismo lector, por supuesto que podrá tener dos “flp1_Contabilidad_Año1994_Enero_Cuentas”. Pero, en este contexto, cada disquete es una unidad diferente. Con dispositivos de

CAPÍTULO 5. Software: Manejo de programas

¿Cuál podría ser la organización completa del disco duro que estamos viendo?. Algo así como el diagrama siguiente. Observe en él que pueden existir directorios vacíos. No todos los directorios contienen ficheros. Yo creo mi organización, mi sistema de directorios, y ya iré dejando los datos en sus correspondientes ficheros dentro de alguno de los directorios cuando corresponda. Y observe que tampoco nadie le obliga a que todos los subdirectorios tengan la misma estructura, nombres, ni nada similar. El usuario es el que decide como organiza sus datos.

Observe también que un directorio puede contener otros directorios o ficheros. Pero no puede contener datos directamente. Los datos estarán en los ficheros que estén dentro del directorio.



Los nombres completos de las diferentes cosas que tenemos aquí, y lo que son, serían como sigue:

hdsk1_Contabilidad_Año1993_Enero_Cuentas	Fichero	(DATOS)
hdsk1_Contabilidad_Año1993_Enero_Informes	Fichero	(DATOS)
hdsk1_Contabilidad_Año1993_Febrero_Cuentas	Fichero	(DATOS)
hdsk1_Contabilidad_Año1993_Febrero_Informes	Fichero	(DATOS)
hdsk1_Contabilidad_Año1993_Abril_	Directorio vacío	
hdsk1_Contabilidad_Año1994_Enero_	Directorio vacío	
hdsk1_Contabilidad_Año1994_Marzo_	Directorio vacío	
hdsk1_Facturación_Año1993_Junio_	Directorio vacío	
hdsk1_Facturación_Año1994_Cuentas	Fichero	(DATOS)
hdsk1_Facturación_Año1994_Informes	Fichero	(DATOS)

almacenamiento que admiten el intercambio del soporte de la información, el nombre de la unidad es variable. Depende de en que lector lo coloque. El nombre de la unidad pasa a ser algo temporal y no cuenta para la norma anterior.

hdsk1_Normas_Año1990_Contables	Fichero	(DATOS)
hdsk1_Normas_Año1990_Organización	Fichero	(DATOS)

Además de, por supuesto, cualquier subnivel que queramos considerar. Es decir, que también está el directorio “hdsk1_Contabilidad_” y en él podríamos perfectamente crear un fichero llamado “Resumen”, lo que nos daría el nombre completo “hdsk1_Contabilidad_Resumen”.

Volviendo a lo de localizar a nuestro Juan Pérez, la adición de directorios nos permite tener el análogo al distrito postal, la calle, el número, el piso, y hasta las horas en que se le puede encontrar. Más que suficiente.

¿Que le parece un poco complicado?. No se preocupe. Uno se habitúa rápidamente, ya que la cosa es útil y merece la pena. Además, pondremos más ejemplos en el capítulo 7, y entonces los tomaremos de la vida real. Hasta entonces.

Como casi siempre que el agobio nos sofoca, eso indica que estamos próximos al final del apartado. Terminaremos éste, también como siempre, con otro par de cosas. La primera de ellas consiste en que lo de los ficheros jerárquicos es la forma más normal que adoptan los sistemas de ficheros en la actualidad. Cuando hace falta poder acceder con rapidez a la información contenida en el ordenador, se montan sistemas más complicados que crean índices de los contenidos de los distintos ficheros de modo que la información se pueda localizar de manera prácticamente inmediata. Se habla entonces de ficheros indexados, y no volverán a salir.

La segunda cuestión, más importante, se refiere al contenido de los ficheros. Recuerde que puedo tener en ellos textos, números, gráficos, programas de ordenador, lo que sea. Suele recurrirse a algún artificio para poder distinguir cual es el tipo de datos que contiene un fichero. Podemos citar varios.

Una posibilidad consiste en agruparlos en directorios con un nombre determinado. Podríamos establecer que cualquier fichero dentro de un directorio que se denomine “prog”, a cualquier nivel, contiene programas, y, análogamente, cualquiera en uno llamado “graf” contendría gráficos, o en uno “text” contendría textos. Con este esquema, “hdsk1_Contabilidad_prog_plan” sería obviamente un programa, en tanto que “hdsk1_graf_Año1994_Cuentas” sería un gráfico, y “hdsk1_txt_Resumen” sería un texto.

Un esquema análogo se basa en el empleo, no de nombres de directorios, sino de extensiones a los nombres de ficheros. Ahora “hdsk1_Contabilidad_plan.prog” sería un programa, en tanto “hdsk1_Año1994_Cuentas.graf” sería un gráfico, y “hdsk1_Resumen.txt” sería un texto. Puede verse que, para indicar que se trata de una extensión y no de un directorio o un nombre de fichero, hemos usado un punto como elemento de separación.

Finalmente es posible marcar los ficheros internamente. Se habla entonces de “cabeceras”, y los ficheros contienen unos bytes iniciales que indican su tipo de datos. Nada permite distinguirlos desde el exterior, pero los programas se encargan de ver si pueden ocuparse de ellos cuando se los intenta usar.

El que Vd. deba enfrentarse a uno u otro esquema depende del sistema operativo que vaya a emplear, o quizá simplemente del programa que use. Todo es posible, hay libertad casi

CAPÍTULO 5. Software: Manejo de programas

total en el campo del software. Incluso pueden usarse todas las alternativas simultáneamente, que es lo que suele ocurrir en la práctica. Es por eso por lo que las tan ansiadas recetas maravillosas son imposibles de dar.

Resignación, y pasemos a otra cosa tras llamar su atención sobre el hecho de que indicar el tipo de datos que contiene un fichero, sea en la forma que sea, no es sino un intento de añadir de sobaquillo un pelín de “significado” a nuestros datos en bruto, para permitir que el ordenador no se líe tanto al tener que manejarlos y sepa un poquito por dónde le va a dar el aire.

Una sesión con un experto

Luis Ricardo, a nuestros pies

Comenzamos el capítulo dedicándonos a mirar por encima del hombro de un auténtico neófito haciendo esfuerzos denodados por apoderarse del control de la bestia. Nada mejor para terminar, ahora que sabemos lo que sabemos, que echar una miradita al ejemplo contrario, el de un verdadero virtuoso enfrentado a una situación similar.

¿Qué es lo que sabe un virtuoso para ponerse delante de un programa desconocido con alguna garantía de éxito? Casi nada. La sabiduría consiste, según un antiguo adagio, en ir quitando cosas diariamente, y la del virtuoso está casi en blanco. Los datos técnicos están, en enorme número eso sí, y se consideran cuando es necesario, pero no sirven como guía. Un virtuoso sabe:

- Para qué sirve el programa que va a utilizar.
- Conoce, o puede suponer, como se designan y almacenan los datos y programas, es decir, las características del sistema de ficheros.
- Entiende la terminología informática. Puede figurarse qué quiere decir el lenguaje particular de un programa, no siempre claro.
- Conoce las características generales de manejo de las interfases de usuario habituales, y es capaz de reconocer, y en consecuencia usar, la o las que el programa emplee.
- Sabe qué quiere hacer.
- Sabe como conseguir información adicional.
- Sabe que va a tener que perder un cierto tiempo aprendiendo, que va a cometer errores, y que analizarlos e intentar explicar porqué se han producido va a ayudarle (mucho) a aprender.

Los procedimientos a su disposición para aprender lo necesario son asimismo pasmosamente simples. Sus fuentes de información son, por este orden:

- Leer la pantalla del ordenador en todo momento.
- Si es necesario, mirar las ayudas.
- Si lo anterior falla, mirar los manuales.
- Como último recurso, preguntar a alguien que sepa más. Puede ser una persona, preferiblemente amiga, o, en algunos casos, un servicio de consulta telefónica o una conferencia de expertos accesible mediante modem⁴⁶.

Y sus recursos ante una situación desesperada consisten en:

Una sesión con un experto

- Buscar formas de soslayar el problema a base de inventiva y experiencia.
- Suicidarse, lo que suele ser equivalente en el caso de los problemas informáticos que nos ocupan a darse por vencido, suponer que no hay forma de hacer lo que queremos con el programa que estamos usando, y aparcar el tema para mejor ocasión. Quizá hasta la aparición de otro programa que resuelva el problema.

Trabajar con un programa de ordenador sobre unos datos para hacer algo requiere siempre una serie de etapas muy simples. A saber:

- Se ejecuta el programa. Para esto se le localiza dentro del sistema de ficheros y, mediante cualquier método a nuestra disposición, se pone en marcha.
- Se le alimenta con los datos. Si se trata de crear algo nuevo, no hace falta más que ponerse a ello. Si se trata, como ocurre en muchas ocasiones, de continuar un trabajo ya empezado, entra en juego otra etapa fundamental: abrir ficheros. Localizamos el fichero donde tengamos guardado nuestro trabajo, y le decimos al programa que lo lea y prepare para realizar modificaciones. Esto simplemente pone una copia en memoria y nos deja trabajar con ella⁴⁷.
- Se trabaja sobre los datos. Se añaden datos nuevos. Se modifican los viejos. Se los imprime para conseguir una copia en papel. Se “exportan”, es decir, se guardan en otro formato al empleado normalmente, para hacerlos accesibles a otros programas. Se abren nuevos ficheros de los que se cogen datos adicionales que se incorporan a aquello que estamos construyendo. Quizá si se encuentran en otro formato, se “importan” datos. En resumen, se produce. Durante esta fase, se almacenan periódicamente las nuevas versiones. Cada vez que hagamos una pausa por ejemplo, o cada 5 minutos⁴⁸.
- Hemos terminado. Tenemos el trabajo hecho. Se graba la versión final. Se cierra el fichero. Se sale del programa. Fin.

46 Las compañías telefónicas están haciendo un excelente negocio con los ordenadores. No existen estadísticas, pero son frecuentes conversaciones de bastante minutos y hasta horas de duración en las que alguien intenta ayudar a un colega en situación más o menos desesperada. Si Vd. cuenta con un modem, uno de esos artilugios que permiten conectar ordenadores entre sí a través de la línea telefónica, tiene incluso acceso a mesas redondas en las que gente experta deja interesantes conocimientos casi imposibles de adquirir de otro modo, o responde cuando puede a las dudas que los pobres mortales plantean. La cuenta del teléfono es entonces aún mayor.

47 Abrir un fichero en realidad significa tan sólo que el ordenador va a poder acceder a su contenido. No necesariamente se va a leer completamente en memoria. Es mucho más normal que se lea el trozo adecuado en cada momento, lo que por otra parte permite manipular ficheros de tamaño muy superior al de la memoria RAM de nuestro ordenador. De todos modos, funcionalmente es casi lo mismo.

48 En el capítulo 9 volveremos probablemente sobre ello. Por ahora, hay que tener en cuenta que Vd. está trabajando sobre lo que es una copia en memoria. Si el ordenador se cuelga, se va la luz, o cualquier otro desastre similar, Vd. se queda sin el trabajo que no haya almacenado. Es por esto por lo que es ABSOLUTAMENTE NECESARIO grabar regular y frecuentemente lo que vamos haciendo.

CAPÍTULO 5. Software: Manejo de programas

Algunos comentarios sobre éste sencillo proceder. Obsérvese que estamos constantemente localizando, abriendo, cerrando, grabando, importando, y exportando ficheros. ¿Entiende porqué no basta con conocer la IU y es necesario manejarse bien asimismo con el sistema de ficheros?. Lo de cerrarlos es el proceso contrario a abrirlos. El programa se desconecta del fichero con el que está trabajando. Quita su copia de la memoria. No es algo tan baladí. En algunos sistemas operativos es absolutamente necesario que se cierre un fichero si Vd. quiere volver a abrirlo y encontrarlo como lo dejó. Todo fichero abierto debe cerrarse antes de salir del programa. No lo olvide.

En cualquier momento que Vd. grabe un fichero (o una parte de él, ya que también eso es posible), podrá elegir el nombre con el que se almacenará en esta ocasión. Si lo graba con el mismo con el que estaba, la nueva versión sustituye a la vieja. Si quiere conservar las dos, o las quince que vaya produciendo, deberá cambiar el nombre cada vez. Recuerde que no puede haber dos ficheros con el mismo nombre completo.

En ocasiones un poco complicadas, especialmente tras quince horas de trabajo ininterrumpido y con la perspectiva de otras 200 por delante para completar lo que Vd. desea hacer, es bastante probable que termine deseando poder automatizar la operación del ordenador. Que, con dos pulsaciones de teclas, o una sola orden dada a través de la línea de órdenes, el ordenador se encargue de abrir 2.000 ficheros, trocearlos, empalmarlos, modificarlos, y hacer el trabajo por Vd. No es mala idea. Requiere, de una u otra forma, programar el ordenador. No es en absoluto tan inmediato de hacer como lo anterior, hacen falta bastantes más conocimientos, y no lo trataremos en este libro. Recuerde, simplemente, que ya avisamos en su momento sobre el hecho de que, al usar un ordenador personal, era bastante probable que Vd. deseara hacer algo más que ser un modesto operador de consola.

No basta para usar un ordenador con soltura con conocer lo anterior. Hace falta también una cierta disposición mental, una tranquilidad ante lo imprevisto, capacidad de análisis de lo que el ordenador va haciendo y de sus respuestas, elaborar procedimientos sobre la marcha para hacer lo que queremos en un momento dado. Cosas que se aprenden con la experiencia.

Para ver si le cogemos el aire es para lo que necesitamos mirar dentro de la cabeza de un experto. Pasemos a ello. Supongamos que Pablo, el amigo de nuestro desdichado neófito del principio, debe enfrentarse sin manuales a un procesador de textos nuevo, regalado recientemente por su amante esposa⁴⁹, y que quiere modificar ligeramente un texto suyo para obtener una nueva versión que nuestro neófito pueda usar. Piensa entregar a éste una copia en papel, para que vea como debe quedar, y un disquete con el fichero, para que no tenga que introducir de nuevo los datos y pueda a su vez practicar con comodidad.

Esto es más o menos lo que iría pasando por la cabeza de Pablo: “Bueno, me olvidé los manuales de TLW 4.0 en casa de la suegra. Intentemos ver si puedo usarlo sin ellos. Al fin y al cabo, he estado empleando TLW 3.2 durante bastante tiempo. No debería haber problema. En primer lugar tengo que instalar el programa en el disco duro de mi ordenador⁵⁰. Vamos a ello. ¿Dónde están los disquetes?. ¡Ah!, allí los veo, en la caja roja. Tres disquetes tan sólo, la cosa irá rápida. Cojamos el primero. ¿Dónde lo meto y dónde creo el

49 Esto es, evidentemente, imaginario. Las amantes esposas no suelen tener excesiva sensibilidad informática, y es muy poco frecuente que regalen programas.

Una sesión con un experto

programa?. Bueno, suelen preferir instalarse desde flp1_, por tanto lo pondremos en el primer lector de disquetes. En cuanto a donde lo dejo, no tengo más que un disco duro. Espero que el programa cree su propio directorio y que el nombre que le dé me convenga. Si no es así, ya lo cambiaré sobre la marcha. ¿Cómo se llama el programa de instalación?. Veamos, hagamos un directorio de flp1_...”

Tras teclear algo así como “dir flp1_” y pulsar la tecla gorda de la derecha esa, el ordenador sacaría en pantalla un listado de los ficheros contenidos en flp1_. Supongamos que los distintos tipos de ficheros se designan mediante extensiones de su nombre, como en una de las posibilidades que citamos antes. Y supongamos que lo que Pablo ve es algo así como:

```
Instalar.prog
Instalar.dat
Instalar.graf
Leeme.ya
TLW_
IMPRESORAS_
```

“Bien, bien. Dos directorios, uno probablemente con ficheros del programa, y otro con datos de impresoras. Y en el directorio raíz, el programa de instalación, que no puede ser otro que Instalar.prog. No hay más. Si algo va mal, miraremos el Leeme.ya. Por lo pronto vamos a probar a ejecutarlo”.

Con el programa de instalación corriendo alegremente, es probable que Pablo viera rápidamente en la pantalla de su ordenador, algo así como:

```
¿Desea instalar TLW 4.0 en hdsk1_TLW4_?
```

y pensaría: “Claro que sí. Correcto el nombre del directorio. No me interfiere con nada de lo que tengo. Adelante pues. Le digo que sí”. El ordenador empezaría a cacharrear, guiado por Instalar.prog e iría pidiendo los dos discos adicionales cuando fuera necesario. Pablo los metería en su momento en el primer lector de disquetes, y la cosa seguiría sola durante algunos minutos más. Al terminar, la pantalla de su ordenador podría mostrarle un nuevo mensaje:

```
Instalación terminada sin errores
```

“Vale, ya está. Veamos como se ejecuta”. Nuevo directorio, nueva búsqueda entre lo que apareciera en pantalla, y hay un hdsk1_TLW4_Tlw4.prog. “Conque esas tenemos. Bien. A ejecutarlo. De entrada, una pantalla más bonita que en la versión anterior. ¡Pero si trabaja en modo gráfico!. ¡Y tenemos un GUI!. Bien, veamos si han conservado las teclas de función viejas. F3 servía para abrir un fichero. Sigue funcionando. No está mal. Lo que quiero es una nueva versión de hdsk1_TEXTOS_Carta.TLW3. Lo abrimos. Veo que ahora los tipos

50 No es estrictamente necesario que un programa se encuentre en el disco duro para que se pueda usar. Es, eso sí, muy conveniente. Al proceso de copiarlo desde disquetes, el soporte en que se distribuyen normalmente, al disco duro, se le denomina “instalación del programa”. No se habla simplemente de copia de ficheros por cuanto, aunque puede consistir tan sólo en eso, es frecuente que el proceso sea más complejo.

CAPÍTULO 5. Software: Manejo de programas

de letra se ven tal cual. Muy bonito. En la primera línea está mi nombre. Me desplazo por el documento con las teclas de cursor usando el editor de pantalla completa con que, por supuesto, sigue funcionando, borro mi nombre, escribo el de Ricardo. Vamos a ver si hay algo más. No recuerdo si mi nombre aparecía en algún otro sitio de las diez páginas. Busquémoslo. F5 era buscar. ¿Cómo, que no marcha ahora?. Algo han cambiado. Veamos si la ayuda sigue saliendo con CTRL+H. Sí. Vamos a ver. ‘Pulsar B para temas que empiecen con esa letra’. Ya está. Lo veo: ‘Búsqueda de textos: Acceda al menú principal con CTRL+M y seleccione la opción’. Menú principal, me muevo con el ratón, marco ‘Búsqueda’, le doy al botón del roedor. Editor de líneas para poner el texto a buscar. ‘Pablo’. Bueno, al final había otro. Lo cambio por Ricardo. No hay que cambiar nada más. A imprimirle una copia. Impresora enchufada y en línea. CTRL+M para menú principal de nuevo, selecciono ‘Imprimir’. Vaya, se ha atascado la hoja otra vez...”. Tras resolver el atasco: “Volvamos a ello. CTRL+M, imprimir. Ahora sí. Bueno, sólo me queda grabarle una copia. Cojamos un nuevo disquete. Lo meto en flp2_ que está vacío. Menú principal, grabar, flp2_ejemplo.TLW3. Ya está. Por si no lo ve le voy a grabar una copia en ASCII. Menú principal, exportar, flp2_ejemplo.ASC. Bien, espero que con esto le valga. Me voy a quedar una copia por si acaso. Grabar hdsk1_TEXTOS_ejemplo.TLW4. Cierro el fichero. Menú principal, ficheros, cerrar. Ya está. Menú principal, salir. Espero que esto le sirva de algo. Tal vez no debería haberle grabado la versión ASCII, quizá lo desorienta más. Anda tan perdido... En fin, que pregunte qué es eso. Así se lo explicaré.”

Y eso sería probablemente todo. Nada complicado. No ponerse nervioso en caso de problemas, tranquilidad, pensar, y conocer una buena serie de ideas generales. Esa es toda la receta a seguir.

Querido LAO, ha demostrado Vd. una fuerza de voluntad admirable al ser capaz de tragar el montón de páginas que, como se advirtió en su momento, integran los capítulos 4 y 5. Espero que haya sido de provecho. Me reafirmo en la advertencia de que si no lo ha entendido intente leerlo de nuevo. Cuando sea, ahora mismo, o tras acabar de leer el libro. Son capítulos importantes. Tanto que ahora ya estamos preparados para pasar a ocuparnos de ordenadores de la vida real. El PeCé nos espera impaciente en el próximo capítulo.