CAPÍTULO 4

Software: Principios generales

Software *El alma de la bestia*

Armados con nuestros ya amplios conocimientos sobre la naturaleza de los ordenadores, puede que nos hayamos sentido tentados de dar el libro por terminado y pasar a mayores, enfrentándonos en combate singular al que quizá nos espera en la habitación de al lado desde hace un par de días. El cosquilleo que sentimos en los dedos es un síntoma claro de que lo que en realidad nos apetece es aporrear teclados, arrastrar ratones, y extasiarnos ante pantallas multicolores.

Paciencia. Si el anterior es nuestro caso, sólo es posible recomendar paciencia. Mucha, incluso. Es, por el momento, más lo que desconocemos que lo que sabemos. Para demostrarlo baste recordar que hasta ahora muy poco se ha dicho sobre el software. Y no es que no fuera importante, como sin duda el LAO recuerda. Como cuanto antes acabemos el temario propuesto, antes podrá dar rienda suelta a sus bajos instintos e irse a manejar de verdad el ordenador, vamos a ver si avanzamos algo y empezamos en este capítulo con tan interesante tema.

Ya dimos una definición de software, por lo que no vamos a insistir sobre ello. A estas alturas debería estar claro de qué vamos a tratar. Es probable que el LAO haya contestado para sí, veloz como el rayo: "¡De los programas!. Las instrucciones que componían el mecanismo de las distintas máquinas en que se podía convertir un ordenador. De aquello que le da su flexibilidad. ¡Por fin!". Pues bien, estimado y querido LAO, apúntese un 5 rapado en su libreta de notas. No se trata solo de esto. Casi ni siquiera es lo principal.

No cabe duda de que los programas son importantes. Les dedicaremos algún capítulo medio en exclusiva, y hasta en éste diremos un montón de cosas sobre ellos, pero no lo son todo. En mi opinión, una parte no despreciable de los problemas en que la gente se mete cuando empieza a usar un ordenador tienen su origen en la forma en que normalmente se desdeña la importancia de los otros componentes del software.

Es probable, a la vista principalmente de los esquemas que hemos ido poniendo en las páginas anteriores, que tengamos la fundada sospecha de que el autor da una importancia considerable a los datos. Si es así, apúntese un positivo adicional en su libreta de notas, con lo que ya vamos por el 6 no tan rapadillo. En el concepto de software, recuerde, están incluidos todos los intangibles, todo aquello relacionado con el funcionamiento del ordenador que no es posible tocar y, además de los programas, están los DATOS (y otro buen montón de cosas, como modos de funcionamiento, documentaciones, y demás etcéteras).

En este capítulo nos ocuparemos ante todo de como se organizan los datos en el interior de un ordenador. Hablaremos también de como se pone en marcha, de sus distintos modos de funcionamiento, de como se estructuran los programas, de la forma en que se comporta habitualmente (lo que podríamos llamar su personalidad), y hasta volveremos ligeramente hacia atrás para completar algún temilla pendiente sobre hardware. Terminaremos viendo

Título: *Léeme.ya (Readme.1st) Manual mínimo crítico para PeCés*
Autor: *Luis Membrado Giner* **Editor:** *John Pigeon Publisher*
ISBN: *84-605-7033-9* **Dep. Legal:** *Z-3314-97*
Formato: *17x24 cm, 771 páginas* **PVP:** *5.000 Pts (30'05 euros)*

como hay que proceder para poder manejar un ordenador con soltura, asunto que supongo de interés y que desarrollaremos en el capítulo siguiente. No me diga que no suena interesante y que no puede dejar durante un tiempo más el ordenador en paz. ¿A que casi está convencido de esperar a terminar el libro antes de enchufarlo?.

Tan sólo un pequeño aviso antes de entrar en materia. Este capítulo y el siguiente son fundamentales. Paciencia y atención, por lo tanto. Si no los entiende, léalos dos veces. Y si aún no se acaba de aclarar, hágalo una vez más. Mejor que domine las ideas generales antes de que comencemos con un ordenador concreto, lo que haremos a partir del capítulo seis. Pero tal vez fuera mejor que antes de volver a leerlos terminara el libro de una tirada sin prestar demasiada atención a si le va quedando algún punto oscuro por el medio. El libro es gordo, volver atrás difícil, y es probable que algunas de sus dudas de dentro de poco se aclaren solas al leer cosas que vienen después. También puede ocurrir que entonces, con una visión completa de todo lo que aquí puede encontrar, esa segunda lectura sea mucho más provechosa. Y quizá quiera extenderla a aquellos otros capítulos que le hayan resultado particularmente problemáticos. En fin, Vd. verá.

Enchufando el ordenador — *Despertando a la bestia*

Como no deseo en absoluto ofender a todo aquel LAO que intenta poner lo mejor de sí mismo en seguir este difícil texto, y ya lo he puesto bastante verde al recriminarle su olvido de la importancia de los datos, vamos a darle temporalmente una razón parcial y a empezar diciendo un par de tontadas sobre los programas, que al parecer es lo que todo el mundo espera cuando se habla de software.

Es preciso comenzar por aclarar que para que el ordenador empiece a marchar van a hacer falta un buen montón (de programas), dado que el hardware, abandonado a su suerte, es totalmente incapaz de funcionar incluso al mínimo nivel[1]. Si pensamos que un teclado es un teclado y una pantalla es una pantalla y una CPU una CPU, y que con enchufarlos la cosa tiene que chutar ya de algún modo, estamos en un gravísimo error. Tanto que estoy por suspender fulminantemente al lector que haya osado pensar así y dejarlo para septiembre. El que así piense no ha terminado de asimilar la naturaleza del ordenador como máquina, y más vale que intente cambiar de punto de vista y colocarse en la piel de algo que podría hacer cosas pero no sabe como. Sin programas, nuestro ordenador no sabe usar su memoria, ni hacer girar un disquete, ni siquiera sabe leer la pulsación que nosotros hagamos de una tecla.

El proceso que todo ordenador sigue para ponerse en marcha es absolutamente ilustrativo de lo que estoy diciendo, y es tan importante que hasta tiene nombre propio. Se denomina "boot", que podríamos traducir por "botar" (como los barcos) o, de forma mucho menos poética, como "proceso de arranque". Si partimos de un ordenador más o menos completo, tal como lo hemos dejado al final del capítulo anterior, en este momento tenemos un teclado, un ratón, una pantalla, quizá una impresora, y, por supuesto, la unidad central con su CPU, su memoria, y uno o varios periféricos de almacenamiento, pongamos un disco duro y un disquete. Bien. Si lo enchufamos a la red y pulsamos el interruptor de encendido, lo que

1 No olvide que era una materia inerte, informe, sin mecanismos.

conseguimos, repito, no es en principio nada más que una pantalla en blanco (o en negro), un teclado y un ratón inertes, y poco más.

Para que la cosa se ponga de verdad en marcha, hace falta que se carguen (es decir, que se lean de donde estén y se coloquen en la memoria del ordenador, dispuestos para ser utilizados) y se ejecuten (o corran, esto es que el ordenador lleve a cabo las instrucciones contenidas en ellos), una serie de programas que le enseñen como funcionar desde el principio. La pregunta en este caso es de donde diablos se van a poder cargar los programas si el ordenador no funciona, al parecer.

Y la respuesta es obvia. ¿Recuerda lo que dijimos sobre distintos tipos de memoria?. ¿Se acuerda que había una que era permanente, que era lo que el ordenador sabía hacer por sí mismo, sin necesidad de nada más?. ¿Se acuerda, en resumen, de la memoria ROM?. Bueno, pues todo ordenador tiene una ROM que servirá al menos para iniciar el proceso de encendido. En ella estarán las instrucciones mínimas para empezar a funcionar. Por lo menos contendrá un programita capaz de verificar que todo está en buen estado, y de iniciar a continuación la carga de otra serie de programitas que se encargan de hacer que teclado, pantalla, memoria, discos y, en resumen, todos los distintos componentes, funcionen. A estos pequeños programas que contienen las instrucciones para hacer marchar periféricos o, en general, cualquier componente del ordenador, se les denomina "drivers"[2].

Mientras que el programa de autocomprobación no necesita seguir funcionando una vez que ha cumplido su labor, los "drivers" deben seguir en memoria y funcionando mientras el ordenador esté trabajando, a no ser que queramos quedarnos sin pantalla, sin teclado, o sin lo que sea. Todo programa en ejecución estará en la memoria del ordenador. Recuerde que la CPU sólo podía coger y dejar datos directamente de la memoria[3].

Una vez cargados los "drivers", y, por tanto, en condiciones de funcionamiento los diversos componentes del ordenador, la cosa no ha terminado. Será preciso cargar algún programa adicional que me permita hacerlos funcionar coordinadamente, repartiendo los recursos del ordenador entre ellos. Hará falta por ejemplo asignar zonas de la memoria RAM para que todos puedan trabajar, y mantener un control continuo de su estado para que no se produzcan errores. Se crean así lo que podemos llamar "variables del sistema".

Como, además, querremos poderle mandar cosas a nuestro chisme, será también preciso que de algún modo el ordenador nos deje darle órdenes para que él las cumpla. De nuevo

---

2 No suele traducirse. Una posible traducción, bastante literal además, sería la de "pilotos", en el sentido de conductores. Los programas piloto se encargarán de conducir los distintos componentes del ordenador. Otra traducción bastante frecuente y también correcta es "controladores". No me gusta, ya que puede confundirse con los controladores de los que hablamos en el capítulo anterior.

3 Si nuestro ordenador es de los que tienen mucha ROM, es probable que dentro de ella esté no sólo el programa de autocomprobación, sino también los drivers y todos los programas que vamos a citar a continuación. Estos ordenadores más "listos" sí que son capaces de funcionar por sí mismos, pero no olvide que es tan sólo porque los programas necesarios están cargados de forma permanente en la memoria ROM.

otro programa, el que suele denominarse, por cierto horriblemente, "procesador de comandos"[4], se encarga de eso.

Y como, además, querremos poder almacenar y recuperar datos con los que el ordenador trabaje, habrá que arbitrar mecanismos para que pueda hacerlo. Otro programita, el "sistema de ficheros"[5], se encargará de esto. El proceso de encendido, el "arranque" del ordenador, puede darse por terminado cuando todos estos programas adicionales han sido asimismo cargados y ejecutados si es necesario.

Cuando todo este complejo proceso ha terminado, lo que tenemos es ni más ni menos que un ordenador que no hace nada. Pero que está dispuesto a escuchar nuestras órdenes y a cumplirlas. Sigue siendo algo bastante informe, sin un objetivo definido, pero al menos los componentes funcionan y existe algún mecanismo para podernos comunicar con él y darle instrucciones adicionales. Por ahora nos conformaremos con dejar el ordenador en este estado semicatatónico e intentaremos aclarar algunas cosas.

La primera de las cuales consiste en que el conjunto de programitas que hemos estado presentando es lo que se denomina, y agárrese a la silla para no caerse, "sistema operativo". Concepto importantísimo que podemos intentar definir diciendo que es el conjunto de programas encargados de hacer funcionar a nivel básico un ordenador[6]. Como tarjeta de visita nos ha permitido ponerlo en marcha, pero nos prestará otros muchos servicios. Por ahora digamos tan sólo, y preste atención, que el conjunto de hardware más sistema operativo define ya un ordenador completo.

Y la segunda es que, como normalmente y hasta que se cargan en memoria para hacer andar al chisme, una buena parte de estos programitas se encuentran almacenados en discos, duros o blandos[7], es frecuente que nos topemos con éste y aquel "sistema operativo en disco". En inglés "Disk Operating System" o, si lo abreviamos como suele ser costumbre, DOS[8]. ¿Le suena de algo el nombre?.

---

4 Lo de comando es una pésima traducción del inglés "command". Podríamos traducir más que correcta y fácilmente la palabreja por "órdenes", pero algún sino fatal ha llevado a que se use con excesiva frecuencia lo de comando, con lo que más parece que estemos involucrados en algún tipo de acción militar.

5 Quizá a algún lector, de los que ya han usado un PeCé, esto le sorprenda por cuanto considere que la forma en que se organizan los archivos es única. Nada más falso. Es otro componente de software, tan variable como los demás y que en ocasiones puede elegirse a voluntad del usuario. Incluso para el PeCé existen diversas alternativas.

6 ¿Qué incluye un sistema operativo?. Es imposible dar una lista. Lo único que puede decirse es que AL MENOS debe incluir lo que se ha citado anteriormente. La generosidad del programador de un sistema operativo, y su concepto de lo que debe ser una "funcionalidad mínima", hacen que existan sistemas operativos descomunales que incluyen prácticamente todo o casi, y otros que no incluyen nada o casi. No hay que apresurarse a preferir los primeros frente a los segundos, ya que no siempre es una sabia elección.

7 Por si no lo recuerda, en ordenadores con mucha ROM, que los hay, lo normal será que el sistema operativo se encuentre almacenado en ella.

8 Como cuando comenzaron a aparecer los DOS para microordenadores, hace cosa de unos 15

Con el hardware y el sistema operativo ya tenemos un ordenador en orden de marcha, aunque todavía no sea capaz de hacer nada concreto aparte de permitirnos darle órdenes para que sepa qué tiene que hacer. En este momento, el ordenador ya funciona, pero como puede verse ha sido tras cargar una buena cantidad de software. ¿Quiere ver cuantas cosas, entre programillas y variables del sistema, hay en la memoria de un ordenador casi inútil?. Nada más fácil que mostrar un listado parcial de su contenido. Helo aquí:

Dirección	Usuario	Tamaño	Tipo
0:0000		A0000h, 655.360	——— RAM ———
0:0000	———	400h, 1.024	Vectores interrumpidos
40:0000	———	100h, 256	Zona de datos de la ROM BIOS
50:0000	DR DOS	200h, 512	Zona de datos DOS
70:0000	DR BIOS	B30h, 2.864	Controladores de dispositivo
70:050B	PRN		Controlador disp. incorporado
70:051D	LPT1		Controlador disp. incorporado
70:052F	LPT2		Controlador disp. incorporado
70:0541	LPT3		Controlador disp. incorporado
70:0553	AUX		Controlador disp. incorporado
70:0565	COM1		Controlador disp. incorporado
70:0577	COM2		Controlador disp. incorporado
70:0589	COM3		Controlador disp. incorporado
70:059B	COM4		Controlador disp. incorporado
70:0602	CLOCK$		Controlador disp. incorporado
70:0645	CON		Controlador disp. incorporado
70:0671	A:-C:		Controlador disp. incorporado
123:0000	DR DOS	11B0h, 4.528	Sistema
123:0048	NUL		Controlador disp. incorporado
23E:0000	DR DOS	110h, 272	Sistema
24F:0000	DR DOS	18620h, 99.872	Sistema
251:0000	D:	B4E0h, 46.304	Controlador disp. de carga
DDF:0000	DR DOS	1E00h, 7.680	15 búfer(es) de disco
F41:15D0	DR DOS	DE0h, 3.552	Código BIOS DR DOS
E7C:3000	DR DOS	9440h, 37.952	Código kernel DR DOS
1AB1:0000	COMMAND	1600h, 5.632	Programa
1C11:0000	COMMAND	110h, 272	Entorno
1C22:0000	MEM	50h, 80	Entorno
1C27:0000	MEM	13AE0h, 80.608	Programa
2FD5:0000	———	702B0h, 459.440	DISPONIBLE

No es momento de preocuparnos del significado de tan bonito cuadro o de como lo hemos obtenido. Baste por ahora ver que hay un buen montón de cosas en la memoria del chisme, todas a la vez.

Al proceso de poner algo en marcha en informática, puede ser un programa aislado o, como en nuestro caso actual, el propio ordenador, se le suele denominar “inicialización”. Comprende siempre la carga de los módulos de software necesarios, la comprobación de los

---

años, los discos (flexibles por supuesto, no digamos ya los duros) eran algo caro y casi exótico, no está muy claro que el significado fuera “sistema operativo EN disco” y no “sistema operativo DE disco”, ya que una de sus misiones consistía precisamente en permitir usar los tales discos. Felizmente, semejante cuestión carece de interés en la actualidad, si bien siguen encargándose de tan oscura, aunque importante, tarea.

recursos disponibles, su reparto según las necesidades, y la creación de variables de control[9] que permitan su funcionamiento de forma coordinada. La importancia de semejante proceso es difícil de exagerar, ya que en ocasiones algo hace que, por ejemplo, las variables del sistema almacenen valores incorrectos, o la memoria no contenga lo que la CPU espera, u otros miles de desastres posibles y entonces...

El ordenador colgado — *Muerte y resurrección de la bestia*

...Entonces puede pasar de todo. En pantalla aparecen algo así como fuegos artificiales, sale una "H" cuando pulsamos una "V" o, simple y llanamente, todo se queda congelado y nada funciona. Se dice entonces, literalmente, que el ordenador se ha "colgado"[10].

¿Qué puede hacerse?. ¿Hay que correr a la tienda más próxima a comprar otro ordenador y esperar que, equipados de nuevo, tengamos más suerte la próxima vez?. ¿Nos toca suicidarnos para expiar nuestra torpeza?. Nada de eso. Simplemente, ha llegado el momento de hacer un "reset", palabra que hay que aprender y que podemos traducir tranquilamente por "reinicialización". Si lo prefiere, hay que salir y volver a entrar. Vamos, que hay que arrancar el ordenador de nuevo.

¿Porqué?. ¿A qué se debe que así curemos los terribles males que sin duda aquejan a ese bicho inmundo que, tras costarnos una no despreciable cantidad de dinero, se empeña en no funcionar?. La explicación es sumamente sencilla.

Lo que hacemos con un reset es aprovechar una de las propiedades de la RAM. Dijimos en su momento que este tipo de memoria era volátil. Si dejábamos de suministrar corriente al equipo, la RAM se borraba totalmente de forma automática. Si, por la razón que sea, los valores contenidos en la RAM del ordenador son erróneos y esto lleva a que no podamos hacer nada para volver a escribirlos correctamente, siempre nos queda el recurso de apagar, dejar que la memoria RAM se borre sola, y encender de nuevo. Perderemos, eso sí, cualquier dato de interés que para nuestra desgracia estuviera en la RAM y no hubiéramos almacenado[11], pero así se copiarán de nuevo, desde la ROM o desde el disco, los valores correctos y volveremos a tener un ordenador operativo[12].

---

9 Sobre qué es exactamente una variable volveremos antes de que acabe el capítulo. Paciencia.

10 En sistemas más complejos que los microordenadores personales, como por ejemplo aquellos en los que varios están conectados entre sí (las llamadas "redes"), se suele emplear también la expresión "caerse". Se habla entonces de que "el sistema (o la red) se ha caído".

11 Un reset no es una operación deseable. Tiene problemas. Se pierden los datos que no hayamos guardado. Se pierde tiempo, ya que a todo ordenador le cuesta un ratillo arrancar. Se pierden los nervios, ya que tener que hacer un reset quiere decir que el chisme no funciona bien y conviene intentar arreglarlo, lo que puede llevar a largas horas de meditación y algunos miles de probatinas diversas. O sea que un ordenador que se cuelga no es, en definitiva, ninguna broma.

12 En su momento veremos con más detalle el proceso de arranque en un PeCé. Observaremos entonces que es perfectamente posible almacenar en disco (por supuesto no en la ROM) valores incorrectos de las variables del sistema que ineludiblemente bloquearán el equipo ya en el proceso de arranque. Que no cunda el pánico. También ésta triste situación, como explicaremos entonces,

Es tan habitual que tengamos que "resetear" el ordenador, que la cosa ha dado lugar a algún chiste al respecto explotando el carácter punto menos que mágico que los informáticos dan a semejante recurso. No resisto la tentación de contar uno. Iban tres amigos, un químico, un físico, y un informático, de excursión en un coche de vigésima mano que, por supuesto, estaba ya bastante lejos de su mejor momento. Al subir un puerto particularmente empinado y tras algún titubeo, el coche se detuvo. El físico aprovechó de inmediato la ocasión para dar una explicación al acontecimiento. "Esto se debe", dijo, "a que el cambio de altitud, al disminuir la presión atmosférica, ha ocasionado el bloqueo del carburador pues ahora el gradiente de presiones ya no es lo suficientemente grande". Acto seguido, el químico, que no iba a dejar que fuera el físico el único que diera las explicaciones, opinó: "Nada de eso. El problema es el menor contenido de oxígeno del aire, asociado seguramente a un octanaje incorrecto de la gasolina. Esto lleva a que la combustión no sea adecuada". Finalmente, el informático propuso: "¿Y si salimos y volvemos a entrar?".

Razones profundas — *Biología molecular de la bestia*

Pero bueno. ¿Cómo es posible que un ordenador se cuelgue?. A los humanos, de vez en cuando se nos cruzan los cables, pero no andamos confundiendo jota con bolero cada dos por tres. Ciertamente. Para entender esto, poner de manifiesto una característica absolutamente básica de los ordenadores que va a ser otro de los hilos conductores del texto, y comenzar con los datos, es preciso que buceemos algo más en las interioridades del bicho y nos enfrentemos al bit y al código ASCII. Cara a cara. Con valor. No asustarse.

La cuestión a aclarar es la de como se almacenan las cosas dentro del ordenador o, para ser más precisos, en su memoria[13]. En realidad, ya la contestamos al hablar de hardware. Las señales eléctricas eran muy fáciles de manipular y detectar, y hacer circuitos para ello no tenía mayor complicación. El interior del ordenador, y la memoria en particular, es un amasijo de circuitos eléctricos. Nada tiene de extraño pues que almacenemos cosas en el ordenador como señales eléctricas. El que tengamos o no tengamos un voltaje o una corriente puede utilizarse para indicar algo. A nivel cacharrería la cosa no tiene misterios. La cuestión real es el significado que le demos a que por un hilo pase o no una corriente.

Comencemos por lo más elemental. ¿Qué información podemos meter en el caso anterior, de que haya o no un voltaje o una corriente?. Porque, recordemos, se trata de manipular información, no voltajes. Bueno, pues que haya o no voltaje (o corriente) es equivalente a decir sí o no. Mirando si hay o no voltaje en un punto, tengo dos posibles valores, o lo hay o no lo hay, ni más ni menos que las mismas posibilidades a que me enfrento al tener que contestar sí o no a cualquier pregunta que se pueda plantear. ¿Tengo frío?. Pues miro si hay voltaje. Si lo hay, tengo frío. Si no lo hay, no tengo frío. ¿Quiero fregar los platos?. Pues miro si hay voltaje. Si lo hay, quiero fregar los platos. Si no lo hay, pues no. ¿Quiero que

---

tiene solución.

13 Lo que sigue es aplicable a todo tipo de memoria, es decir, a la RAM, la ROM, y hasta a los periféricos de almacenamiento. Asimismo hacer notar que hablamos de señales eléctricas, pero podríamos igualmente referirnos a señales ópticas (especialmente si hablamos de periféricos de almacenamiento) o a agujeros en una tarjeta perforada. Se ruega pues que se intente generalizar.

salga un punto en la pantalla de mi ordenador?. Pues miro si hay voltaje. Si lo hay, sí quiero. Si no lo hay, no. ¿Me quiere la chica de mis amores?. Pues miro si hay voltaje, como con la margarita. Si lo hay, me quiere, si no lo hay, no me quiere.

A esta cantidad de información, a un sí o un no, en informática se la llama un "bit". No se traduce y se dice exactamente igual en castellano. La traducción directa, de todos modos, es bastante ilustrativa. Un "bit" es un "poco", o un "algo", de información. El "bit" es nuestro átomo informático, la mínima cantidad de información posible con la que montaremos todo lo demás.

De todos modos, un "bit" es más bien demasiado poco. Podemos decir sí o no, pero no podemos decir, por ejemplo, que sí, que queremos un punto en la pantalla del ordenador pero lo queremos de color rojo, o que sí, que me casaría con Vd. pero sólo si es una rica y guapa heredera y está forrada y me garantiza una vida desahogada en lo sucesivo (o sea, bastante improbable). ¿Qué hacer si en informática queremos tratar estos casos?. ¿Cómo meter en el ordenador este tipo de información?. Pues la respuesta es obvia, y de nuevo se ha probado en otros innumerables campos del saber humano. Si algo tiene sólo dos posibilidades, se hacen grupos, y un grupo en el que cada elemento tiene dos posibilidades tiene ya bastantes más. Tantas más cuanto mayor sea el grupo. Si hacemos grupos de 4 elementos, tenemos 16 posibilidades. Con grupos de 8, 256. Si son de 16, nos vamos a 65.536. Y si son de 32, llegamos a $4.294_{1}967.296$[14].

Si en lugar de mirar el voltaje en un punto miramos el voltaje en cada elemento de un grupo de 4 puntos, podríamos responder a una pregunta de 16 formas diferentes. O de 256, si miramos un grupo de 8 puntos. Son posibles así mayores sutilezas. Y mientras no se nos acaben los puntos, siempre podremos aumentar el tamaño del grupo para albergar posibilidades adicionales[15].

Igual que llamamos "bit" a un elemento de información, se asignan diversos nombres a sus grupos. Un grupo de 4 bits es un "nibble", término que no se traduce y se emplea escasa-

---

14 Para aquellos con inclinaciones matemáticas que se estén preguntando por la procedencia de semejantes numerajos, la fórmula que permite calcularlos es simplemente 2^n, donde n es el número de elementos en el grupo. Se trata de calcular el número de variaciones con repetición de x elementos tomados de n en n, y como aquí tenemos sólo dos elementos (sí o no, blanco o negro, o, siento decir esto pero al fin y al cabo está Vd. leyendo una nota al pie que se supone material avanzado, 0 o 1 si lo hacemos en notación matemática binaria) pues eso, 2^n. El cálculo de este tipo de variaciones ha despertado el interés de los matemáticos desde hace mucho tiempo pues está vinculado a importantes asuntos con gran trascendencia en la vida diaria. Un ejemplo, que supongo de su interés, es el de cuantas quinielas de fútbol en pleno al 15 son posibles. Grupos de 15 elementos, con tres posibilidades cada uno (1, X, o 2) dan la respetable cifra de 3^{15}, es decir $14_{1}348.907$. O sea que si Vd. rellena una columna tiene una posibilidad entre 14 millones de acertar. Algo es algo, pero no es fácil.

15 Quizá esté Vd., apreciado LAO, con la mosca ligeramente detrás de la oreja pues le parezca que le suena de algo esto. Ciertamente. Dijimos que en informática, cosas como un Kilo-algo eran 1.024 algos, y cosas como un Mega-algo eran $1_{1}048.576$ algos, y la procedencia de estos caprichosos múltiplos es ni más ni menos que el ser potencias de dos (2 a la 10 y 2 a la 20, respectivamente, para Kilos y Megas). Del bit y sus múltiplos es de donde sale la querencia por las potencias de dos de los informáticos.

mente, pero un grupo de 8 bits es un "byte", término que sí se puede traducir, que ya conocíamos, y que se emplea muchísimo más. Se puede traducir lo de "byte" por "octeto", pero por diversos motivos es una palabra muy poco popular. Se suele decir "byte" (pronúnciase "bait", si está interesado) y tal vez se acuerde Vd. de los Kilobytes y los Megabytes. En realidad el byte es el elemento práctico de información que se maneja habitualmente, y esto por muy buenas razones que pasamos a exponer a continuación, y que nos llevan directamente al otro punto fundamental de horrible nombre que hemos citado: el código ASCII.

Ya hemos dicho que el ordenador es una máquina que procesa datos de tipo general. Facturas, recibos, cuentas científicas, cartas, y un buen montón de etcéteras. Y eso desde que empezaron a existir como tales en los años 40. ¿De qué manera podemos meter cualquier cosa dentro de un ordenador?. ¿Cómo arreglarnoslas para una vez poner un número y otras una carta al director de un banco por habernos perdido nuestra pasta?. La respuesta, de nuevo, es obvia y ha sido ensayada con anterioridad. Una vez más, del mismo modo que lo hacemos en un papel. Ya hablamos del alfabeto. Pues si metemos el alfabeto en el ordenador, podremos escribir dentro de él lo que nos dé la gana, sea lo que sea, y ya está. Mejor dicho, el alfabeto, los signos de puntuación, las cifras necesarias para escribir números, y alguna cosilla más. ¿Y cómo metemos todo esto en el ordenador?[16]. Bueno, pues veamos cuantas cosas queremos meter. Unas 27 letras. 10 cifras. ¡Ah!, pero como quiero letras mayúsculas y minúsculas serán el doble, o sea, unas 54. Signos de puntuación variados (comas, puntos y comas, comillas, espacios en blanco, paréntesis...) y algunos otros extras hacen que sean claramente más de 64. Quizá con 128 posibilidades podríamos con todo. Éste es el mismo número que dan grupos de 7 bits, pero 7 es un número informáticamente feo ya que no es potencia de 2. Es mejor 8, así como más redondo. Si hacemos grupos de 8 bits y a cada uno de los posibles le asignamos un significado, es decir, le hacemos corresponder una letra, un signo de puntuación, una cifra, o un lo que sea, pues ya está. Y como los grupos de 8 bits son los bytes, basta y sobra con asignar a cada byte un significado de este tipo.

¿Ha visto Vd. películas de espías?. Para que el enemigo no se entere si intercepta una carta, se ponen en clave. Uno de los trucos más típicos consiste en cifrarlas[17]. A cada letra le asignamos un número, y ya está. Por ejemplo, decimos que la "c" es el 11, la "a" el 28, y la "s" el 32. Si queremos decir "casa", escribiremos "11 28 32 28". Mandamos el "11 28 32 28" a nuestro contacto en tierra extraña, al que, además, hemos hecho llegar previamente la clave, por supuesto por separado y con todo tipo de precauciones. Cuando recibe la ristra de números, coge su libro de claves, deshace el cambio, y lee, perfectamente, "casa". A las claves también se las conoce como códigos.

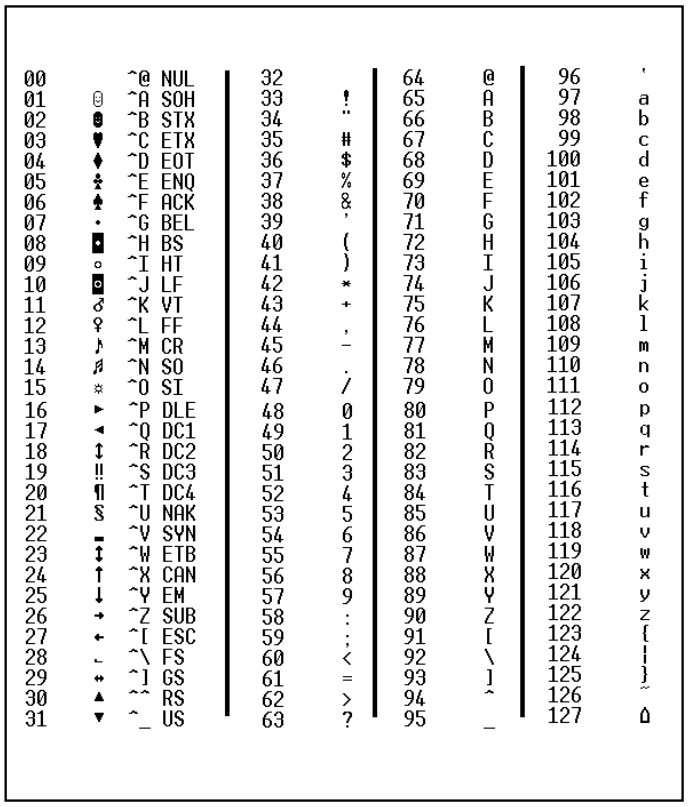
El código ASCII no es nada más que una clave lo suficientemente completa como para poder escribir dignamente casi cualquier cosa. Para hacernos una idea precisa, veámoslo:

---

16 Dijimos que el ordenador era una máquina que procesaba datos alfanuméricos. No quería decir nada más que esto, que conocía el alfabeto y las cifras y se podía escribir dentro de él.

17 El truco es, de nuevo, viejísimo. Me parece recordar que Edgar Allan Poe tiene un relato titulado precisamente "El misterio de la carta cifrada", en el que se cuenta esto y además como proceder para descifrar este tipo de cosas. Ande, léalo.

Su importancia, aparte de su utilidad intrínseca, radica ante todo en que es estándar. Todo el mundo tiene la clave y puede leer las ristras de números escritas con él dentro de un



00		^@	NUL	32		64	@	96	`		
01	☺	^A	SOH	33	!	65	A	97	a		
02	☻	^B	STX	34	"	66	B	98	b		
03	♥	^C	ETX	35	#	67	C	99	c		
04	♦	^D	EOT	36	$	68	D	100	d		
05	♣	^E	ENQ	37	%	69	E	101	e		
06	♠	^F	ACK	38	&	70	F	102	f		
07	•	^G	BEL	39	'	71	G	103	g		
08	◘	^H	BS	40	(	72	H	104	h		
09	○	^I	HT	41	)	73	I	105	i		
10	◙	^J	LF	42	*	74	J	106	j		
11	♂	^K	VT	43	+	75	K	107	k		
12	♀	^L	FF	44	,	76	L	108	l		
13	♪	^M	CR	45	-	77	M	109	m		
14	♫	^N	SO	46	.	78	N	110	n		
15	☼	^O	SI	47	/	79	O	111	o		
16	►	^P	DLE	48	0	80	P	112	p		
17	◄	^Q	DC1	49	1	81	Q	113	q		
18	↕	^R	DC2	50	2	82	R	114	r		
19	‼	^S	DC3	51	3	83	S	115	s		
20	¶	^T	DC4	52	4	84	T	116	t		
21	§	^U	NAK	53	5	85	U	117	u		
22	▬	^V	SYN	54	6	86	V	118	v		
23	↨	^W	ETB	55	7	87	W	119	w		
24	↑	^X	CAN	56	8	88	X	120	x		
25	↓	^Y	EM	57	9	89	Y	121	y		
26	→	^Z	SUB	58	:	90	Z	122	z		
27	←	^[	ESC	59	;	91	[	123	{		
28	∟	^\	FS	60	<	92	\	124	|		
29	↔	^]	GS	61	=	93	]	125	}		
30	▲	^^	RS	62	>	94	^	126	~		
31	▼	^_	US	63	?	95	_	127	⌂		

ordenador. Lo de ASCII no quiere decir nada más que "American Standard Code for Information Interchange", o sea, "código americano estándar para el intercambio de información". No sirve para pasar mensajes secretos, pero sí para meter dentro del ordenador cualquier cosa de forma que todo el mundo pueda leerla, lo que no es poco. Si miramos la tabla anterior con un poco de atención, nos daremos cuenta de algunos detalles graciosos. Por ejemplo, que los primeros 31 números son raros. No representan letras. Lo que viene luego es más normal. El 32 es un espacio en blanco, un signo de puntuación tan importante como cualquier otro. Luego hay más. Luego hay cifras. Luego letras mayúsculas, luego minúsculas, luego algún otro signo de puntuación. Termina con una cosa rara de nuevo y de 128 en adelante, nada, está vacío. ¡Cáspita!. ¿Y eso?.

Parece que proceden dos o tres aclaraciones. Las cosas raras son lo que se llaman "caracteres de control", que básicamente representan acciones en un teletipo, ya que el código en realidad se pensó para manejarlos. En esta zona tenemos el tabulador (9), fin de línea o retorno de carro (13), avance de línea (10), retroceso (8), avance de página (12), y otras lindezas similares[18]. Bastante necesario todo ello, aunque, hay que reconocerlo, algo raro. La forma en que se representan (^A, por ejemplo) es asimismo curiosa y nos llevará a algún

18 Las abreviaturas de la tabla están en inglés, como es lógico y natural. 8=Backspace (retroceso), 13=Carriage Return (retorno de carro), 12=Form Feed (avance de página), por poner algún

comentario posterior[19]. No parece que haya que aclarar mucho la zona normal, aparte de recalcar que es un código americano. No hay "ñ" en inglés, ni letras acentuadas, ni diéresis. No están, no las busquéis. Resulta francamente chocante la enorme zona vacía del final. Es justo la mitad de las 256 posibilidades que teníamos. Hay varios motivos para ello, y uno que puede servirnos consiste en decir que, con el fin de adaptarlo a otros idiomas, esta zona se reserva para dar cabida a eñes, acentos, o lo que queramos[20]. Pero lo que metamos allí ya no es tan estándar como el resto. En un PeCé se llena de varias maneras, y una de las más normales es la que puede verse en la figura siguiente[21], donde ya tenemos casi todo lo que se puede pedir.

Pues esto es todo. Ya sabemos como se meten en la memoria del ordenador los textos que queramos. Simplemente empleamos el código ASCII. En una serie de bytes de la memoria ponemos una serie de números (en realidad series de voltajes en grupos de 8, que representan números del 0 al 255[22] y los leemos con nuestro libro de claves. Es por esto por lo que la memoria se suele medir en bytes, pues cada uno de ellos representa habitualmente un carácter. Es también por esto por lo que anteriormente hicimos el ejercicio de la hoja de papel y por lo que identificamos una de sus cuadrículas con un carácter y éste con un byte. Vienen a ser lo mismo. Es muy importante que cuando se hable de bytes, Vd. piense en letras. Un byte representa la misma cantidad de información que una letra o, en general, un carácter, y para nosotros es mucho más comprensible de este modo.

Ahora que sabemos todo esto, volvamos al tema con que hemos empezado el apartado. ¿Y qué tiene que ver todo esto con que el ordenador se cuelgue?, tal vez nos estemos preguntando. Muy sencillo. El código ASCII es tan sólo uno de los muchos códigos que el ordenador maneja simultáneamente. Incluso para codificar textos existen varios más[23]. No cabe duda

---

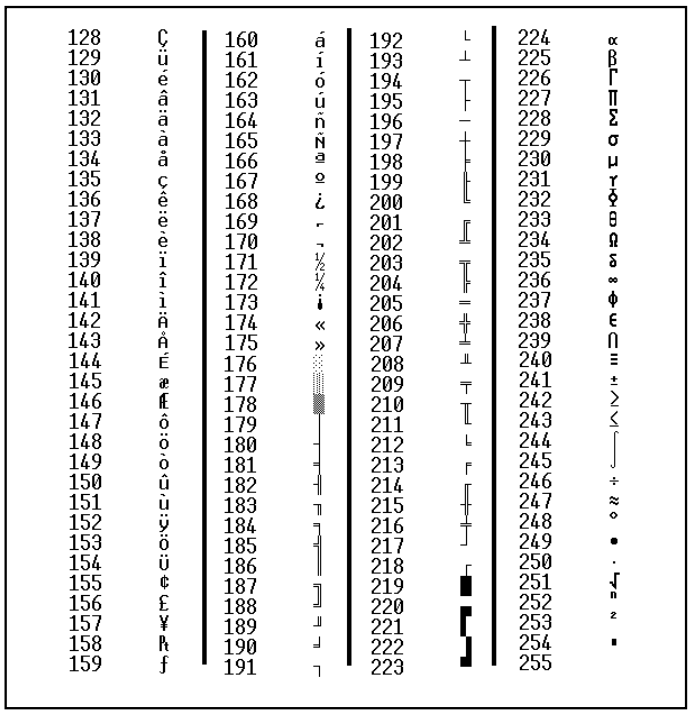
ejemplo.

19 Los caracteres de control tienen más que decir de lo que parece, y seremos medio amigos suyos a no mucho tardar.

20 Otro motivo es que cuando se creó se empleaban sólo 7 bits de los 8 que componen un byte. En realidad, el código ASCII es de 7 bits. El octavo bit se usaba para comprobar que el "intercambio de datos" se hubiera producido correctamente, o sea, que no hubiera errores. Se solía hacer mediante lo que se llama un "control de paridad", que no explicaremos. La mejora de los equipos hizo la cosa innecesaria, y el espacio libre se empleó provechosamente como se ha expuesto en el texto principal.

21 En un PeCé, esta zona se puede llenar de varias maneras, según definamos lo que se denominan "páginas de códigos". No entraremos en ello, ni ahora, ni cuando hablemos de los PeCés con más detalle. El ejemplo que se muestra corresponde a una de la llamadas "páginas de códigos internacionales" y es de uso muy corriente en nuestro país.

22 ¿Cómo hacemos para que un grupo de voltajes, o sea de s_eléctricas, represente un número?. En realidad es tan sólo una forma de verlo. Cambiamos mentalmente a código binario y miramos la memoria suponiendo que cada bit es un 0 o un 1 de un número en esta base. Es más general que el código ASCII, y a veces conviene verlo así. Diremos por tanto que, en esencia, las s_eléctricas en la memoria del ordenador representan números. No me voy a extender más sobre bases de numeración ni zarandajas similares, y si Vd. no entiende lo anterior, no se preocupe. No volverá a salir. Y si quiere más detalles, mire el glosario.



de que el ASCII es muy importante, yo casi diría que el principal de todos ellos, pero hay más.

Y los hay que no codifican textos. Otro que hace falta conocer obligatoriamente, el "código máquina", redefine por completo las tablas anteriores para almacenar en los mismos valores del 0 al 255 significados completamente distintos. Con él ya no son letras, sino instrucciones para la CPU. Los programas para que el ordenador maneje el teclado, por ejemplo, o para que haga cualquier otra cosa, estarán inevitablemente en código máquina. Y tanto los textos como las instrucciones para la CPU están todos a la vez en la memoria del ordenador, el único sitio, recuerde, en que la CPU los puede manejar con soltura. En la misma memoria aunque en diferentes posiciones, en diferentes cuadrículas, como tuvimos oportunidad de ver con el listado de la memoria que mostramos anteriormente. El ordenador distingue unas de otras porque las variables del sistema le dicen qué hay en cada sitio.

Pero si por casualidad una de estas variables cambia su valor, o escribimos un byte codificado en ASCII en una zona en la que el ordenador cree que tiene instrucciones en código máquina, pues pasa lo mismo que cuando intentamos leer un mensaje cifrado con una clave que no es la suya. Que el error es monumental. El ordenador, como cualquier humano, se

---

23 Por ejemplo, si tenemos un PeCé trabajando en Windows, no se emplea normalmente el código ASCII sino el ANSI, que es diferente. Si salimos de Windows, en el mismo PeCé, curiosamente, volvemos al ASCII. O tal vez, en algún ordenador, tengamos que enfrentarnos al código EBDIC. Quizá pronto Windows se pase a Unicode, que emplea dos bytes en lugar de uno para cada carácter. Y todos ellos son tan sólo distintas formas de codificar caracteres alfanuméricos.

confunde. Y como maneja 20 o 30 códigos diferentes simultáneamente dentro de la misma memoria, pues no es tan raro que estas cosas, por muy desagradables que resulten, ocurran. Por lo tanto, el que el ordenador se cuelgue, cuando es por motivos de software[24], no suele ser sino el resultado de una confusión sobre como tiene que leer un byte.

Terminemos el apartado pidiendo un esfuercillo para intentar recordar el máximo posible de lo que aquí se ha dicho. Es quizá algo abstracto, tal vez ligeramente difícil, pero muy importante. Hay que acordarse de que existe el bit porque tiene la costumbre de aparecer donde menos se piensa, y hay que acordarse de que el código ASCII existe porque, a pesar de todos los esfuerzos por sepultarlo y ocultarlo a los ojos del mundo, sigue siendo una herramienta fundamental en cuanto queremos hacer algo mínimamente raro con un ordenador. Tal como pasar datos, un texto por ejemplo, de un ordenador, o incluso de un programa, a otro. Es un código rústico pero útil y lo suficientemente potente, que todo el mundo entiende. Como su propio nombre indica, el "Standard Code for Information Interchange" sigue realizando un más que apreciable trabajo cuando se trata de intercambiar información.

Los buses de nuevo, por favor — *Aún más tripas de la bestia*

Ya sé que no es éste el capítulo, y esto va a ser tan sólo una pequeña digresión ya anunciada, pero ahora que sabemos lo que sabemos podemos entender más cabalmente que es eso de los buses e incluso intentar citar algunos más de ellos, ligeramente distintos a los que ya hemos presentado.

Habíamos dicho que los buses eran una especie de vía definitiva a la expansión del ordenador. Un sitio donde conectar con facilidad tarjetas para ampliar nuestro equipo. Bien, en realidad esto es únicamente una parte del total, aquella que resulta de someter a los buses en bruto a un proceso de estandarización. ¿Y qué son los buses en bruto?. Pues una característica básica de la arquitectura del ordenador.

Como ahora sabemos, los bits son la forma elemental en que se manipulan los datos en su interior y, físicamente, no son más que voltajes o corrientes en un circuito. Un bit necesita un alambre, un conductor donde estar, y los buses, en principio, no son más que los grupos de conductores por los que circulan los bits.

Si queremos manipular 4 bits a la vez, ponemos 4 hilos. El que lleguen 4 u 8 bits de vez a la CPU, que es la que tiene que procesarlos, es decir, que haya un bus de 4 bits o de 8 (de 4 u 8 hilos, si quiere), no es irrelevante en cuanto a la velocidad de operación del aparato. Todo lo contrario. La anchura de los buses de una CPU es un parámetro importantísimo en su rendimiento.

---

24 También es posible que la razón sea un fallo del hardware. Por ejemplo, un sobrecalentamiento de los circuitos, por la razón que sea, suele conducir inevitablemente a colgadas monumentales que, además, no se solucionan volviendo a encender el equipo a menos que lo dejemos enfriar un buen rato.

Hay CPU de 8 bits. Las hay de 16. Las hay de 32 y hasta de 64, y por supuesto, las más rápidas son aquellas que tienen los buses más anchos. ¿Le suena lo de "Tengo un ordenador de 32 bits"?. No es más que esto. Quiere decir que la CPU tiene buses de 32 bits, y es capaz de procesarlos todos de vez. Una CPU de 8 bits puede hacer las mismas cosas, pero necesita ir 4 veces a por los datos para coger 32 bits.

Si asociamos buses y velocidad del reloj (¿se acuerda?), podemos decir que al comparar la velocidad real de dos CPU, una de 32 y otra de 8 bits, con la misma frecuencia de reloj la primera será cuatro veces más rápida. El producto "frecuencia del reloj por anchura del bus" es un buen indicador de la velocidad de la CPU de un ordenador. Una vez más, la cosa es en realidad algo más complicada, y hay buses internos, de direcciones, y de datos[25], pero eso es otra historia.

Cuando estos buses primitivos del ordenador se intentan estandarizar para que cumplan su papel como vehículo de expansión, aparecen como no podría ser de otra forma buses estándar de distintas anchuras y con distintas frecuencias de reloj, con las mismas consecuencias que hemos citado. Los hay de 8, de 16, y de 32 bits[26]. Y los de 32 bits son capaces, evidentemente, de funcionar a mayor velocidad. Pondremos ejemplos concretos para PeCés algo más adelante. Hasta entonces.

La "personalidad" del ordenador — *El nombre de la bestia: Luis Ricardo*

Agotados por nuestra experiencia con el bit y el ASCII, quizá auténticamente exhaustos y a punto de la rendición total tras tener que volver a lidiar de nuevo con los buses y el hardware, tal vez no estemos del mejor humor. Vamos a ver si lo podemos arreglar con algo suavecito[27].

Nada mejor para esto que intentar penetrar con un poco de humor en la personalidad del ordenador, en como se comporta cuando trabajamos con él, momento al que nos vamos acercando inexorablemente.

---

25 Para los muy curiosos, el bus interno de una CPU es el conjunto de hilos que emplea para procesar datos. El bus de datos es el conjunto de hilos que emplea para coger y dejar datos de la memoria, por donde realmente circulan estos. Y el bus de direcciones es el conjunto de hilos que emplea para referirse a las distintas posiciones de memoria (para "direccionar" la memoria). No tienen porqué ser igual de anchos. Hay CPU con buses internos de 32 bits, de datos de 16 bits, y de direcciones de 24 bits. Por ejemplo el Intel 80386SX. Esto da lugar a una serie de limitaciones y peculiaridades que ignoraremos por ahora.

26 No hay que confundir el número de bits de un bus con el número de hilos que lo integran en realidad. Un bus de 8 bits (o de 16, o de 32) suele tener al menos 8 hilos, pero además hay otros, en número bastante variable, que se encargan de ayudar a que la cosa funcione. Son las "líneas de control", que no llevan datos pero hacen falta. Y trucos como la multiplexación permiten reducir el número de hilos necesarios para transportar un número determinado de bits. Como ve, la cosa puede liarse tanto como haga falta, si así se desea.

27 Bit ¡¡puaf!!. ASCII ¡¡bujj!!. Y además, por si fuera poco, vuelta a los buses. Quizá estemos pensando hace rato si no es posible hacerlo algo más distraído. Sí, por suerte, es posible.

¿Cómo se comporta la bestia?. ¿Qué podemos esperar de la personalidad del monstruo?. Si queremos dar una imagen gráfica, nada mejor que otro monstruo para ello. Ha llegado el momento de presentar a Luis Ricardo. Es necesario hacerlo pues ha pasado mucho tiempo desde que era un personaje popular, cotidiano, y por todos conocido. Sea. Presentémoslo pues.

Luis Ricardo, o el Monstruo de Sanchezstein, como tal vez los ancianos de la tribu recordarán, era el personaje principal de un antiguo programa concurso infantil de TVE, sin duda orquestado por los servicios de espionaje del gobierno con el fin de localizar precoces talentos informáticos de la época.

Al grito de "Luis Ricardo cantidubi, cantidubidubidá, ¡ya!", este simpático monstruo (caracterizado como un pequeño y afable Frankenstein e interpretado por un actor cuyo nombre lamentablemente no recuerdo y que un colega sensible y con mejor memoria que la mía cree que era José Carabias) obedecía sin discusión y literalmente las órdenes que se le daban, y demostraba las sutiles diferencias existentes entre "Gira a la derecha. ¡Ya!" y "Da un cuarto de vuelta a la derecha. ¡Ya!". En el primer caso, se ponía a dar vueltas a la derecha sin parar hasta que el desolado concursante dijera "Para. ¡Ya!".

El objetivo del concurso era conseguir, mediante órdenes similares a las anteriores, que Luis Ricardo realizara una tarea dada. Una tarea, además, que normalmente era muy simple a escala humana. Algo así como "Hay que ir hasta aquella puerta, abrirla, volverla a cerrar, y luego regresar al punto de partida". La realización de semejante estupidez daba mucho más juego de lo que los enterados de turno pudieran creer, y el programa resultaba muy entretenido.

Los problemas con que se enfrentaban los concursantes eran la literalidad del comportamiento de Luis Ricardo y su escaso número de habilidades innatas. No sabía "Ve hasta la puerta", sólo sabía "Da un paso adelante. ¡Ya!" y "Para. ¡Ya!". Luis Ricardo no tenía capacidad de juicio, iniciativa, tolerancia, ni base cultural alguna. No discutía, no pensaba, no añadía ni quitaba nada. Si lo entendía, hacía ni más ni menos que lo que se le decía. E inmediatamente. Y muchas veces, ¡ay!, no era esto lo que el desdichado concursante deseaba.

La razón de este comportamiento insospechado y la dificultad del juego radica en que, y no es broma, a través de la educación y la mayor o menor inteligencia de cada uno, los humanos tenemos mecanismos de comunicación muy sofisticados. Cuando un jefe le dice a su secretaria(o) algo así como: "Por favor, Pepa(e), envíe una carta de disculpa al individuo de ayer", está dando por conocido un 99% de la orden. La secretaria(o) debe saber cual es el modelo de carta, el de sobre, donde está la máquina de escribir, como se envía el correo, cual es el destinatario, y hasta cual es el nombre de su jefe entre otra multitud de cosas. Nada de esto está explícitamente en la orden. Hasta se podría hacer más corto: "Mande una disculpa al individuo de ayer", pero en esta ocasión el jefe, de naturaleza más bien sensible, prefiere aprovechar la antedicha eficacia de la comunicación para añadir: "Sé que podría ordenárselo, pero prefiero pedírselo, por educación y para que vea que la valoro, porque sé que Vd. funciona y lo va ha hacer pronto y bien". Todo muy complejo en realidad, muy sutil. Y muy poco informático.

Habituados, querámoslo o no, a este tipo de comunicación, no es extraño que los humanos tengamos problemas a la hora de tratar con el ordenador, en medida que depende asimismo del carácter, inteligencia, formación, e historia personal de cada uno. Pero bueno, ¿no

estábamos hablando de Luis Ricardo?. Si pero no, querido LAO, pues, como hemos dicho, la personalidad de ambos coincide absolutamente, y los problemas de comunicación que existían con Luis Ricardo son los mismos que hay que afrontar al manejar un ordenador[28].

De hecho, podemos decir que el objetivo del concurso era realizar un programa de ordenador sobre la marcha, utilizando, eso sí, un modelo muy avanzado. Nada menos que a Luis Ricardo[29]. Si ya tenemos un poco de visión intuitiva del asunto, quizá convenga precisar algo lo anterior y traducirlo de forma ligeramente más seria al campo de la informática.

Concretemos. Para que un ordenador haga algo hay que detallarle totalmente lo que debe hacer, descompuesto en órdenes elementales, en un lenguaje que él entienda, y, además, esas órdenes deben ser absolutamente correctas, pues el ordenador no las va a discutir y las va a ejecutar de inmediato. "Autodestrúyete". Pues bueno, pues ya está. Hala, a comprar otro[30].

Para poder mandarle algo a un ordenador hay que ser capaz de descomponer lo que queramos que haga de esta forma, y eso representa una tarea importante[31]. Gracias a Dios, con el ordenador es posible almacenar las cosas, y así, una vez hayamos conseguido que haga algo útil, si hemos guardado la serie de órdenes, no tendremos más que decirle que las lea de nuevo y las vuelva a hacer, algo mucho más sencillo. Los programas no son más que estas órdenes enlatadas listas para su uso, y gracias a ellos, en última instancia gracias a su capacidad de almacenamiento, los ordenadores son mínimamente útiles[32]. Ya que estamos, y como cualquiera se figurará, llamase "programar" a la tarea de realizar programas.

---

28 El problema del manejo del ordenador puede plantearse, y quizá en algún momento lo hagamos así, como un problema de comunicación, o, si se quiere, de lenguaje. Hay que saber como habla el ordenador para mandarle cosas, o él tiene que saber como hablamos nosotros para entendernos, o algo entre medio. Nada de ello, en mayor o menor cuantía, es tan fácil como pueda parecer.

29 Visto como ordenador, Luis Ricardo era ¡capaz de ser manejado de viva voz por cualquier concursante!. Funcionaba con un lenguaje de programación en modo interpretado, es decir que hacía las órdenes en cuanto se le daban. No se colgaba jamás, por si fuera poco. No debería hacer falta decir que no existe ordenador ni lejanamente parecido en este momento.

30 Por suerte, los ordenadores actuales incorporan un mínimo de mecanismos de seguridad que evitan estos potencialmente enojosos accidentes...

31 Esto hace que sea difícil mandarle al ordenador un montón de cosas interesantes que los humanos sabemos hacer, pero no tenemos muy claro como las hacemos. Para desarrollar un programa es necesario no sólo saber qué queremos hacer, sino también como, con todo detalle, tiene que hacerlo el ordenador. Y si no sabemos ni como lo hacemos nosotros.... Este es otro motivo por el que los ordenadores no hacen todo lo que querríamos que hicieran, y por el que difícilmente van a ser capaces de hacerlo nunca.

32 Un ordenador sin mecanismos de almacenamiento no podría hacer esto. Sin periféricos de almacenamiento, un ordenador es una lata. Aprovecho la ocasión para llamar su atención sobre la importancia, nunca exagerada cuando de ordenadores se trata, de guardar el trabajo que con él se realice para el futuro.

Para terminar este filosófico apartado, nada mejor que introducir otra filosófica cuestión, también fundamental, y que será otro de los hilos conductores del texto en muchos momentos por venir.

Hemos dicho que el problema con Luis Ricardo puede verse como una cuestión de comunicación[33]. Podríamos decir que su raíz está a su vez en algo más profundo, puesto ya de manifiesto cuando hablamos del bit y el código ASCII. El ordenador era un chisme que procesaba información, datos. No hemos dicho en ningún momento que procesara significados. Y ésta es una diferencia fundamental con los humanos que, en mi modesta opinión, tendemos a funcionar justo al revés. Procesamos más significados que otra cosa. Un dato vale de muy poco si no quiere decir nada, y un humano se permite habitualmente el lujo de no recordar con gran precisión la información. Basta que se acuerde más o menos de ella y de lo que quería decir. En cambio, en un ordenador, los significados no existen como tales. Son siempre algo que le es impuesto externamente a los datos[34]. Algo que nosotros añadimos cuando los vemos. El ordenador jamás sabe qué quieren decir las cosas que maneja, e incluso los códigos no son más que una manera de indicarle como nos tiene que presentar la información para que a nosotros nos sea más fácil interpretarla, "leerla" al fin y al cabo[35].

Estas dos enormes limitaciones, la excesiva necesidad de detalle y la incapacidad de procesar significados, hacen que sea bastante difícil que se logre hacer que un computador funcione de modo "inteligente"[36], con discernimiento, raciocinio, y otra serie de cosas que podemos pedirles a algunos humanos.

Pues bien, por aclararnos y dejarlo establecido para lo que queda de libro, éste es el funcionamiento que podemos esperar de un ordenador, ésta es su personalidad, y así hay que aceptarlo, sin más. Podemos decir, en definitiva, que el ordenador tiene una maravillosa memoria gracias a la cual nada se le olvida, y que hace lo que le mandemos, muy deprisa, sin discutir, sin rechistar, y sin descanso. Pero es un zoquete y un tarugo en lo que a los significados se refiere, y hay que mandarle, además, las cosas absolutamente masticadas, con un detalle insultante. Viene a ser, al parecer, como uno de esos "idiotas sabios" con grandes habilidades pero para un campo muy limitado. Tratar con ellos, en caso de ser similar al trato con el ordenador, debe ser más bien desesperante.

---

33 No es tan raro esto. Los problemas de comunicación son universales y universalmente graves. ¿Nunca le ha dicho a Vd. la prójima, con profunda cara de duelo, algo así como: "No me entiendes, José Ramón"?.

34 Es habitual, de hecho, que el que haya construido un programa haya metido una serie de códigos en él, es decir que le haya dado un significado a los datos. Todo programa define o usa estructuras de datos con significados específicos. Esto no cambia la cosa ya que siguen siendo significados impuestos desde el exterior aunque ahora almacenados previamente, que el ordenador en realidad ignora y que nosotros, si queremos, podemos obviar o cambiar por otros.

35 En realidad, esto mismo podría decirse de cualquier otro soporte de información (este texto, esta página, por ejemplo), pero la variabilidad de los códigos en el ordenador hace que en ellos el problema se manifieste, mientras que en muchos otros casos queda eterna y piadosamente oculto.

36 El tema de la inteligencia artificial, como se llama a esto, requiere bastante más atención. Saldrá tangencialmente en algún otro momento, y hay una nota en el glosario.

Las ventajas del ordenador... *Las habilidades de Luis Ricardo*

Bueno, sabemos como es Luis Ricardo. Vayamos a lo positivo. ¿Cuales son las ventajas de trabajar con él?. En realidad la pregunta se puede contestar bastante rápidamente diciendo que las que se derivan de trabajar con una máquina dotada de memoria, capacidad de almacenamiento, programable, y rápida. Si le está pareciendo que trato de escurrir el bulto, nada de eso.

El que tenga una memoria (RAM) implica que aquello que nosotros produzcamos, tal como una carta, un dibujo, una contabilidad, o una serie de fichas con los datos del personal de nuestra fábrica, estará dentro de la citada RAM al menos mientras estamos trabajando en ello. En el ordenador no trabajamos directamente con las cosas que queremos hacer, sino con una imagen suya en memoria. Y las podemos modificar con enorme facilidad y mínimo coste. Pasar una línea de texto de un sitio a otro. Suprimir todo un párrafo, o añadir dos palabras. Como no está en papel, simplemente cambio la imagen de mi carta. Cuando me guste del todo, esté completamente como yo quiera, ya la pasaré a papel en la impresora. La capacidad de realizar modificaciones fácilmente es lo que aporta la memoria, y la primera gran ventaja.

El que tenga capacidad de almacenamiento tampoco es despreciable. Cuando yo escribo una carta con una máquina de escribir empleo digamos una hora, invertida ante todo en mecanografiar. En lenguaje informático, en introducir los datos de mi documento. Si tengo que escribir otra carta a otra persona, con el mismo modelo, emplearé otra hora más en volver a meter todos los datos. Con un ordenador me basta almacenar la primera carta, que me ha costado también una hora. La recupero de donde la tenga almacenada, probablemente un disquete, modifico las diez palabras que son distintas, cosa que es posible con facilidad por el punto uno expuesto anteriormente y, Voilá!, carta número dos completa. Tiempo de producción de dos cartas: una hora+un minuto (recuperar)+cinco minutos (modificar). La cosa marcha. Más cuando podemos almacenar cualquier cosa e incluso alguien puede hacerlo por nosotros pasándonos, vendiéndonos, o dejándonos acceder a sus datos, con lo que ni siquiera tendríamos que crearlos desde cero la primera vez. Segunda ventaja, los periféricos de almacenamiento aportan la capacidad de reutilizar los datos. La capacidad de realizar modificaciones de la RAM, asociada al almacenamiento de los datos para su reutilización, son dos ventajas enormes y complementarias.

Por su parte, el que el ordenador sea programable, nos permite poner un digno colofón. Tenemos los datos. ¿Porqué no hacer que sea el ordenador el que haga las cosas?. Si hay que imprimir cinco mil cartas iguales, y tenemos por un lado nombres y direcciones, y por otro el modelo a emplear, no estaría mal que las montara él solito. Cualquier procesador de textos moderno, un programa al fin y al cabo, nos permite hacerlo. Metemos el modelo. Una hora. Metemos los nombres y direcciones. Otra hora. Se lo mandamos. Tres segundos. Dos horas más para hacerlo, pero que las trabaja el ordenador y yo me voy a tomar cervezas. Cinco mil cartas en cuatro horas de las que yo he trabajado dos. No está nada, pero que nada mal, sobre todo porque el ordenador no va a meter la pata equivocándose al teclear, y el modelo va a ser exactamente igual en todas las copias, sin errores de mecanografiado. Además, el ordenador no protesta, no se cansa, y tiene una maravillosa memoria que hace que nada se le olvide. Los trabajos repetitivos, enormemente aburridos y propensos a montones de errores, se le dan muy bien. Podemos hacer por ejemplo que esté diez días seguidos revisando un fajo de fichas enorme para clasificarlas e imprimir todas aquellas que empiecen por “P”. Lo hará y no dirá esta boca es mía. Y no se dejará ninguna.

Y no sólo es capaz de hacer cosas idiotas pero latosas, lo que ya estaría francamente bien. Basta que alguien haya programado algo para que, con el programa, yo pueda hacerlo también. Yo no necesito saber como hacer una contabilidad. Si tengo un programa que lo haga, hecho por algún experto en el tema, sólo tengo que poner los datos. El ordenador, empleando el saber de otro, lo hará. En este aspecto, el ordenador puede proporcionarme nuevas habilidades. La capacidad de realizar modificaciones de la RAM, asociada al almacenamiento de los datos para su reutilización y a la capacidad de programación son las tres ventajas, enormes y complementarias, que el ordenador pone a nuestro alcance.

Y si a lo anterior añadimos la velocidad de proceso, todo junto, nos permite realizar simulaciones. Esto es, probar alternativas cuyo resultado se calcula automáticamente, que yo veo, juzgo, y me quedo con la que me interesa. Cuando cambio una línea de sitio estoy haciendo una simulación, viendo como quedaría mi texto con la frase aquí en lugar de allá. O puedo ver si me ahorraré dinero haciendo mi declaración de renta individual en lugar de hacerla conjunta. El ordenador calcula los resultados, me los muestra, y yo elijo. Hace falta la rapidez porque debe mostrarlos de inmediato. Si cada modificación necesita de dos horas para recalcularse, apañados estamos. Las simulaciones se pueden hacer difícilmente a mano. Como mucho, uno le puede echar imaginación al asunto e intentar suponer como quedaría. El ordenador lo calcula completamente, con todo detalle, y me lo enseña.

Si queremos ponerlo de otro modo, por si quedan dudas, con el ordenador podemos evitar cualquier repetición del trabajo y hacer que, si se lo sabemos mandar, el ordenador haga con nuestros datos lo que queramos, incluso cosas que nosotros no sabemos exactamente como hacer o que nos llevarían mucho tiempo. Hay, eso sí, que sacar datos y órdenes de algún lado. El ordenador no crea información. La transforma.

Y sus inconvenientes... *Las limitaciones de Luis Ricardo*

En primer, segundo, y tercer lugar, las tres primeras ventajas anteriores pueden convertirse en problemas. Como muchas otras cosas en este mundo cruel. Trabajar con datos en memoria lleva a que la mayor parte del tiempo tengamos más borradores que otra cosa. Poder modificar sin talento hace que corramos el peligro de no llegar nunca a nada definitivo, terminado. Con un ordenador, el 90% de las veces las cosas sólo están acabadas cuando tenemos una copia en papel. La impresora se puede romper en cualquier momento, o se nos puede acabar la tinta un viernes por la noche dejándonos tirados hasta el lunes, o podemos perder accidentalmente nuestros datos.

Podemos construir un almacén de datos enorme, pero esto hace necesario que nos preocupemos de aprender como construirlo, como mantenerlo, y como usarlo. Hay que meter algo de tiempo sólo en tener los datos localizados, ordenados, listos para su empleo inmediato. Saber que tengo algo, si no sé dónde está ni como recuperarlo, no sirve más que para que tengamos un buen motivo para pegarnos cabezazos en una pared por torpes. Podemos, finalmente, mandarle cosas al ordenador. Pero hay que saber como. Y eso requiere aún más aprendizaje y cierta habilidad que no todo el mundo posee.

Si seguimos con nuestro balance de tiempos, los que hemos citado antes son suponiendo que sabemos hacer las cosas. Las primeras veces que lo intente, la cosa no será así. Vd. va a perder un montón de tiempo aprendiendo a hacer que todo marche. La primera vez que intente reutilizar una carta, va a emplear no dos, sino diez horas. Eso sí, la segunda vez, ya

lo hará en la hora y seis minutos. Más o menos lo mismo, incluso algo peor, para las 5.000 cartas a base de mandárselo al ordenador. Y aunque Vd. no necesite conocer con todo detalle como hacer una contabilidad, sí que es mucho mejor que tenga una idea general bastante completa. Simplemente para poder entender los resultados que el ordenador le proporcione. Recuerde que no manejaba significados. Los significados los debe poner el usuario. Hay que partir de la base de que el ordenador sólo es rentable a medio y largo plazo, cuando ya se han adquirido los conocimientos y se han almacenado los datos necesarios para hacer que nos devuelva la inversión en tiempo y esfuerzo que deberemos meter en él al principio.

Hay algún problema más, que podemos intentar hacer ver comparando el trabajo a realizar para escribir un texto en el ordenador y en una máquina de escribir. Con la máquina, lo que hacemos es ir imprimiendo sobre la marcha lo que pensamos. Vamos produciendo un documento absolutamente real carácter a carácter. No es fácil corregir, no tenemos más que un tipo de letra, no podemos meter gráficos en el texto (habría que dejar un hueco y luego pegarlos tras sacarlos de otro sitio), no podemos reutilizar los datos, ni hacer que la máquina escriba sola. Pero todos tenemos claro como funciona. Es muy evidente. O como se gusta de decir en informática, intuitivo. Que, traducido, quiere decir que todo el mundo sabe como hacerlo.

Con el ordenador, en cambio, construimos la imagen de una página, que imprimimos completa cuando terminamos, no letra a letra. Sólo al imprimir tenemos un documento real. Podemos poner gráficos. Tenemos distintos tipos de letra que el ordenador sabe como manejar pero sobre los que tenemos que decidir cual emplear, en que tamaño, si de espaciado proporcional o fijo. Podemos hasta hacer que el ordenador compruebe la ortografía, para evitar que el resultado final tenga errores. Podemos guardarlo para otra vez. Pero todo eso exige que sepamos que es posible, qué significa que lo hagamos, y como mandárselo al ordenador. No es tan inmediato ni intuitivo como la máquina de escribir. Podríamos quitar estas posibilidades, y hacer que haga exactamente lo mismo que nuestra vieja máquina y sea igual de intuitivo. Al fin y al cabo para algo es programable. Pero a cambio de perder todas las ventajas que aporta.

El problema del empleo del ordenador y la necesidad de adquirir nuevos métodos de trabajo surge quizá como fruto de una ¿sutil? combinación entre el intento de hacerlo lo más parecido posible a la herramienta que sustituye, su diferencia intrínseca con ella, y la necesidad de incorporar sus nuevas posibilidades, lo que aumenta la complejidad de la tarea en sí misma.

Desarrollo de programas — *Enseñando a Luis Ricardo*

Ya sabemos que para que el ordenador haga alguna cosa hay que suministrarla los programas correspondientes. Ya sabemos que los programas son instrucciones sumamente detalladas y en un lenguaje que el ordenador pueda entender, almacenadas para ponérselas pasar cuando sea preciso.

Si estudiamos el proceso que se sigue en realidad para producir y vender programas industrialmente[37], aprenderemos un buen montón de cosas más. Vamos a ello.

---

37 Porque, ésta es otra, al menos un 70% de los programas que empleemos en un ordenador son

Para hacer un programa, lo primero es saber qué queremos que haga el ordenador. El proceso comienza estudiando las necesidades de nuestros potenciales clientes y viendo qué es lo que más necesitan o lo que vamos a vender mejor. Trabajo que llevan a cabo los expertos en marketing o los responsables de la línea de productos de nuestra empresa editora de software. Una vez decidido qué hay que hacer[38], entra en juego el primer oficio auténticamente informático. Un analista, algo así como un arquitecto de la programación, se encarga de ver la mejor manera de hacerlo con un ordenador y del diseño general. También de la supervisión del proceso de construcción del programa y de la introducción de las modificaciones que hagan falta sobre la marcha.

¿Quién se encarga de poner las cosas en un lenguaje que el ordenador entienda?. El o los programadores. A escala industrial, un programa es desarrollado habitualmente por un equipo controlado por un analista. Ellos son los albañiles que hacen el edificio. Un programador no suele hacer nada más que partes del programa, que son ensambladas por programadores más expertos o analistas no principales, algo así como capataces de obra o aparejadores. También los analistas saben programar. Son normalmente programadores muy experimentados y con conocimientos adicionales.

Bien, en algún momento, se obtiene algo lejanamente parecido a lo que se quería. A diferencia de lo que ocurre al construir un edificio, un programa no se termina de una vez. La primera versión no funciona nunca. Está llena de errores que hay que quitar. A esta primera versión, aún de desarrollo y no comercial, se la llama "versión alfa". Es como si nuestro edificio nos hubiera quedado torcido y lleno de agujeros. Hay que arreglarlo.

Se pasa entonces a emprender una serie de pruebas y modificaciones destinadas a corregir los defectos más graves y eliminar los errores más profundos. Lo que se denomina la depuración del programa. Enderezamos el edificio, tapamos los agujeros más descomunales, y tenemos un programa en "versión beta". Esta es una versión aún no definitiva, inestable (se cuelga demasiado, algunas cosas no funcionan), todavía con demasiados errores y que puede sufrir mejoras significativas, pero que ya es casi operativa.

Vuelta a depurar, empezando otro ciclo de pruebas, en el que se puede solicitar el empleo del programa por gente de fuera de la empresa (usuarios cualificados, programadores externos o similares), y tras un período de tiempo más o menos dilatado, se eliminan los errores importantes y se llega a un producto vendible.

Nuestro programa, al que vamos a llamar "The last word"[39], o "TLW" por abreviar, se lanza al mercado bajo la denominación "The last word v1.0". Lo de v1.0[40] quiere decir que

---

desarrollados por empresas que hacen negocio con ello.

38 Pongamos por ejemplo que facilitar la producción de documentos escritos. Los programas que hacen esto son los denominados "procesadores de textos", como Vd., querido LAO, sin duda sospechaba. Podría ser cualquier otra de mil cosas. La ya citada contabilidad, la declaración de renta, el control del tráfico en una gran ciudad, la creación de un sistema operativo nuevo para un tipo de ordenador dado...

39 Traducción: "La última palabra". Algo tétrico, pero podría valer para un procesador de textos.

40 Aprovecho la ocasión para comunicar que los países anglosajones tienen la peculiar costumbre

es la primera versión de un producto que se desea mantener mucho tiempo a la venta y que se va a ir poniendo al día periódicamente.

¿Está nuestro "TLW 1.0" libre de errores ya?. Desgraciadamente, no. Es incluso muy posible que aún le queden errores gordos. Tan gordos que los comentarios que llegan después de las primeras ventas de verdad, con pasta de por medio, son considerablemente hirientes y llegan a poner en duda la honorabilidad de los empleados de nuestra editora de software. Incluso la de sus parientes. Para detener el desastre, nuestra empresa, que no ha cesado de trabajar en el programa, decide lanzar rápidamente una versión corregida y ligeramente aumentada. La denomina "versión a", con lo que nuestro producto es ahora el "TLW 1.0a"[41]. Y ya no se cuelga mucho, funciona razonablemente, y se vende casi sin efectos secundarios. ¿Y los pobres desgraciados que compraron la versión 1.0?, tal vez nos estemos preguntando. Con el fin de conservarlos como clientes, y porque realmente la culpa es suya al vender algo en malas condiciones, nuestra empresa decide cambiarles gratis su versión vieja por la nueva.

A partir de aquí, "TLW" emprende una dilatada carrera llena de éxitos. Se suceden las versiones a lo largo del tiempo. Cuando se introducen modificaciones menores, que no afectan al diseño general, se cambia la parte decimal del número de versión. La versión 1.0a es sustituida por la 1.1. Cuando se realizan cambios importantes, se cambia la parte entera. Aparecen así las versiones "2.0", "3.0", "3.1", "3.1a", "3.2" (la versión 3 fue una auténtica joya, muy adelantada a su tiempo, gran éxito de ventas, y sólo tuvo que ser mínimamente puesta al día un par de veces antes tener que sustituirla), "4.0", "5.0"... y podemos seguir indefinidamente[42]. Durante su vida útil, un buen montón de usuarios lo emplean en sus ordenadores y producen y manipulan sus textos gracias a él, admirando cotidianamente su potencia, solidez, y facilidad de uso. Nuestra empresa se hace, con toda justicia, de oro, y reparte con prodigalidad sus abundantes beneficios entre sus accionistas y, ya puestos, sus empleados.

¿Qué lecciones podemos sacar de este hermoso cuentecito, absolutamente basado en la vida real?. En primer lugar, que un buen programa es algo con una larga vida, sometido a una evolución constante. Un programa, en realidad, no se termina nunca. Simplemente, cuando es preciso, se retira del mercado. Mientras tanto se le está modificando constantemente, actualizando, corrigiendo. Es éste un hecho bastante diferente al que se da en la producción de herramientas. Allí se funde un martillo y ya está. Nos quedamos con el modelo para una larguísima temporada. Hacer moldes es caro, lleva mucho tiempo, y las pruebas son muy costosas. El proceso de producción del software es muchísimo más barato. No cuesta casi nada hacer una modificación, probarla, ver si funciona, y si lo hace, incorporarla al producto

---

de separar los decimales con un puntito, el mismo que a nosotros nos sirve para marcar los millares. Lo anterior debería leerse por lo tanto "TLW 1'0", pero esto de la numeración es otra de las cosas que jamás se traducen.

41 Un antiguo chiste electrónico-informático sostiene que ningún producto está realmente terminado hasta su versión "a". Una postdata de algún escéptico llega a mantener que un programa quizá no lo está ni siquiera en su versión "b", caso de existir.

42 Algunos programas de la vida real van por su versión 12 o más.

final[43]. Una gran ventaja del ordenador era su flexibilidad, y ésta es posible por la facilidad de modificación del software.

Y en segundo lugar, conocer los oficios informáticos. A saber: analista, programador, y operador de consola. No hemos dicho que era esto último, pero es fácil. El rimbombante "operador de consola" designa simplemente a aquel que usa un programa para meter datos en el ordenador. Es el más humilde menester y no exige más saber informático que conocer el funcionamiento de un programa y, muy probablemente, algo de mecanografía[44]. En el caso del usuario de un ordenador personal, es preciso aclarar que Vd., querido LAO, va a poder ejercer todos ellos, hasta el de responsable de la línea de productos, y en ocasiones simultáneamente. Y que, quiéralo o no, al menos va a ser Vd. el operador de consola de su ordenador. Por cierto, ¿que tal anda de mecanografía?.

Es vital conocer la escala de tiempo en que nos movemos. Desde que se anuncia un programa importante y se comienza a trabajar en él hasta que está a la venta pueden pasar perfectamente entre uno y dos años. La aparición de la "versión a", en caso de existir, puede tardar unos seis meses más. Las sucesivas versiones suelen salir cada año, más o menos.

¿Cómo tiene que ser un buen programa?. La preguntita se las trae, es una de esas preguntas del millón que requieren en realidad algunas decenas de páginas para empezar a incarles el diente, y eso es lo que haremos mucho más adelante. Por ahora nos conformaremos con exponer una retahíla de características aceptadas universalmente como deseables. Pequeño, para que no ocupe mucho sitio. Rápido, para que su empleo sea agradable y produzca resultados a buen ritmo. Sólido, esto es que no se cuelgue nunca, para evitar un buen montón de problemas. Coherente, es decir, que reaccione de forma previsible a las diferentes operaciones y evite sorpresas y una problemática desorientación en los usuarios. Fácil de empleo. Bien integrado en un ordenador, utilizando los modos de operación habituales en el resto de los programas que probablemente estamos usando y sacando partido de los recursos del equipo. Todo alejamiento de este ideal crea problemas al usuario en mayor o menor cuantía.

Puede que Vd. opine que operador de consola es lo único que va a hacer y desea ser, pero en un ordenador personal es bastante probable que, de una u otra forma, acabe haciendo algo similar a programar. El porqué diantres desearía Vd. complicarse la vida de ese modo es algo que ya hemos insinuado varias veces, pero ahora mismo hay que saber qué mecanismos lo hacen posible[45].

Un programa podría controlar directamente el funcionamiento de todos los componentes del ordenador. Al construir nuestro "TLW v1.0" podríamos incluir en él las instrucciones

---

43 Idealmente. La cosa no es tan fácil en la práctica, pero podemos dejarlo así.

44 De nuevo esto es la teoría. En mi opinión no está de más saber alguna cosa adicional de ordenadores, principalmente los conceptos generales que estamos intentando exponer aquí. Lo de la mecanografía es absolutamente necesario en el 90% de los casos. Sólo se salvan un poco aquellos que emplean programas de diseño gráfico.

45 El caso antes citado de creación automática de una serie de cartas puede verse como un ejemplo de programación, si bien de realización muy simple. ¿Le basta con esto?.

necesarias para manejar la memoria y gestionar los discos y el teclado, por ejemplo. Hacerlo así requeriría un importante conocimiento de las tripas del chisme, complicaría el programa, y lo haría muy complejo. Muy poca gente podría hacerlo. Por suerte, el sistema operativo está ahí para algo más que para poner el ordenador en marcha. En realidad, este invento se encarga de aislar el hardware y ofrecerle al usuario unos servicios que lo hacen no sólo útil sino también más fácilmente programable. El ordenador que nosotros vemos y podemos manejar es aquel que los programas que estamos ejecutando nos dejan ver. Es posible programar porque podremos decirle al ordenador que escriba un fichero (ya veremos qué es eso muy pronto) y algún programa adicional, en este caso el sistema operativo, sabrá como hacerlo. Mucha más gente puede de este modo desarrollar programas perfectamente válidos.

Aparece así el concepto, enormemente importante, de "ocultamiento de la información[46]". Como el ordenador es muy complejo, se va envolviendo sucesivamente en distintas capas de software que dejan ver tan sólo aquellos detalles que puedan interesar, y ocupándose en la sombra de todo lo demás. Se manejan sólo esos detalles, se emplean sólo esas puertas de acceso, y se consigue de esta forma hacer cosas que no serían posibles si hubiera que saberlo todo con todo detalle y encargarse de todo. Las capas de software van modificando el aspecto de Luis Ricardo y enseñándole distintas habilidades que son las que, en realidad, se utilizan.

La primera de estas capas, la que envuelve al hardware lo suficiente como para hacerlo utilizable, es el sistema operativo. Sin él, hacer cualquier programa es casi imposible, y es por eso por lo que en todo ordenador funcional debe haber uno corriendo en la sombra. Los programas se construyen por lo tanto no para un hardware, sino para un sistema operativo, y necesitan que éste se esté ejecutando para poder ejecutarse a su vez. Dependen de él por cuanto manejan el hardware a través de los recursos que éste pone a su disposición. Es el sistema operativo, y no el hardware, el que define el funcionamiento del ordenador. En consecuencia, lo unifica. Varios ordenadores, con diferente hardware pero corriendo el mismo sistema operativo, son funcionalmente iguales entre sí.

Sobre esta primera capa podemos poner muchas otras, civilizando cada vez más nuestro ordenador y alejándonos progresivamente de la aridez del hardware y sus señales eléctricas. Si lo hacemos, como suele ser habitual, nuestro hipotético programa dependerá por supuesto de la presencia de todas ellas para poder funcionar. Terminemos, para calmar cualquier posible ansiedad, diciendo que la programación que un usuario normal pueda terminar haciendo será sobre un ordenador equipado de un buen montón de capas intermedias, y mucho más fácil de lo que hasta ahora pueda parecer.

Modos de funcionamiento — *Luis Ricardo se multiplica (o no)*

El sistema operativo se está revelando como lo que es, una parte importantísima del invento informático que estamos intentando explicar. No sólo ha puesto el ordenador en marcha sino que hemos visto que va a definir como funciona nuestro chisme. Bueno, pero una vez

46 En inglés "information hiding". Teóricamente, su deber es facilitar la vida del usuario del ordenador. Tan noble propósito no es, lamentablemente y como casi siempre, tan fácil de lograr como pudiera pensarse.

en funcionamiento todos chutarán más o menos igual, quizá pensemos. Si y no, ya que aparecen importantes diferencias en función del que pongamos. Podemos elegir el sistema operativo para un ordenador, cambiando del modo más profundo posible su funcionamiento, del mismo modo que podemos elegir cualquier otro tipo de programa. La única limitación es, simplemente, que exista para nuestro hardware.

Según el sistema operativo que carguemos, el ordenador funcionará de varios modos que conviene conocer. El más simple es el monousuario-monotarea. Quiérese decir con esto que los recursos del ordenador se dedican en su totalidad a una sola persona (usuario[47]), y a hacer una sola cosa a la vez (tarea). El ordenador está siempre ocupándose de nuestros deseos y de hacer lo que le hemos mandado. Cuando acaba, nos deja pedir otra cosa.

Ya hemos visto que un ordenador hacía muchas cosas por segundo, era muy veloz. Dedicar toda su atención y recursos a una sola cosa puede parecer algo así como desaprovechar una potencia muy respetable. Si el ordenador tiene que estar diez minutos esperando que a nosotros se nos ocurra la frase que vamos a escribir en una carta, son diez minutos que no ha hecho nada. Por esto, hace ya tiempo que aparecieron sistemas operativos que permiten trabajar bien con varios usuarios a la vez o haciendo varias cosas a la vez[48].

Aparecen así los sistemas multiusuario o multitarea. Un sistema trabajando en modo multiusuario deja que varias personas, equipada cada una de su teclado y su pantalla[49], compartan una única unidad central[50]. Esta última se "multiplica", y parece como si cada usuario tuviera su propio ordenador. Tal milagro aparente no tiene nada de maravilloso. La cosa consiste en que el sistema operativo emplea los tiempos muertos de un usuario en atender a otro. Va pasando de uno a otro tan deprisa que no se nota.

El mismo mecanismo que permite que el ordenador atienda a varios usuarios puede hacer que atienda a uno sólo pero le deje hacer varias cosas a la vez. Un sistema multitarea reparte su trabajo entre varios programas, cargados a la vez en la memoria, que el ordenador va ejecutando simultáneamente. Si tenemos un sólo usuario, hablamos de un sistema monousuario-multitarea. Si, además, dejamos que varias personas accedan a la vez, tenemos un sistema multiusuario-multitarea. Podemos así, mientras el ordenador hace algo largo y tedioso, continuar usando el ordenador para otras cosas. Podemos, por ejemplo, dejar que reciba un fax a través de la línea telefónica, o compruebe si un programa que yo he hecho (larguísimo) es correcto, mientras estamos redactando una carta o comprobando una contabilidad.

---

47 También se dice en ocasiones "puesto", en lugar de usuario. Es lo mismo.

48 El aprovechar mejor los recursos es una muy pequeña parte de la justificación de este tipo de funcionamiento. Otras, entre muchas, son cubrir mejor las necesidades de comunicación entre equipos, incluso entre programas dentro de un mismo equipo o, simplemente, permitir una operación más cómoda, segura, y potente.

49 Un terminal llamábamos a esto. Lo que un usuario necesita para comunicarse con un ordenador.

50 Por si no se acuerda, en la unidad central estaban la CPU, la memoria, y probablemente los periféricos de almacenamiento. Las tripas del chisme, en resumen.

Un tipo de sistemas multiusuario eran los antaño llamados "de tiempo compartido", y estuvieron muy de moda. Fueron el primer intento de popularizar la informática. Cualquiera con un terminal se podía conectar, incluso a distancia, y se le facturaba el importe del tiempo que usara el ordenador. La CPU y la memoria eran cosas tan caras que sólo se podía tener una para muchos. Los microprocesadores revolucionaron el panorama, y aunque todavía existen los sistemas de este tipo, su interés ha quedado reducido a aquellos casos en los que se quiere acceder a ordenadores tan potentes que no son económicos a escala personal. Se siguen empleando, por lo tanto, dentro de grandes instalaciones informáticas (bancos, grandes empresas, centros de cálculo, acceso a superordenadores, y demás etcéteras enormes). De la popularización se ha encargado la microinformática, donde cada usuario tiene un ordenador barato para él sólo.

Hay tantos microordenadores en la actualidad, que coordinarlos entre sí se ha convertido en una necesidad. El propósito al final es similar al de una instalación multiusuario, tener a un buen montón de gente trabajando a la vez con una instalación informática común en la que se puedan compartir datos, programas, recursos (impresoras por ejemplo), y comunicarse, pero el modo de lograrlo es justo el inverso. En lugar de compartir una única unidad central, lo que hacemos es conectarlos de modo que uno pueda acceder, por ejemplo, al disco duro o a la impresora del otro. Aparecen así las redes, y el que funcionen depende, una vez más, de que pongamos un sistema operativo adecuado[51].

El último tipo de funcionamiento interesante de conocer es lo que se denomina "tiempo real". Decimos que un ordenador trabaja en tiempo real cuando es capaz de detectar algo, calcular sus efectos, y si es necesario actuar para evitarlos antes de que se produzcan[52]. La cosa puede ser tan banal como controlar las existencias de un almacén con reposición mensual de modo que no se nos agote ningún suministro. Haciendo pedidos mensuales no hay que esforzarse mucho para que la cosa funcione. El término se suele aplicar, sin embargo, a situaciones más comprometidas. Hacer que un ordenador controle cuando un avión se va a caer para que automáticamente manipule los mandos y lo evite. Permitir que entre el momento en que oprimo el disparador de una cámara fotográfica y el momento en que se abre el obturador, la cámara decida la exposición adecuada para mi fotografía. Detectar una avería en una planta de proceso químico y evitar que se produzcan vertidos tóxicos al exterior. Conseguir (no es un sistema informático, pero la idea es la misma) que desde que me doy el bofetón con un coche hasta que me llegan los golpes, el cinturón de seguridad se tense y el airbag se hinche para evitar en lo posible el chandrío. Hay sistemas operativos que hacen que el ordenador trabaje de esta manera.

Hasta aquí, todo es muy bonito. Una llamada de atención para terminar. No basta con poner un sistema operativo para que cualquier ordenador se convierta en un "multiusuario-multitarea-tiempo real" plausible. Si ponemos doscientas personas a compartir los recursos de

---

51 Para que un sistema trabaje en red, puede ponerse un sistema operativo adecuado. También es posible cargar programas adicionales (extensiones) en un sistema operativo monousuario. No siempre está claro el límite entre ambos enfoques, especialmente para PeCés. Lo mismo puede decirse, también para PeCés, en el caso de la multitarea o el multiusuario.

52 Si quiere una definición más formal, se dice que operamos en tiempo real cuando el tiempo requerido para el procesado de la información es inferior al tiempo de respuesta del sistema que estamos controlando.

un ordenador, la cosa empieza a fallar. Cada usuario sigue trabajando como si tuviera un ordenador completo pero muy, muy lento. Hace falta que el chisme tenga la potencia suficiente. Y esto se traduce en una CPU superpotente, cantidades ingentes de memoria RAM y de capacidad de almacenamiento, y algunos otros requisitos. Tiene que haber un mutuo ajuste entre el tipo de funcionamiento que deseamos, el sistema operativo a emplear, y el hardware en el que se ejecuta[53].

Modo texto y modo gráfico — *Luis Ricardo linotipista o Luis Ricardo pintor*

Podemos hacer que el ordenador nos muestre la información en forma de texto o gráfica. Que lo hagamos de uno u otro modo tiene muchas más consecuencias de lo que se pueda pensar, es una distinción fundamental en realidad, y, por tanto, hay que dedicarle algo de atención.

El modo texto consiste simplemente en que la pantalla del ordenador[54] está formada por un determinado número de celdas, digamos 25 líneas con 80 caracteres cada una[55], en las que podemos poner cualquiera de los caracteres de nuestro famoso código ASCII[56] que se visualizan con una forma fija.

En este modo de trabajo todo tiene la apariencia típica de los textos de las máquinas de escribir antiguas. Un sólo tipo de letra con todos los caracteres ocupando el mismo espacio. En un ordenador, de todos modos, es posible que podamos hacer algunos malabarismos con la presentación. Poner unos caracteres blancos sobre fondo negro, otros negros sobre fondo blanco (video inverso), otros con mayor brillo (alta intensidad), otros intermitentes, y hasta tal vez de diferentes colores. Esto permite resaltar alguna parte de la información que el ordenador nos presenta. Aún así, podemos sacarle un montón de defectos con toda facilidad. Nada de ver dibujos, nada de ver tipos o tamaños de letra diferentes.

En modo gráfico, la pantalla del ordenador pasa a ser un conjunto de puntos preferiblemente muy finos. Denomínaseles técnicamente "pixels[57]". En lugar de una pantalla de 25x80 caracteres tenemos una de 640x480 pixels, por ejemplo[58]. En lugar de escribir a máquina,

---

53 De todos modos, existen múltiples ejemplos que demuestran que es posible tener un monousuario-multitarea más que digno en equipos poco sofisticados. Todos los ordenadores personales deberían funcionar de este modo.

54 Los modos texto y gráfico se aplican a los periféricos de salida. Hablamos de pantallas, pero podríamos hablar asimismo de impresoras en modo texto o modo gráfico.

55 Este es, ni más ni menos, el tamaño habitual de la pantalla de un ordenador moderno trabajando en modo texto normal. Se habla en realidad de 25 líneas por 80 columnas.

56 En el que si le echamos otra miradita podremos ver hasta caracteres, llamados caracteres gráficos (176 a 223) que sirven para dibujar cosas, en nuestro caso cuadros, a base de llenar con ellos celdas adyacentes. No podemos dibujar casi nada más, e incluso esto sería imposible si estuviéramos manejando un ASCII estricto, que no los incluye.

57 Abreviatura del inglés "picture element" literalmente traducible como "elemento de un cuadro". En resumen, cada uno de los puntos de color que forman una imagen.

dibujamos. Podemos hacer lo mismo que con un lápiz o un pincel, pintar cualquier cosa. Textos, que, por supuesto, vamos a tener que seguir manejando, solo que ahora podrán aparecer en distintos tipos y tamaños. Dibujos de cualquier clase, que ya van a ser posibles. Igual de fácil que sacarle faltas al modo texto, es ver las ventajas del modo gráfico. No debería ser preciso decir que, simplemente, en modo gráfico y con una definición y un número de colores suficiente, vamos a poder meter en la pantalla de nuestro ordenador todo aquello que sea dibujable. Es un modo de trabajo flexible, potente, bonito, y de aplicación general.

No es que con un ordenador que trabaje en modo texto no podamos manipular distintos tipos y tamaños de letra, o no podamos procesar gráficos. Hay que diferenciar claramente entre el tipo de información que se procesa, que puede ser de tipo gráfico, y la presentación que de ella hagamos, que puede verse limitada al modo texto. Es simplemente que no vamos a poder visualizarlos, con lo que las habilidades de simulación del ordenador quedan muy limitadas. Es mejor poder ver las cosas tal como van a quedar mientras las estamos modificando.

Por ejemplo, un procesador de textos podrá mostrarnos, al trabajar en un ordenador en modo texto, los subrayados de un color, los superíndices de otro, las letras pequeñas en video inverso, y las grandes en alta intensidad, pero habrá que poner imaginación. En modo gráfico lo veremos tal cual, no hará falta imaginar nada, y no meteremos la pata creyendo que tenemos una cosa cuando en realidad tenemos otra. A esto de ver las cosas tal como quedarán es a lo que se le denomina "WYSIWYG", y si Vd. desea saber de donde sale semejante barbaridad puede mirar el glosario.

Ha sido evidente ver los fallos del modo texto y las ventajas del modo gráfico. Algo menos evidente es hacerlo al revés, ver las ventajas del primero y los problemas del segundo. Intentémoslo, ya que no carece de interés en los tiempos que corren.

En primer lugar hay que ver como metemos los gráficos en el ordenador. El modo más habitual es el denominado "bitmap" o mapa de bits. Cambiamos de código. Los bits de la memoria ya no se usan agrupados en bytes, cada uno de los cuales es un carácter como cuando almacenábamos textos, sino que a cada punto de la pantalla se le asignan los necesarios para contener el número de colores que permitamos a un pixel. Si dejamos que cada punto tenga 2 colores (blanco o negro), un bit bastará. En un byte meteremos 8 puntos. Si dejamos que tengan 16 colores harán falta 4 bits, y en un byte meteremos sólo dos puntos.

El modo texto es rústico, hay que admitirlo. Pero partamos de la base de que una buena parte de la información que manejamos en un ordenador está en esta forma. Aunque mi ordenador no me deje hacer gráficos de ninguna de las maneras, va a ser perfectamente válido para al menos el 85% de las tareas habituales[59]. Los textos, como información que

---

58 También ésta es la resolución normal de la pantalla de un ordenador moderno en modo gráfico. Aunque quizá debiéramos hablar mejor de 800x600 puntos.

59 Lamento tener que repetir los ejemplos de nuevo, siempre estamos con lo mismo. Son perfectamente posibles las contabilidades, facturaciones, control de existencias, proceso de textos....

luego presentaremos como sea menester, tienen grandes ventajas. Son enormemente frecuentes, y podemos hacer que el ordenador los busque, clasifique, y, si es preciso, modifique con facilidad. No puede decirse lo mismo de los datos gráficos. En una amalgama de puntos no es sencillo aislar partes con sentido. Es muy difícil buscar o clasificar datos gráficos en función de algún criterio[60]. La necesidad de los datos tipo texto puede demostrarse fácilmente subrayando que hay ordenadores que no admiten trabajar con gráficos, pero ahora mismo es inconcebible un ordenador que no manipule textos.

Podemos irnos explicando el porqué todos los ordenadores tienen un modo texto o, al menos manejan este tipo de información. ¿Pero porqué no emplear siempre una presentación gráfica?. La razón es la ya citada de que hay que ajustar la potencia del equipo con el uso que de él vayamos a hacer. Emplear una visualización gráfica aumenta muchísimo el volumen de información a procesar y requiere equipos más potentes. Ver cuanto lo sobrecarga exactamente depende de los gráficos concretos que empleemos y es un ejercicio interesante. Vamos con él.

El precio del modo gráfico

Luis Ricardo glotón

Para hacer comparaciones, cualquier soporte vale. Nuestra sufrida hoja de papel nos va a venir de nuevo de perlas. Ya vimos cual era la capacidad de un DIN A4 cuando se trataba de almacenar texto. Unas 60 líneas de 80 columnas. Unos 5KB, más o menos. Veamos qué volumen de datos tenemos en la misma cantidad de papel cuando metemos gráficos. O sea, qué significa querer meter en el ordenador la cantidad de información de una de las hojas de nuestro libro de ilustraciones predilecto.

Hay que suponer una definición de la imagen. Comencemos con algo mínimamente digno, unos 300 puntos por pulgada[61], que es lo que dan las impresoras actuales, mucho menos de lo que tenemos en el libro en cuestión. Con márgenes, la superficie útil de nuestro A4 vendrá a ser de unas 7x10 pulgadas, lo que hace 70 pulgadas cuadradas. Unos 18x25 cm. A razón de 90.000 puntos por pulgada cuadrada, un total de $6_1$300.000 puntos.

Hora de considerar los colores. En blanco y negro puros un punto es un bit (2 posibilidades). Nuestra hoja tendrá una capacidad de $6_1$300.000/8=787.500 bytes, cosa de 0'75 Megas. Si suponemos 256 colores o niveles de gris, cada punto será un byte (256 posibilidades). Nuestra hoja almacenará ahora $6_1$300.000 bytes, unos 6 Megas. Supongamos 65.536 niveles de gris o colores, algo ya medio decente. Son 16 bits, 2 bytes, los que nos permiten almacenar semejante número de posibilidades. La hoja contendrá ahora $6_1$300.000x2=$12_1$600.000 bytes, unos 12 Megas.

---

60 Un programa podría seleccionar aquellos gráficos que presentaran formas redondeadas y tuvieran un color dado como predominante. Algún programa de estos existe, pero no puedo decir si funciona bien. Lo que sí que puedo decir es que un ejercicio tan simple como leer, a los ordenadores se les atraganta con persistencia. Como todo el mundo sabe, consiste en coger una imagen gráfica (por ejemplo este párrafo) y extraer el texto en él representado. Se llama a esto "O.C.R." (ver glosario), y si los ordenadores supieran hacerlo bien, sería un chollo.

61 La pulgada es una unidad de longitud típica de países anglosajones que equivale a 2'54 cm.

Y si queremos color de verdad, con matices suficientes para degradados suaves y similares, hemos de ir a colores en 24 bits, con 16$_1$777.216 posibilidades, lo que se llama "True Color" (ver glosario). Lo suficiente para no distinguirlo de una foto o una pintura. Con este número de colores, nuestra hoja almacenará 6$_1$300.000x3=18$_1$900.000 bytes, casi 20 Megas. Éste es el caso más real, lo que solemos tener en una hoja con un gráfico a color que ocupe casi toda su superficie. Sin olvidar que estamos haciendo el cálculo para una resolución mediocre.

Si queremos resolución igual a la real de las imprentas, hay que trabajar con 1.200 puntos por pulgada. A esta resolución habría que multiplicar las cifras anteriores por 16 (4 veces más, al cuadrado). 302$_1$400.000 es la cifra para el número de colores más alto.

Pongamos un breve resumen de la cantidad de información contenida en un A4 en los distintos modos:

Texto	4.800 bytes
Gráficos blanco y negro, 300 puntos/pulgada . . .	787.500 bytes
Gráficos 256 colores, 300 puntos/pulgada . . .	6$_1$300.000 bytes
Gráficos 65.536 colores, 300 puntos/pulgada .	12$_1$600.000 bytes
Gráficos 24 bits, 300 puntos/pulgada	18$_1$900.000 bytes
Gráficos 24 bits, 1200 puntos/pulgada	302$_1$400.000 bytes

En primer lugar, espero que tengamos ahora mismo mayor respeto por nuestra modesta hoja de papel, capaz sin titubeos de almacenar semejantes barbaridades, y en segundo, espero nos demos cuenta de lo que supone en un ordenador pasar al modo gráfico. Aumentar los volúmenes de la información a procesar en cantidades exuberantes.

Sólo con 256 colores y una resolución mediocre, el volumen de información con respecto al modo texto es 1.312 veces mayor. El ordenador tiene ser capaz de trabajar 1.312 veces más deprisa para trabajar igual de rápido que en modo texto. Y capaz de almacenar 1.312 más datos que antes si queremos archivar nuestros gráficos "bitmap". Espero que estas cantidades sean lo suficientemente obvias para que no haya que recordarlas cuando hablemos de según que cosas en Windows o similares.

Y también que se entienda porqué existen ordenadores que todavía trabajan en modo texto, o que haya gente que, pudiendo elegir en su ordenador ambos modos de presentación, renuncie a la seducción de los gráficos. Con poca memoria, una CPU de las más bien lentas, no mucha capacidad de almacenamiento y, en consecuencia, poco dinero, es perfectamente posible hacer un monousuario-multitarea en modo texto que funcione a velocidad más que aceptable y me permita hacer la mayor parte de las cosas para las que sirve un ordenador. Excepto dibujar.

Con esto, podemos dar por terminada la cuestión de los distintos modos de funcionamiento. Tal vez alguien esté preguntándose si era necesario verlo, cuando hemos dicho que íbamos a dedicarnos principalmente al PeCé que, como muchos LAO sabrán, es ante todo un monousuario-monotarea. En primer lugar, es preciso tener una idea general, sea o no directamente aplicable a un PeCé. Y en segundo lugar, es que es directamente aplicable. El PeCé tiene cierta tendencia a la ubicuidad, está invadiendo terrenos que antes no eran suyos, y puede ahora mismo trabajar en prácticamente todos los modos que se han expuesto. Esto son cosas de software, ante todo del sistema operativo, y el hardware de un PeCé tiene a su

disposición un montón de sistemas operativos distintos en la actualidad. Y con toda seguridad, tendrá aún más en el futuro.

Variables y ficheros — *El almacén de Luis Ricardo*

Casi hemos terminado el capítulo, impaciente LAO. Nos faltan un par de cosillas que vamos a intentar presentar rápidamente porque el siguiente capítulo va a ser todavía más interesante. Esto se anima. Hay que ver ahora donde y como se guardan los datos en un ordenador.

Dado que hemos insistido en la importancia del almacenamiento de datos, habrá que concretar los mecanismos con que tan trascendental tarea se lleva a la práctica. Comencemos diciendo que se guardarán de dos formas. La primera consistirá en dejar los datos en la memoria. Si son datos nuestros, sólo podrán estar en la RAM, pero si son datos que el ordenador emplea podrán estar también en ROM. Los datos que se guardan en memoria se denominan "variables". Simplemente. Las "variables del sistema" son aquellas variables que el ordenador emplea para su funcionamiento. El se las entiende con ellas, en principio. Pero nosotros también podemos crear las nuestras. En ocasiones se habla de "variables del usuario". Por ejemplo, el número de días que hay en el mes de enero. Guardamos el 31 en memoria, y eso es una variable. No basta con guardarla. Hay que poder acceder a ella, saber donde está para poder usarla cuando queramos. Ya que, como vimos, los bytes de la memoria están numerados (es como la CPU los maneja), podríamos simplemente referirnos a ellas por su número, lo que llamábamos la "dirección", y decir que "en el byte 24.536 tenemos la variable con los días del mes de enero". No es una buena idea hacerlo así[62]. Es mucho mejor nombrar las cosas. Las variables son posiciones de memoria a las que les damos un nombre y contienen un dato. Con este sistema, llamaríamos por ejemplo "enero" a la variable que contuviera el número de días de ese mes. Eso es casi todo. Fácil. Lo de ponerles nombre no es una tontería. Facilita enormemente la localización de los datos[63], y conviene por ello que sean lo más explícitos posible para evitar errores estúpidos.

Como seguramente habremos pensado, el problema de guardar los datos en RAM consiste en que su contenido se borra al apagar el ordenador. No podremos construir un almacén de datos permanente a base de variables[64]. Pero es posible guardarlos en los periféricos de almacenamiento, donde sí que quedarán a salvo de verdad. Un conjunto de datos, almacenados en un periférico de almacenamiento, es lo que se denomina un fichero. Y también le ponemos un nombre, con el fin de facilitar su localización, recuperación, e identificar qué contiene. Podríamos crear un fichero "enero", meter dentro el 31, y leerlo cuando fuera preciso. Fácil de entender de nuevo, espero. Por ahora solo añadiremos que siempre ten-

---

62 No es buena idea pero en ocasiones se emplea. Se usa sobre todo para las variables del sistema.

63 Además, permite que un dato se pueda colocar, según convenga, en cualquier sitio de la memoria y no en una dirección fija tal vez ya ocupada por otra cosa. Esta es una razón muy poderosa para ponerles nombres.

64 Salga de esta nota inmediatamente si no quiere liarse. Lo cierto es que, a base de emplear memorias de bajo consumo montadas de forma que puedan manejarse como disquetes (tarjetas de memoria continua), SÍ es posible almacenar casi permanentemente las variables y crear un almacén de datos con ellas. Ordenadores hay que emplean este sistema.

dremos un "sistema de ficheros", un conjunto de programas y especificaciones encargados de todo lo relativo al modo en que se manipulan tan importantes entidades en un ordenador dado.

Y ahora, ocupémonos del formato en que se guardan los datos, tanto en las variables como en los ficheros. El formato no es más que la distribución exacta de la información en la memoria, sea RAM, ROM, o un disquete. La cosa es engañosamente simple. No hay reglas, y cada usuario puede guardarlos como le plazca. Pongamos un ejemplo. Yo podría guardar mi "enero" en formato ASCII. Simplemente, lo escribiría empleando ese código. Cogería dos posiciones de memoria, o los dos primeros bytes de mi fichero (también el tamaño de un fichero, o las posiciones de los datos dentro de él, se expresan en bytes[65]) y en el primero pondría el ASCII para el tres y en el segundo el ASCII para el uno. La pareja de bytes "51 48" haría el trabajo maravillosamente. Cuando guardamos los valores de este modo, decimos que empleamos formato ASCII. Escribimos nuestro dato con ese código, cada carácter ocupa un byte, y ya está.

El formato ASCII tiene muchas ventajas. Es muy fácil de entender. Es muy fácil visualizar, cuando sea menester, el contenido de un fichero o una variable con herramientas muy sencillas[66]. Es fácil que cualquier programa pueda leer y escribir los datos en ASCII, con lo que es enormemente universal. Podemos usarlo para guardar cualquier cosa que se pueda escribir, con lo que es más que razonablemente potente.

Quizá algún LAO audaz (y probablemente experimentado) esté pensando algo similar a "Bien. Nos ponemos todos de acuerdo en usar ASCII para almacenarlo todo. Así yo podré leer cualquier fichero de cualquiera en mi ordenador sin ninguna dificultad". Bueno, no es mala idea. Pero hay tipos de datos a los que el formato ASCII no parece adaptarse. Ya vimos que los gráficos se almacenaban como puntos de colores en el número de bits que hiciera falta. Y esto desde luego no es un formato ASCII. "Bueno, ya está bien de sacar faltas", contesta mi más que ligeramente curtido LAO, "Pues nos ponemos de acuerdo en dos formatos. Para textos, el ASCII. Para gráficos, como con 24 bits por punto parece que había colores más que suficientes para cualquier cosa, pues tres bytes por piel y punto final. Con dos formatos todos contentos". Bueno, con esto cubrimos ya los gráficos, pero ¿y otros datos potencialmente interesantes como imágenes animadas y sonidos estereofónicos?. "Vale, pues cuatro formatos". Pero es que no está claro como, incluso con el ASCII, vamos a poder meter cosas como distintos tipos de letra y, además, tal vez pasado mañana aparezca un nuevo tipo de dato con el que no hemos contado que sea necesario procesar. Al fin y al cabo, el ordenador es una cosa muy flexible y puede que interese aplicarlo a vaya Vd. a saber qué. "Bueno. Ya está bien. Adiós".

No es fácil, no, el tema de los formatos. Aparte de los distintos tipos de datos que vamos a procesar, puede que interese guardar un mismo dato con formas diferentes. Los datos los

---

65 Ya dijimos en su momento que convenía ver los periféricos de almacenamiento como extensiones de la memoria del ordenador.

66 ¿Y porqué ha de ser menester?. Pues simplemente porque no tengamos ni idea de qué diablos hemos metido en algo que se llama TB_2_IDX.ACW y queramos echar una ojeada a ver si nos suena.

manejan los programas, y en la práctica ambos están tan interrelacionados que podemos decir que están hechos uno para el otro y ninguno funciona por separado[67]. La forma en que trabaja un programa depende de como estructure los datos, el programa se construye para manejar los dispuestos de ese modo, y procesar aquellos organizados de otra forma exige convertirlos de un formato a otro. Si se conoce cual es el otro, cosa nada fácil con varios millones de formatos diversos flotando alrededor. Además, puede ser que la conversión no sea tan sencilla, que cueste mucho tiempo, o incluso que pueda hacerse pero en el proceso se pierdan según que cosas.

Convendría poner un ejemplo para concretar de qué estamos hablando exactamente. Veamos como podemos meter en un fichero la información necesaria para almacenar un texto con la posibilidad de usar distintos tipos de letra y similares[68]. La información que queremos guardar puede ser algo así como:

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

y podríamos almacenarlo escribiendo en ASCII, dentro de nuestro fichero, lo siguiente:

Tipo: Courier, Tamaño: 10, Texto: "Qué bonito es Candanchú con nieve,", Tipo: Times, Tamaño: 10, Modo: Negrita, Texto: "y qué bonitas vistas tiene."

O, puesto que estamos empleando tanto sitio para los tipos y tamaños como para el texto en sí, podríamos intentar abreviar y escribirlo como:

T:Courier, S:10, "Qué bonito es Candanchú con nieve,", T:Times, S:10, M:N, "y qué bonitas vistas tiene"

O abreviarlo aún más, crear un código para nosotros solos en el que el 0 quisiera decir "Tipo: Courier", el 1 "Tipo: Times", y ambos debieran ir seguidos por su tamaño y modo, caso de haberlo. Con esto quedaría:

0,10,"Qué bonito es Candanchú con nieve,",1,10,N,"y qué bonitas vistas tiene"

Cualquiera de los tres formatos anteriores contiene exactamente la misma información que la frase original pintada como estaba. Nosotros podremos hacer un programa que maneje los datos en el tercer formato, el más compacto y probablemente más rápido, pero no es

---

67 Todo aquel que haya programado alguna vez sabe que un 50% del programa y su funcionamiento consiste en decidir las estructuras de los datos a manipular. Equivocarse en esto puede llevar a un programa lento, farragoso, y para el que algunas cosas son imposibles de hacer. Y a la inversa. Niklaus Wirth, uno de los grandes académicos informáticos, inventor de lenguajes como Pascal, Modula 2, y otro montón de cosas, tiene un libro titulado "Algoritmos+ Estructuras de datos=Programas". Lo de "Algoritmo" es la denominación que se le da en jerga matemático-informática a un procedimiento detallado para hacer algo y, con esto explicado, no hay más que añadir a título tan expresivo.

68 Tal vez se esté preguntando el porqué de tanta insistencia sobre ejemplos con textos. Hay varios motivos. Son los más simples de entender por cotidianos y por tener los formatos más sencillos. Además, los ordenadores se emplean la mayor parte del tiempo para manipular textos.

fácil que un programa hecho para el primer formato pueda leer el tercero. Se confundirá. Se liará. Se colgará.

Si queremos que diversos programas puedan intercambiar este tipo de información habrá que ponerse de acuerdo en una forma de guardar los datos, sea cual sea. Habrá que usar un formato estándar.

Vimos, hace ya mucho tiempo, que en el ordenador almacenábamos información de muchas formas distintas, con varios códigos. Código máquina, ASCII, ANSI... Ahora estamos viendo que podemos disponer los datos de diversas maneras y que empleamos, además de lo anterior, distintos e innumerables formatos. Y tanto unos como otros incompatibles. Y el ordenador no entiende de significados y no puede intentar deducir qué quiere decir un formato, ni siquiera aislar por sí solo la información relevante[69].

Si a Vd., querido LAO, esto le huele a problemas, no dude de su buen olfato. Baste por ahora decir que para tratar de resolverlos nos volveremos a encontrar con nuestro ya viejo amigo el código ASCII y que, por tanto, no nos vendría mal ver qué tipo de información se maneja o puede manejar en un ordenador con él. Los textos, sin ningún enriquecimiento, pueden y suelen estar en ASCII. Los textos con florituras están en ASCII en la parte texto y de cualquier manera en las florituras, pero pueden pasarse a ASCII si renunciamos a ellas. O también si no renunciamos, aunque en este caso es un poco más complicado y se acaba con algo similar al ejemplo de antes. Los números no están casi nunca en ASCII, pero pueden convertirse fácilmente. Los gráficos y otros datos más complejos no lo están jamás pero son susceptibles de terminar de esta forma[70]. ¡Caramba!. Ahora resulta que el ASCII puede emplearse para todo.

El gran farsante — *Luis Ricardo nos engaña*

En inglés “The great pretender”, famosa y clásica canción que tal vez el LAO recuerde. Y, como última pincelada sobre la personalidad de Luis Ricardo, también de necesario conocimiento.

Podemos decir con total justificación que el ordenador, equipado con sus programas correspondientes y ya funcionando, va a ser un aparato que nos va a querer tomar el pelo de forma constante, que va a intentar que creamos que está haciendo lo que en realidad no hace. Su incapacidad para procesar significados[71] lleva a que todo lo que el ordenador pueda

---

69 Para el ordenador, un fichero o una variable es una ristra de bits. Si no hay un programa que le diga si tiene que agruparlos de dos en dos, de ocho en ocho, o que los ignore sin más, no tienen ningún sentido. Es el programa con que manejamos un fichero el que le dice al ordenador como lo tiene que mirar. Y ante un formato que nuestro programa desconoce estamos en las mismas que al principio.

70 Lo que se hace es definir auténticos lenguajes, con las ordenes en ASCII, que van encargándose de describir el dato en cuestión. Algo así como “Cuadro de 800x600. Izquierda a derecha. Arriba a abajo. 256 colores. Rojo, verde, rojo, azul, negro,.....”. Puede hacerse bastante mejor, pero es algo así.

71 Inevitable. Que el ordenador no procese significados es asimismo inevitable. Para ser general, el

lograr sea intentar presentarnos los datos bajo una apariencia digerible para nosotros, pero nada más.

Este juego de apariencias que hemos arañado al hablar del código ASCII y que consiste en que un número, el 65 por ejemplo, unas veces va a querer decir "A", otras "B", otras "J", otras "x", en ocasiones va a ser incluso una línea de 3 cm de color azul marino desde la esquina superior izquierda de la pantalla en dirección Sureste, y otras hasta una orden para que "Luis Ricardo cantidubi, cantidubidubidá ¡ya!, lo que sea", se aplica no sólo a los datos, sino también a la forma en que el ordenador nos deja comunicarnos con él. Y con efectos tan o más temibles dada la trascendencia de la cuestión.

Todo en el ordenador es variable, todo apariencia. Y ésta puede cambiar en cualquier momento, a voluntad de alguien[72]. No estamos diciendo nada nuevo. El ordenador no era sino una máquina capaz de adoptar cualquier forma, convertirse en cualquier cosa. No debemos sorprendernos de que haga aquello para lo que se diseñó.

El ordenador sólo tiene forma temporalmente, y por eso no conviene hacerse ideas muy rígidas. El mismo hardware, con otro programa, igual incluso manipulando los mismos datos y haciendo la misma cosa, puede pasar en 3 segundos de ser algo inhóspito y desagradable a ser un prodigio de seducción multicolor. Dijimos que los programas le dan al ordenador su mecanismo y su flexibilidad. También le dan su apariencia. El mecanismo define qué va a hacer el ordenador. La apariencia va a influir sobre todo en la manera en que nos comunicamos con él y le decimos qué tiene que hacer.

Opciones por defecto — *¡Luis Ricardo, defínete!*

Enfrentados a un baile de máscaras, a un juego de espejos en el que todo puede ser pero no es de ninguna forma particular, nuestra salud mental peligra y el invento en cuestión es inservible si no conseguimos precisarlo un poco, darle una apariencia lo suficientemente concreta como para que podamos al menos empezar a trabajar con él. Hay que evitar que el ordenador pueda hacer dos mil cosas pero no llegue a hacer ninguna en la práctica. Tenemos que arbitrar mecanismos para que todo se defina, tanto la forma en que se manipulan los datos como la manera de mandarle cosas a Luis Ricardo.

El problema se resuelve inicialmente, aunque tan sólo de forma parcial, mediante lo que se denominan "opciones por defecto". Una opción por defecto no es aquella que el ordenador elige si está defectuoso o averiado (craso error sería interpretarlo así), sino aquella que, a falta de otra indicación, adopta automáticamente[73].

---

ordenador debe procesar datos brutos. Un dato con significado es algo excesivamente concreto, dependiente del contexto cultural, sofisticado, más elaborado que el dato en sí mismo. Un ordenador que procesara significados sería muy poco general.

72 Si ese alguien fuera siempre el usuario, el ordenador sería una cosa absolutamente dócil, maravillosa. Desgraciadamente, ese alguien es muchas veces el que ha enlatado las instrucciones (el programador) y no siempre la razón del cambio está clara. Lo que lleva a desorientación y problemas en abundancia.

73 Hay un cierto paralelismo con el papel de la ROM. Las opciones por defecto son las que el

Su existencia permite que, al cargar un programa, por lo menos tengamos algo. Con suerte, incluso algo manejable. Y rara vez algo cómodo, bonito, útil, y a nuestro gusto. Porque es que no pueden pedírsele peras al olmo, y en ocasiones las opciones por defecto son inadecuadas y el programa no funciona. Puede por ejemplo intentar visualizar las cosas en color en un monitor monocromo, o querer usar un ratón y que tengamos tan sólo un teclado. Es necesario entonces cambiar las opciones que el programa ha elegido automáticamente por otras que se ajusten mejor a nuestro equipo o, caso éste también muy probable, que simplemente se adapten mejor a nuestro gusto, que para eso hemos pagado el ordenador y podemos hacer con él lo que queramos. Este cambio de opciones puede hacerse de forma temporal, por probar cosas a ver que tal, o puede hacerse casi permanente, hasta que lo queramos cambiar de nuevo[74]. No hay más que decirle al programa que las grabe.

El proceso de ajuste de las opciones es lo que se denomina configuración. Parece algo banal a primera vista. Pero no lo es. Asunto grave y casi omnipresente, aparece tanto en software como en hardware, y en ambos a todos los niveles. Es uno de los grandes problemas a los que hay que enfrentarse[75].

El que avisa no es traidor. Dijimos que el tema iba a ser largo e importante. Espero que hayamos aprendido un montón de cosas interesantes, aunque nos haya quedado un pelín metafísico a ratos. Tal vez no pueda ser de otro modo. Todo lo que sigue será más aplicado. Continuamos yendo de lo general a lo concreto y en el capítulo siguiente, ahora que sabemos esto, veremos como se emplea el software. Qué hay que saber para manejar un ordenador con soltura.

Por despertar su interés, veremos que con sólo entender dos cosas, exactamente dos, cualquier ordenador quedará rendido a nuestros pies. Si puede resistir sin abalanzarse apresuradamente y de forma inmediata sobre un capítulo que promete divulgar tan portentosa receta, es que ha llegado el momento de que vuelva a irse de cañas con los amigos. Descanse un rato o dos. Se lo ha ganado. Y, ya regenerado, vuelva para enfrentarse con el capítulo 5.

---

programa selecciona por sí solo. Lo que él sabe hacer.

74 Esto no siempre es cierto, pero es muy frecuente y viene muy bien.

75 Ver “configuración” en el glosario.