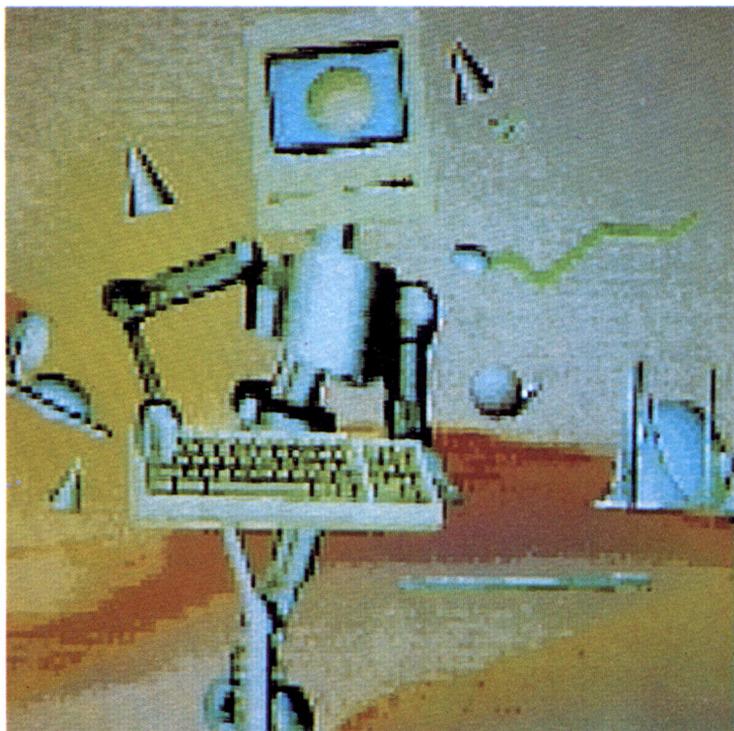


# GRAN BIBLIOTECA AMSTRAD



## INTELIGENCIA ARTIFICIAL

---

¿PIENSAN LAS MÁQUINAS?



**GRAN BIBLIOTECA**  
**AMSTRAD**

**4**

**INTELIGENCIA ARTIFICIAL**



# INTRODUCCIÓN



Desde antaño, el hombre ha intentado imitar mediante artilugios diversos el comportamiento humano, inicialmente el mecánico. Desde las bailarinas de las antiguas cajas de música, hasta los complejos robots de las cadenas de montaje actuales, pasando por sofisticadas muñecas que corren, ríen y hasta nadan, un sin fin de reproducciones humanas han sido el fruto de su genio creador.

Con el avance tecnológico, sus aspiraciones en este sentido se han ido sintetizando en la obtención de un mecanismo que simule la más preciosa cualidad del hombre, lo que le distingue del resto de los animales que pueblan la Tierra: la inteligencia.

La difusión de la microinformática ha puesto a disposición de la práctica mayoría de los usuarios, una herramienta que les permite hacer realidad su sueño dorado: reproducir en su ordenador los comportamientos y mecanismos inteligentes del hombre... o al menos, algo muy próximo a ellos.

Las tendencias actuales en la investigación de la inteligencia artificial se centran fundamentalmente en dos campos: la inteligencia aplicada y la teórica.

En su primera vertiente, las investigaciones se encauzan principalmente a través del perfeccionamiento y nuevo diseño de sistemas expertos en determinadas disciplinas, de los cuales tendremos oportunidad de hablar más adelante con mayor detenimiento.

Por otra parte, coexiste con este movimiento de investigación otro menos «práctico» si queremos, pero no por ello carente de interés, e incluso nos atreveríamos a asegurar que más apasionante, cuyos descubrimientos son al fin y a la postre la base de las realizaciones prácticas en sistemas de inteligencia artificial (IA).

En lo que al usuario de cualquier Amstrad concierne, hay que tener bien presente, que la simulación de un comportamiento tan extraordinariamente complejo como el del hombre es algo que está totalmente reservado, por el momento, a los potentísimos «macroordenadores» actuales, ya que es preciso para su tratamiento disponer de grandes cantidades de memoria y altas velocidades de proceso, para conseguir con ello tan sólo acercarse levemente a lo que en el hombre es una manifestación natural y espontánea de la inteligencia en todo su esplendor, en la total amplitud de este término.

Así pues, ¿qué aspiraciones puede mantener el usuario de un «deminuto» Amstrad en este campo que actualmente sobrepasa los grandes «superordenadores»? Claramente ninguna en la faceta práctica, si bien es cierto que no le está prohibida su entrada a la investigación a menor escala, más bien en el área de la teorización, que como anteriormente hemos aseverado, se convierte en algo absolutamente fascinante.

Recordemos nuestra ilusión cuando conseguíamos edificar un castillo mediante las piezas de un simple juego de construcción ¿estábamos acaso entonces tan lejos de ser arquitectos?

Este libro nos introducirá en el apasionante mundo de la IA. Conoceremos sus fundamentos, sus logros e incluso algunas de sus anécdotas, viendo plasmados en forma de programas sus más importantes descubrimientos, como pequeños arquitectos de juguete. Indudablemente, llegaremos a conocer suficientemente el tema como para entablar una conversación sobre el mismo y teorizar sobre él, con la experiencia de aquel que ha intentado hacer razonar a su máquina, aunque sólo sea mínimamente.

Estaremos desde luego en condiciones de ampliar nuestros conocimientos, y lo que es más interesante y divertido: de experimentar por nuestra cuenta y cómo no, de asombrarnos ante la aparente inteligencia de un simple montón de circuitos.

# EL SISTEMA DE INTELIGENCIA HUMANO

**E**l humano, como ser eminentemente social, nunca puede ser estudiado en ninguno de sus aspectos, y menos en el concerniente a la inteligencia, aislado del mundo que le rodea, ya que precisamente la interacción con este medio propicia el desarrollo de su capacidad pensante.

Sin duda, el primer paso para enseñar a un ordenador mediante un programa a resolver una ecuación de segundo grado es aprender a resolverla uno mismo. Es por ello que para adentrarnos en el mundo de la IA (Inteligencia Artificial), para enseñar a nuestro ordenador a «pensar», el primer paso debe ser comprender como pensamos nosotros mismos. Pero ¿por qué enseñar a la máquina nuestro modo de pensar, nuestro «sistema de inteligencia»? ¿es acaso perfecto ese sistema?, ¿es que el hombre no se equivoca?

La respuesta a esta pregunta es una sola palabra: ZEPOL. ¿Que no sabemos lo que es ZEPOL? Hagamos un esfuerzo. ¿No conseguimos imaginar lo que es ZEPOL?

Lógico, ¡porque ZEPOL no es nada! Era un simple juego de demostración para apoyar la siguiente afirmación: somos incapaces de imaginar algo sobre lo cual no tengamos un conocimiento previo a través del medio que nos rodea.

Del mismo modo, no conocemos otro sistema de pensar inteligentemente que no sea el humano, y somos por tanto incapaces de imaginar otro distinto; de ahí que nos veamos forzados a enseñar a la máquina precisamente nuestro modo de razonar.

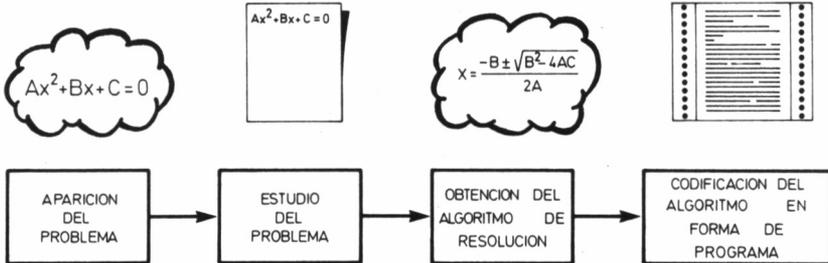
## EL MEDIO QUE NOS RODEA

A grandes rasgos, podríamos decir que la vida de todo hombre se basa en un proceso cíclico y repetitivo dividido en tres etapas: adquisición de conocimientos, almacenamiento de los mismos y utilización de ellos para la resolución de problemas; y no sólo es un problema una compleja integral matemática, sino cualquiera de las múltiples situaciones que a lo largo del día nos exigen la toma de una decisión, como cruzar una calle o entrar en un cine a ver una película.

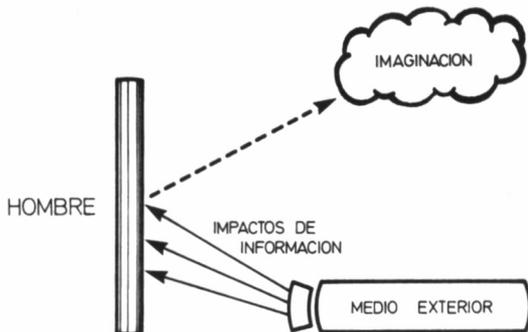
Este proceso es muy similar al de fabricación de cualquier objeto por el hombre, pongamos por ejemplo un mueble.

En nuestra fábrica hacen su entrada multitud de materias primas: clavos, colas, diferentes tipos de maderas e incluso diseños de muebles. Todo esto equivale a la toma de datos exterior en el proceso vital del ser humano, gracias a una función que podríamos llamar *percepción*.

Acto seguido, los artículos son almacenados en una gigantesca nave, con objeto de que dejen el suficiente espacio libre en el taller para el cómodo desenvolvimiento de los operarios. Este segundo proceso,



4.1.1. Proceso general de resolución de un problema por ordenador.



■ 4.1.2. *La imaginación es un simple espejo del medio que nos rodea.*

extrapolado al área del conocimiento humano, se puede denominar *memorización*.

El siguiente paso es la llegada de un cliente que nos encarga la confección de un mueble. Este proceso es de relativa complejidad en el hombre, e involucra los factores asociados *motivación-voluntad*.

A continuación, es necesario transportar al taller todos aquellos elementos necesarios para la confección del mueble; incluso puede darse el caso de que no sepamos fabricarlo, es decir, desconozcamos su diseño, o que carezcamos de alguno de sus elementos indispensables, ya sea porque nunca lo tuvimos o porque se agotó. Este traslado de elementos desde el almacén hasta el taller, tiene su símil en el hombre en el proceso de la *imaginación*, entendida como recuerdo de lo ya percibido, a lo cual se une además la falta de conocimiento o percepción y por supuesto, el *olvido* de lo anteriormente conocido o percibido.

Por último, los operarios confeccionan el mueble, con todos los elementos en el taller, incluido su diseño y libro de instrucciones de montaje. Este es sin duda el proceso más importante en el hombre y se conoce bajo el nombre de *razonamiento*.

## LA PERCEPCIÓN

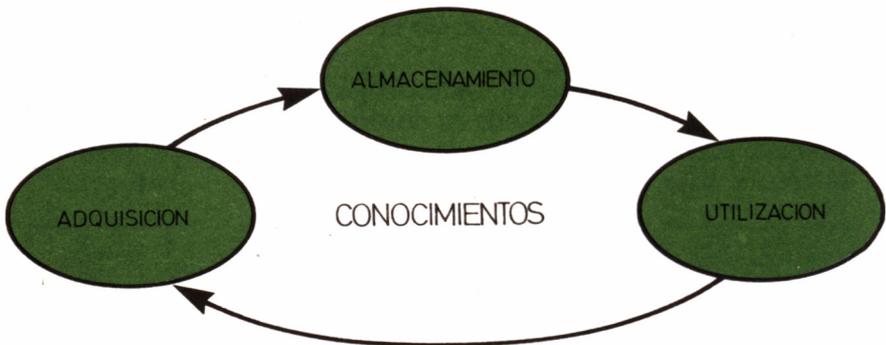
Si afirmáramos que lo percibido nos llega a través de nuestros

cinco sentidos: vista, oído, gusto, tacto y olfato, estaríamos cometiendo un error de precisión, algo imperdonable al estar tratando el tema informático, frente a la conocida precisión del ordeandor.

Ahora es importante que hagamos un inciso para fijarnos en algo aparentemente carente de importancia, pero de gran relevancia en la percepción. La inmensa mayoría de los lectores del párrafo anterior, tras hacer una cita a la Informática, algo que se relaciona directa y subconscientemente con los ordenadores, habremos entendido precisamente esto (ordenador) como última palabra del mencionado párrafo; nada más lejos de la realidad, pues la claridad tipográfica no deja lugar a dudas: la palabra que cierra el primer párrafo de este apartado es ordeandor, y no ordenador.

Como podemos comprobar, puede existir una notable diferencia entre lo que nos llega por los sentidos, el de la vista en nuestro experimento (ordeandor) y lo que finalmente percibimos (ordenador). Así pues, hemos de distinguir claramente entre la *sensación*, es decir, lo que captan nuestros sentidos, y la *percepción*, lo que finalmente percibe nuestro intrincado cerebro.

Usualmente, en el proceso sensitivo se distinguen tres fases fundamentales: la física, la fisiológica y la psicológica. La primera etapa se corresponde con la excitación física de alguno de nuestros órganos sensitivos, desencadenándose en él reacciones físico-químicas; tal es el caso de las experimentadas en el ojo al recibir la luz reflejada del libro en nuestra lectura. La segunda etapa se desarrolla a lo largo del sistema nervioso, haciendo llegar hasta el cerebro la sensación proveniente del órgano sensitivo, pasando de célula en célula a través de la sinápsis. La última etapa se desarrolla en el cerebro a la llegada de impulso nervioso.



#### 4.1.3. Proceso inteligente humano.

Hasta el momento, la reproducción de un proceso sensitivo humano en un ordenador escapa a la finalidad de este libro, dado que entra más en el campo de la Robótica que en el de la Informática; no obstante, queda indudablemente muy lejano, al menos según las perspectivas actuales, el día en que se puedan crear mecanismos sensitivos tan perfectos como los humanos. Aun así, los avances tecnológicos han propiciado resultados de consideración, ya que algunos robots actuales pueden reconocer la voz, sintetizarla, e incluso disponer de los sentidos del tacto, gusto u olfato, o más correctamente hablando, de la simulación de los mismos.

En todo caso, la fase psicológica es la más importante de las tres mencionadas en el proceso perceptivo, dado que es aquella en que tomamos conciencia de lo sentido. De hecho, en multitud de ocasiones las sensaciones no pasan de la segunda fase, desvaneciéndose como fantasmas para la conciencia humana.

Así por ejemplo, cuando nos encontramos en la confortable butaca de un cine asistiendo a la proyección de una película que nos fascina, pongamos «Tron» o «Juegos de guerra» para entrar en situación, no somos conscientes del ruido producido por nuestro vecino de butaca al revolverse en un golpe de emoción de la película, o que en el cine no reina la completa oscuridad, sino que unas pequeñas lucecillas en el suelo velan por la integridad física de los consabidos espectadores de última hora, o de los padres que apresuradamente se dirigen al servicio acompañados de sus criaturas a la mitad de la proyección, y lo que puede ser aún más conflictivo, no reparamos en que el brazo que estamos pellizcando en los momentos de excitación culminante no es el de la butaca, sino el de nuestro acompañante.



4.1.4. Detalle de un proceso inteligente.



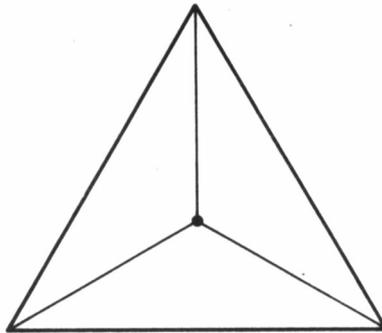
— 4.1.5. *La pelota se acerca a nosotros.*

## MÁS ALLÁ DE LA SENSACIÓN

Con lo dicho anteriormente, queda explicado el fenómeno de la sensación, como un simple reflejo de un objeto; ahora bien, según hemos podido comprobar en nuestro experimento anterior con la palabra «ordeandor», lo percibido no está sólo compuesto por lo sentido.

Así por ejemplo, en la lectura, muchas veces no llegamos a captar más que unas pocas letras de las que componen una palabra, lo cual no nos impide percibir a la perfección el contenido del texto. Este es también el caso de cualquier representación en el plano que nos pueda hacer percibir una imagen en profundidad. Observemos cómo la pelota se acerca a nosotros (fig. 4.1.5).

O esta curiosa perspectiva, «a vista de pájaro» de la pirámide de Keops (fig. 4.1.6).

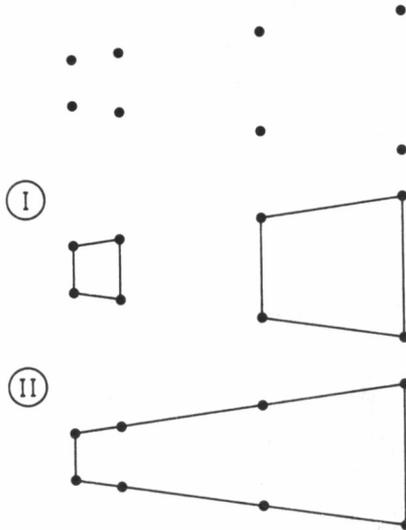


— 4.1.6. *La pirámide de Keops a vista de pájaro.*

No obstante, pese a la ilusión perceptiva, pronto puede llegar la desmitificación sensitiva que convierta la bola acercándose peligrosamente en dos pequeños círculos unidos por una flecha de izquierda a derecha, y la misteriosa Gran Pirámide en una simple figura geométrica, en la que se ilustra la posición del baricentro de un triángulo.

Evidentemente, cada vez que percibimos algo lo hacemos a través de los sentidos, debido a lo cual en el mecanismo de percepción se ven mezcladas no sólo multitud de sensaciones, sino también recuerdos de percepciones pasadas. Así, al degustar un delicioso postre, no sólo percibimos a través del paladar su sabor, sino que también se une a ello el murmullo de los comensales, el olor de la cocina cercana, la atrayente presentación del plato, la fría sensación en la mano al tocar la copa... y lo que es más, también la subjetiva apreciación de «el agradable ambiente», la felicidad de celebrar un cumpleaños o el recuerdo de haber pasado excelentes momentos en ese restaurante.

Constatando esta realidad, algunos investigadores aseguran que el mecanismo perceptivo en el hombre se desarrolla basándose en el entorno de lo sentido, es decir, que el mundo exterior no se nos presenta tal como es, sino a través de símbolos que se denominan formas de la percepción. Esto nos hace pensar en las palabras que Alan Turing, uno de los padres de la inteligencia artificial, pronunció hace ya



4.1.7. La ley de la proximidad nos hace asociar los puntos según la primera figura, pero no en la segunda forma.

tiempo: «Todo lo que puede hacerse con símbolos, puede hacerse también con un ordenador programable».

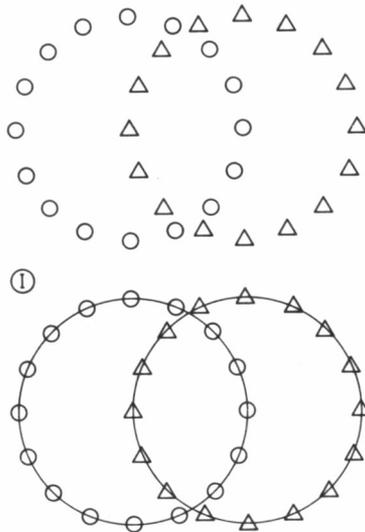
## EL GESTALTISMO

La teoría de la percepción anteriormente expuesta es la defendida por los llamados psicólogos gestaltistas (*Gestalt* en alemán significa forma) y el mecanismo de producción de estas formas perceptivas ha sido ampliamente estudiado, lo cual permite enunciar una serie de reglas; unas de ellas dependen directamente del objeto percibido y se agrupan bajo la denominación común de leyes objetivas y las otras dependen del sujeto perceptor y se conocen como leyes subjetivas.

Las leyes objetivas son de extraordinario interés para nosotros, ya que rigen en la mayoría de los casos no sólo a los individuos, sino también a muchos animales, seres que en principio no se consideran inteligentes:

1.<sup>a</sup> *Ley de proximidad*: Tendemos a agrupar los objetos que se encuentran juntos (fig. 4.1.7).

2.<sup>a</sup> *Ley de semejanza*: Tendemos a agrupar objetos que son semejantes entre sí (fig. 4.1.8).

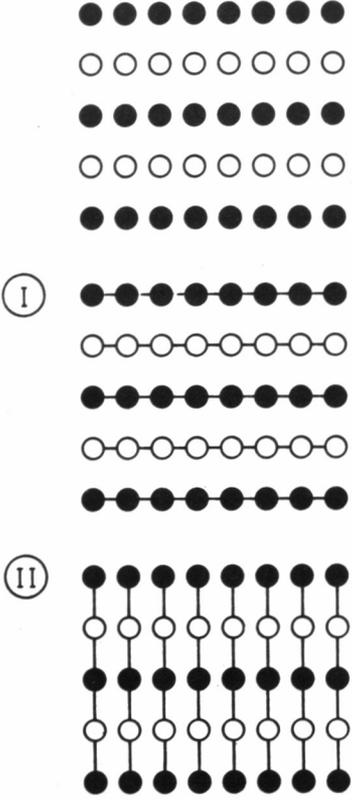


4.1.8. Ilustración de la ley de semejanza.

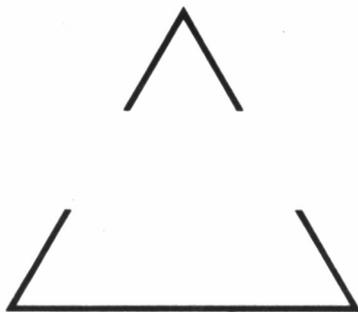
3.<sup>a</sup> *Ley de contraste:* Por antagonismo a la ley de semejanza, tendemos a agrupar los objetos completamente diferentes entre sí (fig. 4.1.9).

4.<sup>a</sup> *Ley de compensación:* Con frecuencia tendemos a compensar o completar un determinado dato sensible (fig. 4.1.10).

5.<sup>a</sup> *Ley de figura-fondo:* En toda percepción hay que distinguir una figura y un fondo sobre el cual destaca la figura. Realmente, en esta simple teoría se basa el foco óptico. El jarrón de Rubin (fig. 4.1.11) es un buen ejemplo de esta ley, ya que según consideremos un fondo oscuro o claro apreciaremos dos figuras completamente distintas: un jarrón o dos caras enfrentadas.



4.1.9. La ley de asociación por contraste nos hace percibir preferiblemente nos hace percibir preferiblemente la primera figura en vez de la segunda.



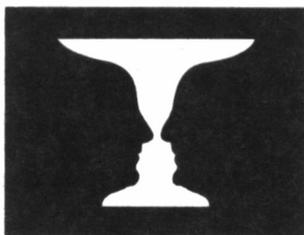
■ 4.1.10. *La ley de compensación nos hace percibir en la figura un triángulo inexistente.*

6.<sup>a</sup> *Ley del movimiento ilusorio:* Esta ley tiene una aplicación evidente en el cine, en el cual apreciamos un movimiento partiendo de una sucesión de imágenes que se encuentran en completo reposo en el celuloide.

Por otra parte, las dos principales leyes subjetivas son:

1.<sup>a</sup> *Ley de la predisposición:* Los sujetos tienen tendencia a interpretar un determinado objeto según como se le predisponga. Por ejemplo, se puede casi escuchar como el tic-tac del despertador se convierte en «Pi-lar».

Esta ley se aplica con frecuencia en ordenadores cuyo sistema de sintetización de voz deja bastante que desear. Así, si por ejemplo el sintetizador emite la palabra *verde*, simultáneamente la pantalla se torna de ese color, o bien se escribe la palabra. De esta forma, el operador comprenderá perfectamente el sonido emitido por el ordenador, pero no realmente por una calidad en la sintetización, sino porque se encontraba predispuesto a ello; de hecho, lo más probable



■ 4.1.11. *El Jarrón de Rubín nos sirve como ilustración de la ley de figura-fondo.*

es que cualquier observador ajeno al experimento, situado en la parte trasera del monitor, aunque más cerca del altavoz, no consiga distinguir la palabra pronunciada por el ordenador.

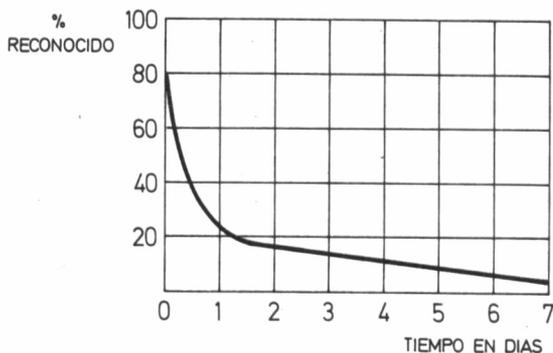
2.<sup>a</sup> *Ley de las tendencias*: Esta última ley explica que ciertas situaciones del sujeto influyen en las formas de la percepción. Así por ejemplo, en la oscuridad de una habitación, un niño atemorizado da forma de monstruos imaginarios a las sombras que entre ruidos pasean por la estancia.

## LA MEMORIA

Anteriormente establecimos un símil entre el mecanismo inteligente en el hombre y el proceso de fabricación de un mueble, pero de él sólo hemos estudiado la recogida de materiales: la percepción, ahora bien, cómo se contempla el aspecto de la motivación o voluntad, lo que nos impulsa a confeccionar el mueble, o la memorización e imaginación, que nos llevan a almacenar los materiales y traer al taller los adecuados para la confección de cada pieza.

La motivación o voluntad en un sistema artificialmente inteligente debe estar en principio clara: la acción del hombre solicitando una respuesta de la máquina, directa o indirectamente, debe ser lo que inicie un desencadenamiento del proceso pensante en éste.

Así, para no vernos en la desagradable situación de convertirnos en Blade Runners, caza-cyborgs (CYbernetic ORganism, organismo cibernético) Isaac Asimov en su libro «Yo, robot» propone las llamadas tres leyes de la Robótica:



4.1.12. Curva de Ebbinghaus.

1.<sup>a</sup> Un robot no puede dañar a un ser humano o, a través de la inacción, permitir que se dañe a un ser humano.

2.<sup>a</sup> Un robot debe obedecer las órdenes dadas por los seres humanos, excepto cuando tales órdenes estén en contra de la primera ley.

3.<sup>a</sup> Un robot debe proteger su propia existencia siempre y cuando esta protección no entre en conflicto con la primera y la segunda ley.

No nos debe extrañar que comencemos a entrar en el tema de la Cibernética, ya que según Oparín, la definición de ésta es «la ciencia de los medios de percepción, transmisión, conservación, elaboración y utilización de información por los equipos de dirección, independientemente de su naturaleza material concreta, sean ellos de metal o de carne, es decir, máquinas u organismos».

No obstante, para nuestra desgracia ya existen robots que infringen las leyes de Asimov. Por ejemplo, el Prowler es un robot centinela destinado a patrullar áreas críticas, como oleoductos o bases militares; su nombre responde a las siglas *Programmable Robot Observer Hith Logical Enemy Response* (Robot Observador Programable Con Respuesta Lógica al Enemigo). La respuesta lógica al enemigo es tan lacónica como dos ametralladoras M60.

Retomando el tema de la memoria, ésta es sin duda algo de cuya emulación tenemos clara conciencia en la Informática. En seguida la asociamos con ROMs, RAMs y otros conceptos básicos aprendidos. No obstante, la memoria humana, además de extraordinariamente vasta, dada la cantidad de datos heterogéneos que puede almacenar, es portentosamente dinámica. Y en su dinamismo entra directamente en juego el olvido.

Dejando al margen el olvido patológico, es decir, aquel producido por alguna causa física externa (amnesia), son dos las teorías fundamentales para la explicación del olvido natural: pasiva y activa.

Según la primera corriente, encabezada por el psicólogo francés Ribot, somos sujetos pasivos víctimas del olvido, sin que podamos hacer nada por evitarlo o manipularlo conforme a nuestras necesidades. Por el contrario, Freud opina que el individuo maneja su olvido.

Dé hecho, es frecuente decir que lo malo se olvida pronto, y en efecto tenemos el acierto de recordar lo agradable de una determinada experiencia, tendiendo a olvidar lo desagradable.

No obstante, también existe un olvido real involuntario, aun de aquellas cosas que deseamos recordar; por ello, la verdad parece estar en una pequeña mezcla de las dos teorías. Así pues, unido al olvido

voluntario, existe una cierta degeneración de la memoria en relación directa al tiempo que lleva almacenada la información. La curva de Ebbinghaus (fig. 4.1.12) es un ejemplo de ello, aunque no es en absoluto universalizable y es sólo aplicable a la memorización de una lista de palabras bajo determinadas circunstancias.

Este tipo de mecanismo podría ser también adoptado para el almacenamiento masivo por ordenador de conocimientos de llamemos segundo orden, de forma que se pudieran desechar los más antiguos en pro de los recién llegados por motivos de escasez de memoria.

## EL ROBOT

Con el programa ROBOT tendremos un primer contacto con la inteligencia artificial. Este programa ilustra la relación de un supuesto robot con el mundo exterior, a través de mecanismos perceptivos in-existentes que serán simulados en la pantalla.

Así pues, supongamos que nos encontramos manejando un sofisticado robot, capaz de reconocer los colores de tres cubos situados sobre una mesa (azul, naranja y verde, por orden de claridad en un monitor monocromo) y desde luego de manipularlos de acuerdo con las órdenes recibidas de un humano a través del teclado.

Descubriremos que el autómatas aparenta una cierta inteligencia, no sólo por entender nuestras órdenes sin necesidad de atenernos a unas normas de sintaxis muy severas, sino por ser capaz de efectuar una manipulación de cierta complejidad con habilidad, e incluso con pericia. Hay que tener en cuenta que este tipo de manejo de volúmenes es el que aprenden los niños con los juegos de construcciones.

En realidad, el programa está basado en uno de los clásicos de la inteligencia artificial: el SHRDLU. Este programa con tan complicado nombre fue creado en 1971 por Terry Winograd y servía para demostrar como una máquina podía aparentar inteligencia moviéndose en un entorno bastante concreto, en este caso, la manipulación de una serie de figuras geométricas sobre una mesa.

Las acciones que podía llegar a realizar SHRDLU eran realmente complejas, como lo demuestra este fragmento de una conversación cualquiera entre Terry y su robot imaginario:

coge el cubo azul

CUAL DE ELLOS?

el grande

DE ACUERDO. PERO PREVIAMENTE HE DE APARTAR LA PIRAMIDE QUE TIENE ENCIMA

pon el cubo pequeño azul dentro de la caja roja

DE ACUERDO

cuántos bloques hay fuera de la caja?

TRES

hay algún bloque grande detrás de la pirámide

SI, EL BLOQUE GRANDE ROJO

puede una pirámide sostener un cubo?

NO

quién sostiene la pirámide?

LA MESA

cuántas cosas hay encima de los bloques rojos

NO ESTOY MUY SEGURO DEL SIGNIFICADO DE LA  
EXPRESION «ENCIMA» EN LA FRASE «ENCIMA DE  
LOS CUBOS ROJOS» QUIERES DECIR:

1 - INMEDIATAMENTE ENCIMA

2 - POR ENCIMA A CUALQUIER ALTURA

inmediatamente encima

DE ACUERDO

has cogido algún cubo antes de mover el rojo

SI. EL AZUL

cuándo lo cogiste

CUANDO ESTABA PONIENDO EL CUBO ROJO DE-  
TRAS DE LA PIRAMIDE AZUL

por qué hiciste eso?

PARA APARTARLO

por qué querías apartarlo?

PARA PODER MOVER EL CUBO ROJO

por qué querías mover el cubo rojo?

PARA PONERLO DETRAS DE LA PIRAMIDE AZUL

por qué querías hacer eso?

PORQUE TU ME LO ORDENASTE

No pretendemos que nuestro robot 'VODBIR' (Visto Oscuro Derecha aBajo Izquierda aRibera) se parezca al SHRDLU original, salvo en su extrañísimo nombre; no obstante, podremos pasar un rato entretenido comprobando cómo comprende nuestras órdenes y manipula los cubos informándonos de los pasos que sigue en la consecución de su tarea.

## CÓMO TRABAJA 'VODBIR'

En lo referente al proceso del lenguaje natural, el primer paso

seguido por VODBIR es cambiar a mayúsculas la frase entrada y desproveerla de cualquier carácter diferente a una letra. Acto seguido, toma la frase y la descompone en posibles «sub-órdenes» buscando palabras como Y o TAMBIEN; de esta forma, el robot es capaz de aceptar órdenes compuestas, como por ejemplo:

pon el cubo azul sobre el naranja y deja el amarillo encima del azul

Una vez tomada la frase simple a analizar (pon el cubo azul sobre el naranja) se procede a analizar el tipo de manipulación requerida, que será de dos clases fundamentales: tomar un cubo o dejarlo en un determinado lugar. El modo de orden más compleja (pon el cubo... sobre...) se entiende como una composición de las dos órdenes básicas. Así por ejemplo, las frases siguientes son equivalentes:

pon el cubo azul sobre el naranja  
coge el cubo azul y déjalo sobre el naranja

Si la acción a ejecutar no es reconocida, el programa responderá con un tajante «NO ENTIENDO EL SIGNIFICADO DE LA EXPRESION ‘...’». Por el contrario, una vez comprendida la acción, se pasa a evaluar el cubo que es objeto del movimiento, suponiendo que es el que tiene en las manos en caso de que no se especifique otra cosa:

coge el cubo naranja  
COJO EL CUBO NARANJA  
déjalo sobre el azul  
DEJO EL CUBO SOBRE EL CUBO AZUL  
deja el cubo  
NO SE A QUE CUBO TE REFIERES  
toma el cubo naranja  
COJO EL CUBO NARANJA  
deja el cubo  
DEJO EL CUBO

Cumplir determinadas órdenes le puede llevar a un conflicto, como por ejemplo coger cubos que se encuentran debajo de otros; en este caso el problema lo resuelve de forma bastante interesante: se «autoordena» las operaciones previas a realizar, descompuestas de nuevo en las dos básicas. Así por ejemplo, si sobre el cubo azul está el verde y le ordenamos que sitúe encima del primero el que sostiene

(naranja), dejará el cubo naranja, que ya tiene en las manos, cogerá el verde, lo apartará y entonces retomaré la orden inicial, volviendo a coger el naranja, y ahora sí depositándolo sobre el azul.

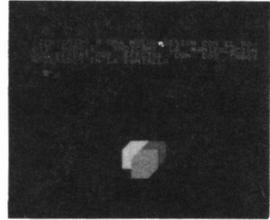
Como podemos comprobar, el primer movimiento ha sido bastante estúpido, pero se ha debido a que no ha tomado conciencia del problema que le ha ocasionado la orden hasta que no ha cumplido su primera fase.

Por contra, en ocasiones VODBIR se comporta de una forma excesivamente eficaz, con el sano fin de evitarse problemas. Así por ejemplo, si le pedimos que sitúe el cubo azul bajo el naranja, preferirá emplazar el segundo sobre el primero.

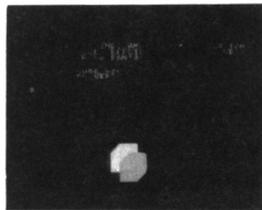
Por lo demás, nuestro robot es bastante competente, y además de situar los cubos a la izquierda, derecha, delante, detrás o encima de otro, admite un lenguaje suficientemente flexible, aunque desde luego siempre dentro del restringido ámbito de la manipulación de bloques.

Para terminar, veamos un ejemplo de conversación mantenida con 'VODBIR'. Las órdenes han sido tecleadas en minúsculas y las respuestas del ordenador en mayúsculas.

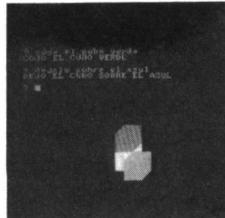
BIENVENIDO A UNA NUEVA SESION CON EL ROBOT 'VODBIR'. INDICAME LA MANIPULACION QUE DESEAS QUE EFECTUE CON LOS CUBOS MOSTRADOS EN LA PANTALLA.



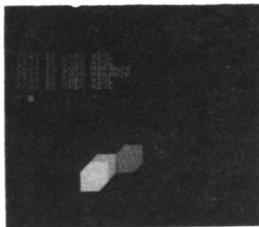
coge el cubo verde  
COJO EL CUBO VERDE



dejalo sobre el azul  
DEJO EL CUBO SOBRE EL AZUL



toma el cubo azul y ponlo delante del naranja  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL  
COJO EL CUBO NARANJA  
DEJO EL CUBO NARANJA  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL



ponlo delante del verde  
NO SE A QUE CUBO TE REFIERES

coge el cubo verde y dejalo a la derecha del naranja  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE



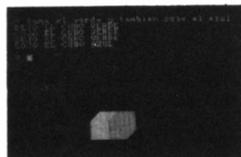
cogrlo  
NO COMPRENDO LA EXPRESION 'COGRLO'

cogelo  
NO SE A QUE CUBO TE REFIERES

toma el verde y tambien el azul  
COJO EL CUBO VERDE  
NO COMPRENDO LA EXPRESION 'EL AZUL'



toma el verde y tambien coge el azul  
DEJO EL CUBO VERDE  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE  
COJO EL CUBO AZUL



actuaste de una manera algo estúpida no crees  
NO COMPRENDO LA EXPRESION 'ACTUASTE DE UNA MANERA ALGO ESTUPIDA NO CREES'

deja el cubo verde  
TENGO EN LAS MANOS EL CUBO AZUL

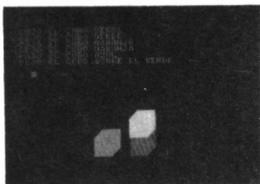
dejalo  
DEJO EL CUBO AZUL



deja el bloque naranja encima del azul y  
pon el verde encima del naranja  
COJO EL CUBO NARANJA  
DEJO EL CUBO SOBRE EL AZUL  
COJO EL CUBO VERDE  
DEJO EL CUBO SOBRE EL NARANJA



pon el cubo verde debajo del azul  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE



DEJO EL CUBO SOBRE EL NARANJA

pon el cubo verde debajo del azul  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE  
COJO EL CUBO NARANJA  
DEJO EL CUBO NARANJA  
COJO EL CUBO AZUL  
DEJO EL CUBO SOBRE EL VERDE



deja el cubo azul a la izquierda del nar-  
anja  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL  
COJO EL CUBO NARANJA  
DEJO EL CUBO NARANJA  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL

deja el cubo verde detras del naranja y  
pon el cubo azul delante del naranja  
COJO EL CUBO VERDE  
DEJO EL CUBO VERDE  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL  
COJO EL CUBO NARANJA  
DEJO EL CUBO NARANJA  
COJO EL CUBO AZUL  
DEJO EL CUBO AZUL

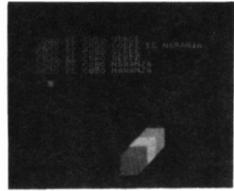


deja el cubo sobre la mesa  
NO SE A QUE CUBO TE REFIERES

coge el cubo verde  
COJO EL CUBO VERDE

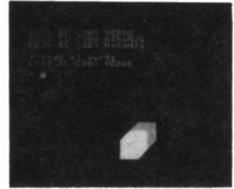
dejalo a la derecha del verde  
 NO PUEDO DEJAR EL CUBO AL LADO DE SI MIS  
 MO

tiralolo en la mesa  
 DEJO EL CUBO VERDE



deja el bloque azul encima de la figura  
 naranja ademas deja el verde sobre el na  
 ranja y tambien mueve el naranja

COJO EL CUBO AZUL  
 DEJO EL CUBO SOBRE EL NARANJA  
 COJO EL CUBO VERDE  
 DEJO EL CUBO VERDE  
 COJO EL CUBO AZUL  
 DEJO EL CUBO AZUL  
 COJO EL CUBO VERDE  
 DEJO EL CUBO SOBRE EL NARANJA  
 COJO EL CUBO VERDE  
 DEJO EL CUBO VERDE  
 COJO EL CUBO NARANJA  
 DEJO EL CUBO NARANJA



coge el verde  
 COJO EL CUBO VERDE

coge el azul  
 DEJO EL CUBO VERDE  
 COJO EL CUBO AZUL

hasta pronto  
 NO COMPRENDO LA EXPRESION 'HASTA PRONTO'



#### 4.1.28. Programa ROBOT versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /--- ROBOT VODBIR (c)1986 GRAN BIBLIOTECA AMSTRAD ---
40 /----- CPC-464 -----
50 /-----
60 /
70 DIM m(4,3,3):n$(1)="NARANJA":n$(2)="AZUL":n$(3)="VERDE":I
NK 1,15:INK 2,20:INK 3,9:MODE 1:WINDOW 1,40,1,8:WINDOW#1,1,4
0,9,25
80 FOR i=1 TO 3
90 p=INT(RND*2+0.5)+1:a=INT(RND*2+0.5)+1
100 IF m(1,a,p)=0 THEN m(1,a,p)=i:GOTO 130
110 IF m(2,a,p)=0 THEN m(2,a,p)=i:GOTO 130
120 m(3,a,p)=i
130 NEXT
140 GOSUB 1230
150 PRINT"BIENVENIDO A UNA NUEVA SESION CON EL RO-";

```

```

160 PRINT'BOT 'VODBIR'. INDICAME LA MANIPULACION';
170 PRINT'QUE DESEAS QUE EFECTUE CON LOS CUBOS';
180 PRINT'MOSTRADOS EN LA PANTALLA.'
190 PRINT CHR$(7)
200 INPUT f$:f%=UPPER$(f%)
210 i=1:WHILE i<=LEN(f%):a=ASC(MID$(f%,i,1)):IF (a<65 OR a>9
0) AND a<>32 THEN f%=LEFT$(f%,i-1)+MID$(f%,i+1) ELSE i=i+1
220 WEND
230 IF LEFT$(f%,1)=' ' THEN f%=MID$(f%,2):GOTO 230
240 IF f%='' THEN 190
250 RESTORE 260:FOR i=1 TO 5:READ b$:s=INSTR(f%,b%):IF s<>0
THEN 270 ELSE NEXT
260 DATA Y ADEMAS,Y TAMBIEN,ADEMAS,TAMBIEN,Y
270 IF s=0 THEN ff%=f%:f%='':GOTO 290
280 ff%=LEFT$(f%,s-1):f%=MID$(f%,s+LEN(b%))
290 IF LEFT$(ff%,1)=' ' THEN ff%=MID$(ff%,2):GOTO 290
300 RESTORE 310:FOR i=1 TO 8:READ b$:IF LEFT$(ff%,LEN(b%))=b
% THEN ff%=MID$(ff%,LEN(b%)+1):GOTO 320 ELSE NEXT:GOTO 470
310 DATA COGE,TOMA,AGARRA,TEN,HAZTE CON,ATRAPA,CAPTURA,QUITA
320 IF LEFT$(ff%,1)=' ' THEN ff%=MID$(ff%,2):GOTO 320
330 GOSUB 1480
340 FOR i=1 TO 3:IF LEFT$(ff%,LEN(n$(i)))=n$(i) THEN 360 ELS
E NEXT
350 PRINT'NO SE A QUE CUBO TE REFIERES':GOTO 190
360 IF m=0 THEN 390
370 f%=m:GOSUB 1360
380 f%='COGE '+ff%+' Y '+f%:ff%='DEJA '+n%:GOTO 300
390 f=:ci=i:GOSUB 1360
400 IF m(h+1,a,p)=0 THEN 430
410 f%=m(h+1,a,p):GOSUB 1360
420 f%='DEJA '+n%+' Y COGE '+ff%+' Y '+f%:ff%='COGE '+n%:GOT
O 300
430 f=:ci=m:GOSUB 1360:m(h,a,p)=0:PRINT'COJO EL CUBO 'n%
440 GOSUB 1230
450 IF f%='' THEN 190
460 GOTO 230
470 RESTORE 480:FOR i=1 TO 9:READ b$:IF LEFT$(ff%,LEN(b%))=b
% THEN ff%=MID$(ff%,LEN(b%)+1):GOTO 490 ELSE NEXT:GOTO 1220
480 DATA DEJA,SUELTA,PON,UBICA,SITUA,DESPLAZA,MUEVE,COLOCA,T
IRA
490 IF LEFT$(ff%,1)=' ' THEN ff%=MID$(ff%,2):GOTO 490
500 GOSUB 1480
510 FOR i=1 TO 3:IF LEFT$(ff%,LEN(n$(i)))=n$(i) THEN 550 ELS
E NEXT
520 IF m<>0 THEN i=m:GOTO 560
530 PRINT'NO SE A QUE CUBO TE REFIERES'
540 GOTO 190
550 IF m=0 THEN f%='DEJA '+ff%+' Y '+f%:ff%='COGE '+ff%:GOTO
300
560 s=INSTR(ff%,' ')
570 IF s=0 THEN ff%='':GOTO 590
580 ff%=MID$(ff%,s+1)
590 IF i=m THEN 620
600 n%=n$(m)
610 IF ff%='' THEN PRINT'TENGO EN LAS MANOS EL CUBO '+n%:GOT
O 190
620 IF ff%<>'' THEN 720
630 IF hor+ver=0 THEN 690
640 IF hor=0 THEN 670
650 med=med+1:IF med=1 THEN 690
660 a=2:FOR p=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
670 med=med+1:IF med=1 THEN 690
680 p=2:FOR a=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
690 a=INT(RND*2+0.5)+1:p=INT(RND*2+0.5)+1:IF m(1,a,p)<>0 THE
N 690
700 m(1,a,p)=m

```

```

710 f=m:GOSUB 1360:PRINT'DEJO EL CUBO 'n$:m=0:GOTO 440
720 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 720
730 RESTORE 740:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 750 ELSE NEXT:GOTO 880
740 DATA POR ENCIMA DEL,POR ENCIMA DE,ENCIMA DEL,ENCIMA DE,S
OBRE,EN
750 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 750
760 GOSUB 1480
770 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 800 ELS
E NEXT
780 IF LEFT$(ff$,4)='MESA' THEN 630
790 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
800 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO SOBRE SI MISMO'
:GOTO 190
810 f=i:GOSUB 1360
820 IF m(h+1,a,p)<>0 THEN 850
830 PRINT'DEJO EL CUBO SOBRE EL '+n$
840 m(h+1,a,p)=m:m=0:GOTO 440
850 an$=n$:f=m(h+1,a,p):GOSUB 1360:mem$=n$(m)
860 f$='COGE '+n$+' Y DEJALO Y COGE '+mem$+' Y DEJALO SOBRE
'+an$+' Y '+f$:ff$='DEJALO':GOTO 300
870 GOTO 440
880 RESTORE 890:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 900 ELSE NEXT:GOTO 970
890 DATA POR DEBAJO DEL,POR DEBAJO DE,DEBAJO DEL,DEBAJO DE,D
EBAJO,BAJO
900 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 900
910 GOSUB 1480
920 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 940 ELS
E NEXT
930 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
940 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO BAJO SI MISMO':
GOTO 190
950 f=i:GOSUB 1360:mem$=n$(m)
960 f$='COGE '+ff$+' Y DEJALO SOBRE'+mem$+' Y '+f$:ff$='DEJA
LO':GOTO 300
970 ex$=ff$
980 RESTORE 990:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1000 ELSE NEXT:GOTO 1010
990 DATA A LA IZQUIERDA DEL,A LA IZQUIERDA DE,A IZQUIERDA DE
L,A IZQUIERDA DE,POR LA IZQUIERDA DEL,POR LA IZQUIERDA DE,IZ
QUIERDA DEL,IZQUIERDA DE,IZQUIERDA
1000 hor=-1:ver=0:GOTO 1100
1010 RESTORE 1020:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1030 ELSE NEXT:GOTO 10
40
1020 DATA A LA DERECHA DEL,A LA DERECHA DE,A DERECHA DEL,A D
ERECHA DE,POR LA DERECHA DEL,POR LA DERECHA DE,DERECHA DEL,D
ERECHA DE,DERECHA
1030 hor=1:ver=0:GOTO 1100
1040 RESTORE 1050:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1060 ELSE NEXT:GOTO 10
70
1050 DATA POR DETRAS DEL,POR DETRAS DE,DETRAS DEL,DETRAS DE,
DETRAS,ATRAS
1060 hor=0:ver=1:GOTO 1100
1070 RESTORE 1080:FOR i=1 TO 10:READ b$:IF LEFT$(ff$,LEN(b$)
)=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1090 ELSE NEXT:GOTO 1
220
1080 DATA POR DELANTE DEL,POR DELANTE DE,DELANTE DEL,DELANTE
DE,DELANTE,POR ALANTE DEL,POR ALANTE DE,ALANTE DEL,ALANTE D
E,ALANTE
1090 hor=0:ver=-1
1100 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1100
1110 GOSUB 1480
1120 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 1140 E

```

```

LSE NEXT
1130 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
1140 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO AL LADO DE SI
MISMO':GOTO 190
1150 f=i:GOSUB 1360:ch=hica=a:cp=p
1160 IF hor+a<4 AND hor+a>0 AND ver+p<4 AND ver+p>0 THEN 118
0
1170 mem=n$(m):n=n$(m(ch,ca,cp)):f$='COGE '+n$+' Y DEJALO
Y COGE '+mem$+' Y DEJALO '+ex$+' Y '+f$:ff$='DEJALO':GOTO 30
0
1180 FOR h=1 TO 3:IF m(h,hor+a,ver+p)=0 THEN 1190 ELSE NEXT
1190 m(h,hor+a,ver+p)=m:n=n$(m):m=0:hor=0:ver=0:med=0
1200 PRINT'DEJO EL CUBO 'n$
1210 GOTO 440
1220 PRINT'NO COMPRENDO LA EXPRESION ''+ff$+'':GOTO 190
1230 CLS#1
1240 FOR h=1 TO 3
1250 FOR a=1 TO 3
1260 FOR p=3 TO 1 STEP -1
1270 f=m(h,a,p)
1280 IF f=0 THEN 1340
1290 x=50*a+25*p+100
1300 y=50*h+25*p-50
1310 MOVE x,y,f:DRAW 50,0:DRAW 0,50:DRAW -50,0:DRAW 0,-5
0:MOVE x+45,y+2:FILL f
1320 MOVE x+50,y+50:DRAW 25,25:DRAW -50,0:DRAW -25,-25:MO
VE x+45,y+52:FILL f
1330 MOVE x+50,y:DRAW 25,25:DRAW 0,50:MOVE x+54,y+8:FILL f
1340 NEXT p,a,h
1350 RETURN
1360 h=0
1370 FOR i=1 TO 3
1380 FOR j=1 TO 3
1390 FOR k=1 TO 3
1400 IF m(i,j,k)<>f THEN 1460
1410 h=i:a=j:p=k
1420 x=50*a+25*p+100
1430 y=50*h+25*p-50
1440 n=n$(f)
1450 k=3:j=3:i=3
1460 NEXT k,j,i
1470 RETURN
1480 RESTORE 1490:FOR i=1 TO 2:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1500 ELSE NEXT
1490 DATA EL,LA
1500 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1500
1510 RESTORE 1520:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1530 ELSE NEXT
1520 DATA BLOQUE,PIEZA,CUBO,FIGURA,CUADRADO,DADO
1530 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1530
1540 RETURN

```

#### 4.1.29. Programa ROBOT versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /--- ROBOT VOOBIR (c)1986 GRAN BIBLIOTECA AMSTRAD ----
40 /----- CPC 664/6128 -----
50 /-----
60 /
70 DIM m(4,3,3):n$(1)="NARANJA":n$(2)="AZUL":n$(3)="VERDE":I
NK 1,15:INK 2,20:INK 3,9:MODE 1:WINDOW 1,40,1,8:WINDOW#1,1,4
0,9,25

```

```

80 FOR i=1 TO 3
90 p=INT(RND*2+0.5)+1:a=INT(RND*2+0.5)+1
100 IF m(1,a,p)=0 THEN m(1,a,p)=i:GOTO 130
110 IF m(2,a,p)=0 THEN m(2,a,p)=i:GOTO 130
120 m(3,a,p)=i
130 NEXT
140 GOSUB 1230
150 PRINT'BIENVENIDO A UNA NUEVA SESION CON EL RO-';
160 PRINT'BOT 'VODBIR'. INDICAME LA MANIPULACION';
170 PRINT'QUE DESEAS QUE EFECTUE CON LOS CUBOS';
180 PRINT'MOSTRADOS EN LA PANTALLA.'
190 PRINT CHR$(7)
200 INPUT f$:f$=UPPER$(f$)
210 i=1:WHILE i<=LEN(f$):a=ASC(MID$(f$,i,1)):IF (a<65 OR a>90) AND a<>32 THEN f$=LEFT$(f$,i-1)+MID$(f$,i+1) ELSE i=i+1
220 WEND
230 IF LEFT$(f$,1)=' ' THEN f$=MID$(f$,2):GOTO 230
240 IF f$='' THEN 190
250 RESTORE 260:FOR i=1 TO 5:READ b$:s=INSTR(f$,b$):IF s<>0 THEN 270 ELSE NEXT
260 DATA Y ADEMAS,Y TAMBIEN,ADEMAS,TAMBIEN,Y
270 IF s=0 THEN ff$=f$:f$='':GOTO 290
280 ff$=LEFT$(f$,s-1):f$=MID$(f$,s+LEN(b$))
290 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 290
300 RESTORE 310:FOR i=1 TO 8:READ b$:IF LEFT$(ff$,LEN(b$))=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 320 ELSE NEXT:GOTO 470
310 DATA COGE,TOMA,AGARRA,TEN,HAZTE CON,ATRAPA,CAPTURA,QUITA
320 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 320
330 GOSUB 1480
340 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 360 ELSE NEXT
350 PRINT'NO SE A QUE CUBO TE REFIERES':GOTO 190
360 IF m=0 THEN 390
370 f=m:GOSUB 1360
380 f$='COGE '+fff$+' Y '+f$:fff$='DEJA '+n$:GOTO 300
390 f=i:ci=i:GOSUB 1360
400 IF m(h+1,a,p)=0 THEN 430
410 f=m(h+1,a,p):GOSUB 1360
420 f$='DEJA '+n$+' Y COGE '+fff$+' Y '+f$:fff$='COGE '+n$:GOTO 300
430 f=ci:m=ci:GOSUB 1360:m(h,a,p)=0:PRINT'COJO EL CUBO 'n$
440 GOSUB 1230
450 IF f$='' THEN 190
460 GOTO 230
470 RESTORE 480:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 490 ELSE NEXT:GOTO 1220
480 DATA DEJA,SUELTA,PON,UBICA,SITUA,DESPLAZA,MUEVE,COLOCA,T IRA
490 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 490
500 GOSUB 1480
510 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 550 ELSE NEXT
520 IF m<>0 THEN i=m:GOTO 560
530 PRINT'NO SE A QUE CUBO TE REFIERES'
540 GOTO 190
550 IF m=0 THEN f$='DEJA '+fff$+' Y '+f$:fff$='COGE '+fff$:GOTO 300
560 s=INSTR(ff$, ' ')
570 IF s=0 THEN ff$='':GOTO 590
580 ff$=MID$(ff$,s+1)
590 IF i=m THEN 620
600 n$=n$(m)
610 IF ff$='' THEN PRINT'TENGO EN LAS MANOS EL CUBO '+n$:GOTO 190
620 IF ff$<>' ' THEN 720
630 IF hor+ver=0 THEN 690
640 IF hor=0 THEN 670

```

```

650 med=med+1:IF med=1 THEN 690
660 a=2:FOR p=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
670 med=med+1:IF med=1 THEN 690
680 p=2:FOR a=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
690 a=INT(RND*2+0.5)+1:p=INT(RND*2+0.5)+1:IF m(1,a,p)<>0 THE
N 690
700 m(1,a,p)=m
710 f=m:GOSUB 1360:PRINT'DEJO EL CUBO 'n$:m=0:GOTO 440
720 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 720
730 RESTORE 740:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 750 ELSE NEXT:GOTO 880
740 DATA POR ENCIMA DEL,POR ENCIMA DE,ENCIMA DEL,ENCIMA DE,S
OBRE,EN
750 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 750
760 GOSUB 1480
770 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 800 ELS
E NEXT
780 IF LEFT$(ff$,4)='MESA' THEN 630
790 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
800 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO SOBRE SI MISMO'
:GOTO 190
810 f=i:GOSUB 1360
820 IF m(h+1,a,p)<>0 THEN 850
830 PRINT'DEJO EL CUBO SOBRE EL '+n$
840 m(h+1,a,p)=m:m=0:GOTO 440
850 an$=n$:f=m(h+1,a,p):GOSUB 1360:mem$=n$(m)
860 f$='COGE '+n$+' Y DEJALO Y COGE '+mem$+' Y DEJALO SOBRE
'+an$+' Y '+f$:ff$='DEJALO':GOTO 300
870 GOTO 440
880 RESTORE 890:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 900 ELSE NEXT:GOTO 970
890 DATA POR DEBAJO DEL,POR DEBAJO DE,DEBAJO DEL,DEBAJO DE,D
EBAJO,BAJO
900 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 900
910 GOSUB 1480
920 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 940 ELS
E NEXT
930 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
940 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO BAJO SI MISMO':
GOTO 190
950 f=i:GOSUB 1360:mem$=n$(m)
960 f$='COGE '+ff$+' Y DEJALO SOBRE'+mem$+' Y '+f$:ff$='DEJA
LO':GOTO 300
970 ex$=ff$
980 RESTORE 990:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1000 ELSE NEXT:GOTO 1010
990 DATA A LA IZQUIERDA DEL,A LA IZQUIERDA DE,A IZQUIERDA DE
L,A IZQUIERDA DE,POR LA IZQUIERDA DEL,POR LA IZQUIERDA DE,IZ
QUIERDA DEL,IZQUIERDA DE,IZQUIERDA
1000 hor=-1:ver=0:GOTO 1100
1010 RESTORE 1020:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1030 ELSE NEXT:GOTO 10
40
1020 DATA A LA DERECHA DEL,A LA DERECHA DE,A DERECHA DEL,A D
ERECHA DE,POR LA DERECHA DEL,POR LA DERECHA DE,DERECHA DEL,D
ERECHA DE,DERECHA
1030 hor=1:ver=0:GOTO 1100
1040 RESTORE 1050:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1060 ELSE NEXT:GOTO 10
70
1050 DATA POR DETRAS DEL,POR DETRAS DE,DETRAS DEL,DETRAS DE,
DETRAS,ATRAS
1060 hor=0:ver=1:GOTO 1100
1070 RESTORE 1080:FOR i=1 TO 10:READ b$:IF LEFT$(ff$,LEN(b$)
)=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1090 ELSE NEXT:GOTO 1
220
1080 DATA POR DELANTE DEL,POR DELANTE DE,DELANTE DEL,DELANTE

```

```

DE,DELANTE,POR ALANTE DEL,POR ALANTE DE,ALANTE DEL,ALANTE D
E,ALANTE
1090 hor=0:ver=-1
1100 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1100
1110 GOSUB 1480
1120 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 1140 E
LSE NEXT
1130 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
1140 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO AL LADO DE SI
MISMO':GOTO 190
1150 f=i:GOSUB 1360:ch=h:ca=a:cp=p
1160 IF hor+a<4 AND hor+a>0 AND ver+p<4 AND ver+p>0 THEN 118
0
1170 mem$=n$(m):n$=n$(m(ch,ca,cp)):f$='COGE '+n$+' Y DEJALO
Y COGE '+mem$+' Y DEJALO '+ex$+' Y '+f$:ff$='DEJALO':GOTO 30
0
1180 FOR h=1 TO 3:IF m(h,hor+a,ver+p)=0 THEN 1190 ELSE NEXT
1190 m(h,hor+a,ver+p)=m:n$=n$(m):m=0:hor=0:ver=0:med=0
1200 PRINT'DEJO EL CUBO 'n$
1210 GOTO 440
1220 PRINT'NO COMPRENDO LA EXPRESION ''+ff$+'':GOTO 190
1230 CLS#1
1240 FOR h=1 TO 3
1250 FOR a=1 TO 3
1260 FOR p=3 TO 1 STEP -1
1270 f=m(h,a,p)
1280 IF f=0 THEN 1340
1290 x=50*a+25*p+100
1300 y=50*h+25*p-50
1310 MOVE x,y,f:DRAW 50,0:DRAW 0,50:DRAW -50,0:DRAW 0,-5
0:MOVE x,y:DRAW 25,25:DRAW 50,0
1320 MOVE x+50,y+50:DRAW 25,25:DRAW -50,0:DRAW -25,-25:MO
VE x+25,y+25:DRAW 0,50
1330 MOVE x+50,y:DRAW 25,25:DRAW 0,50
1340 NEXT p,a,h
1350 RETURN
1360 h=0
1370 FOR i=1 TO 3
1380 FOR j=1 TO 3
1390 FOR k=1 TO 3
1400 IF m(i,j,k)<>f THEN 1460
1410 h=i:a=j:p=k
1420 x=50*a+25*p+100
1430 y=50*h+25*p-50
1440 n$=n$(f)
1450 k=3:j=3:i=3
1460 NEXT k,j,i
1470 RETURN
1480 RESTORE 1490:FOR i=1 TO 2:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1500 ELSE NEXT
1490 DATA EL,LA
1500 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1500
1510 RESTORE 1520:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1530 ELSE NEXT
1520 DATA BLOQUE,PIEZA,CUBO,FIGURA,CUADRADO,DADO
1530 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1530
1540 RETURN

```

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /--- ROBOT VODBIR (c)1986 GRAN BIBLIOTECA AMSTRAD ---
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70
80 k1s$=CHR$(27)+'E'+CHR$(27)+'H':PRINT k1s$
90 DIM m(4,3,3):n$(1)="NARANJA":n$(2)="AZUL":n$(3)="VERDE"
100 FOR i=1 TO 3
110 p=INT(RND*2+0,5)+1:a=INT(RND*2+0,5)+1
120 IF m(1,a,p)=0 THEN m(1,a,p)=i:GOTO 150
130 IF m(2,a,p)=0 THEN m(2,a,p)=i:GOTO 150
140 m(3,a,p)=i
150 NEXT
160 GOSUB 1230
170 PRINT'BIENVENIDO A UNA NUEVA SESION CON EL ROBOT 'VODBIR
' . INDICAME LA MANIPULACION QUE DESEAS';
180 PRINT'QUE EFECTUE CON LOS CUBOS MOSTRADOS EN LA PANTALLA
.
190 PRINT CHR$(7)
200 INPUT f$:f$=UPPER$(f$)
210 i=1:WHILE i<=LEN(f$):a=ASC(MID$(f$,i,1)):IF (a<65 OR a>9
0) AND a<>32 THEN f$=LEFT$(f$,i-1)+MID$(f$,i+1) ELSE i=i+1
220 WEND
230 IF LEFT$(f$,1)=' ' THEN f$=MID$(f$,2):GOTO 230
240 IF f$='' THEN 190
250 RESTORE 260:FOR i=1 TO 5:READ b$:s=INSTR(f$,b$):IF s<>0
THEN 270 ELSE NEXT
260 DATA Y ADEMAS,Y TAMBIEN,ADEMAS,TAMBIEN,Y
270 IF s=0 THEN ff$=f$:f$='':GOTO 290
280 ff$=LEFT$(f$,s-1):f$=MID$(f$,s+LEN(b$))
290 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 290
300 RESTORE 310:FOR i=1 TO 8:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 320 ELSE NEXT:GOTO 470
310 DATA COGE,TOMA,AGARRA,TEN,HAZTE CON,ATRAPA,CAPTURA,QUITA
320 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 320
330 GOSUB 1550
340 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 360 ELS
E NEXT
350 PRINT'NO SE A QUE CUBO TE REFIERES':GOTO 190
360 IF m=0 THEN 390
370 f=m:GOSUB 1430
380 f$='COGE '+ff$+' Y '+f$:ff$='DEJA '+n$:GOTO 300
390 f=i:ci=i:GOSUB 1430
400 IF m(h+1,a,p)=0 THEN 430
410 f=m(h+1,a,p):GOSUB 1430
420 f$='DEJA '+n$+' Y COGE '+ff$+' Y '+f$:ff$='COGE '+n$:GOT
O 300
430 f=ci:m=ci:GOSUB 1430:m(h,a,p)=0:PRINT'COJO EL CUBO 'n$
440 GOSUB 1230
450 IF f$='' THEN 190
460 GOTO 230
470 RESTORE 480:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 490 ELSE NEXT:GOTO 1220
480 DATA DEJA,SUELTA,PON,UBICA,SITUA,DESPLAZA,MUEVE,COLOCA,T
IRA
490 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 490
500 GOSUB 1550
510 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 550 ELS
E NEXT
520 IF m<>0 THEN i=m:GOTO 560
530 PRINT'NO SE A QUE CUBO TE REFIERES'
540 GOTO 190
550 IF m=0 THEN f$='DEJA '+ff$+' Y '+f$:ff$='COGE '+ff$:GOTO
300
560 s=INSTR(ff$, ' ')

```

```

570 IF s=0 THEN ff$="" :GOTO 590
580 ff$=MID$(ff$,s+1)
590 IF i=m THEN 620
600 n$=n$(m)
610 IF ff$="" THEN PRINT'TENGO EN LAS MANOS EL CUBO '+n$:GOT
0 190
620 IF ff$(<)>"" THEN 720
630 IF hor+ver=0 THEN 690
640 IF hor=0 THEN 670
650 med=med+1:IF med=1 THEN 690
660 a=2:FOR p=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
670 med=med+1:IF med=1 THEN 690
680 p=2:FOR a=1 TO 3:IF m(1,a,p)=0 THEN 700 ELSE NEXT
690 a=INT(RND*2+0.5)+1:p=INT(RND*2+0.5)+1:IF m(1,a,p)<>0 THE
N 690
700 m(1,a,p)=m
710 f=m:GOSUB 1430:PRINT'DEJO EL CUBO 'n$:m=0:GOTO 440
720 IF LEFT$(ff$,1)=" " THEN ff$=MID$(ff$,2):GOTO 720
730 RESTORE 740:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 750 ELSE NEXT:GOTO 880
740 DATA POR ENCIMA DEL,POR ENCIMA DE,ENCIMA DEL,ENCIMA DE,S
OBRE,EN
750 IF LEFT$(ff$,1)=" " THEN ff$=MID$(ff$,2):GOTO 750
760 GOSUB 1550
770 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 800 ELS
E NEXT
780 IF LEFT$(ff$,4)="MESA" THEN 630
790 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
800 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO SOBRE SI MISMO'
:GOTO 190
810 f=i:GOSUB 1430
820 IF m(h+1,a,p)<>0 THEN 850
830 PRINT'DEJO EL CUBO SOBRE EL '+n$
840 m(h+1,a,p)=m:m=0:GOTO 440
850 an$=n$:f=m(h+1,a,p):GOSUB 1430:mem$=n$(m)
860 *f$="COGE '+n$+' Y DEJALO Y COGE '+mem$+' Y DEJALO SOBRE
'+an$+' Y '+f$:ff$='DEJALO':GOTO 300
870 GOTO 440
880 RESTORE 890:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 900 ELSE NEXT:GOTO 970
890 DATA POR DEBAJO DEL,POR DEBAJO DE,DEBAJO DEL,DEBAJO DE,D
EBAJO,BAJO
900 IF LEFT$(ff$,1)=" " THEN ff$=MID$(ff$,2):GOTO 900
910 GOSUB 1550
920 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 940 ELS
E NEXT
930 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
940 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO BAJO SI MISMO'
:GOTO 190
950 f=i:GOSUB 1430:mem$=n$(m)
960 f$="COGE '+ff$+' Y DEJALO SOBRE'+mem$+' Y '+f$:ff$='DEJA
LO':GOTO 300
970 ex$=ff$
980 RESTORE 990:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))=b
$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1000 ELSE NEXT:GOTO 1010
990 DATA A LA IZQUIERDA DEL,A LA IZQUIERDA DE,A IZQUIERDA DE
L,A IZQUIERDA DE,POR LA IZQUIERDA DEL,POR LA IZQUIERDA DE,IZ
QUIERDA DEL,IZQUIERDA DE,IZQUIERDA
1000 hor=-1:ver=0:GOTO 1100
1010 RESTORE 1020:FOR i=1 TO 9:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1030 ELSE NEXT:GOTO 10
40
1020 DATA A LA DERECHA DEL,A LA DERECHA DE,A DERECHA DEL,A D
ERECHA DE,POR LA DERECHA DEL,POR LA DERECHA DE,DERECHA DEL,D
ERECHA DE,DERECHA
1030 hor=1:ver=0:GOTO 1100
1040 RESTORE 1050:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1060 ELSE NEXT:GOTO 10
70
1050 DATA POR DETRAS DEL,POR DETRAS DE,DETRAS DEL,DETRAS DE,
DETRAS,ATRAS

```

```

1060 hor=0:ver=1:GOTO 1100
1070 RESTORE 1080:FOR i=1 TO 10:READ b$:IF LEFT$(ff$,LEN(b$
)=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1090 ELSE NEXT:GOTO 1
220
1080 DATA POR DELANTE DEL,POR DELANTE DE,DELANTE DEL,DELANTE
DE,DELANTE,POR ALANTE DEL,POR DELANTE DE,ALANTE DEL,ALANTE D
E,ALANTE
1090 hor=0:ver=-1
1100 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1100
1110 GOSUB 1550
1120 FOR i=1 TO 3:IF LEFT$(ff$,LEN(n$(i)))=n$(i) THEN 1140 E
LSE NEXT
1130 PRINT'NO SE DONDE DEJAR EL CUBO':GOTO 190
1140 IF i=m THEN PRINT'NO PUEDO DEJAR EL CUBO AL LADO DE SI
MISMO':GOTO 190
1150 f=i:GOSUB 1430:ch=h:ca=a:cp=p
1160 IF hor+a<4 AND hor+a>0 AND ver+p<4 AND ver+p>0 THEN 118
0
1170 mem$=n$(m):n$=n$(m(ch,ca,cp)):f$='COGE '+n$+' Y DEJALO
Y COGE '+mem$+' Y DEJALO '+ex$+' Y '+f$:ff$='DEJALO':GOTO 30
0
1180 FOR h=1 TO 3:IF m(h,hor+a,ver+p)=0 THEN 1190 ELSE NEXT
1190 m(h,hor+a,ver+p)=m:n$=n$(m):m=0:hor=0:ver=0:med=0
1200 PRINT'DEJO EL CUBO '+n$
1210 GOTO 440
1220 PRINT'NO COMPRENDO LA EXPRESION '+ff$+'':GOTO 190
1230 PRINT CHR$(27)'j'
1240 PRINT CHR$(27)'X'CHR$(32)CHR$(32)CHR$(47)CHR$(122)
1250 PRINT kls$
1260 FOR h=3 TO 1 STEP -1
1270 PRINT TAB(41)'-----'
1280 FOR p=3 TO 1 STEP -1
1290 PRINT TAB(37+p)'/ ';
1300 FOR a=1 TO 3
1310 IF m(h,a,p)=0 THEN PRINT' ';
1320 IF m(h,a,p)=1 THEN PRINT'N';
1330 IF m(h,a,p)=2 THEN PRINT'A';
1340 IF m(h,a,p)=3 THEN PRINT'V';
1350 NEXT
1360 PRINT' /'
1370 NEXT
1380 PRINT TAB(37)'-----';
1390 NEXT
1400 PRINT CHR$(27)'X'CHR$(48)CHR$(32)CHR$(48)CHR$(122)
1410 PRINT CHR$(27)'k';
1420 RETURN
1430 h=0
1440 FOR i=1 TO 3
1450 FOR j=1 TO 3
1460 FOR k=1 TO 3
1470 IF m(i,j,k)<>f THEN 1530
1480 h=i:a=j:p=k
1490 x=50*a+25*p+100
1500 y=50*h+25*p-50
1510 n$=n$(f)
1520 k=3:j=3:i=3
1530 NEXT k,j,i
1540 RETURN
1550 RESTORE 1560:FOR i=1 TO 2:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1570 ELSE NEXT
1560 DATA EL,LA
1570 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1570
1580 RESTORE 1590:FOR i=1 TO 6:READ b$:IF LEFT$(ff$,LEN(b$))
=b$ THEN ff$=MID$(ff$,LEN(b$)+1):GOTO 1600 ELSE NEXT
1590 DATA BLOQUE,PIEZA,CUBO,FIGURA,CUADRADO,DADO
1600 IF LEFT$(ff$,1)=' ' THEN ff$=MID$(ff$,2):GOTO 1600
1610 RETURN

```

# LA INTELIGENCIA

**L**a inteligencia ha sido un tema estudiado por los hombres desde tiempos remotos, y así, en épocas con un especial florecimiento cultural, como por ejemplo la Grecia Antigua, a su tratamiento se dedicaron filósofos a lo largo de toda su vida. Frutos de esta continua investigación han sido teorías como la que a continuación se expone.

El hombre, mediante la asociación de tres funciones: concepto, juicio y razonamiento es capaz de extraer una conclusión inteligente a partir de unos datos, es decir, de discurrir una idea fruto de la razón sobre cualquier grupo de conceptos, no sólo tangibles, sino también abstractos. Estas tres funciones, también denominadas procesos lógicos, son las que conforman la llamada inteligencia abstracta, patrimonio exclusivo del hombre, y ausente en los animales.

No obstante, antes de continuar por este camino, existe una importante puntualización que efectuar ¿qué ocurre con la inteligencia de los animales? Frecuentemente éstos se comportan de una manera inteligente; así por ejemplo; el chimpancé que no puede alcanzar un plátano obstruido por los barrotes de su jaula, es capaz de ayudarse de un palo para atraparlo.

De hecho, no todos los animales son igualmente inteligentes. Existen realmente grandes diferencias entre un delfín, catalogado como el animal más inteligente que puebla nuestro mundo (después del hombre, por supuesto) y una gallina. Si por ejemplo situáramos comida fuera de una jaula, abierta por el extremo opuesto a la ubicación del alimento, observaríamos cómo un pájaro se obstinaría en golpearse contra la jaula intentando alcanzar las migas, revoloteando de un lado a otro, pero sin dar nunca la espalda ni perder de vista la comida. No obstante, un animal como el perro sería perfectamente capaz de escapar al encierro por la salida posterior para alcanzar su objetivo.

¿Nos encontramos ante un problema de inteligencia o simplemente de instinto? Cada uno es muy dueño de apreciar el problema desde su propia óptica, pero hay que considerar que una de las cualidades que habitualmente se han considerado distintivas de la inteligencia frente al instinto ha sido la del aprendizaje, la capacidad de adaptación a nuevas situaciones.

Una gallina continúa incubando sus huevos aunque éstos hayan sido sustituidos por piedras. Evidentemente, nos hallamos ante un comportamiento claramente instintivo, dado que la adaptabilidad a las circunstancias actuales brilla por su ausencia.

No obstante, animales de superior desarrollo cerebral son muy capaces de aprender y no sólo de esto, sino también de adaptar su comportamiento a las circunstancias reinantes. Si bien es cierto que los animales son capaces de actuar ante situaciones tan complejas como las contempladas por el cerebro humano, no es menos cierto que sí están dotados de una cierta inteligencia, denominada por algunos filósofos inteligencia práctica, en contraposición a la inteligencia abstracta de la que hace gala el género humano.

En este sentido, es importante hacer notar que algunos pensadores como Bergson distinguen dos fases en la trayectoria evolutiva del hombre: por una parte, se habla del *homo faber* y por otra del *homo sapiens*. Fundamentalmente, la diferencia entre ambos estriba en que el primero se movía a un nivel de inteligencia práctica, mientras que la segunda fase se considera a partir de que el hombre pudo desarrollar su inteligencia abstracta.

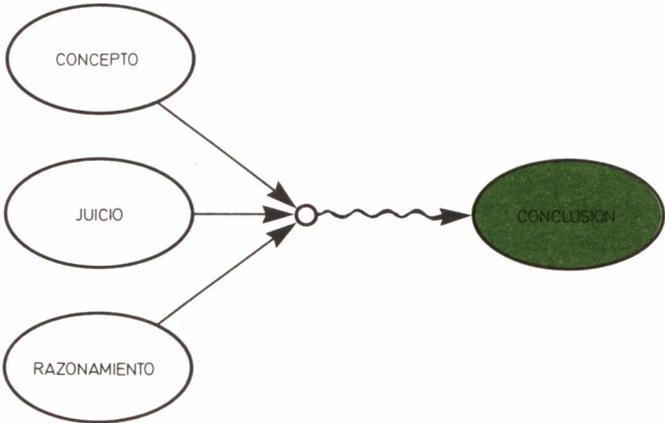
Leamos atentamente el siguiente texto, extraído del libro «Antropología» de E. Tylor: «Los salvajes sólo conocen la trayectoria de un proyectil lo bastante bien para hacer la puntería y saben cuánta mayor fuerza manda un hacha montada en un mango largo que en un mango corto; mas apenas pueden inducir de estas ideas prácticas un principio o ley.»

Nos encontramos ante los dos conceptos antes mencionados: la inteligencia práctica y la inteligencia abstracta. No obstante, estamos hablando de hombres del siglo XX, ¿es que los maoríes neozelandeses carecen de inteligencia? ¿No son hombres como los universitarios españoles? ¿No han llevado un proceso evolutivo comparativamente igual en cuanto al tiempo?

Evidentemente, restringir el uso de la palabra inteligencia a los universitarios occidentales es algo bastante deplorable y discriminatorio. En todo caso, podríamos llegar a afirmar que se comportan de un modo algo menos inteligente que los universitarios, de forma similar a como el delfín es más inteligente que la gallina, sin pretender por supuesto caer en el agravio comparativo.

Así pues, desde la completa carencia de inteligencia a la demostración más esplendorosa de la misma, existe un avance progresivo y no bruscos saltos desde la estupidez al instinto, de éste a la manifestación de la inteligencia práctica y por fin a la inteligencia abstracta.

El desarrollo de un punto a otro se obtiene a partir de múltiples factores. Algunos de ellos físicos, como la propia capacidad cerebral, o el porcentaje de células cerebrales que se ponen en funcionamiento; se dice que el hombre sólo aprovecha aproximadamente el 18 por 100 de las mismas. Por otra parte, además de estos factores puramente «hardware», nos encontramos con otros problemas como el desarrollo cultural, la adquisición de conocimientos mediante el aprendizaje, etc...



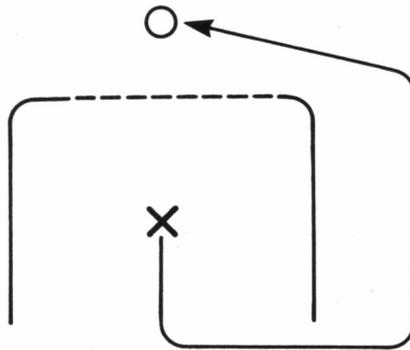
■ 4.2.1. El hombre extrae conclusiones a partir de la asociación de tres funciones: concepto, juicio y razonamiento.

Básicamente por tanto, un mismo bebé humano no alcanzará el mismo desarrollo de su inteligencia en algún lugar perdido de la selva amazónica, que en una familia acomodada española. Esto nos lleva a una conclusión de extraordinaria importancia: hablando de inteligencia no debemos acudir a generalizaciones o afirmaciones tajantes como «el hombre es inteligente» o «los ordenadores no son inteligentes». Por el contrario, hemos de reparar en otras consideraciones como ¿puede un ordenador llegar a mostrar inteligencia práctica? o más aún ¿puede alcanzar atisbos de inteligencia abstracta?

Por otra parte, existe una última diferencia entre la simulación de inteligencia en un ordenador y la realidad humana, y es la característica vertical del proceso evolutivo inteligente. Expliquemos esto mejor.

## INTELIGENCIA VERTICAL

Es fácil apreciar un cierto brote de inteligencia en una máquina, sobre todo práctica, pero sólo sobre un determinado aspecto relativamente restringido, mientras que su desconocimiento de otras materias es casi absoluto.



- ALIMENTO
- RECINTO OPACO
- - - RECINTO TRASLUCIDO
- X ANIMAL
- COMPORTAMIENTO INTELIGENTE
- COMPORTAMIENTO INSTINTIVO

4.2.2. *Algunos animales presentan actitudes inteligentes ante ciertos estímulos.*

Así, es muy posible que un complejo programa de ajedrez que nos acaba de propinar la paliza del siglo, sea un inútil completo en el manejo de cubos, aunque realmente sea más inteligente que el programa ROBOT 'VODBIR' con que finalizábamos el capítulo anterior.

En contraposición a este sistema, el humano tiene un desarrollo de la inteligencia «horizontal», ya que va profundizando más paralelamente en multitud de ramas y no exhaustivamente en una sola.

Evidentemente, el problema de simular la horizontalidad en un ordenador es de capacidad, y en la actualidad es prácticamente imposible imaginar una máquina con la suficiente memoria como para atesorar todos los conocimientos sobre la ingente cantidad de temas que un humano puede acumular durante su vida.

No obstante, aunque este problema de capacidad sea hoy por hoy insalvable, volviendo a la realidad del tratamiento de campos restringidos, vamos a comprobar con el siguiente programa como un Amstrad puede llegar a tratar problemas típicos de la inteligencia práctica.

En la ejecución de ROBOT 'VODBIR' observamos cómo su sistema de funcionamiento le hacía caer en el error de dar pasos innecesarios, como por ejemplo coger un cubo que inmediatamente tenía que dejar para apartar otro que le estorbaba. Veremos gracias al programa 'HANNOI' otro comportamiento del ordenador bastante más inteligente, dentro de la resolución de un problema práctico.

## LAS TORRES DE HANNOI

He aquí un típico juego que sólo puede ser resuelto de una manera inteligente, y que no implica ninguna complejidad en lo referente al reconocimiento o manejo de formas y volúmenes, sino que simplemente hace hincapié en los pasos seguidos para la resolución de un determinado problema.

El programa representa tres plataformas, en una de las cuales se hiergue una pila con un cierto número de bloques, que constituyen una construcción similar a una antigua pirámide azteca.

El problema consiste en trasladar la torre de la plataforma número tres (la de la derecha de la pantalla) en la que se encuentra inicialmente, a la plataforma uno (la de la izquierda de la pantalla). Este trabajo ha de ser efectuado en el menor número de movimientos posible, y sólo es necesario respetar tres reglas:

1. Se puede transportar un sólo bloque en cada movimiento.
2. No se puede desplazar ningún bloque que tenga por encima

algún otro en la misma torre; del mismo modo, cualquier bloque trasladado a una torre se sitúa encima de ésta.

3. En ningún momento ni torre se puede depositar un bloque de mayores dimensiones sobre otro más pequeño.

Así pues, por cada movimiento el programa nos solicitará dos datos: la torre de origen y la de destino. Esto quiere decir, que tomará el bloque superior de la torre de partida y lo depositará como último de la torre de llegada. Evidentemente, el movimiento será efectuado siempre y cuando sea posible, es decir, en caso de que el bloque superior de la torre de origen sea de menor tamaño que el de la de destino.

La plataforma intermedia nos servirá de apoyo indispensable en el trasiego de las piezas, y se atiene a todas las normas anteriormente explicadas sobre entrada, salida y condiciones de apilamiento de las mismas.

Al comienzo, el programa nos pregunta si deseamos ser nosotros quienes resolvamos el problema, o preferimos ser espectadores de los inteligentes movimientos del ordenador. Así mismo, nos interroga sobre el número de bloques de que se compone la torre, comprendido entre dos y siete, considerándose esta última una cantidad suficientemente alta e ilustrativa de la complejidad creciente que puede llegar a adquirir el problema. Además, en base a este último dato, el programa nos informa sobre el número óptimo de movimientos en que se puede llegar a resolver el problema.

## ¿CÓMO PIENSA EL ORDENADOR?

Sería conveniente que no leyéramos este apartado hasta haber introducido el programa y comprobado no sólo las aptitudes del ordenador para la resolución del problema, sino el considerable apuro en el que nos podemos ver, al intentar nosotros mismos solucionarlo con una torre de cuatro o cinco pisos. Ni que decir tiene, que sin leer el resto de este epígrafe, resolver el problema con siete alturas en el número óptimo de jugadas es prácticamente imposible.

El hecho de que el programa en base a la altura, nos indique inmediatamente el número óptimo de jugadas para la resolución del problema, nos habrá hecho pensar sin duda en lo sencilla que puede ser la solución del mismo. Tan sólo le hace falta restar uno, a dos elevado al número de pisos, para saber la cantidad de movimientos que es imprescindible efectuar ( $2 \uparrow h-1$ , donde  $h$  es el número de bloques que componen la torre).

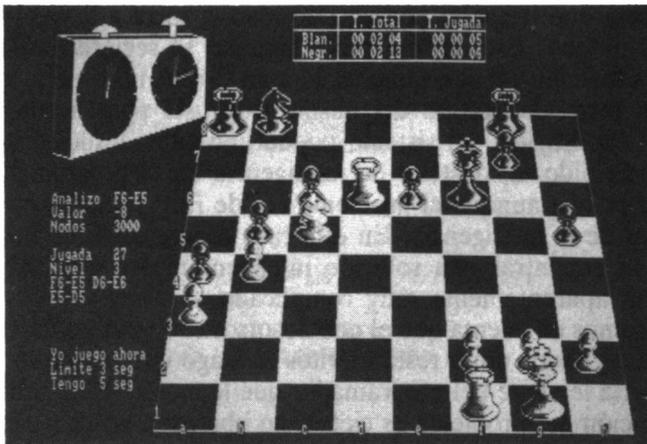
Pues bien, si cuantificar los movimientos que se deben realizar es algo que se consigue con una sola línea de programa, la inteligencia real del mismo también reside en una sola línea:

$$560 \quad f = p - \frac{sg^*(t(x,y)/2) <> t(x,y)}{\sqrt{2} + \frac{sg^*(t(x,y)/2) = t(x,y)}{\sqrt{2}}}$$

En realidad, las 3 K de programa se emplean en la representación del problema en la pantalla, la toma de los datos iniciales y la infraestructura de toma de datos y depuración de los mismos necesaria para que el programa admita también el concurso de un jugador humano. Su diagrama de bloques es el siguiente:

- 10-340 Presentación.
- 350-510 Jugador humano.
- 520-640 Jugador electrónico.
- 650-680 Fin del problema.
- 690-880 Subrutinas para movimientos y representación de las torres.

Como podemos apreciar, el bloque denominado «jugador electrónico», en el cual reside la inteligencia del programa ocupa tan sólo 12 líneas, de las cuales 11 son las destinadas a que el ordenador respete las reglas del juego y notifique por la pantalla el movimiento escogido.



4.2.3. Un buen programa de ajedrez puede servir de muestra del desarrollo «vertical» de la inteligencia en los ordenadores.

Ahora veremos de qué forma una sola línea puede resolver un problema de relativamente tanta complejidad. Simplemente, asigna a cada altura un sentido de movimiento constante; dicha asignación es alternativa, comenzando como «derecha» el bloque mayor. Así por ejemplo, en una torre de tres alturas, el bloque de longitud 6 y el de 2, es decir, el primero y el último (alturas 1 y 3) se desplazarán siempre hacia la derecha, mientras que el de longitud 4 (altura 2) se moverá siempre hacia la izquierda.

Para evitar el problema de llegar al tope derecho o izquierdo de la pantalla, las torres se deben considerar dispuestas en triángulo, como si en la pantalla se obtuviera una vista lateral de las mismas. Así, cuando llegando un bloque a la torre 3 deba desplazarse a la derecha, irá a parar a la 1, y de forma similar, desplazándose a la izquierda desde la 1, llegará a la 3 (líneas 570 y 580 del programa).

Nosotros podemos marcar una flecha imaginaria, que indique el sentido de desplazamiento en cada bloque sin ningún inconveniente; comenzando derecha, izquierda, derecha... desde abajo hacia arriba. Una vez sabido el sentido en que siempre se deben desplazar los bloques, sólo hemos de empezar a moverlos siguiendo las reglas del juego, comenzando por el primero de ellos (el mayor).

Así por ejemplo, el desarrollo de un juego con dos alturas será muy simple: el bloque 1 se mueve hacia la derecha y el 2 hacia la izquierda. Comenzamos intentando desplazar el 1 hacia la derecha, pero observamos que previamente debemos quitarle de encima el 1, que ha de ir a parar a su izquierda; si este movimiento no fuera posible, habría que despejar con anterioridad la torre a la que va a ir a parar el bloque. Este mismo procedimiento se seguirá, hasta conseguir la situación en que tanto el bloque 2 esté libre para moverse, como la torre de destino del mismo esté lista para aceptarlo.

Esta explicación del sistema puede habernos desilusionado, tras haber percibido un destello de inteligencia en el programa, pero la realidad es que aunque el problema sea de fácil solución no deja de depender de la inteligencia. En este caso concreto, el mérito ha consistido en encontrar una solución inteligentemente sencilla para un problema aparentemente muy complicado.

Tengamos en cuenta que el que nosotros conozcamos los trucos de un prestidigitador no le resta méritos al mago que los urdió. Que su inteligencia le viene del programador que le mostró el algoritmo para la resolución del problema es cierto; no obstante, no es menos cierto que la inteligencia que le permite al arquitecto edificar una casa, también ha nacido de la de los profesores que en la escuela de arquitectura le enseñaron los sistemas de construcción.

De hecho, ahora que ya conocemos el truco, podemos llegar a impresionar a los conocidos con nuestra extraordinaria habilidad. No tenemos más que dibujar tres rayas verticales en un papel, y conseguir una moneda de peseta, otra de 5, de 25 y de 50; he aquí un magnífico tablero para la experimentación de las TORRES DE HANNOI con cuatro alturas.

#### 4.2.6. Programa HANNOI versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /-- TORRES DE HANNOI (c)1986 GRAN BIBLIOTECA AMSTRAD --
40 /----- CPC 464/664/6128 -----
50 /-----
60 /
70 MODE 1
80 PRINT"?QUIERES QUE SEA YO QUIEN RESUELVA EL?"
90 PRINT"PROBLEMA?"
100 FOR i=0 TO 999
110 x$=INKEY$
120 IF x$="" THEN 160
130 IF UPPER$(x$)="S" THEN ju=0:GOTO 190
140 IF UPPER$(x$)="N" THEN ju=1:GOTO 190
150 i=999
160 NEXT
170 PRINT CHR$(7);
180 GOTO 100
190 PRINT
200 INPUT"?QUE ALTURA DEBE TENER LA TORRE";h
210 h=INT(h)
220 IF h<2 OR h>7 THEN PRINT:PRINT"PRUEBA CON OTRA ENTRE 2 Y
7":GOTO 190
230 DIM t(3,h)
240 FOR i=1 TO h
250 t(3,i)=h-i+1
260 NEXT
270 t(3,0)=h
280 PRINT:PRINT"EL NUMERO OPTIMO DE MOVIMIENTOS PARA LA"
290 PRINT"RESOLUCION DEL PROBLEMA SON:"2^h-1
300 PRINT:PRINT"PULSA UNA TECLA PARA COMENZAR"
310 IF INKEY$="" THEN 310
320 MODE 2
330 WINDOW 1,80,5,24
340 WINDOW#1,1,80,2,3
350 WINDOW#2,1,80,25,25
360 WINDOW#3,1,80,1,1
370 PRINT#2,"////////TORRE: 1////////"SPC(10)"////////TORRE: 2////
//"SPC(10)"////////TORRE: 3////////";
380 GOSUB 740
390 IF ju=0 THEN 570
400 m=m+1:PRINT#3:PRINT#3,"MOVIMIENTO NUMERO"m;
410 WINDOW SWAP 0,1
420 CLS
430 LOCATE 1,1
440 INPUT"?TORRE DE ORIGEN";p
450 p=INT(p)
460 IF p<1 OR p>3 THEN PRINT CHR$(7):GOTO 430
470 IF t(p,0)=0 THEN PRINT CHR$(7):GOTO 430
480 LOCATE 40,1

```

```

490 INPUT'?TORRE DE DESTINO';f
500 f=INT(f)
510 IF f<1 OR f>3 THEN PRINT CHR$(7):GOTO 430
520 IF t(f,0)=0 THEN 540
530 IF t(f,t(f,0))<t(p,t(p,0)) OR p=f THEN PRINT'MOVIMIENTO
ERRONEO'CHR$(7);GOTO 430
540 PRINT'MOVIMIENTO ADMITIDO';:FOR i=0 TO 99:NEXT
550 WINDOW SWAP 0,1
560 GOSUB 820:IF t(1,0)<h THEN 400 ELSE 700
570 IF h/2 <> h\2 THEN sg=1 ELSE sg=-1
580 FOR a=h TO 1 STEP -1:p#=CHR$(a)
590 GOSUB 880
600 IF t(x,0)<y THEN p#=CHR$(t(x,y+1))+p#:GOTO 590 ELSE p#=x
610 f=p-sg*(t(x,y)/2<>t(x,y)\2)+sg*(t(x,y)/2=t(x,y)\2)
620 IF f=4 THEN f=1
630 IF f=0 THEN f=3
640 IF t(f,0)=0 THEN 660
650 IF t(f,t(f,0))<ASC(p#) THEN p#=CHR$(t(f,1))+p#:GOTO 590
660 p#=MID$(p#,2)
670 m=m+1:PRINT#3:PRINT#3,'MOVIMIENTO NUMERO'm;:CLS#1:PRINT#
1,'TORRE DE ORIGEN:'p TAB(40)'TORRE DE DESTINO:'f:GOSUB 820:
PRINT#1,'PULSA UNA TECLA':WHILE INKEY$='':WEND
680 IF t(1,0)=h THEN 700
690 IF p#<' THEN 590 ELSE NEXT
700 CLS#1
710 PRINT#1,'** PROBLEMA SOLUCIONADO **'
720 LOCATE 1,1
730 END
740 CLS
750 FOR i=1 TO 3
760 FOR j=1 TO t(i,0)
770 LOCATE 30*(i-1)+11-t(i,j),21-j
780 PRINT STRING$(2*t(i,j),CHR$(143));
790 NEXT
800 NEXT
810 RETURN
820 t(f,t(f,0)+1)=t(p,t(p,0))
830 t(f,0)=t(f,0)+1
840 t(p,t(p,0))=0
850 t(p,0)=t(p,0)-1
860 GOSUB 740
870 RETURN
880 FOR j=1 TO 3
890 FOR k=1 TO t(j,0)
900 IF t(j,k)=ASC(p#) THEN x=j:y=k:k=t(j,0):j=4
910 NEXT
920 NEXT
930 RETURN

```

#### 4.2.7. Programa HANNOI versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /-TORRES DE HANNOI (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 PRINT CHR$(27)*X'CHR$(32)CHR$(32)CHR$(63)CHR$(122)
90 k1s#=CHR$(27)+'E'+CHR$(27)+'H':PRINT k1s#
100 PRINT'?QUIERES QUE SEA YO QUIEN RESUELVA EL PROBLEMA?'
110 FOR i=0 TO 999

```

```

120 x$=INKEY$
130 IF x$="" THEN 170
140 IF UPPER$(x$)="S" THEN ju=0:GOTO 200
150 IF UPPER$(x$)="N" THEN ju=1:GOTO 200
160 i=999
170 NEXT
180 PRINT CHR$(7);
190 GOTO 110
200 PRINT
210 INPUT"?QUE ALTURA DEBE TENER LA TORRE";h
220 h=INT(h)
230 IF h<2 OR h>7 THEN PRINT:PRINT"PRUEBA CON OTRA ENTRE 2 Y
7":GOTO 200
240 DIM t(3,h)
250 FOR i=1 TO h
260 t(3,i)=h-i+1
270 NEXT
280 t(3,0)=h
290 PRINT:PRINT"EL NUMERO OPTIMO DE MOVIMIENTOS PARA LA RESO
LUCION DEL PROBLEMA SON:"2^h-1
300 PRINT:PRINT"PULSA UNA TECLA PARA COMENZAR"
310 IF INKEY$="" THEN 310
320 PRINT kls$
330 PRINT CHR$(27)*X*CHR$(63)CHR$(32)CHR$(33)CHR$(122)
340 PRINT"////////TORRE: 1////////SPC(10)////////TORRE: 2////////
SPC(10)////////TORRE: 3////////";
350 GOSUB 700
360 IF ju=0 THEN 520
370 PRINT CHR$(27)*X*CHR$(32)CHR$(32)CHR$(33)CHR$(122)
380 m=m+1:PRINT:PRINT"MOVIMIENTO NUMERO"m
390 PRINT CHR$(27)*X*CHR$(34)CHR$(32)CHR$(34)CHR$(122)
400 PRINT kls$
410 INPUT"?TORRE DE ORIGEN";p
420 p=INT(p)
430 IF p<1 OR p>3 THEN PRINT CHR$(7):GOTO 400
440 IF t(p,0)=0 THEN PRINT CHR$(7):GOTO 400
450 INPUT"?TORRE DE DESTINO";f
460 f=INT(f)
470 IF f<1 OR f>3 THEN PRINT CHR$(7):GOTO 400
480 IF t(f,0)=0 THEN 500
490 IF t(f,t(f,0))<t(p,t(p,0)) OR p=f THEN PRINT"MOVIMIENTO
ERRONEO"CHR$(7);FOR i=0 TO 499:NEXT:GOTO 400
500 PRINT"MOVIMIENTO ADMITIDO";FOR i=0 TO 99:NEXT
510 GOSUB 800:IF t(1,0)<h THEN 370 ELSE 660
520 IF h/2 <> h\2 THEN sg=1 ELSE sg=-1
530 FOR a=h TO 1 STEP -1:p$=CHR$(a)
540 GOSUB 860
550 IF t(x,0)<>y THEN p$=CHR$(t(x,y+1))+p$:GOTO 540 ELSE p=x
560 f=p-sg*(t(x,y)/2<>t(x,y)\2)+sg*(t(x,y)/2=t(x,y)\2)
570 IF f=4 THEN f=1
580 IF f=0 THEN f=3
590 IF t(f,0)=0 THEN 610
600 IF t(f,t(f,0))<ASC(p$) THEN p$=CHR$(t(f,1))+p$:GOTO 540
610 p$=MID$(p$,2)
620 m=m+1:PRINT CHR$(27)*X*CHR$(32)CHR$(32)CHR$(33)CHR$(122)
:PRINT:PRINT"MOVIMIENTO NUMERO"m
630 PRINT CHR$(27)*X*CHR$(34)CHR$(32)CHR$(35)CHR$(122):PRINT
kls$:PRINT"TORRE DE ORIGEN:"p:PRINT"TORRE DE DESTINO:"f:PRI
NT"PULSA UNA TECLA":GOSUB 800:WHILE INKEY$="" :WEND
640 IF t(1,0)=h THEN 660
650 IF p$<>"" THEN 540 ELSE NEXT
660 PRINT CHR$(27)*X*CHR$(34)CHR$(32)CHR$(34)CHR$(122):PRINT
kls$
670 PRINT"* PROBLEMA SOLUCIONADO *"
680 PRINT CHR$(27)*X*CHR$(37)CHR$(32)CHR$(57)CHR$(122)
690 END
700 PRINT CHR$(27)*X*CHR$(37)CHR$(32)CHR$(57)CHR$(122)

```

```

710 PRINT k1$;PRINT CHR$(27)*f*
720 FOR j=1 TO h
730 FOR i=3 TO 1 STEP -1
740 PRINT CHR$(27)*H*;;FOR k=1 TO 26-j:PRINT:NEXT:PRINT TAB(
30*(i-1)+11-t(i,j));
750 PRINT STRING$(2*t(i,j),'*');
760 NEXT
770 NEXT
780 PRINT CHR$(27)*e*
790 RETURN
800 t(f,t(f,0)+1)=t(p,t(p,0))
810 t(f,0)=t(f,0)+1
820 t(p,t(p,0))=0
830 t(p,0)=t(p,0)-1
840 GOSUB 700
850 RETURN
860 FOR j=1 TO 3
870 FOR k=1 TO t(j,0)
880 IF t(j,k)=ASC(p$) THEN x=j:y=k:k=t(j,0):j=4
890 NEXT
900 NEXT
910 RETURN

```

# EL ORDENADOR AL HABLA

**L**a capacidad de comunicación mediante el lenguaje natural, es otra de las características que distinguen al hombre del resto de los seres no inteligentes, y también una de las facetas que más intensamente han sido estudiadas por los técnicos en IA, para su reproducción en un ordenador. No obstante, este objetivo queda aún lejano, dada la enorme complejidad del mecanismo comunicativo humano.

Muchas son las formas de comunicación que ha encontrado el hombre a lo largo del tiempo. Desde los lenguajes puramente simbólicos, como la música, hasta los códigos tipo morse. En todo caso, esto no simplifica el problema, pues sólo se trata de sustituir las letras o fonemas por otro tipo de representaciones, siendo aún el lenguaje natural, en sus distintos idiomas, la forma de comunicación por excelencia entre los humanos.

Así pues, el estudio de este lenguaje nace de la necesidad de enseñárselo a las máquinas con el fin de facilitar nuestra interacción con ellas, y llevados a un último extremo, conseguir el contacto directo con un ordenador, como si de otro humano se tratase. Este es el único camino para que de una máquina se pueda decir que es suficientemente inteligente.

El gran investigador en IA Alan Turing, al cual ya hemos hecho referencia con anterioridad, proponía una curiosa prueba para llegar a afirmar que una máquina era inteligente. A un lado de un tabique, sentaremos un hombre ante un terminal, que estará conectado al otro lado de la pared con otros dos terminales; uno de ellos controlado por un humano y el otro por un ordenador. Si puestos en comunicación ambos lados de la pared, el primero de los hombres es incapaz de distinguir si su comunicación se desarrolla con el otro humano o con la máquina, ésta se puede considerar inteligente.

Realmente, el hecho de llegar a conseguir en algún momento que el ordenador pueda comprender nuestros requerimientos en lenguaje natural, supondría romper la barrera probablemente más importante que nos separa de él.

## LA COMUNICACIÓN HUMANA

Las investigaciones sobre el lenguaje natural comienzan a mediados de este siglo, orientándose hacia el campo concreto de la traducción. En un principio, esta tarea se concibe como algo automático, y por tanto realizable por los ordenadores en base a un diccionario bilingüe y una gramática propia de cada idioma, que construía las frases añadiendo las terminaciones a las palabras.

Los paupérrimos resultados obtenidos en el transcurso de diez años, convencieron a los investigadores que se encontraban sobre una pista errónea. A partir de entonces, se tomó auténtica conciencia que la traducción de lenguajes naturales necesitaba bastante más que un simple diccionario y una serie de reglas gramaticales, y que de hecho el primer problema a resolver era el de la comprensión del lenguaje, para a partir de este punto y con una base suficientemente sólida, plantearse la posibilidad de conseguir una traducción, con los problemas de manejo de dos lenguajes que ello suponía.

En la comunicación intervienen diferentes factores, que fueron estudiados independientemente: por una parte podemos hablar de comunicación oral y por otra de la escrita, consiguiéndose resultados bien distintos en unos y otros.

El estudio de la fonética ha llevado a bastantes éxitos en el campo de la comunicación oral, y además de sistemas más o menos sofisticados que permiten el reconocimiento de voz y palabras, están al alcance de cualquiera los sintetizadores de voz, que llegan a conseguir gran calidad en la generación de fonemas y vocablos, pudiendo incluso im-

primir énfasis, variar timbres, entonaciones, etc... Esto es debido a que el estudio anatómico nos ha permitido fijar claramente la forma en que el hombre emite sonidos.

No son tantos los logros conseguidos en la comprensión del lenguaje escrito, dejando a un lado los problemas inherentes a la fonética que aporta el lenguaje oral, como que este último suele ser más incompleto e incorrectamente estructurado.

## LA PROBLEMÁTICA DEL LENGUAJE

El problema principal nace de la incapacidad de representar en el ordenador el mundo que le rodea. Observemos la siguiente frase:

Fernando me rompió la muñeca

Esta misma oración, analizada en dos situaciones diferentes, puede llevarnos a extraer conclusiones completamente distintas. Por ejemplo, puede ser pronunciada por Carlos cuando se acerca a nosotros, permitiéndonos apreciar su brazo en cabestrillo, o por una niña mientras entre pucheros exhibe en su cochecito de muñecas la prueba de las fechorías de su hermano.

Lógicamente, el ordenador se encontrará con el mismo problema de comprensión de la frase que nosotros mismos, pero en una situación de desventaja sustancial: no tiene el conocimiento del contexto que nos ha capacitado para extraer una conclusión correcta.

Por el momento, este escollo se salva restringiendo el ámbito de actuación del ordenador, haciéndole vivir en un pequeño mundo, como a nuestro robot 'VODBIR' del primer capítulo.

Por otra parte, hemos de tener en cuenta que no sólo la semántica (significado de las palabras) presenta un problema de contexto, sino que la complejidad del hombre, le lleva a atribuir determinado sentido a frases cuyo significado literal en un contexto dado, aparentemente, no deja lugar a dudas. Supongamos por ejemplo, que yendo por la calle preguntamos a un transeúnte «¿Sabe usted qué hora es?», y éste, ni corto ni perezoso, nos contesta «Sí», continuando su camino. Evidentemente, nuestra pretensión era saber la hora, no enterarnos de si el otro hombre no se había olvidado el reloj en casa; no obstante, el significado de las palabras y la construcción de la frase, no transmitían nuestras pretensiones reales.

Por otra parte, en la comprensión de las frases, además de las dos facetas mencionadas, significado de las palabras y contexto en que

han sido pronunciadas, existe un tercer elemento de vital importancia: la estructura de la oración. Así, del mismo modo que en la frase anterior, dos palabras con una misma forma podían tener un diferente significado, en el siguiente, un mismo mensaje se presenta con dos estructuras superficiales distintas:

Fernando vio a Carlos  
o  
Carlos fue visto por Fernando

Podemos decir que el proceso completo de comprensión de una oración por el ordenador, se consigue a través de tres análisis (*Par-sing*) diferenciados:

- Análisis sintáctico, de la estructura de la oración.
- Análisis semántico, del significado de las palabras.
- Análisis interpretativo o de contexto.

## EL PROBLEMA FILOSÓFICO

Siguiendo más o menos de cerca las normas aquí expuestas, algunos diseñadores han llegado a obtener procesadores de lenguaje natural de cierta eficacia.

De ellos, el más conocido en cuanto a aplicación general es LIFER, creado por Gary Hendrix en 1977, y que permite ser aplicado a un dominio concreto, llegándose a obtener resultados bastante satisfactorios como interface de lenguaje natural para sistemas expertos y accesos a bases de datos como LADDER.

En cuanto a un sistema de aplicación concreta, la palma se la lleva sin duda alguna ELIZA; el programa creado en 1966 por el profesor Weizenbaum, que simulaba la conversación con un psiquiatra de la escuela rogeriana. Más adelante, tendremos ocasión de hablar de él. Su importancia radica fundamentalmente en que planteó una nueva problemática al proceso del lenguaje natural, e incluso generalizando, a toda la inteligencia artificial. ¿Entendía realmente ELIZA lo que se le decía?

A esta polémica se le dedicó mucha atención en los comienzos de la investigación sobre IA, produciendo una fuerte sacudida en las convicciones de algunos investigadores, como el propio Weizenbaum. Actualmente, parece ser que este problema ha dejado de ser tenido en consideración y ya nadie se pregunta si el robot PROWLER que conocimos en el primer capítulo posee un instinto asesino.

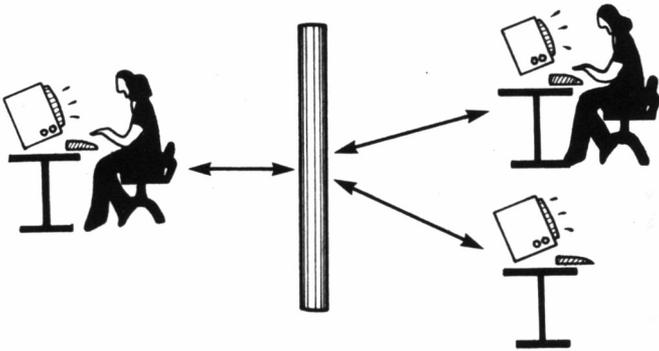
no, si ELIZA entiende nuestros problemas o 'VODBIR' se siente satisfecho al finalizar adecuadamente un manejo de cubos.

La aplicación más palpable de la IA, los llamados sistemas expertos, son empleados en la actualidad para colaborar con el hombre en la resolución de problemas complejos. No importa si realmente tienen conciencia de que nos están ayudando, o si los robots entienden lo que ven y oyen, o más aún, si realmente se puede decir que estos artefactos ven u oyen.

Las actuales tendencias de la IA afirman que es inteligente todo aquel sistema que se comporte como tal. Así pues, si un ordenador está actuando de una manera inteligente, aunque sea en un universo restringido, demos por sentado que es inteligente, sin detenernos a pensar si realmente está dotado de entendimiento. Actuar así sería perder un tiempo precioso que podría emplearse en la ampliación de la inteligencia de la máquina, sea ésta real o ficticia.

## APLICACIONES PRÁCTICAS

Dejando aparte ELIZA, son bastantes los sistemas de proceso de lenguaje que se han aplicado a algunas áreas particulares del conocimiento humano, actuando como sistemas expertos que más adelante estudiaremos. Tal es el caso del BASE-BALL de Bert Green, LUNAR de William Woods, o el propio SHRDLU de Terry Winograd, al que ya hicimos referencia con anterioridad.



4.3.1. Ilustración del experimento de Turing sobre inteligencia de un ordenador.

Aún sin pretender acercarnos a la magnificencia de estos sistemas, desde nuestro Amstrad podemos llevar a cabo modestos intentos para que el ordenador nos entienda en nuestro propio lenguaje. ¿Qué mejor para comenzar que enseñarle a leer? Este pequeño proyecto nos llevará bien poco tiempo, cosa que no podemos decir de cuando nosotros aprendimos.

La aplicación concreta del programa, puede ser la ruptura con guión de una palabra por fin de línea en un procesador de texto, o bien la traducción de una palabra a sus correspondientes sonidos, para ser emitidos por un sintetizador de voz.

Repararemos en un hecho curiosísimo: la correcta separación de las sílabas se consigue interpretando las palabras ¡al revés! Efectivamente, la lectura se efectúa de derecha a izquierda. Dejando aparte este pequeño detalle, el desarrollo del programa es bastante simple, como apreciaremos en la siguiente explicación.

Una vez introducida la palabra, se somete a un leve tratamiento, despojándola de los signos de puntuación y otros caracteres ajenos al alfabeto, aunque respetando la presencia de la eñe, y de la u con diéresis (ü), que se introducirá mediante un corchete abierto. Acto seguido, la palabra es convertida a mayúsculas e invertida, para comenzar lo que propiamente constituye el proceso de separación en sílabas. Este se lleva a cabo según ciertas reglas que a continuación estudiaremos, y posteriormente se reinvierte la palabra y se eliminan las separaciones de diptongos y triptongos.

Como podemos comprobar, aunque no se encuentre contemplada la inclusión de acentos, ésta no presenta ningún problema, y tan sólo afecta al área de tratamiento de diptongos y triptongos.

Las reglas para silabación son las siguientes:

— Conversión fonética

- ñ pasa a sonido @.
- Y pasa a I cuando va precedida o antecede a una vocal.
- SH pasa a sonido O.
- CH pasa a sonido O.
- CA pasa a KA.
- CE pasa a ZE.
- CI pasa a ZI.
- CO pasa a KO.
- CU pasa a KU.
- CK pasa a K.

- GE pasa a JE.
- GI pasa a JI.
- H desaparece.
- QU pasa a K.
- V pasa a B.
- X pasa a S.
- LL pasa a Y.
- RR pasa a sonido 1.
- GUE pasa a GE.
- GUI pasa a GI.
- [ pasa a U.

— Conversión silábica (c. = carácter)

- Inversión de la palabra.
- Si 1.<sup>er</sup> c. vocal
  - Si 2.<sup>o</sup> c. vocal → sílaba = 1.<sup>er</sup> c.
  - Si 2.<sup>o</sup> c. cons. → proseguir en (2) considerando c. = c. - 1.
- Si 1.<sup>er</sup> c. consonante
- (1) ● Si 2.<sup>o</sup> c. cons. → se tratan 1.<sup>er</sup> y 2.<sup>o</sup> c. como una sola consonante y se reinicia la evaluación.
  - Si 2.<sup>o</sup> c. vocal
    - Si 3.<sup>er</sup> c. vocal → sílaba = 2.<sup>o</sup> y 1.<sup>er</sup> c. más posible cons. residual procedente de (1).
  - (2) ● Si 3.<sup>er</sup> c. cons.
    - Si 3.<sup>er</sup> y 4.<sup>o</sup> c. forman el grupo consonántico: BL, BR, CL, CR, DR, FL, FR, GL, GR, KL, KR, PL, PR o TR → sílaba = 4.<sup>o</sup>, 3.<sup>er</sup>, 2.<sup>o</sup> y 1.<sup>er</sup> c. más posible cons. residual procedente de (1).
    - Si 4.<sup>o</sup> c. consonante líquida (primer carácter) → sílaba = 4.<sup>o</sup>, 3.<sup>er</sup>, 2.<sup>o</sup> y 1.<sup>er</sup> c. más posible cons. residual procedente de (1).
    - Otro caso → sílaba = 3.<sup>er</sup>, 2.<sup>o</sup> y 1.<sup>er</sup> c. más posible cons. residual procedente de (1).
  - Reinversión de la palabra.
- (3) — Eliminación e diptongos: grupo formado por una vocal fuerte (a, e, o) y una débil (i, u) o dos débiles.
  - Repetición de (3) para eliminación de triptongos (dip-  
tongo de diptongo).

Pese a que las reglas parecen en principio complicadas, no lo son tanto, y buena prueba de ello es el relativamente bajo número de

líneas que se precisan para la consecución del proceso. Durante su ejecución, el programa nos irá mostrando sus progresos, como podemos ver en los siguientes ejemplos que anteceden al listado del programa LEER.

#### 4.3.8. EJECUCION DE DEMOSTRACION DEL PROGRAMA LEER.

/VEHEMENTE/  
/BEEMENTE/  
ETNEMEEB  
ET/  
ET/NEM/  
ET/NEM/E/  
BE/E/MEN/TE  
BE/E/MEN/TE

/ALMOHADA/  
/ALMOADA/  
ADAOMLA  
AD/  
AD/A/  
AD/A/OM/  
AL/MO/A/DA  
AL/MO/A/DA

/CABIZBAJO/  
/KABIZBAJO/  
OJABZIBAK  
OJ/  
OJ/AB/  
OJ/AB/ZIB/  
KA/BIZ/BA/JO  
KA/BIZ/BA/JO

/BIZCOCHO/  
/BIZK000/  
000KZIB  
00/  
00/OK/  
BIZ/KO/00  
BIZ/KO/00

/OBSERVADOR/  
/OBSERBADOR/  
RODABRESBO  
ROD/  
ROD/AB/  
ROD/AB/RES/  
OB/SER/BA/DOR  
OB/SER/BA/DOR

/ARROGANTE/  
/A10GANTE/  
ETNAG01A  
ET/  
ET/NAG/  
ET/NAG/01/  
A/10/GAN/TE  
A/10/GAN/TE

/AHORCAR/  
/AORKAR/  
RAKROA  
RAK/  
RAK/RO/  
A/OR/KAR  
A/OR/KAR

/HABITACION/  
/ABITAZION/  
NOIZATIBA  
NO/  
NO/IZ/  
NO/IZ/AT/  
NO/IZ/AT/IB/  
A/BI/TA/ZI/ON  
A/BI/TA/ZION

/RENACIMIENTO/  
/RENAZIMIENTO/  
OTNEMIZANER  
OT/  
OT/NE/  
OT/NE/IM/  
OT/NE/IM/IZ/  
OT/NE/IM/IZ/AN/  
RE/NA/ZI/MI/EN/TO  
RE/NA/ZI/MIEN/TO

/OBSCURO/  
/OBSKURO/  
ORUKSBO  
OR/  
OR/UK/  
OBS/KU/RO  
OBS/KU/RO

### 4.3.9. Programa LEER versión CPC.

```

10 MODE 2
20 INPUT "INTRODUCE LA PALABRA A CODIFICAR: ",p$
30 IF p$="" THEN END
40 p$=UPPER$(p$)
50 p=INSTR(p$,CHR$(161)):IF p<>0 THEN MID$(p$,p,1)="@":GOTO 50
50 p=INSTR(p$,CHR$(171)):IF p<>0 THEN MID$(p$,p,1)="@":GOTO 60
70 p=2:p$="/"+p$+"/":WHILE p<=LEN(p$)-1
80 IF ASC(MID$(p$,p))<64 OR ASC(MID$(p$,p))>91 THEN p$=LEFT$(p$,p-1)+MID$(p$,p+1):GOTO 100
90 p=p+1
100 WEND
110 p=INSTR(p$,"Y"):IF p=0 THEN 140
120 z=ASC(MID$(p$,p+1)):IF z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<>85 THEN MID$(p$,p)="I":GOTO 110
130 z=ASC(MID$(p$,p-1)):IF z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<>85 THEN MID$(p$,p)="I":GOTO 110
140 IF LEN(p$)=2 THEN ip$="-":GOTO 610
150 PRINT p$
160 FOR i=0 TO 18
170 READ b$,s$
180 p=INSTR(p$,b$):IF p<>0 THEN p$=LEFT$(p$,p-1)+s$+MID$(p$,p+LEN(b$)):GOTO 180
190 NEXT
200 DATA SH,0,CH,0,CA,KA,CE,ZE,CI,ZI,CO,KO,CU,KU,CK,K,GE,JE,GI,JI,H,"",QU,K,V,B,X,S,LL,Y,RR,1,GUE,GE,GUI,GI,C,U
210 IF LEN(p$)=2 THEN ip$="-":GOTO 610
220 PRINT p$
230 c$=""
240 ip$="":FOR i=2 TO LEN(p$)-1:ip$=MID$(p$,i,1)+ip$:NEXT
250 PRINT ip$
260 r$=""
270 z=ASC(ip$):IF z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<>85 THEN 310
280 IF LEN(ip$)=1 THEN l=1:GOTO 450
290 z=ASC(MID$(ip$,2)):IF z=65 OR z=69 OR z=73 OR z=79 OR z=85 THEN l=1:GOTO 450
300 f=1:GOTO 360
310 IF LEN(ip$)=1 THEN l=1:GOTO 450
320 z=ASC(MID$(ip$,2)):IF z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<>85 THEN r$=LEFT$(ip$,1):ip$=MID$(ip$,2):GOTO 270
330 IF LEN(ip$)=2 THEN l=2:GOTO 450
340 z=ASC(MID$(ip$,3)):IF z=65 OR z=69 OR z=73 OR z=79 OR z=85 THEN l=2:GOTO 450
350 f=0
360 z$=MID$(ip$,3-f,2)
370 RESTORE 410
380 FOR i=0 TO 13
390 READ b$:IF z$=b$ THEN l=4-f:GOTO 450
400 NEXT
410 DATA LB,RB,LC,RC,RD,LF,RF,LG,RG,LK,RK,LP,RP,RT
420 IF LEN(ip$)=4-f THEN z=ASC(MID$(ip$,4-f))
430 IF LEN(ip$)=4-f AND z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<>85 THEN l=4-f:GOTO 450
440 l=3-f
450 c$=c$+r$+LEFT$(ip$,l)+"/"
460 IF LEN(ip$)<l+1 THEN 500
470 ip$=MID$(ip$,l+1)
480 PRINT c$
490 GOTO 260
500 ip$="":FOR i=1 TO LEN(c$):ip$=MID$(c$,i,1)+ip$:NEXT
510 ip$=MID$(ip$,2)
520 PRINT ip$

```

```

530 f=0:p=0
540 p=p+1:p=INSTR(p,ip$,'/'):IF p=0 THEN 610
550 z1=ASC(MID$(ip$,p-1)):IF z1<>65 AND z1<>69 AND z1<>73 AN
D z1<>79 AND z1<>85 THEN 540
560 z2=ASC(MID$(ip$,p+1)):IF z2<>65 AND z2<>69 AND z2<>73 AN
D z2<>79 AND z2<>85 THEN 540
570 d=0:IF z1=73 OR z1=85 THEN d=1
580 IF z2=73 OR z2=85 THEN d=d+1
590 IF d<>0 THEN ip$=LEFT$(ip$,p-1)+MID$(ip$,p+1)
600 GOTO 540
610 PRINT ip$
620 PRINT
630 RUN 20

```

#### 4.3.10. Programa LEER versión PCW.

```

10 k1$=CHR$(27)+'E'+CHR$(27)+'H':PRINT k1$
20 INPUT'INTRODUCE LA PALABRA A CODIFICAR:',p$
30 IF p$='' THEN END.
40 p$=UPPER$(p$)
50 p=INSTR(p$,CHR$(161)):IF p<>0 THEN MID$(p$,p,1)='@':GOTO
50
60 p=INSTR(p$,CHR$(171)):IF p<>0 THEN MID$(p$,p,1)='@':GOTO
60
70 p=2:p$='/'+p$+'/' :WHILE p<=LEN(p$)-1
80 IF ASC(MID$(p$,p))<64 OR ASC(MID$(p$,p))>91 THEN p$=LEFT$(
p$,p-1)+MID$(p$,p+1):GOTO 100
90 p=p+1
100 WEND
110 p=INSTR(p$, 'Y'):IF p=0 THEN 140
120 z=ASC(MID$(p$,p+1)):IF z<>65 AND z<>69 AND z<>73 AND z<>
79 AND z<>85 THEN MID$(p$,p)='I':GOTO 110
130 z=ASC(MID$(p$,p-1)):IF z<>65 AND z<>69 AND z<>73 AND z<>
79 AND z<>85 THEN MID$(p$,p)='I':GOTO 110
140 IF LEN(p$)=2 THEN ip$='-':GOTO 610
150 PRINT p$
160 FOR i=0 TO 18
170 READ b$,s$
180 p=INSTR(p$,b$):IF p<>0 THEN p$=LEFT$(p$,p-1)+s$+MID$(p$,
p+LEN(b$)):GOTO 180
190 NEXT
200 DATA SH,0,CH,0,CA,KA,CE,ZE,CI,ZI,CO,KO,CU,KU,CK,K,GE,JE,
GI,JI,H,'',QU,K,V,B,X,S,LL,Y,RR,1,GUE,GE,GUI,GI,I,C,U
210 IF LEN(p$)=2 THEN ip$='-':GOTO 610
220 PRINT p$
230 c$=''
240 ip$='':FOR i=2 TO LEN(p$)-1:ip$=MID$(p$,i,1)+ip$:NEXT
250 PRINT ip$
260 r$=''
270 z=ASC(ip$):IF z<>65 AND z<>69 AND z<>73 AND z<>79 AND z<
85 THEN 310
280 IF LEN(ip$)=1 THEN l=1:GOTO 450
290 z=ASC(MID$(ip$,2)):IF z=65 OR z=69 OR z=73 OR z=79 OR z=
85 THEN l=1:GOTO 450
300 f=1:GOTO 360
310 IF LEN(ip$)=1 THEN l=1:GOTO 450
320 z=ASC(MID$(ip$,2)):IF z<>65 AND z<>69 AND z<>73 AND z<>7
9 AND z<>85 THEN r$=LEFT$(ip$,1):ip$=MID$(ip$,2):GOTO 270
330 IF LEN(ip$)=2 THEN l=2:GOTO 450
340 z=ASC(MID$(ip$,3)):IF z=65 OR z=69 OR z=73 OR z=79 OR z=
85 THEN l=2:GOTO 450

```

```

350 f=0
360 z$=MID$(ip$,3-f,2)
370 RESTORE 410
380 FOR i=0 TO 13
390 READ b$:IF z$=b$ THEN l=4-f:GOTO 450
400 NEXT
410 DATA LB,RB,LC,RC,RD,LF,RF,LG,RG,LK,RK,LP,RP,RT
420 IF LEN(ip$)=4-f THEN z=ASC(MID$(ip$,4-f))
430 IF LEN(ip$)=4-f AND z<>65 AND z<>69 AND z<>73 AND z<>79
AND z<>85 THEN l=4-f:GOTO 450
440 l=3-f
450 c$=c$+r$+LEFT$(ip$,l)+'/'
460 IF LEN(ip$)<l+1 THEN 500
470 ip$=MID$(ip$,l+1)
480 PRINT c$
490 GOTO 260
500 ip$="":FOR i=1 TO LEN(c$):ip$=MID$(c$,i,1)+ip$:NEXT
510 ip$=MID$(ip$,2)
520 PRINT ip$
530 f=0:p=0
540 p=p+1:p=INSTR(p,ip$,'/'):IF p=0 THEN 610
550 z1=ASC(MID$(ip$,p-1)):IF z1<>65 AND z1<>69 AND z1<>73 AN
D z1<>79 AND z1<>85 THEN 540
560 z2=ASC(MID$(ip$,p+1)):IF z2<>65 AND z2<>69 AND z2<>73 AN
D z2<>79 AND z2<>85 THEN 540
570 d=0:IF z1=73 OR z1=85 THEN d=1
580 IF z2=73 OR z2=85 THEN d=d+1
590 IF d<>0 THEN ip$=LEFT$(ip$,p-1)+MID$(ip$,p+1)
600 GOTO 540
610 PRINT ip$
620 PRINT
630 RUN 20

```

## EL CÓDIGO SOUNDEX

Para finalizar, vamos a estudiar un curioso algoritmo que nos será de vital importancia en el ejercicio final de este capítulo: el código SOUNDEX.

Los lenguajes naturales varían con gran frecuencia, y existen palabras de dudosa ortografía que nos pueden llevar a un problema a la hora de recuperar información de un banco de datos. El inglés es desde luego un idioma bastante propicio a este tipo de errores, sobre todo por la pronunciación bastante similar de palabras con diferente ortografía.

La solución más clara a este problema es intentar encontrar los sonidos de las palabras en vez de buscar las letras. De hecho, éste es uno de los pasos seguidos por el programa LEER, cuyo listado acabamos de ver. Hacia el año 1980 se desarrolló en los Estados Unidos un código fonético, que aseguraba que las palabras con un sonido similar recibieran un mismo código, de forma que pudieran ser localizadas

posteriormente en una base de datos. Este código se denominó *Soundex*, y fue empleado con éxito en el proceso del censo estadounidense.

Las reglas Soundex son muy simples:

— Conservar la primera letra de cada palabra como primer carácter del código.

A partir del segundo carácter.

— Suprimir los signos de puntuación.

— Suprimir las vocales y las letras H, Q, W e Y.

— Codificar el resto de las letras según las siguientes normas:

● B, F, P y V = 1

● C, G, J, K, S, X y Z = 2

● D y T = 3

● L = 4

● M y N = 5

● R = 6

— Si existen varias letras consecutivas con el mismo código, se tomará únicamente el primero.

— El código resultante deberá tener siempre cuatro caracteres; será completado con ceros por la derecha cuando no llegue a esta longitud, o trasladado procusteanamente (¡cortado por las malas!) cuando los exceda.

Evidentemente, la aplicación directa del código Soundex sobre palabras castellanas es algo prácticamente inútil. No obstante, hemos diseñado un código particular que conoceremos como GBA, de cuya validez tendremos constancia en el programa corrector ortográfico que finaliza este capítulo.

Las normas de elaboración del código GBA son la conversión de determinados caracteres o grupos de ellos a otros:

PT → T

BS → S

OO → O

GE → JE

S → Z

C → K

Q → K

P → B

L → R

ü (l) → U

CT → T

CC → Z

CE → ZE

GI → JI

CH → X

H → Desaparece

V → B

D → Z

RR → R

O → U

NM → N

EE → E

CI → ZI

X → Z

HU → GU

QU → K

M → N

LL → Y

GU → BU

y las terminaciones:

Consonante → Desaparece  
AA → A                      ZU → U

ZA → A  
UU → U

## ORTOGRAFÍA

El programa basado en el código GBA, nuestra versión del Soundex, se compone de dos partes bien diferenciadas: la gestión de la base de datos y el corrector ortográfico propiamente dicho.

El sistema de funcionamiento del programa es bastante simple. La sección de gestión de la base de datos, se encarga de procesar un vocabulario introducido por el usuario, dando la posibilidad de entrada de nuevas palabras, baja y listado de las ya existentes. Este diccionario es transferido al disco en forma de fichero secuencial al inicio y fin de ejecución del programa, permaneciendo durante su funcionamiento en el RAM-DISC, de ahí su gran velocidad de ejecución. Es por tanto de gran importancia, que tras efectuar alguna modificación en el vocabulario, abandonemos el programa mediante la opción 3 del menú principal (SALIDA DE PROGRAMA), dado que es la única forma en que estas alteraciones son almacenadas en el disco. Así pues, este programa no puede ser ejecutado en los modelos 464, 472 ó 664.

Los registros de la base de datos se componen por tres campos: la palabra en sí, su codificación GBA y una descripción opcional del significado del vocablo.

Al efectuarse una consulta ortográfica, se averigua el código de la palabra a investigar, mostrándose en la pantalla todas aquellas presentes en el diccionario que tengan el mismo código.

La capacidad total es de 1.007 vocablos, asignándose 65 caracteres a cada registro. Realmente, sin la ayuda de descripciones adicionales, se habría podido duplicar la capacidad del diccionario, aunque esto es algo que ya queda «a gusto del programador». Para una mayor rapidez en la gestión de la base de datos, la memoria se divide en compartimentos de distinta capacidad para cada letra, según un estudio de las frecuencias con que las palabras en castellano comienzan por una determinada letra:

A → 10.25 %	J → 1.12 %	R → 5.18 %
B → 4.05 %	K → 0.11 %	S → 5.29 %
C → 13.59 %	L → 2.81 %	T → 5.97 %
CH → 1.01 %	LL → 0.22 %	U → 0.67 %

D → 6.08 %	M → 6.08 %	V → 2.93 %
E → 7.77 %	N → 1.46 %	W → 0.11 %
F → 3.15 %	Ñ → 0.11 %	X → 0.11 %
G → 3.26 %	O → 1.80 %	Y → 0.22 %
H → 2.59 %	P → 9.69 %	Z → 0.78 %
I → 2.93 %	Q → 0.56 %	

Dado que el código GBA consta tan sólo de 16 caracteres, la distribución absoluta de los 1.007 registros ha sido la siguiente:

A → 103	J → 23	U → 26
B → 168	K → 91	W → 6
E → 79	N → 83	X → 11
F → 32	R → 81	Y → 10
G → 21	T → 61	Z → 18
I → 32		

A la hora de hacer un listado hemos de recordar que se efectuará una presentación secuencial a partir de un determinado código que indicaremos mediante **INPUT**; dicho código será un carácter válido de GBA (entre A y Z).

Por último, aunque la capacidad del diccionario es para bastantes fines suficientemente amplia, podemos crear diferentes bases de datos para distintos temas. Al final de este capítulo tenemos una muestra de léxico de uso general compuesto íntegramente por palabras de dudosa ortografía.

Recordemos seleccionar la opción 2 del menú de entrada (LECTURA DE FICHERO) cuando queramos volcar en la memoria un diccionario ya creado, bien sea para su consulta o para la ampliación o modificación del mismo, y la primera opción (CREACION NUEVO FICHERO) cuando deseemos generar un diccionario completamente nuevo. En ambos casos se nos solicitará un nombre de fichero, que se deberá asignar sin extensión, pues el programa les añade automáticamente el distintivo «.DIC».

### 4.3.3. LEXICO DE PALABRAS DE DUDOSA ORTOGRAFIA.

PALABRA	CODIGO	DESCRIPCION
*****	*****	*****
ABACO	ABAKU	calculadora
ABALORIO	ABARURIU	bisuteria
ABANICO	ABANIKU	para dar aire
ABATIR	ABATI	derribar

ABDOMEN	ABZUNE	tripa
ABECEDARIO	ABEZEZARIU	letras
ABEDUL	ABEZU	arbol
ABEJORRO	ABEJURU	mosca
AVIAR	ABIA	preparar
ABLANDAR	ABRANZA	poner blando
ABOBADO	ABUBAU	atontado
ABNEGACION	ABNEGAZIU	resignacion
ABOCHORNAR	ABUXURNA	poner en ridiculo
ABOLIR	ABURI	quitar
ABONAR	ABUNA	pagar
ABORRECER	ABUREZE	detestar
ABOGADO	ABUGAU	leyes
ABRIGAR	ABRIGA	guarecer
ABREVADERO	ABREBAZERU	bebedero
ACCESO	AZEU	acercarse
ABSORBER	AZURBE	chupar
ABSURDO	AZURU	loco
ABUELO	ABUERU	padre de los padres
ABUELA	ABUERA	madre de los padres
ACABAR	AKABA	terminar
ACEITE	AZEITE	liquido oleoso
ACROBATA	AKRUBATA	equilibrista
AHINCO	AINKU	teson
AHORA	AURA	tiempo
AHUYENTAR	ABUYENTA	hacer huir
AHORRAR	AURA	atesorar
AJENO	AJENU	del otro
AJETREO	AJETREU	lio
ALGARABIA	ARGARABIA	alegria
ALIENADO	ARIENAU	loco
ALMOHADA	ARNUA	para dormir
APROXIMAR	ABRUZINA	acercar
APTITUD	ATITU	capacidad
ACTITUD	ATITU	comportamiento
ASFIXIA	AZFIZIA	ahogo
AUTOPSIA	AUTUBZIA	analisis forense
AZAHAR	AZAA	flor del almendro
ABAD	ABA	fraile
AERODROMO	AERUZRUNU	aeropuerto
ALCOHOL	ARKUU	compuesto quimico
HABER	ABE	poseer
HABICHUELA	ABIXUERA	legumbre
HACIA	AZIA	preposicion
HALLAR	AYA	encontrar
ADICTO	AZITU	dependiente
HAMBRE	ANBRE	necesidad fisiologica
HALO	ARU	corona
HASTA	AZTA	preposicion
HATILLO	ATIYU	ropa
HAYA	AYA	arbol
ADVERTENCIA	AZBERTENZIA	consejo
ALBORNOZ	ARBURNU	ropa
BANDERA	BANZERA	insignia
BACALAO	BAKARAU	pez
BACHILLER	BAXIYE	estudios
BALANCE	BARANZE	resumen
BALON	BARU	esferico
BANDIDO	BANZIU	malhechor
BARAJA	BARAJA	naipes
BARNIZ	BARNI	laca
BARULLO	BARUYU	jaleo
BELFO	BERFU	hado
BICEPS	BIZEB	musculos
BISTURI	BIZTURI	instrumental medico
BIZOCHO	BIZKUXU	dulce
BOHEMIO	BUENTU	vida libre e irregular
BOGAVANTE	BUGABANTE	marisco
BOXEAR	BUZEA	lucha

BOYA	BUYA	cuerpo flotante
BREVIARIO	BREBIARIU	libro de rezos
BUHO	BU	ave nocturna
BUHONERIA	BUUNERIA	tienda portatil
BULBO	BURBU	tallo
BUQUE	BUKE	barco grande
BURGUES	BURBUE	persona acaudalada
GUIAR	BUIA	dirigir
GUAYABERA	BUAYABERA	chaquetilla corta
HUEVO	BUEBU	de la gallina
HUESO	BUEU	esqueleto
HURRACA	BURAKA	ave
PACTO	BATU	convenio
PAJE	BAJE	criado
PAPAGAYO	BABAGAYU	ave tropical
PARABOLA	BARABURA	curva
PARADAJICO	BARAZUJIKU	absurdo
PARVULO	BARBURU	inocente
PASAJERO	BAZAJERU	viajero
PATIZAMBO	BATIZANBU	piernas torcidas
PAVIMENTO	BABINENTU	suelo
PAYASO	BAYAU	volatinero
PEJIGUERA	BEJIBUERA	molesto
PELLIZA	BEYIA	prenda de abrigo
PERENNE	BERENNE	duradero
PERFECCION	BERFEZIU	perfecto
PERSONAJE	BERZUNAJE	sujeto
PERSPECTIVA	BERZBETIBA	representacion
PEYORATIVO	BEYURATIBU	empeorar
PIGMEO	BIGNEU	raza humana
PLEBEYO	BREBEYU	sin nobleza
PREAMBULO	BREANBURU	rodeo
PRIVILEGIO	BRIBIREJIU	ventaja
PROFILAXIS	BRUFIRAZI	preservacion
PROHIBIR	BRUBI	vedar
PROJIMO	BRUJINU	comunidad
PROVERBIO	BRUBERBIU	refran
PROXIMO	BRUZINU	cercano
PROYECTO	BRUYETU	plan
PUGNA	BUGNA	pelea
VALIJA	BARIJA	maleta
VACIAR	BAZIA	sacar
VACUNA	BAKUNA	virus
VADO	BAU	pasaje
VAHO	BAU	vapor
VAIVEN	BAIBE	oscilacion
VAJILLA	BAJIYA	servicio de mesa
VALLA	BAYA	cercado
VASALLO	BAZAYU	subdito
VASTAGO	BAZTAGU	descendiente
VAYA	BAYA	burla
VEJIGA	BEJIGA	organo
VENDAVAL	BENZABA	viento fuerte
VENTRILOCUO	BENTRIRUKU	estomago de la laringe
VERRACO	BERAKU	cerdo
VERRUGA	BERUGA	abultamiento
VERTEBRA	BERTEBRA	hueso
VICTORIA	BITURIA	vencer
VICTIMA	BITINA	sacrificado
VICEVERSA	BIZEBERA	al contrario
VOLUPTUOSO	BURUTUU	placeres
VOZARRON	BUZARU	voz ronca
VOLTAJE	BURTAJE	electricidad
VIVAZ	BIBA	energico
PULVERIZAR	BURBERIZA	desintegrar
EBULLICION	EBUYIZIU	hervor
ECLIPSE	EKRIBZE	desaparicion
EFLUVIO	EFRUBIU	emanacion
ELIXIR	ERIZI	medicamento maravilloso

EMBOLIA	ENBURIA	obstrucion de un vaso sanguineo
ENREVESADO	ENREBEZAU	complicado
ENMOHECER	ENNUEZE	criar moho
EPILEPSIA	EBIREBZIA	enfermedad
ERIZO	ERIU	animal
ESLABON	EZRABU	parte de una cadena
ESTORBO	EZTURBU	obstruccion
EXCAVAR	EZKABA	quitar
EXCUSA	EZKUA	pretexto
EXIGIR	EZIJJ	demandar
EXPRIMIR	EZBRINI	extraer
EXUBERANCIA	EZUBERANZIA	exceso
HEBILLA	EBIYA	trozo de metal
HEBRA	EBRA	porcion de hilo
HECTAREA	ETAREA	medida de superficie
HELICE	ERIZE	espiral
HELICOPTERO	ERIKUTERU	avion
HEMOGLOBINA	ENUGRUBINA	sangre
HEMORRAGIA	ENURAJIA	perdida de sangre
HERBIVORO	ERBIBURU	animal que come hierba
HERRAR	ERA	poner herraduras
HERVIR	ERBI	agitacion
HETERODOXO	ETERUZU	no catolico
HEXAGONO	EZAGUNU	figura geometrica
FACTIBLE	FATIBRE	posible
FERVOR	FERBU	devocion
FICCION	FIZIU	invencion
FLACCIDO	FRAZIU	flojo
FUELLE	FUEYE	instrumento para dar aire
FRUCTIFICAR	FRUTIFIKA	producir
FRIVOLO	FRIBURU	inconstante
GABACHO	GABAXU	frances
GABINETE	GABINETE	habitacion
GALLARDO	GAYARU	valiente
GAMBERRO	GANBERU	libertino
GARAJE	GARAJE	cochera
GAVIOTA	GABIUTA	ave
GNOMO	GNUNU	ser fantastico
GRAVOSO	GRABU	molesto
GRILLO	GRIYU	animal
HIDROFOBIA	IZRUFUBIA	terror al agua
HILVAN	IRBA	costura
ICEBERG	IZEBER	masa de hielo
ILEGIBLE	IREJIBRE	incomprensible
IMBERBE	INBERBE	sin barba
IMPERFECCION	INBERFEZIU	no perfecto
IMPROBO	INBRUBU	excesivo
IMPROVISAR	INBRUBIZA	sin preparacion
INACCESIBLE	INAZEZIBRE	no accesible
INCOHERENCIA	INKUERENZIA	falta de coherencia
INCUBAR	INKUBA	empollar
INGRAVIDO	INGRABIU	sin gravedad
INEXPERTO	INEZBERTU	sin experiencia
INHIBIRSE	INIBIRZE	supresion de la conciencia
INSCRIPCION	INZKRIBZIU	grabacion
INSOMNIO	INZUNNIU	desvelo
INSPECCION	INZBEZIU	indagacion
INTERCEPTAR	INTERZETA	detener
INTERRUPCION	INTERUBZIU	detencion
INYECTAR	INYETA	pinchar
IRREFLEXION	IREFREZIU	sin reflexion
ISTMO	IZTNU	accidente geografico
GELATINA	JERATINA	sustancia alimenticia
GEMA	JENA	piedra preciosa
GINEBRA	JINEBRA	bebida alcoholica
GIMNASIO	JINNAZIU	recinto deportivo
GESTA	JEZTA	acciones destacadas
JABATO	JABATU	cachorro de jabali
JACTARSE	JATARZE	alabarse

JEHOVA	JEUBA	Dios en hebreo
JEFE	JEFE	superior
JAZZ	JAZ	musica
JENGIBRE	JENJIBRE	planta india
JERGA	JERGA	lengua
JERGON	JERGU	colchon
JERINGUILLA	JERINBUIYA	para inyecciones
JEROLGIFICO	JERUGRIFIKU	tipo de escritura
JILGUERO	JIRBUERU	pajaro
JOROBA	JURUBA	corcova
JOVIAL	JUBIA	alegre
JOYERIA	JUYERIA	tienda
JUZGAR	JUZGA	deliberar
CABIZBAJO	KABIZBAJU	pensativo
CALUMNIA	KARUNNIA	falsa acusacion
CAPTURAR	KATURA	atrapar
CARRUAJE	KARUAJE	coche
CLARABOYA	KRARABUYA	ventana en el techo
CLAVICORDIO	KRABIKURZIU	instrumento musical
CLIMAX	KRINA	punto culminante
COEXISTIR	KUEZIZTI	existencias en un mismo tiempo
COHIBIR	KUIBI	reprimir
COLAPSO	KURABU	sincope
COMPLEJIDAD	KUNBREJIZA	diversidad
COMPUNGIRSE	KUNBUNJIRZE	afligirse
CONEXION	KUNEZIU	enlace
CONTRAHECHO	KUNTRAEXU	deforme
CONVOY	KUNBU	escolta
CONYUGE	KUNYUJE	consorte
CUCHILLO	KUXIYU	utensilio que corta
CONVEXO	KUNBEU	forma esferica
CRUCIFIXION	KRUZIFIZIU	sacrificio
CRILLO	KRIYU	animal
KIOSCO	KIUZKU	tenderete
KILOVATIO	KIRUBATIU	medida electrica
QUEHACER	KEAZE	ocupacion
QUEBRAR	KEBRA	doblar
QUEJIDO	KEJIU	voz lastimosa
QUERELLA	KEREYA	discordia
QUIETUD	KIETU	calma
QUINCALLA	KINKAYA	mercancias de metal
QUISQUILLA	KIZKIYA	melindre
CO@O	KU@U	
MAGNITUD	NAGNITU	dimension
MAGNESIO	NAGNEZIU	mineral
MAGULLAR	NABUYA	golpear
MAHONESA	NAUNEA	salsa
MAJESTAD	NAJEZTA	titulo
MAIZ	NAI	planta
MALEVOLO	NAREBURU	perverso
MALHUMORADO	NARBUNURAU	enfadado
MALIGNO	NARIGNU	no beneficioso
MANCEBO	NANZEBU	muchacho
MANIVELA	NANIBERA	manubrio
MAULLAR	NAUYA	gato
MAXIMO	NAZINU	grande
MAYORDOMO	NAYURZUNU	criado
MEJILLA	NEJIYA	carrillo
MEJUNJE	NEJUNJE	cosmetico
MEDLLO	NEUYU	medula
METALURGIA	NETARURJIA	ciencia del metal
MEZCLAR	NEZKRA	unir
MICROBIO	NIKRUBIU	microscopico
MIXTO	NIZTU	mezclado
MOHIN	NUI	mueca
MOJIGATO	NUJIGATU	finje humildad
MOLLA	NUYA	parte magra
MOZALBETE	NUZARBETE	mozo

MUELLE	NUEYE	elastico
NABO	NABU	planta
NAVAJA	NABAJA	cuchillo
NIEBLA	NIEBRA	fenomeno atmosferico
NEGLIGENCIA	NEGRIJENZIA	descuido
NEURALGIA	NEURARJIA	dolor
NIHILISMO	NIIRIZNU	sin creencias
NOCIVO	NUZIBU	perjudicial
NOCTURNO	NUTURNU	perteneciente a la noche
NOSTALGIA	NUZTARJIA	recuerdo
NOVATO	NUBATU	nuevo
NOVILLO	NUBIYU	toro
NUPCIAS	NUBZIA	esponsales
NOVELA	NUBERA	obra literaria
MAYUSCULO	NAYUZKURU	mayor de lo normal
LABERINTO	RABERINTU	enredo
LACAYO	RAKAYU	criado
LARVA	RARBA	insecto
LAPSO	RABU	espacio de tiempo
LECCION	REZIU	trozo de una escritura
LEGITIMO	REJITINU	genuino
LEJIA	REJIA	disolucion
LENGUAJE	RENBUAJE	modo de expresarse
LEVADIZO	REBAZIU	se puede levantar
LEVITA	REBITA	tipo de vestidura
LAVATIVA	RABATIBA	ayuda
LEVE	REBE	corto
LEXICO	REZIKU	diccionario
LEYENDA	REYENA	cuento
LIBERTAD	RIBERTA	no esclavo
LINAJE	RINAJE	familia
LITIGIO	RITIJIU	pleito
LIVIANO	RIBIANU	ligero
LOBO	RUBU	animal
LOGARITMO	RUGARITNU	funcion matematica
LUXACION	RUZAZIU	dislocacion
RABEL	RABE	instrumento musical
RABIZA	RABIA	cabo corto
RAPSODIA	RABUZIA	poema
RAPTAR	RATA	secuestrar
RAYA	RAYA	pez
REACCION	REAZIU	oposicion
REBATIR	REBATI	discutir
REBOTAR	REBUTA	botar repetidamente
REBOZAR	REBUZA	cubrir
REBUZNO	REBUZNU	asno
RECEPCION	REZEBZIU	recibimiento
RECOVECO	REKUBEKU	curva
REDACCION	REZAZIU	escritura
REDUCCION	REZUZIU	disminucion
REFLEXIVO	REFREZIBU	pensativo
REHACER	REAZE	volver a hacer
REHEN	REE	prenda
REHUSAR	REBUZA	rechazar
REIVINDICAR	REIBINZIKA	pedir
RELIEVE	RERIEBE	importante
REPTIL	RETI	animal
REPUGNANTE	REBUGNANTE	asco
RESBALAR	REZBARA	perder el equilibrio
RESIGNACION	REZIGNAZIU	conformidad
RUMBA	RUNBA	musica
TABANO	TABANU	insecto
TABURETE	TABURETE	mueble
TACTO	TATU	sentido corporal
TAHONA	TAUNA	panaderia
TANGIBLE	TANJIBRE	se puede tocar
TAPARRABO	TABARABU	calzon
TAXATIVO	TAZATIBU	limitar
TECNICO	TEKNIKU	con conocimientos

TEJER	TEJE	entrelazar hilos
TODAVIA	TUZABIA	hasta el momento
TORAX	TURA	pecho
TOXICO	TUZIKU	venenoso
TRABAR	TRABA	sujetar
TRAYECTO	TRAYETU	recorrido
TRAVIESO	TRABIEU	inquieto
TRAVESIA	TRABEZIA	camino
TRILLAR	TRIYA	quebrantar
TRIPTICO	TRITIKU	tipo de libro o pintura
TRIVIAL	TRIBIA	sin importancia
TOCAYO	TUKAYU	con el mismo nombre
HOLA	URA	saludo
HORQUILLA	URKIYA	pieza de alambre
OBCECAR	UBZEKA	cegar
OBJECION	UBJEZIU	impugnacion
OBSCENO	UZZENU	deshonesto
OBSCURO	UZKURU	sin luz
OBSERVAR	UZERBA	mirar
OBTENER	UBTENE	conseguir
OBVIO	UBBIU	facil
OJO	UJU	vista
OLIVA	URIBA	aceituna
OLLA	UYA	cocina
OMBLIGO	UNBRIGU	cicatriz redonda
OMNIBUS	UNNIBU	autobus
ONIX	UNI	agata
OPCION	UBZIU	eleccion
OQUEIDAD	UKEZA	hueco
ORFEBRE	URFEBRE	platero
ORGULLO	URBUYU	excesiva estimacion
OVAL	UBA	figura
OVINO	UBINU	ganado lanar
OVILLO	UBIYU	madeja
OXIGENO	UZIJENU	aire
UBERRIMO	UBERINU	abundante
UBICAR	UBIKA	situar
OBSTACULO	UZTAKURU	impedimento
CHABACANO	XABAKANU	sin gusto
CHAVAL	XABA	joven
CHIRIMOYA	XIRINUYA	fruta
CHIVATA	XIBATA	acusica
CHUBASCO	XUBAZKU	chaparron
LLAGA	YAGA	herida
LLAMA	YANA	fuego
LLANA	YANA	superficie igual
LLANTA	YANTA	rueda
LLAVE	YABE	para abrir puertas
LLOVER	YUBE	caer agua
YODO	YU	metaloides
YEGUA	YEBUA	hembra de caballo
YERBA	YERBA	hierba
YATE	YATE	barco
CERBATANA	ZERBATANA	arma de caza
CINC	ZIN	mineral
CIVICO	ZIBIKU	ciudadano
DECEPCION	ZEZEBZIU	perder la ilusion
DEHESA	ZEEA	prado
DESAHOGADO	ZEZAUGAU	sin apuros
DESHOJAR	ZEZUJA	quitar hojas
DESMAYAR	ZEZNAYA	desfallecimiento
DIFTERIA	ZIFTERIA	enfermedad respiratoria
DISTRACCION	ZIZTRAZIU	entretenerse
SABANA	ZABANA	ajuar
SABANDIJA	ZABANZIJA	reptil
SABIHONDO	ZABIUNU	sabio de mentirijilla
SABER	ZABE	conocer
SABOTAJE	ZABUTAJE	perjudicar
SABUESO	ZABUEU	perseguidor

SALIVA	ZARIBA	boca
SALVAJE	ZARBAJE	silvestre
SAVIA	ZABIA	alimento de las plantas
SAYO	ZAYU	casaca
SELVA	ZERBA	bosque extenso
SEPTIMO	ZETINU	numero
SEXENIO	ZEZENIU	seis anualidades
SIGILO	ZIJIRU	ocultacion
SINTAXIS	ZINTAZI	gramatica
SOBERBIA	ZUBERBIA	orgullo
SOSLAYAR	ZUZRAYA	rehuir
SUAVE	ZUABE	blando
SUBDITO	ZUBZITU	dependiente
SUBLEVAR	ZUBREBA	motin
SUBRAYAR	ZUBRAYA	recalcar
SUBSANAR	ZUZANA	excusar
SUBTERRANEO	ZUBTERANEU	debajo tierra
SUBYUGAR	ZUBYUGA	avasallar
ZURRAR	ZURA	vencer
ZURDO	ZURU	mano izquierda
ZANAHORIA	ZANAURIA	planta
SUBJETIVO	ZUBJETIBU	espíritu del hombre

4.3.6. Programa ORTOGRAFIA versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /---- DICCIONARIO (c)1986 GRAN BIBLIOTECA AMSTRAD ----
40 /----- CPC 6128 -----
50 /-----
60 /
70 CLS:r%=0;!BANKOPEN,65:r$=SPACE$(65):FOR i=1 TO 1007;!BANK
WRITE,@r%,r$,i:NEXT
80 RESTORE 230:DIM m$(16),m(16):FOR i=1 TO 16:READ m$(i),m(i)
):NEXT
90 MODE 1
100 LOCATE 15,8:PRINT "FICHERO"
110 PRINT TAB(15)"*****"
120 PRINT:PRINT
130 PRINT TAB(6)"1.- CREACION NUEVO FICHERO"
140 PRINT
150 PRINT TAB(6)"2.- LECTURA DE FICHERO"
160 f$=INKEY$:IF f$<'1' OR f$>'2' THEN 160
170 LOCATE 1,24:INPUT "?NOMBRE DEL FICHERO";a$
180 IF a$="" THEN 90
190 IF LEN(a$)>8 THEN PRINT "DEMASIADO LARGO" :GOTO
170
200 IF INSTR(a$,".") THEN PRINT "SIN ESPECIFICAR EXTENSION":
GOTO 170
210 IF INSTR(a$," ") THEN PRINT "NO DEBE CONTENER ESPACIOS":
GOTO 170
220 a$=UPPER$(a$)+".DIC"
230 DATA A,103,B,160,E,79,F,32,G,21,I,32,J,23,K,91,N,83,R,81
,T,61,U,26,W,6,X,11,Y,10,Z,180,@
240 IF f$='1' THEN re$=STRING$(16,CHR$(60)):PRINT "CREANDO E
L NUEVO FICHERO "a$ ":OPENOUT a$:PRINT#9,re$:CLOSEOUT:!BAN
KWRITE,@r%,re$,0:GOTO 420
250 PRINT "LEYENDO EL FICHERO "a$ "

```

```

260 OPENIN a$.
270 INPUT#9,re$.
280 :BANKWRITE,@r%,re$,0
290 s=0
300 CLS:PRINT'CODIGO', 'OCUPADAS', 'LIBRES':PRINT'*****', '***
*****', '*****':PRINT
310 FOR i=1 TO 16
320 fic=ASC(MID$(re$,i))-60:PRINT m$(i),fic,m(i)-fic
330 FOR j=1 TO fic
340 INPUT#9,r$.
350 :BANKWRITE,@r%,r$,s+j
360 NEXT
370 s=s+m(i)
380 NEXT
390 CLOSEIN
400 PRINT:PRINT'PULSE UNA TECLA'
410 WHILE INKEY$='':WEND
420 CLS
430 LOCATE 14,7:PRINT 'ORTOGRAFIA'
440 PRINT TAB(14)'*****'
450 PRINT:PRINT
460 PRINT TAB(7)'1.- GESTION BASE DE DATOS'
470 PRINT
480 PRINT TAB(7)'2.- CONSULTA ORTOGRAFICA'
490 PRINT
500 PRINT TAB(7)'3.- SALIDA DE PROGRAMA'
510 x$=INKEY$:IF x$<'1' OR x$>'3' THEN 510
520 ON VAL(x$) GOTO 530,1320,1540
530 CLS
540 LOCATE 14,6:PRINT 'BASE DATOS'
550 PRINT TAB(14)'*****'
560 PRINT:PRINT
570 PRINT TAB(11)'1.- INTRODUCCION'
580 PRINT
590 PRINT TAB(11)'2.- BAJA'
600 PRINT
610 PRINT TAB(11)'3.- LISTADO'
620 PRINT
630 PRINT TAB(11)'4.- RETORNO'
640 x$=INKEY$:IF x$<'1' OR x$>'4' THEN 640
650 ON VAL(x$) GOTO 660,800,1000,420
660 CLS
670 INPUT'?PALABRA';p$.
680 IF p$='' THEN 530
690 GOSUB 2020:IF c$='' THEN 660
700 IF ev=m(i) THEN PRINT:PRINT'NO HAY ESPACIO PARA CODIGOS
m$(i)'...':GOTO 770
710 PRINT:PRINT'?DESCRIPCION ('63-LEN(p$)-LEN(c$)'CARACTERES
)';
720 INPUT d$.
730 r$=LEFT$(c$+'#'+p$+'#'+d$+SPACE$(63),65)
740 p=s+ev
750 :BANKWRITE,@r%,r$,p
760 MID$(re$,i,1)=CHR$(ev+61)
770 PRINT:PRINT'PULSE UNA TECLA'
780 WHILE INKEY$='':WEND
790 GOTO 660
800 CLS
810 INPUT'?PALABRA PARA BAJA';p$.
820 IF p$='' THEN 530
830 GOSUB 2020
840 FOR j=s TO s+ev-1
850 :BANKREAD,@r%,@r$,j
860 p1=INSTR(r$,'#')+1:p2=INSTR(p1,r$,'#'):pb$=MID$(r$,p1,p2
-p1)
870 IF pb$<>p$ THEN 950

```

```

880 FOR k=j+1 TO s+ev
890 :BANKREAD,@r%,@r$,k
900 :BANKWRITE,@r%,r$,k-1
910 NEXT k
920 :BANKWRITE,@r%,SPACE$(65)
930 MID$(re$,i,1)=CHR$(59+ev)
940 PRINT:PRINT'PALABRA 'p$' BORRADA':GOTO 970
950 NEXT j
960 PRINT:PRINT'PALABRA 'p$: NO LOCALIZADA'
970 PRINT:PRINT'PULSE UNA TECLA'
980 WHILE INKEY$='':WEND
990 GOTO 800
1000 MODE 2
1010 INPUT'?A PARTIR DE QUE CODIGO';c$
1020 IF c$='' THEN MODE 1:GOTO 530
1030 c$=UPPER$(LEFT$(c$,1))
1040 FOR i=1 TO 16
1050 IF c$=m$(i) THEN 1080
1060 NEXT
1070 GOTO 1000
1080 PRINT:PRINT'?PANTALLA (P) O IMPRESORA (I)?'
1090 x$=INKEY$:IF x$='' THEN 1090
1100 IF UPPER$(x$)<>'P' AND UPPER$(x$)<>'I' THEN 1090
1110 IF UPPER$(x$)='P' THEN ch=0 ELSE ch=8
1120 PRINT:PRINT'PULSE <SPACE> PARA DETENER EL LISTADO'
1130 PRINT:PRINT#ch,'PALABRA', 'CODIGO', 'DESCRIPCION'
1140 PRINT#ch,'*****', '*****', '*****'
1150 PRINT#ch
1160 GOSUB 1970
1170 FOR i=s TO 1007
1180 :BANKREAD,@r%,@r$,i
1190 p1=INSTR(r$, '#'):p2=INSTR(p1+1,r$, '#')
1200 IF p1=0 OR p2=0 OR p2=p1+1 THEN 1270
1210 c$=LEFT$(r$,p1-1)
1220 p$=MID$(r$,p1+1,p2-p1-1)
1230 d$=MID$(r$,p2+1)
1240 FOR j=LEN(d$) TO 1 STEP -1:IF MID$(d$,j,1)=' ' THEN NEX
T:d$='':GOTO 1260
1250 d$=LEFT$(d$,j)
1260 PRINT#ch,p$,c$,d$
1270 IF INKEY$=' ' THEN 1290
1280 NEXT
1290 PRINT:PRINT'PULSE UNA TECLA'
1300 WHILE INKEY$='':WEND
1310 GOTO 1000
1320 MODE 2
1330 INPUT'?PALABRA';p$
1340 IF p$='' THEN MODE 1:GOTO 420
1350 GOSUB 2020:IF c$='' THEN 1320
1360 PRINT
1370 p=s
1380 WHILE p<s+m(i)
1390 :BANKFIND,@r%,c$,p,s+m(i)
1400 IF r%<0 THEN p=s+m(i):GOTO 1500
1410 p=r%
1420 :BANKREAD,@r%,@r$,p:p=p+1
1430 p1=INSTR(r$, '#'):p2=INSTR(p1+1,r$, '#')
1440 IF LEN(c$)<>p1-1 THEN 1500
1450 p$=MID$(r$,p1+1,p2-p1-1)
1460 d$=MID$(r$,p2+1)
1470 FOR j=LEN(d$) TO 1 STEP -1:IF MID$(d$,j,1)=' ' THEN NEX
T:d$='':GOTO 1490
1480 d$=LEFT$(d$,j)
1490 PRINT p$,d$
1500 WEND
1510 PRINT:PRINT'PULSE UNA TECLA'

```

```

1520 WHILE INKEY$="" :WEND
1530 GOTO 1320
1540 CLS
1550 PRINT'GRABANDO EL FICHERO ':a$
1560 OPENOUT a$
1570 PRINT#9,re$
1580 PRINT:PRINT
1590 PRINT'CODIGO', "OCUPADAS", "LIBRES":PRINT'*****', "*****
**', "*****':PRINT
1600 s=0
1610 FOR i=1 TO 16
1620 fic=ASC(MID$(re$,i))-60:PRINT m$(i),fic,m(i)-fic
1630 FOR j=1 TO fic
1640 :BANKREAD,@r%,@r$,s+j
1650 PRINT#9,r$
1660 NEXT
1670 s=s+m(i)
1680 NEXT
1690 CLOSEOUT
1700 END
1710 p$=UPPER$(p$)
1720 p=INSTR(p$,CHR$(161)):IF p<>0 THEN MID$(p$,p,1)="@":GOT
O 1720
1730 p=INSTR(p$,CHR$(171)):IF p<>0 THEN MID$(p$,p,1)="@":GOT
O 1730
1740 p=2:p$="/"+p$+ "/" :WHILE p<=LEN(p$)-1
1750 IF ASC(MID$(p$,p))<64 OR ASC(MID$(p$,p))>91 THEN p$=LEF
T$(p$,p-1)+MID$(p$,p+1):GOTO 1770
1760 p=p+1
1770 WEND
1780 IF LEN(p$)=2 THEN p$="" ELSE p$=MID$(p$,2,LEN(p$)-2)
1790 RETURN
1800 RESTORE 1860
1810 c$=p$:c$="/"+c$+ "/"
1820 FOR i=0 TO 27
1830 READ b$,s$
1840 p=INSTR(c$,b$):IF p<>0 THEN c$=LEFT$(c$,p-1)+s$+MID$(c$
,p+LEN(b$)):GOTO 1840
1850 NEXT
1860 DATA PT,T,CT,T,BS,S,CC,Z,EE,E,OO,O,CE,ZE,CI,ZI,GE,JE,GI
,J,I,X,S,S,Z,CH,X,HU,GU,C,K,H,"",QU,K,Q,K,V,B,M,N,P,B,D,Z,LL
,Y,L,R,RR,R,GU,BU,C,U,O,U
1870 IF LEN(c$)=2 THEN c$="":GOTO 1960
1880 z=ASC(MID$(c$,LEN(c$)-1)):IF z<>65 AND z<>69 AND z<>73
AND z<>79 AND z<>85 AND LEN(c$)>3 THEN c$=LEFT$(c$,LEN(c$)-2
)+"/":GOTO 1940
1890 IF LEN(c$)<5 THEN 1940
1900 IF RIGHT$(c$,3)="ZA/" THEN c$=LEFT$(c$,LEN(c$)-3)+"/A/"
1910 IF RIGHT$(c$,3)="AA/" THEN c$=LEFT$(c$,LEN(c$)-3)+"/A/" :
GOTO 1940
1920 IF RIGHT$(c$,3)="ZU/" THEN c$=LEFT$(c$,LEN(c$)-3)+"/U/"
1930 IF RIGHT$(c$,3)="UU/" THEN c$=LEFT$(c$,LEN(c$)-3)+"/U/"
1940 IF LEN(c$)=2 THEN c$="":GOTO 1960
1950 c$=MID$(c$,2,LEN(c$)-2)
1960 RETURN
1970 s=1
1980 FOR i=1 TO 16
1990 IF m$(i)=LEFT$(c$,1) THEN RETURN
2000 s=s+m(i)
2010 NEXT
2020 GOSUB 1710
2030 GOSUB 1800:IF c$="" THEN 2070
2040 GOSUB 1970
2050 PRINT:PRINT'CODIGO: 'c$
2060 ev=ASC(MID$(re$,i))-60
2070 RETURN

```

### 4.3.7. Programa ORTOGRAFIA versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /--- DICCIONARIO (c)1986 GRAN BIBLIOTECA AMSTRAD ----
40 /----- PCW 8256/8512 -----
50 /--- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS ----
60 /-----
70 /
80 IF FIND$('m:sintax.dic')<>' THEN OPEN'R',1,'m:sintax.dic
',65:FIELD 1,65 AS campo$:FOR i=1 TO 1008:LSET campo$=SPACE$
(65):PUT 1,i:NEXT:CLOSE
90 in$=CHR$(27)+'f':ac$=CHR$(27)+'e':PRINT in$:kls$=CHR$(27)
+'E'+CHR$(27)+'H':DEF FN locate$(f,c)=CHR$(27)+'Y'+CHR$(32+f
)+CHR$(32+c):PRINT CHR$(27)+'x'
100 PRINT kls$:OPEN'R',1,'m:sintax.dic',65:FIELD 1,65 AS cam
po$
110 RESTORE 250:DIM m$(16),m(16):FOR i=1 TO 16:READ m$(i),m(
i):NEXT
120 f=10:c=28:PRINT FN LOCATE$(f,c):'F I C H E R O'
130 PRINT TAB(27)'*****'
140 PRINT:PRINT
150 PRINT TAB(24)'1.- CREACION NUEVO FICHERO'
160 PRINT
170 PRINT TAB(24)'2.- LECTURA DE FICHERO'
180 f$=INKEY$:IF f$<'1' OR f$>'2' THEN 180
190 f=19:c=1:PRINT FN locate$(f,c):PRINT ac$:INPUT '?NOMBRE
DEL FICHERO':a$:PRINT in$
200 IF a$="" THEN 120
210 IF LEN(a$)>8 THEN PRINT 'DEMASIADO LARGO' :GOTO
190
220 IF INSTR(a$,'.') THEN PRINT 'SIN ESPECIFICAR EXTENSION':
GOTO 190
230 IF INSTR(a$,' ') THEN PRINT 'NO DEBE CONTENER ESPACIOS':
GOTO 190
240 a$=UPPER$(a$)+'.DIC'
250 DATA A,103,B,168,E,79,F,32,G,21,I,32,J,23,K,91,N,83,R,81
,T,61,U,26,W,6,X,11,Y,10,Z,180,@
260 IF f$='1' THEN re$=STRING$(16,CHR$(60)):PRINT'CREANDO EL
NUEVO FICHERO 'a$':OPEN'0',2,a$:PRINT #2,re$:CLOSE 2:LSE
T campo$=re$:PUT 1,1008:GOTO 440
270 PRINT 'LEYENDO EL FICHERO 'a$'
280 OPEN'i',2,a$
290 INPUT #2,re$
300 LSET campo$=re$:PUT 1,1008
310 s=0
320 PRINT kls$:PRINT'CODIGO','OCUPADAS','LIBRES':PRINT'*****
*','*****','*****':PRINT
330 FOR i=1 TO 16
340 fic=ASC(MID$(re$,i))-60:PRINT m$(i),fic,m(i)-fic
350 FOR j=1 TO fic
360 INPUT #2,r$
370 LSET campo$=r$:PUT 1,s+j
380 NEXT
390 s=s+m(i)
400 NEXT
410 CLOSE 2
420 PRINT:PRINT'PULSE UNA TECLA'
430 WHILE INKEY$="" :WEND
440 PRINT kls$
450 f=7:c=30:PRINT FN LOCATE$(f,c):PRINT 'O R T O G R A F I
A'
460 PRINT TAB(29)'*****'
470 PRINT:PRINT

```

```

480 PRINT TAB(27)*1.- GESTION BASE DE DATOS*
490 PRINT
500 PRINT TAB(27)*2.- CONSULTA ORTOGRAFICA*
510 PRINT
520 PRINT TAB(27)*3.- SALIDA DE PROGRAMA*
530 x$=INKEY$:IF x$<'1' OR x$>'3' THEN 530
540 ON VAL(x$) GOTO 550,1340,1580
550 PRINT kls$
560 f=:c=30:PRINT FN LOCATE$(f,c);:PRINT 'B A S E   D A T O
S'
570 PRINT TAB(29)******'
580 PRINT:PRINT
590 PRINT TAB(31)*1.- INTRODUCCION*
600 PRINT
610 PRINT TAB(31)*2.- BAJA*
620 PRINT
630 PRINT TAB(31)*3.- LISTADO*
640 PRINT
650 PRINT TAB(31)*4.- RETORNO*
660 x$=INKEY$:IF x$<'1' OR x$>'4' THEN 660
670 ON VAL(x$) GOTO 680,820,1020,440
680 PRINT kls$
690 PRINT ac$:INPUT"?PALABRA";p$:PRINT in$
700 IF p$="" THEN 550
710 GOSUB 2060:IF c$="" THEN 680
720 IF ev=m(i) THEN PRINT:PRINT'NO HAY ESPACIO PARA CODIGOS
'm$(i)'...':GOTO 790
730 PRINT:PRINT"?DESCRIPCION ('63-LEN(p$)-LEN(c$)'CARACTERES
)";
740 PRINT ac$:INPUT d$:PRINT in$
750 r$=LEFT$(c$+'#'+p$+'#'+d$+SPACE$(63),65)
760 p=s+ev
770 LSET campo$=r$:PUT 1,p
780 MID$(re$,i,1)=CHR$(ev+61)
790 PRINT:PRINT'PULSE UNA TECLA*
800 WHILE INKEY$="" :WEND
810 GOTO 680
820 PRINT kls$
830 PRINT ac$:INPUT"?PALABRA PARA BAJA";p$:PRINT in$
840 IF p$="" THEN 550
850 GOSUB 2060
860 FOR j=s TO s+ev-1
870 GET 1,j:r$=campo$
880 p1=INSTR(r$,'#')+1:p2=INSTR(p1,r$,'#'):pb$=MID$(r$,p1,p2
-p1)
890 IF pb$<>p$ THEN 970
900 FOR k=j+1 TO s+ev
910 GET 1,k:r$=campo$
920 LSET campo$=r$:PUT 1,k-1
930 NEXT k
940 LSET campo$=SPACE$(65):PUT 1
950 MID$(re$,i,1)=CHR$(59+ev)
960 PRINT:PRINT'PALABRA 'p$' BORRADA':GOTO 990
970 NEXT j
980 PRINT:PRINT'PALABRA 'p$' NO LOCALIZADA*
990 PRINT:PRINT'PULSE UNA TECLA*
1000 WHILE INKEY$="" :WEND
1010 GOTO 820
1020 PRINT kls$
1030 PRINT ac$:INPUT"?A PARTIR DE QUE CODIGO";c$:PRINT in$
1040 IF c$="" THEN PRINT kls$:GOTO 550
1050 c$=UPPER$(LEFT$(c$,1))
1060 FOR i=1 TO 16
1070 IF c$=m$(i) THEN 1100
1080 NEXT
1090 GOTO 1020

```

```

1100 PRINT:PRINT"?PANTALLA (P) O IMPRESORA (I)?"
1110 x$=INKEY$:IF x$="" THEN 1110
1120 IF UPPER$(x$)<>'P' AND UPPER$(x$)<>'I' THEN 1110
1130 IF UPPER$(x$)='P' THEN ch=0 ELSE ch=8
1140 PRINT:PRINT"PULSE <SPACE> PARA DETENER EL LISTADO"
1150 IF ch THEN LPRINT:LPRINT "PALABRA", "CODIGO", "D E S C R
I P C I O N" ELSE PRINT:PRINT "PALABRA", "CODIGO", "D E S C R
I P C I O N"
1160 IF ch THEN LPRINT "*****", "*****", "*****"
**** ELSE PRINT "*****", "*****", "*****"
1170 IF ch THEN LPRINT ELSE PRINT
1180 GOSUB 2010
1190 FOR i=s TO 1007
1200 GET 1, i: r$=campo$
1210 p1=INSTR(r$, "#"): p2=INSTR(p1+1, r$, "#")
1220 IF p1=0 OR p2=0 OR p2=p1+1 THEN 1290
1230 c$=LEFT$(r$, p1-1)
1240 p$=MID$(r$, p1+1, p2-p1-1)
1250 d$=MID$(r$, p2+1)
1260 FOR j=LEN(d$) TO 1 STEP -1: IF MID$(d$, j, 1)=' ' THEN NEX
T: d$="": GOTO 1280
1270 d$=LEFT$(d$, j)
1280 IF ch THEN LPRINT p$, c$, d$ ELSE PRINT p$, c$, d$
1290 IF INKEY$="" THEN 1310
1300 NEXT
1310 PRINT:PRINT"PULSE UNA TECLA"
1320 WHILE INKEY$="" :WEND
1330 GOTO 1020
1340 PRINT kls$
1350 PRINT ac$: INPUT"?PALABRA"; p$: PRINT in$
1360 IF p$="" THEN PRINT kls$: GOTO 440
1370 GOSUB 2060: IF c$="" THEN 1340
1380 PRINT
1390 p=s
1400 WHILE p<s+m(i)
1410 FOR x=p TO s+m(i)
1420 r%=-1: GET 1, x
1430 IF LEFT$(campo$, LEN(c$))=c$ THEN r%=x: GOTO 1450 ELSE NE
XT
1440 IF r%<0 THEN p=s+m(i): GOTO 1550
1450 p=r%
1460 GET 1, p: r$=campo$: p=p+1
1470 p1=INSTR(r$, "#"): p2=INSTR(p1+1, r$, "#")
1480 IF LEN(c$)<>p1-1 THEN 1530
1490 p$=MID$(r$, p1+1, p2-p1-1)
1500 d$=MID$(r$, p2+1)
1510 FOR j=LEN(d$) TO 1 STEP -1: IF MID$(d$, j, 1)=' ' THEN NEX
T: d$="": GOTO 1530
1520 d$=LEFT$(d$, j)
1530 PRINT p$, d$
1540 WEND
1550 PRINT:PRINT"PULSE UNA TECLA"
1560 WHILE INKEY$="" :WEND
1570 GOTO 1340
1580 PRINT kls$
1590 PRINT"GRABANDO EL FICHERO "; a$
1600 OPEN"O", 2, a$
1610 PRINT #2, re$
1620 PRINT:PRINT
1630 PRINT"CODIGO", "OCUPADAS", "LIBRES": PRINT"*****", "*****
**", "*****": PRINT
1640 s=0
1650 FOR i=1 TO 16
1660 fic=ASC(MID$(re$, i))-60: PRINT m$(i), fic, m(i)-fic
1670 FOR j=1 TO fic
1680 GET 1, s+j: r$=campo$

```

```

1690 PRINT #2,r$
1700 NEXT
1710 s=s+m(i)
1720 NEXT
1730 CLOSE
1740 PRINT:PRINT "PULSE UNA TECLA":WHILE INKEY$="":WEND:PRINT
ac$:PRINT CHR$(27)+"y":END
1750 p$=UPPER$(p$)
1760 p=INSTR(p$,CHR$(161)):IF p<>0 THEN MID$(p$,p,1)="@":GOTO
0 1760
1770 p=INSTR(p$,CHR$(171)):IF p<>0 THEN MID$(p$,p,1)="@":GOTO
0 1770
1780 p=2:p$="/"+p$+"/":WHILE p<=LEN(p$)-1
1790 IF ASC(MID$(p$,p))<64 OR ASC(MID$(p$,p))>91 THEN p$=LEFT$(p$,p-1)+MID$(p$,p+1):GOTO 1810
1800 p=p+1
1810 WEND
1820 IF LEN(p$)=2 THEN p$="" ELSE p$=MID$(p$,2,LEN(p$)-2)
1830 RETURN
1840 RESTORE 1900
1850 c$=p$:c$="/"+c$+"/"
1860 FOR i=0 TO 27
1870 READ b$,s$
1880 p=INSTR(c$,b$):IF p<>0 THEN c$=LEFT$(c$,p-1)+s$+MID$(c$,
p+LEN(b$)):GOTO 1880
1890 NEXT
1900 DATA PT,T,CT,T,BS,S,CC,Z,EE,E,OO,O,CE,ZE,CI,ZI,GE,JE,GI
,J1,X,S,S,Z,CH,X,HU,GU,C,K,H,"",QU,K,Q,K,V,B,M,N,P,B,D,Z,LL,
Y,L,R,RR,R,GU,BU,C,U,O,U
1910 IF LEN(c$)=2 THEN c$="":GOTO 2000
1920 z=ASC(MID$(c$,LEN(c$)-1)):IF z<>65 AND z<>69 AND z<>73
AND z<>79 AND z<>85 AND LEN(c$)>3 THEN c$=LEFT$(c$,LEN(c$)-2
)+"/":GOTO 1980
1930 IF LEN(c$)<5 THEN 1980
1940 IF RIGHT$(c$,3)="ZA/" THEN c$=LEFT$(c$,LEN(c$)-3)+"A/"
1950 IF RIGHT$(c$,3)="AA/" THEN c$=LEFT$(c$,LEN(c$)-3)+"A/"
GOTO 1980
1960 IF RIGHT$(c$,3)="ZU/" THEN c$=LEFT$(c$,LEN(c$)-3)+"U/"
1970 IF RIGHT$(c$,3)="UU/" THEN c$=LEFT$(c$,LEN(c$)-3)+"U/"
1980 IF LEN(c$)=2 THEN c$="":GOTO 2000
1990 c$=MID$(c$,2,LEN(c$)-2)
2000 RETURN
2010 s=1
2020 FOR i=1 TO 16
2030 IF m$(i)=LEFT$(c$,1) THEN RETURN
2040 s=s+m(i)
2050 NEXT
2060 GOSUB 1750
2070 GOSUB 1840:IF c$="" THEN 2110
2080 GOSUB 2010
2090 PRINT:PRINT"CODIGO: "c$
2100 ev=ASC(MID$(re$,i))-60
2110 RETURN

```

## PUESTA EN PRÁCTICA

La efectividad del corrector ortográfico puede que nos asombre en muchas ocasiones, y lo cierto es que sin ánimo de resultar inmodestos funciona bastante bien. Introduzcamos por ejemplo los siguientes vocablos, indicando su correcta ortografía:

ADVENIMIENTO  
HOLA (Saludo)  
OLA (Onda marina)

Si en consulta ortográfica introducimos OLA, el programa nos responderá con sus dos posibles variantes, dado que es el significado de la palabra en su contexto, la que señala su ortografía. Por el contrario, con ADVENIMIENTO apreciaremos una obstinación casi pedante del programa en su resistencia contra nuestra incultura ortográfica. Así, cualquiera de las siguientes palabras serán indicadas por el programa como incorrectas, señalándonos su forma válida (ADVENIMIENTO): HADVENIMIENTO, AZVENIMIENTO, HAZVENIMIENTO, ADBENIMIENTO, AZBENIMIENTO, HAZBENIMIENTO, HADBENIMIENTO, ADVENIMIENTU, AZVENIMIENTU, HAZVENIMIENTU, ADBENIMIENTU, AZBENIMIENTU, HAZBENIMIENTU, HADBENIMIENTU, en definitiva, el programa detecta ¡un amplio repertorio de burradas!

# PROCESO DE LENGUAJE NATURAL

A large, stylized green letter 'S' with a white outline, positioned at the beginning of the first paragraph.

on muchos los sistemas de proceso de lenguaje natural diseñados por el hombre. Para aumentar nuestra pequeña cultura sobre IA, que nos capacite para seguir a la perfección cualquier libro o conversación sobre el tema, haremos ahora mención a algunos de los más renombrados.

Uno de ellos es el SHRDLU, de Terry Winograd, al cual ya nos hemos referido al hablar de la percepción, creando nuestra propia versión: 'VODBIR'. SHRDLU se puede considerar como una mera demostración del punto hasta el que pueden llegar este tipo de sistemas; no obstante, carece de una utilidad práctica palpable.

No es este el caso de LIFER, del cual hablamos ya en el capítulo anterior: un interface de carácter general para el acceso a bases de datos.

## DE LA LUNA AL BASE-BALL

También la era espacial tuvo algo que decir en este sentido, y otro sistema de gran importancia en el tema que nos ocupa es el denominado LUNAR, diseñado por William Woods con el fin de ayudar a

los geólogos a recabar información sobre las muestras de minerales lunares, obtenidos en la exploración del Apolo XII.

Como dato curioso sobre este sistema, añadiremos que dispone de un vocabulario de unas 3.500 palabras, lo cual no le sitúa muy lejos de las 1.000 e incluso 2.000 que puede llegar a alcanzar nuestro corrector ortográfico en un CPC. Téngase en cuenta que este cálculo se efectúa para el RAM-DISC de 64 K de un CPC 6128, ya que el número llegaría a las 3.500 aprovechando a tope la memoria del PCW y más aún con la base de datos soportada por el disco de 1 Mb del PCW 8512.

Dentro del campo de la ayuda a los geólogos, también contamos con el programa PROSPECTOR; pero no sólo a la Geología se destinan este tipo de sistemas, sino también a otros temas algo más lúdicos. Tal es el caso de BASE-BALL, diseñado por Bert Green para informar al usuario, sobre los acontecimientos más relevantes acaecidos en la liga profesional de Base-Ball de los Estados Unidos durante un año.

Todos estos programas, se constituyen en sistemas expertos sobre determinados temas, y aunque destacan por su refinamiento en el sistema de comunicación con el usuario, no es ésta su única misión (a excepción de SHRDLU). En el último capítulo de este volumen, dedicado al estudio de los sistemas expertos, tendremos la oportunidad de conocer más programas de esta clase.

Ahora bien, ningún comentario sobre el proceso del lenguaje natural está completo sin mencionar el programa ELIZA, que merece un tratamiento separado como obra clave en este campo.

## UN PROGRAMA 'ROGERIANO'

He aquí un sistema dedicado única y exclusivamente, al menos en un principio, al diálogo mediante el lenguaje natural con el usuario. En resumen, se trata de una parodia de las técnicas psiquiátricas seguidas por la escuela de Carl Rogers, que propugnan básicamente que el peso de la conversación entre médico y paciente recaiga fundamentalmente en éste último, actuando el psiquiatra como un espejo de las observaciones y sentimientos experimentados por el enfermo.

El proyecto que comenzó de una manera tan simple, e incluso burlona, se llegó a convertir en una auténtica pesadilla para su creador, el profesor Joseph Weizenbaum del Masachussets Institute of Technology (Instituto Tecnológico de Masachussets, MIT), cuando

comenzó a comprobar que las reacciones de los «pacientes» humanos ante ELIZA se salían completamente de lo previsto. Pero comencemos por el principio.

## ELIZA AL HABLA

El MIT es probablemente el centro informático más renombrado en Occidente, contribuyendo al desarrollo universal de esta ciencia con aportaciones como el lenguaje Lisp o programas de la calidad de SHRDLU o ELIZA,

En este ambiente desarrollaba su trabajo como profesor de Informática a mediados de la década de los 60 Joseph Weinzenbaum, que entre 1964 y 1966 tuvo la brillante idea de desarrollar un programa que imitara la conducta de los psiquiatras de la escuela Rogeriana con sus pacientes.

En realidad, la versión inicial de ELIZA estaba compuesta de dos secciones, una primera que analizaba la entrada del usuario y otra que constituía la base de datos para mantener una conversación sobre determinado tema. DOCTOR era uno de estos guiones y dado el impacto que produjo ELIZA, pasó a utilizarse en exclusiva, por lo que con el tiempo se produjo una identificación entre el programa madre y la versión, pasando a conocerse como una misma cosa.

Realmente, el nombre de ELIZA proviene de una particular ambición de su creador. Pretendía que su programa, del mismo modo que la Eliza de Pygmalion, aprendiera a expresarse cada vez mejor. No obstante, sus previsiones se vieron pronto rebasadas y ELIZA comenzó a hablar excesivamente bien.

En muy poco tiempo, el programa se hizo famoso en el MIT, no tanto entre los investigadores como entre los alumnos, que solicitaban angustiosamente tiempo de acceso para resolver sus problemas personales con DOCTOR.

Tras esta desconcertante reacción de los usuarios, la de su creador no lo fue menos, y decidió retirarse durante un par de años para reflexionar sobre el impresionante impacto producido por ELIZA, escribiendo una de las obras cumbre en la investigación de la inteligencia artificial: «El poder de los ordenadores y la razón humana» (*Computer power and human reason*, W. H. Freeman and Co., San Francisco 1976).

Independientemente de las opiniones muy personales vertidas en

el libro por el profesor Weinzenbaum, por otra parte algo inherente a un tema tan «opinable» como la IA, se trata de una obra de gran importancia, en la que el autor lanza una reflexión que algunos investigadores destacan: el profesor no había llegado a calcular el poder de engaño al usuario que se podía conseguir mediante un programa.

Sin embargo ¿dónde está el engaño del que nos habla Weinzenbaum?, ¿acaso no sabía todo aquel que accedía al programa que hablaba con una máquina? quizá, ¿el engaño no provenía tanto del programa como de la reacción aparentemente absurda de los «pacientes»? Uno de los extremos que prácticamente horrorizó al padre de ELIZA, fue descubrir como la gente se sentía fascinada por la conversación con el programa, como si de un humano se tratase. Pero ¿qué hay de malo o extraño en ello? ¿No nos sentimos también cautivados por la construcción de una maqueta de tren en miniatura, tanto como si de nosotros dependiese el tráfico ferroviario de una gran estación «de tamaño natural»?

Realmente, muchas personas pueden opinar que la reacción de Weinzenbaum fue bastante desproporcionada, pero en todo caso esto no le resta el mérito de su fran creación. No obstante, ahí no quedan las afirmaciones del profesor del MIT, y llegan hasta la polemicidad completa cuando asevera que existen determinadas tareas que nunca deberían ser efectuadas por máquinas.

A esta afirmación responde el doctor Kenneth Colby con auténtica vehemencia: ¿acaso el que un ordenador ayude a la gente es peor que dejarla sufrir?

La verdad es que si durante el proceso de la Revolución Industrial hubieran existido muchas personas del parecer de Weinzenbaum, lo más probable es que no hubiéramos conseguido alcanzar nuestro actual grado de desarrollo.

## WEINZENBAUM VS COLBY

Pronto los dos científicos, ambos relacionados por el proyecto ELIZA, vieron en clara oposición sus ideas, hasta su separación a raíz de la publicación de un trabajo sobre DOCTOR por parte de Colby, en el cual se resaltaban sus posibles aplicaciones terapéuticas.

Lo cierto es que probablemente uno de los grandes problemas de Weinzenbaum fue hacer un programa demasiado bueno; un programa que demostraba justamente lo opuesto a aquello que su creador quería dar a entender con él. En principio, Weinzenbaum afirmaba, e incluso llegó a publicar una nota sobre el tema, que un ordenador no

podría llegar a comprender el lenguaje humano, dada la imposibilidad de encontrar una solución general a este problema.

No cabe duda que las reacciones fueron, si desproporcionadas, al menos en cierto modo justificadas, ya que el comportamiento de algunas personas ante el programa fue sorprendente. Por ejemplo, su secretaria, que había colaborado en el desarrollo de ELIZA y por tanto tenía un considerable conocimiento del mismo, habiendo comenzado una conversación con el programa, se vio repentinamente enredada en un diálogo tal, que se le hizo necesario solicitar al profesor que abandonara la habitación para terminar sin embarazo su conversación.

ELIZA empezaba a tomar los rasgos de un ser humano y esta antropomorfización preocupaba seriamente a Weizenbaum. Descubrió con espanto que los accesos al programa por parte de los alumnos del MIT no eran sólo numerosos, sino dilatados (hasta de una hora) e incluso secretos, puesto que se efectuaban a horas intempestivas. Preocupado por el «poder de engaño» del programa, su creador apuntó la posibilidad de conexión de un dispositivo que registrara todas las conversaciones mantenidas, incluso a las altas horas de la madrugada a las que tenían lugar. Tan sólo la sugerencia de esta idea provocó una auténtica revuelta entre los «Elizadictos», dado el carácter de ingerencia en su vida privada que ello suponía.

Por el contrario, mientras que la implicación emocional que ELIZA producía en sus usuarios alarmaba a Weizenbaum, fascinaba a Colby. Este, profesor de Psiquiatría en la Universidad de California, había conocido a Weizenbaum en la Universidad de Stanford, y llamado poderosamente por el tema de la IA, investigó particularmente sobre los efectos y posibles aplicaciones de DOCTOR en un campo que el conocía muy bien.

Colby llegó a sugerir que con un pequeño perfeccionamiento, DOCTOR podría colaborar con psiquiatras en el tratamiento de las enfermedades como una herramienta más de la terapéutica. Esta afirmación, proveniente por otra parte de un profesor de Psiquiatría, chocaba directamente con la tesis de Weizenbaum, según la cual determinadas tareas no podían, ni debían, ser desempeñadas por máquinas.

Otra de las anécdotas más difundidas sobre este programa, asegura que un científico soviético en viaje por los Estados Unidos, se vio en una ocasión enredado en una conversación auténticamente embarazosa sobre sus problemas familiares, con una versión de ELIZA creada por el propio Colby.

En todo caso, no deberíamos preocuparnos excesivamente por el hecho de que es un ordenador quien nos ayuda, a nivel físico o mental; lo realmente importante es que colabora con el hombre, aportándole un beneficio. Aunque esto no deja de ser una simple opinión en la que el lector puede discrepar, lo cierto es que sistemas expertos como MYCIN, que veremos más adelante, ya están colaborando con éxito en el campo médico. ¿Por qué negarnos a que nos ayude?... aunque sólo sea una máquina.

Si no nos preocupa, e incluso nos vanagloriamos de conseguir que un pulmón artificial, un sistema mecánico, mantenga con vida a un hombre que sin su concurso habría fallecido. ¿Por qué alarmarnos de la existencia de una máquina que nos ayude en el tratamiento psíquico? ¿No es acaso esto dar a un simple artefacto el servicio del hombre más importancia de la que realmente tiene?

## EL APRENDIZAJE

**L**a capacidad de aprendizaje es otra de las características que distinguen al hombre del resto de los seres no inteligentes, y será por tanto el próximo objeto de nuestro estudio. Comencemos de una manera agradable: jugando.

He aquí el listado de un programa que, si bien no tiene nada de inteligente, si será suficientemente ilustrativo del tema del aprendizaje. Debemos continuar la lectura a partir del listado, cuando hayamos conseguido ganar durante cinco partidas seguidas al ordenador, o bien cuando no hayamos logrado vencerle tras algún tiempo; aunque hay que hacer una advertencia: no hagamos la trampa de desesperarnos demasiado pronto.

Las reglas del juego son bien fáciles. El ordenador comienza pidiéndonos un número entre 1 y 10; él contestará con otro, sumándole al nuestro una cantidad entre 1 y 10, al cual deberemos responder con otra cifra mayor que la anterior en al menos una unidad y nunca superior a ésta en más de diez unidades y así sucesivamente. Por ejemplo, si el ordenador propone el 15, nosotros deberemos señalar otro número entre 16 y 25. Las entradas erróneas serán rechazadas por el programa con un pitido. El juego lo gana aquel que consiga llegar a decir 100.

#### 4.5.4. Programa EL 100 versión CPC.

```
10 REM EL 100 $ FERNANDO LOPEZ MARTINEZ 1986 $ CPC 6128
20 MODE 0
30 t=0
40 INPUT"dime tu numero ",n
50 IF n>t+10 OR n<t+1 THEN PRINT CHR$(?):GOTO 40
60 IF n=100 THEN PRINT"GANASTE":END
70 t=n
80 CLS
90 n=INT(t/10)
100 n=10*n+(n+1):IF n<t+1 THEN n=INT(t/10)+1:GOTO 100
110 IF n>100 THEN n=100
120 IF n>t+10 THEN n=n-10
130 PRINT "el mio es",n
140 IF n=100 THEN PRINT"GANE":END
150 t=n
160 GOTO 40
```

#### 4.5.5. Programa EL 100 versión PCW.

```
10 REM EL 100 $ FERNANDO LOPEZ MARTINEZ 1986 $ PCW 8256/8512
20 k1s$=CHR$(27)+'E'+CHR$(27)+'H':PRINT k1s$
30 t=0
40 INPUT"dime tu numero ",n
50 IF n>t+10 OR n<t+1 THEN PRINT CHR$(?):GOTO 40
60 IF n=100 THEN PRINT"GANASTE":END
70 t=n
80 PRINT k1s$
90 n=INT(t/10)
100 n=10*n+(n+1):IF n<t+1 THEN n=INT(t/10)+1:GOTO 100
110 IF n>100 THEN n=100
120 IF n>t+10 THEN n=n-10
130 PRINT "el mio es",n
140 IF n=100 THEN PRINT"GANE":END
150 t=n
160 GOTO 40
```

Seguramente, no habremos tardado mucho tiempo en averiguar que el jugador que comience tiene ganada la partida de antemano, ya que el que diga el número 1 tiene la victoria asegurada. Examinemos el proceso de aprendizaje que hemos seguido.

Cuando el ordenador haya alcanzado el 89 durante la primera partida, habremos llegado a la conclusión que el juego estaba perdido, con lo que nuestro objetivo en la siguiente partida habrá sido llegar a este punto, para dejar a la máquina en la misma situación. No obstante, habremos constatado a continuación, que el programa siempre alcanza el número 78 para impedirnos a nosotros decir el 89. De esta forma, llegar a 78 se habrá convertido en el objetivo inmediato.

Las partidas se habrán ido desarrollando encontrando cada vez una meta más baja para alcanzar la victoria: 67, 56, 45, 34, 23, 12 y finalmente 1. La fórmula para la obtención de estos números «ganadores» es bien simple:  $N=10 \cdot A + A + 1$ ,  $\forall A [0,9]$  es decir, el número es igual a 10 multiplicado por A, más A, más 1, para todo A comprendido en el intervalo cerrado entre 0 y 9. La regla mnemotécnica es quizá más sencilla: los números mágicos son aquellos en que la unidad supera en 1 a la decena.

Este sistema de aprendizaje que hemos seguido se denomina *rutina*, aunque tras este término tan rimbombante se esconde otro bastante menos halagüeño: método de «a fuerza de perder»... No nos tenemos que sentir avergonzados, puesto que finalmente hemos conseguido descubrir el secreto de cómo ganar la partida.

Pues bien, veremos ahora un ingenioso experimento en el que un ordenador aprende también por la rutina, o a fuerza de perder, como prefiramos. Lo extraordinario del caso, no es ya el hecho de que el ordenador sea lo suficientemente inteligente como para aprender, sino que esta asombrosa máquina está constituida única y exclusivamente ¡por cajas de cerillas y trocitos de papel!

## LAS CAJAS DE CERILLAS INTELIGENTES

Durante su estudio sobre el proceso de aprendizaje automático, el investigador Donald Michie de la Universidad de Edinburgo, calculó las posibles posiciones que se puede encontrar un jugador de tres en raya (¡al fin y al cabo no son tantas!).

Ni corto ni perezoso, emprendió la dura tarea de construir su rudimentario ordenador. Fue vaciando cajas de cerillas y pintando encima de cada una de ellas una de las posibles situaciones del tablero de las tres en raya, numerando las casillas que quedaban vacías. A continuación, depositó en las cajas un número igual de papeletos con cada uno de los números de casilla que quedaban vacíos en la jugada.

Así por ejemplo, en una jugada en que las casillas vacías fueran la 1 y la 2, habría un número igual de papeles con el 1 que con el 2.

Si construir el ordenador había sido trabajoso, ponerlo a andar debió ser desesperante. La primera jugada consistía en extraer un papel al azar de la caja que tuviera todas las casillas vacías. A continuación, se hacía la jugada indicada por el papeletito, anotándose el movimiento en una hoja. La partida continuaba de esta manera.

Si al final del juego, sus cajas de cerillas habían conseguido ven-

cer, entonces se aumentaban las posibilidades de repetir las mismas jugadas, añadiendo papelitos iguales a los que habían propiciado la victoria en cada caja. Si la partida acababa en empate, no se efectuaba ninguna modificación y si por el contrario, se había perdido, se retiraba un papelito de los correspondientes a las jugadas erróneas en cada una de las cajas.

Si ya de por sí el experimento es asombroso, más lo es su resultado: lento, pero seguro. Después de haber jugado 1.000 partidas, además de haberle crecido la barba a Michie, su ordenador-fosforero conseguía ganar más del 75 por 100 de las partidas. En fin, el que no se lo crea ya sabe lo que tienen que hacer: pasar por el estanco más cercano y...

## LAS TRES EN RAYA

El juego de las tres en raya se ha convertido en un histórico para la inteligencia artificial, y no sólo porque según Michie fuera un juego de los de «para comer cerillas», sino porque nos salvó del desencadenamiento de la Tercera Guerra Mundial. Nos explicamos.

Los cinéfilos informáticos seguro que habrán tenido la oportunidad de ver la película «Juegos de Guerra». En ella, el ordenador WHOPR, tenía encomendado el control del sistema de Defensa Estratégica de los Estados Unidos. Su programador, Falkner, traumatizado por la muerte de su mujer y su hijo en un accidente, decide retirarse del proyecto, y dotado de una nueva identidad es dado oficialmente por muerto.

Un joven Hacker (pirata de sistemas informáticos), intentando acceder al banco de datos de una firma de videojuegos, encuentra la clave para la entrada a WHOPR: JOSHUA (el nombre del hijo de Falkner). Sin tener conciencia de con qué ordenador ha entrado en contacto, le propone jugar, nada más y nada menos, que al juego de la Guerra Termonuclear Mundial; un juego al que el sistema de Falkner ha aprendido con el tiempo a jugar extraordinariamente bien.

Tras multitud de peripecias, el protagonista consigue detener el juego que la computadora se había tomado completamente en serio, de una forma extraordinariamente ingeniosa: descubriendo que el juego de la guerra no tiene ningún ganador, y por tanto es absurdo. ¿Qué cómo lo consigue? Sencillamente por analogía con el juego de las tres en raya.

Este es un juego al cual se entregan febrilmente los estudiantes de

EGB, pero que cuando llegan a ser más mayores abandonan, puesto que siempre quedan empatados. Del mismo modo, JOSHUA comprende que jugar a un juego sin ganador es absurdo, y antes de continuar en su carrera hacia el inicio de la ignición de los misiles, decide comprobar previamente si el juego de la guerra nuclear no tiene vencedor, como efectivamente sucede.

El autor de este libro le debe a Hollywood dos favores. En primer lugar, haber respirado aliviado en su butaca cuando tras minutos de tensión, Joshua deja oír su voz metálica: «Este juego es absurdo. ¿Echamos una partidita de ajedrez, profesor Falkner?» Y por otra parte, haber obtenido la idea para un curioso programa: las tres en raya.

Si bien es cierto que nuestra versión denominada JOSHUA, no llega a jugar a la velocidad que el Séptimo Arte nos hace ver en «Juegos de Guerra», no es excesivamente lento, y sobre todo, podemos apreciar como juega bastante bien, hasta llegar a la triste conclusión, si topa con un jugador tan «inteligente» como él, que el juego no tiene solución.



4.5.1. El sueño de la inteligencia artificial llevado a las pantallas de cine.

Hemos tenido la oportunidad de ver muchas versiones del programa conocido como tres en raya, o sus equivalentes anglosajones: tic-tac-toe o noughts and crosses (ceros y cruces), pero JOSHUA es sin duda el mejor exponente de este género que el autor ha tenido ocasión de presenciar... después desde luego del programa del profesor Falkner.

#### 4.5.6. Programa JOSHUA versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- JOSHUA (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- CPC 464/664/6128 -----
50 /-----
60 /
70 ch=8:!SET,0:!SET,1
80 MODE 0
90 DIM t(3,3)
100 e=0
110 DATA 5,3,2,2,4,2,6,2,8,2,1,1,3,1,7,1,9,1
120 DATA 5,3,1,2,3,2,7,2,9,2,2,1,4,1,6,1,8,1
130 DIM d(18,2)
140 FOR i=1 TO 18
150 READ d(i,1),d(i,2)
160 NEXT
170 n=0
180 FOR i=2 TO 15 STEP 4:IF i=10 THEN i=11
190 b=i+INT(RND(TIME)*4)
200 a=d(b,1):d(b,1)=d(i,1):d(i,1)=a
210 NEXT
220 h=INT(RND*0.5)
230 IF h=1 THEN 680
240 h=0:GOSUB 970:IF n=10 THEN r=0:GOTO 800
250 IF n=1 THEN p=INT(RND*9)+1:f=INT((p-1)/3)+1:c=p-3*(f-1):
GOTO 670
260 IF n=3 AND t(2,2)=0 THEN f=2:c=2:GOTO 670
270 sx=0:px=1
280 FOR k=1 TO 3
290 FOR l=1 TO 3
300 ps=3*(k-1)+l
310 IF t(k,l)=ASC('X') THEN sx=sx+ps
320 IF ps/2<>INT(ps/2) AND t(l,k)=ASC('X') THEN px=0
330 IF t(k,l)<>0 THEN 390
340 t(k,l)=ASC('0')
350 m=3*ASC('0')
360 GOSUB 1110
370 IF r=1 THEN k=3:l=3:GOTO 390
380 t(k,l)=0
390 NEXT l,k
400 IF r=1 THEN 800
410 FOR k=1 TO 3
420 FOR l=1 TO 3
430 IF t(k,l)<>0 THEN 490
440 t(k,l)=ASC('X')
450 m=3*ASC('X')
460 GOSUB 1110
470 IF r=1 THEN f=k:c=l:k=3:l=3:GOTO 490
480 t(k,l)=0

```

```

490 NEXT l,k
500 IF r=1 THEN 670
510 b=0:IF t(2,2)=ASC('X') THEN b=9
520 v=0
530 FOR k=1 TO 9
540 p=d(b+k,1):q=d(b+k,2)
550 f1=INT((p-1)/3)+1:c1=p-3*(f1-1)
560 IF t(f1,c1)<>0 THEN 660
570 t(f1,c1)=ASC('0')
580 m=2*ASC('0')
590 GOSUB 1110
600 v1=3*r+q
610 IF b=9 THEN 640
620 IF sx+p=15 AND px=0 AND n>2 THEN v1=v1+3
630 IF sx-5=p AND px=1 AND n>2 THEN v1=v1+3
640 IF v1>v THEN f=f1:c=c1:v=v1
650 t(f1,c1)=0
660 NEXT
670 t(f,c)=ASC('0')
680 h=1:GOSUB 970:IF n=10 THEN r=0:GOTO 800
690 CLEAR INPUT
700 x$=INKEY$:IF x$="" THEN 700
710 IF x$<'0' OR x$>'9' THEN PRINT CHR$(7);:GOTO 690
720 x=VAL(x$)
730 f=INT((x-1)/3)+1:c=x-3*(f-1)
740 IF t(f,c)<>0 THEN PRINT CHR$(7);:GOTO 690
750 t(f,c)=ASC('X')
760 m=3*ASC('X')
770 GOSUB 1110
780 IF r=1 THEN 800
790 GOTO 240
800 CLS
810 IF r=0 THEN PRINT "EMPATADOS":e=e+1:GOTO 840
820 IF h=1 THEN PRINT "HAS VENCIDO":GOTO 840
830 PRINT "HAS PERDIDO"
840 GOSUB 1010
850 DATA 69,83,84,69,32,74,85,69,71,79,32,69,83,32,65,66,83,
85,82,68,79,46
860 DATA 63,69,67,72,65,77,79,83,32,85,78,65,32,80,65,82,84,
73,68,73,84,65,32,68,69,32,80,79,75,69,82,32,80,82,79,70,69,
83,79,82,32,70,65,76,75,78,69,82,63
870 IF e=5 THEN RESTORE 850:a$="":FOR i=1 TO 22:READ a:a$=a$
+CHR$(a):NEXT:MODE 2:PRINT#ch,a$:PRINT:a$="":FOR i=1 TO 49:R
EAD a:a$=a$+CHR$(a):NEXT:PRINT#ch,a$:LOCATE 1,24:END
880 LOCATE 1,25:PRINT"?OTRA PARTIDITA?";
890 x$=UPPER$(INKEY$):IF x$="" THEN 890
900 IF x$<'S' AND x$<'N' THEN PRINT CHR$(7);:GOTO 890
910 IF x$='N' THEN MODE 2:END
920 FOR i=1 TO 3
930 FOR j=1 TO 3
940 t(i,j)=0
950 NEXT j,i
960 GOTO 170
970 CLS
980 n=n+1
990 IF n=10 THEN RETURN
1000 IF h=1 THEN PRINT "TU MUEVES" ELSE PRINT "YO MUEVO"
1010 FOR i=1 TO 3
1020 FOR j=1 TO 3
1030 LOCATE 4+j*3,7+i*3
1040 IF t(i,j)=0 THEN PRINT CHR$(3*(i-1)+j+48) ELSE PRINT CH
R$(t(i,j))
1050 NEXT j,i
1060 MOVE 172,176:DRAWR 256,0
1070 MOVE 172,224:DRAWR 256,0
1080 MOVE 348,128:DRAWR 0,143

```

```

1090 MOVE 252,128:DRAWR 0,143
1100 RETURN
1110 r=0
1120 FOR i=1 TO 3
1130 s=0
1140 FOR j=1 TO 3
1150 s=s+t(i,j)
1160 NEXT
1170 IF s=m THEN r=1:i=3
1180 NEXT
1190 IF r=1 THEN RETURN
1200 FOR i=1 TO 3
1210 s=0
1220 FOR j=1 TO 3
1230 s=s+t(j,i)
1240 NEXT
1250 IF s=m THEN r=1:i=3
1260 NEXT
1270 IF r=1 THEN RETURN
1280 s=0
1290 FOR i=1 TO 3
1300 s=s+t(i,i)
1310 NEXT
1320 IF s=m THEN r=1:RETURN
1330 s=0
1340 FOR i=1 TO 3
1350 s=s+t(4-i,i)
1360 NEXT
1370 IF s=m THEN r=1
1380 RETURN

```

#### 4.5.7. Programa JOSHUA versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- JOSHUA (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 in$=CHR$(27)+'f':ac$=CHR$(27)+'e':PRINT in$:kls$=CHR$(27)
+'E'+CHR$(27)+'H':DEF FN locate$(x,y)=CHR$(27)+'Y'+CHR$(32+x)
+CHR$(32+y):pq$=CHR$(27)+'x':gr$=CHR$(27)+'y'
90 PRINT pq$:kls$:PRINT FN locate$(13,24)'PULSA UNA TECLA PA
RA COMENZAR':WHILE INKEY$='':time=1+(time MOD 100):WEND
100 RANDOMIZE time:b$=ac$+gr$
110 GOSUB 1450
120 DIM t(3,3)
130 e=0
140 DATA 5,3,2,2,4,2,6,2,8,2,1,1,3,1,7,1,9,1
150 DATA 5,3,1,2,3,2,7,2,9,2,2,1,4,1,6,1,8,1
160 DIM d(18,2)
170 RESTORE 140:FOR i=1 TO 18
180 READ d(i,1),d(i,2)
190 NEXT
200 n=0
210 FOR i=2 TO 15 STEP 4:IF i=10 THEN i=11
220 b=i+INT(RND(TIME)*4)
230 a=d(b,1):d(b,1)=d(i,1):d(i,1)=a
240 NEXT

```

```

250 h=INT(RND*0.5)
260 IF h=1 THEN 710
270 h=0:GOSUB 1000:IF n=10 THEN r=0:GOTO 820
280 IF n=1 THEN p=INT(RND*9)+1:f=INT((p-1)/3)+1:c=p-3*(f-1):
GOTO 700
290 IF n=3 AND t(2,2)=0 THEN f=2:c=2:GOTO 700
300 sx=0:px=1
310 FOR k=1 TO 3
320 FOR l=1 TO 3
330 ps=3*(k-1)+l
340 IF t(k,l)=ASC('X') THEN sx=sx+ps
350 IF ps/2<>INT(ps/2) AND t(l,k)=ASC('X') THEN px=0
360 IF t(k,l)<>0 THEN 420
370 t(k,l)=ASC('0')
380 m=3*ASC('0')
390 GOSUB 1100
400 IF r=1 THEN k=3:l=3:GOTO 420
410 t(k,l)=0
420 NEXT l,k
430 IF r=1 THEN 820
440 FOR k=1 TO 3
450 FOR l=1 TO 3
460 IF t(k,l)<>0 THEN 520
470 t(k,l)=ASC('X')
480 m=3*ASC('X')
490 GOSUB 1100
500 IF r=1 THEN f=k:c=l:k=3:l=3:GOTO 520
510 t(k,l)=0
520 NEXT l,k
530 IF r=1 THEN 700
540 b=0:IF t(2,2)=ASC('X') THEN b=9
550 v=0
560 FOR k=1 TO 9
570 p=d(b+k,1):q=d(b+k,2)
580 f1=INT((p-1)/3)+1:c1=p-3*(f1-1)
590 IF t(f1,c1)<>0 THEN 690
600 t(f1,c1)=ASC('0')
610 m=2*ASC('0')
620 GOSUB 1100
630 v1=3*r+q
640 IF b=9 THEN 670
650 IF sx+p=15 AND px=0 AND n>2 THEN v1=v1+3
660 IF sx-5=p AND px=1 AND n>2 THEN v1=v1+3
670 IF v1>v THEN f=f1:c=c1:v=v1
680 t(f1,c1)=0
690 NEXT
700 t(f,c)=ASC('0')
710 h=1:GOSUB 1000:IF n=10 THEN r=0:GOTO 820
720 x$=INKEY$:IF x$="" THEN 720
730 IF x$<'0' OR x$>'9' THEN PRINT CHR$(7);:GOTO 720
740 x=VAL(x$)
750 f=INT((x-1)/3)+1:c=x-3*(f-1)
760 IF t(f,c)<>0 THEN PRINT CHR$(7);:GOTO 720
770 t(f,c)=ASC('X')
780 m=3*ASC('X')
790 GOSUB 1100
800 IF r=1 THEN 820
810 GOTO 270
820 REM PRINT k1s$
830 IF r=0 THEN PRINT FN locate$(11,46)'E M P A T A D O S':e
=e+1:GOTO 860
840 IF h=1 THEN PRINT FN locate$(11,46)'H A S V E N C I D
O':GOTO 860
850 PRINT FN locate$(11,46)'H A S P E R D I D O'
860 GOSUB 1040

```

```

870 DATA 69,83,84,69,32,74,85,69,71,79,32,69,83,32,65,66,83,
85,82,68,79,46
880 DATA 63,69,67,72,65,77,79,83,32,85,78,65,32,80,65,82,84,
73,68,73,84,65,32,68,69,32,80,79,75,69,82,32,80,82,79,70,69,
83,79,82,32,70,65,76,75,78,69,82,63
890 IF e=5 THEN RESTORE 870:a$="":FOR i=1 TO 22:READ a:a$=a$
+CHR$(a):NEXT:PRINT k1s$:PRINT a$:PRINT:a$="":FOR i=1 TO 49:
READ a:a$=a$+CHR$(a):NEXT:PRINT a$:PRINT FN locate$(21,1);ac
$:END
900 PRINT FN locate$(16,29)"?OTRA PARTIDITA?"
910 x$=UPPER$(INKEY$):IF x$="" THEN 910
920 IF x$(">"S" AND x$("<"N" THEN PRINT CHR$(7);:GOTO 910
930 IF x$="N" THEN PRINT ac$;gr$:END
940 PRINT FN locate$(11,46)SPACE$(22)FN locate$(16,29)SPACE$
(17)
950 FOR i=1 TO 3
960 FOR j=1 TO 3
970 t(i,j)=0
980 NEXT j,i
990 GOTO 200
1000 n=n+1
1010 IF n=10 THEN RETURN
1020 PRINT FN locate$(6,29);
1030 IF h=1 THEN PRINT "T U M U E V E S"CHR$(7) ELSE PRIN
T "Y O M U E V O "
1040 FOR i=1 TO 3
1050 FOR j=1 TO 3
1060 PRINT FN locate$(7+i*2,29+j*4);
1070 IF t(i,j)=0 THEN PRINT CHR$(3*(i-1)+j+48) ELSE PRINT CH
R$(t(i,j))
1080 NEXT j,i
1090 RETURN
1100 r=0
1110 FOR i=1 TO 3
1120 s=0
1130 FOR j=1 TO 3
1140 s=s+t(i,j)
1150 NEXT
1160 IF s=m THEN r=1:i=3
1170 NEXT
1180 IF r=1 THEN RETURN
1190 FOR i=1 TO 3
1200 s=0
1210 FOR j=1 TO 3
1220 s=s+t(j,i)
1230 NEXT
1240 IF s=m THEN r=1:i=3
1250 NEXT
1260 IF r=1 THEN RETURN
1270 s=0
1280 FOR i=1 TO 3
1290 s=s+t(i,i)
1300 NEXT
1310 IF s=m THEN r=1:RETURN
1320 s=0
1330 FOR i=1 TO 3
1340 s=s+t(4-i,i)
1350 NEXT
1360 IF s=m THEN r=1
1370 RETURN
1380 DATA 134,138,138,138,142,138,138,138,142,138,138,138,14
0
1390 DATA 133,32,32,32,133,32,32,32,133,32,32,32,133
1400 DATA 135,138,138,138,143,138,138,138,143,138,138,138,14
1

```

```

1410 DATA 133,32,32,32,133,32,32,32,133,32,32,32,133
1420 DATA 135,138,138,138,143,138,138,138,143,138,138,138,14
1
1430 DATA 133,32,32,32,133,32,32,32,133,32,32,32,133
1440 DATA 131,138,138,138,139,138,138,138,139,138,138,138,13
7
1450 RESTORE 1380:PRINT k1$;FOR i=1 TO 7:FOR j=1 TO 13:READ
a:PRINT FN locate$(i+7,j+30);CHR$(a):NEXT j,i
1460 RETURN

```

En todo caso, para cuando nuestro Joshua descubra que el juego de las tres en raya no tiene solución, nos tiene reservada una curiosa sorpresa. Esta se verá realizada en los poseedores de un CPC y un sintetizador de voz MHT Ingenieros, si antes de cargar el programa ubicamos el software del periférico y añadimos las siguientes líneas al listado:

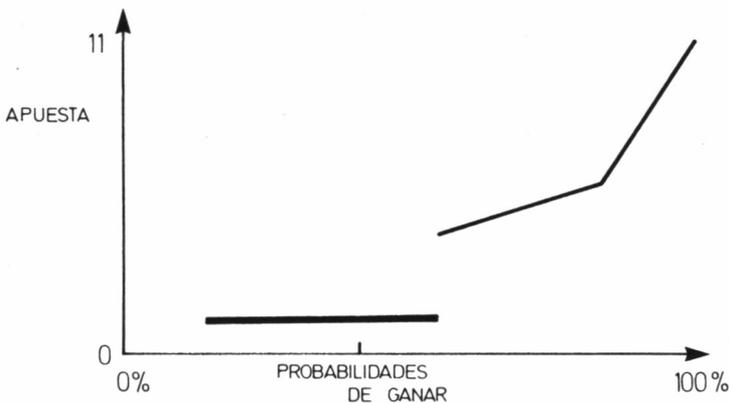
```
20 ch=8: |SET,0: |SET,1
```

La voz metálica de Joshua parecerá inundar de nuevo la habitación: «Este juego es absurdo, ¿echamos una partidita de...»

## ATENCIÓN A LOS FAROLES

Una baraja francesa y al menos dos jugadores inteligentes y arriesgados... la concurrencia suficiente para entregarse al juego del poker. Si no contamos con un oponente versado en las artes del naípe, tomemos asiento a un lado de la mesa, frente a frente con nuestro Amstrad: el digno y astuto adversario que falta para iniciar la partida. Para empezar vamos a darle un repaso a las reglas por las que se va a regir nuestra partida con el Amstrad.

El juego se llevará a cabo utilizando una baraja francesa, con sus cuatro palos (corazones, diamantes, tréboles y picas), incluyendo tanto números como figuras (jack, dama y rey), aunque desechando los dos comodines. En el caso de una partida de dos jugadores, como la que nos ocupa, se suelen eliminar determinados números de los palos. En nuestro caso concreto, el programa contempla cada palo constitui-



■ 4.5.8. Gráfico guía para la subasta.

do por las siguientes cartas: 8, 9, 10, J (jack), Q (dama), K (rey) y A (as).

Al comienzo de la partida, el ordenador sorteará quién será «mano» por primera vez; en las siguientes jugadas, el turno de «mano» irá pasando de uno a otro contendiente. Por supuesto, en el programa, al igual que en el juego real, el «mano» es aquel que se descarta primero.

A continuación, el ordenador reparte cinco cartas al jugador y, desde luego, se entrega otras cinco a él mismo. Al llegar a este punto hay que abrir un breve paréntesis, para explicar concretamente el funcionamiento del programa.

#### 4.5.9 Programa POKER versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- POKER (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- CPC 464/472/664/6128 -----
50 /-AD. XAVIER ALAMAN (c) TU MICRO/EDITORIAL INGELEK 1.985-
60 /-----
70 /
80 ZONE 7
90 DIM k$(32),b(6,10),f(2),d(2,5),a(8,3),p(4),r(14),t(5),e(5
),z(5),c(55),m(2,5):FOR i=1 TO 52:c(i)=0:NEXT
100 DEF FN m(a)=((a/4)-INT(a/4))*0.4
110 DEF FN d(b)=INT(0.3+(10*(b-INT(b))))
120 DATA 1,4,0,1,7,7,1,4,1,2,0,4,5,6,7,2,1,5,0,7,1,1,1,6
130 FOR i=1 TO 3
140 FOR j=1 TO 8
150 READ a(j,i)
160 NEXT j,i
170 m=0
180 DATA 6,0,0,0,0,60,32,2,0,0,22,35,25,0,0,11,6,0,0,0,10,27
,6,0,46,0,6,1,3,0,0,47,25,0,21,0,5,0,2,0,10,45,25,0,20,0,0,0
,0,0,10,45,25,0,20,0,0,0,0,0
190 FOR i=1 TO 6:FOR j=1 TO 10:READ dato:b(i,j)=dato/100:NEX
T j,i
200 RANDOMIZE TIME:mano=INT(0.5+RND):minimo=7:sb=1:cap1=100:
cap2=100:apmax=11:n=(14-minimo)*4
210 MODE 1:LOCATE 5,9:PRINT "?QUIERES JUGAR CON UNA BARAJA D
E VERDAD?"
220 GOSUB 1740:IF UPPER$(r$)='S' THEN sb=0
230 ap=0:FOR i=1 TO 55:c(i)=0:NEXT
240 CLS:LOCATE 12,10:PRINT "ERES MANO":IF mano=1 THEN LOCATE
12,10:PRINT "SOY"
250 CLEAR INPUT:LOCATE 1,25:PRINT "PULSA CUALQUIER TECLA":WH
ILE INKEY$="":WEND
260 PRINT CHR$(7):GOSUB 1470
270 FOR y=1 TO sb+1:jug=y:FOR i=1 TO 5:GOSUB 1900:m(jug,i)=c
arta:NEXT:GOSUB 1800:NEXT
280 IF sb<>0 THEN GOSUB 1510
290 jug=1:GOSUB 510
300 IF mano<>0 THEN GOSUB 870:GOSUB 1600
310 IF mano=0 THEN GOSUB 1600:GOSUB 870
320 FOR y=1 TO sb+1:jug=y
330 FOR i=1 TO 5
340 IF d(y,i)=1 THEN GOSUB 1900:m(y,i)=carta
350 NEXT:GOSUB 1800:NEXT
360 IF sb<>0 THEN jug=2:GOSUB 1510
370 jug=1:GOSUB 510:GOSUB 1410:GOSUB 1060
380 jug=1:GOSUB 1510
390 GOSUB 1450:IF mano=0 THEN mano=1 ELSE mano=0
400 IF cap1>0 AND cap2>0 THEN CLEAR INPUT:LOCATE 1,25:PRINT
"PULSA CUALQUIER TECLA":WHILE INKEY$="":WEND:GOTO 230
410 IF cap1<=0 THEN CLS:LOCATE 8,13:PRINT "!ENHORABUENA! HAS
GANADO":LOCATE 1,24:END
420 CLS:LOCATE 10,13:PRINT "LO SIENTO. HE GANADO":LOCATE 1,2
4:END
430 REM - SUBR. PRESENTA 1 CARTA -
440 cc%=MID$("CDTP",c,1)
450 IF a=11 THEN ac$='J':GOTO 500
460 IF a=12 THEN ac$='Q':GOTO 500
470 IF a=13 THEN ac$='K':GOTO 500
480 IF a=1 THEN ac$='A':GOTO 500
490 ac%=MID$(STR$(a),2)
500 PRINT ac$'.cc$,

```

```

510 REM - EVALUADOR -
520 FOR i=1 TO 5:e(i)=0:t(i)=1:NEXT
530 FOR i=minimo TO 14:r(i)=0:NEXT
540 FOR i=1 TO 4:p(i)=0:NEXT
550 f(jug)=0:esc=0:col=0:estado=8:final=0:figuras=0
560 FOR i=1 TO 5
570 r(INT(m(jug,i)))=r(INT(m(jug,i)))+1-
580 p(FN d(m(jug,i)))=p(FN d(m(jug,i)))+1
590 IF i=5 THEN 650
600 entr=1+(INT(m(jug,i))-INT(m(jug,i+1)))
610 IF entr=3 AND i=4 AND (estado=5 OR estado=6) THEN 640
620 IF entr>=4 THEN entr=1:IF i<>1 AND i<>4 THEN estado=1:GOTO 640
630 estado=a(estado,entr)
640 IF entr=1 THEN e(i)=1
650 NEXT
660 IF estado=5 OR estado=6 THEN e(5)=1:e(4)=0
670 esc=INT(estado/2)
680 FOR i=1 TO 4
690 IF p(i)=3 THEN col=p(i):FOR j=1 TO 5:z(j)=i<>FN d(m(jug,j)):z(j)=-1*z(j):NEXT
700 NEXT
710 IF estado=2 THEN f(jug)=5+m(jug,1)/100-4*(col=5):RETURN
720 IF col=5 THEN f(jug)=7+(FN d(m(jug,1))/10):RETURN
730 FOR i=14 TO minimo STEP -1
740 IF r(i)=3 THEN GOSUB 1790:f(jug)=4+i/100:FOR j=14 TO minimo STEP -1:f(jug)=f(jug)-(2*(r(j)=2)):NEXT:final=1
750 IF r(i)=4 THEN GOSUB 1790:f(jug)=8+i/100:final=1
760 NEXT:IF final<>0 THEN RETURN
770 FOR i=14 TO minimo STEP -1
780 IF r(i)<>2 THEN 840
790 f(jug)=1+i/100:GOSUB 1790
800 FOR k=i-1 TO minimo STEP -1
810 IF r(k)=2 THEN f(jug)=f(jug)+1:FOR j=1 TO 5:t(j)=t(j) AND (k<>INT(m(jug,j))):NEXT
820 NEXT
830 i=minimo
840 NEXT
850 FOR i=1 TO 5:figuras=figuras AND ((1<=INT(m(jug,i))) AND (14<>INT(m(jug,i)))):NEXT:IF figuras<>0 THEN f(jug)=3
860 RETURN
870 REM - DESCARTES -
880 fz=INT(f(1)):farol=0
890 IF fz=5 OR fz=6 OR fz=7 OR fz=9 THEN 1020
900 IF fz=8 OR fz=2 OR fz=4 THEN 1030
910 IF col=4 THEN 1050
920 IF fz=1 AND ((10*FN d(f(1)))>12) THEN 1030
930 IF esc=3 THEN 1040
940 IF fz=1 AND ((10*FN d(f(1)))>(7+(minimo/2))) THEN 1030
950 IF esc=3 THEN 1040
960 IF fz=1 AND col=3 THEN IF RND>0.5 THEN 1050
970 IF fz=1 THEN 1030
980 IF col=3 THEN 1050
990 FOR i=1 TO 5:d(1,i)=1:NEXT
1000 IF RND>0.6 THEN FOR i=1 TO (3+INT(RND*3)):d(1,i)=0:NEXT:farol=1
1010 des2=0:FOR i=1 TO 5:des2=des2+d(1,i):NEXT:LOCATE 1,16:PRINT "YO ME DESCARTO DE"des2:PRINT CHR$(7):RETURN
1020 FOR i=1 TO 5:d(1,i)=0:NEXT:GOTO 1010
1030 FOR i=1 TO 5:d(1,i)=t(i):NEXT:GOTO 1010
1040 FOR i=1 TO 5:d(1,i)=e(i):NEXT:GOTO 1010
1050 FOR i=1 TO 5:d(1,i)=z(i):NEXT:GOTO 1010
1060 REM - SUBASTA -
1070 ap=1:LOCATE 26,8:PRINT 2*ap' ':LOCATE 26,2:PRINT cap1-1' ':LOCATE 26,5:PRINT cap2-1' ':IF prob<0.9 THEN tope=(1.25*prob-0.625)*cap1

```

```

1080 IF prob>=0.9 THEN tope=(5*prob-4)*cap1
1090 IF cap1<=10 AND prob>=0.8 THEN tope=cap1
1100 IF faro1 THEN tope=cap1*0.5*(1+RND)
1110 IF tope<0 THEN tope=0
1120 tope=INT(tope)
1130 IF tope>cap2 THEN tope=cap2
1140 IF mano<>0 THEN 1190
1150 GOSUB 1720:LOCATE 1,13:INPUT "?CUANTO APUESTAS";iap2:IF
iap2>cap1 OR iap2>cap2 OR iap2<0 OR iap2>apmax THEN LOCATE
1,13:PRINT "NO ES POSIBLE":PRINT CHR$(7);:FOR i=0 TO 4
99:NEXT:GOTO 1150
1160 LOCATE 26,5:PRINT cap2-iap2":jug=2:GOSUB 1720
1170 IF iap2<0 THEN 1210
1180 IF tope<=1 THEN LOCATE 1,13:PRINT "YO PASO TAMBIEN
":GOTO 1370
1190 IF tope>1 THEN IF mano=0 OR RND>0.3 THEN 1250
1200 iap1=0:GOTO 1280
1210 IF ap+iap2<tope THEN LOCATE 1,13:PRINT "QUIERO TUS"iap2
:ap=ap+iap2:LOCATE 26,2:PRINT cap1-ap":LOCATE 26,8:PRINT 2
*ap":GOTO 1250
1220 x=ap+iap2-tope
1230 IF x<=0.3*ap OR (x<=0.2*cap1 AND prob>0.3) THEN LOCATE
1,13:PRINT "QUIERO TUS"iap2:ap=ap+iap2:LOCATE 26,2:PRINT cap
1-ap":LOCATE 26,8:PRINT 2*ap:GOTO 1370
1240 LOCATE 1,13:PRINT "NO LAS VEO":cap1=cap1-ap:cap2=cap2+a
p:RETURN
1250 x=tope-ap
1260 IF x>apmax THEN x=apmax
1270 iap1=INT(1+RND*x)
1280 LOCATE 1,16:PRINT "APUESTO"iap1":LOCATE 26,2
:PRINT cap1-ap-iap1":LOCATE 26,8:PRINT iap1+2*ap"
1290 LOCATE 1,19:PRINT "?LAS VES (S/N)?":GOSUB 17
50
1300 IF UPPER$(r$)="N" THEN cap1=cap1+ap:cap2=cap2-ap:RETURN
1310 jug=1:GOSUB 1720:LOCATE 1,16:PRINT "APUESTA VISTA":PRIN
T CHR$(7);:ap=ap+iap1:LOCATE 26,8:PRINT 2*ap":LOCATE 26,5:
PRINT cap2-ap"
1320 LOCATE 1,19:PRINT "?APUESTAS MAS (S/N)?":GOSUB 1750:IF
UPPER$(r$)="N" THEN 1370
1330 IF cap2-ap=0 OR cap1-ap=0 THEN 1370
1340 GOSUB 1720:LOCATE 1,19:INPUT "?CUANTO MAS";iap2:IF ap+i
ap2>cap1 OR iap2<=0 OR ap+iap2>cap2 OR iap2>apmax THEN LOCATE
E 1,19:PRINT "NO ES POSIBLE":PRINT CHR$(7);:FOR i
=0 TO 499:NEXT:GOTO 1340
1350 LOCATE 26,5:PRINT cap2-ap-iap2":LOCATE 26,8:PRINT iap
2+2*ap"
1360 jug=2:GOSUB 1720:GOTO 1210
1370 IF sb<>0 THEN jug=2:GOSUB 510
1380 IF sb=0 THEN INPUT "?QUE JUGADA TIENES?",f(2)
1390 IF f(1)>f(2) THEN cap1=cap1+ap:cap2=cap2-ap:RETURN
1400 cap1=cap1-ap:cap2=cap2+ap:RETURN
1410 REM - PROBABILIDADES -
1420 prob=0
1430 FOR i=1 TO INT(f(1)):prob=prob+b(desc,i):NEXT
1440 prob=prob+((f(1)-INT(f(1)))*100-minimo-1)*b(desc,1+INT(
f(1)))/(14-minimo):RETURN
1450 REM - VARIACION DE PROBS. -
1460 FOR i=1 TO 10:b(desc,i)=(b(desc,i)*10-(i=(1+INT(f(2))))
)/11:NEXT:RETURN
1470 REM - PANTALLA -
1480 CLS
1490 LOCATE 17,2:PRINT "TENGO : "cap1":LOCATE 17,5:PRINT
"TIENES: "cap2":LOCATE 17,8:PRINT "APOSTADO:"ap"
1500 RETURN
1510 REM - DIBUJA CARTAS -
1520 LOCATE 1,10+(12 AND jug=1)

```

```

1530 d=1+(11 AND jug=1)
1540 FOR w=1 TO 5
1550 c=FN d(m(jug,6-w))
1560 a=INT(m(jug,6-w)):IF a=14 THEN a=1
1570 GOSUB 430
1580 NEXT
1590 RETURN
1600 REM - DESCARTE=EL -
1610 IF sb=0 THEN LOCATE 1,10:INPUT "?DE CUANTAS TE DESCARTA
S":desc:desc=desc+1:LOCATE 1,10:PRINT SPACE$(30):RETURN
1620 desc=1
1630 FOR i=1 TO 5
1640 LOCATE 2+7*(i-1),11
1650 PRINT ".^"
1660 LOCATE 1,13:PRINT "?TE DESCARTAS DE ESTA (S/N)?":GOSUB
1750
1670 d(2,6-i)=-1*(UPPER$(r$)="S"):desc=desc+d(2,6-i):IF UPPE
R$(r$)="N" THEN LOCATE 2+7*(i-1),11:PRINT ". "
1680 NEXT
1690 LOCATE 1,13:PRINT "?ESTA BIEN ASI (S/N)?":GOSUB
1750:IF UPPER$(r$)="N" THEN 1620
1700 GOSUB 1470:RETURN
1710 REM - SUBR. BORRAR -
1720 FOR i=13 TO 19 STEP 3:LOCATE 1,i:PRINT SPACE$(30):NEXT
1730 RETURN
1740 REM - SUBR. TECLA -
1750 r$=INKEY$:IF UPPER$(r$)<"S" AND UPPER$(r$)<"N" AND UP
PER$(r$)<"G" THEN GOTO 1750
1760 IF UPPER$(r$)="G" THEN GOSUB 1970:GOTO 1740
1770 PRINT CHR$(7):RETURN
1780 REM - SUBR. AUX. -
1790 FOR j=1 TO 5:t(j)=(i<>INT(m(1,j))):t(j)=-1*t(j):NEXT:RE
TURN
1800 REM - ORDENADOR -
1810 FOR i=1 TO 5
1820 FOR j=i+1 TO 5
1830 IF m(jug,i)>=m(jug,j) THEN 1870
1840 v1=m(jug,i)
1850 m(jug,i)=m(jug,j)
1860 m(jug,j)=v1
1870 NEXT
1880 NEXT
1890 RETURN
1900 REM - REPARTIDOR DE CARTAS -
1910 IF sb=0 THEN LOCATE 1,10:PRINT CHR$(7):INPUT "DAME UNA
CARTA: ",carta:LOCATE 1,10:PRINT SPACE$(30):RETURN
1920 num=55-INT(RND*n)
1930 IF c(num)>0 THEN 1920
1940 c(num)=1
1950 carta=1.1+INT(num/4)+FN m(num)
1960 RETURN
1970 REM - GRABACION -
1980 LOCATE 1,16:INPUT "?SALVAR (S) O CARGAR (C)?",r$
1990 IF UPPER$(r$)="S" THEN OPENOUT "PROBS":FOR k=1 TO 6:FOR
l=1 TO 10:PRINT#9,b(k,l):NEXT l,k:CLOSEOUT:k=i:GOSUB 1710:i
=k:RETURN
2000 IF UPPER$(r$)="C" THEN OPENIN "PROBS":FOR k=1 TO 6:FOR
l=1 TO 10:INPUT#9,b(k,l):NEXT l,k:CLOSEIN:k=i:GOSUB 1710:i=k
:RETURN
2010 GOTO 1970

```

#### 4.5.10. Programa POKER versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- POKER (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 in$=CHR$(27)+'f':act$=CHR$(27)+'e':PRINT in$:kls$=CHR$(27
)+'E'+CHR$(27)+'H':DEF FN locate$(x,y)=CHR$(27)+'Y'+CHR$(32+
x)+CHR$(32+y):pq$=CHR$(27)+'x':gr$=CHR$(27)+'y'
90 PRINT kls$;FN locate$(15,26)'PULSE UNA TECLA PARA COMENZA
R':WHILE INKEY$='':time=1+(time MOD 1000):WEND
100 PRINT pq$;kls$:ZONE 7
110 DIM k$(32),b(6,10),f(2),d(2,5),a(8,3),p(4),r(14),t(5),e(
5),z(5),c(55),m(2,5):FOR i=1 TO 52:c(i)=0:NEXT
120 DEF FN m(a)=(a/4)-INT(a/4)*0.4
130 DEF FN d(b)=INT(0.3+(10*(b-INT(b))))
140 DATA 1,4,0,1,7,7,1,4,1,2,0,4,5,6,7,2,1,5,0,7,1,1,1,6
150 FOR i=1 TO 3
160 FOR j=1 TO 8
170 READ a(j,i)
180 NEXT j,i
190 m=0
200 DATA 6,0,0,0,0,60,32,2,0,0,22,35,25,0,0,11,6,0,0,0,10,27
,6,0,46,0,6,1,3,0,0,47,25,0,21,0,5,0,2,0,10,45,25,0,20,0,0,0
,0,0,10,45,25,0,20,0,0,0,0,0
210 FOR i=1 TO 6:FOR j=1 TO 10:READ dato:b(i,j)=dato/100:NEX
T j,i
220 RANDOMIZE TIME:mano=INT(0.5+RND):minimo=7:sb=1:cap1=100:
cap2=100:apmax=11:n=(14-minimo)*4
230 PRINT kls$;FN locate$(12,19)'QUIERES JUGAR CON UNA BARA
JA DE VERDAD?'
240 GOSUB 1760:IF UPPER$(r$)='S' THEN sb=0
250 ap=0:FOR i=1 TO 55:c(i)=0:NEXT
260 PRINT kls$;FN locate$(8,26):'E R E S M A N O':IF mano=
1 THEN PRINT FN locate$(8,26)' S O Y '
270 PRINT FN locate$(14,24)'PULSA CUALQUIER TECLA':WHILE INK
EY$='':WEND
280 PRINT CHR$(7);:GOSUB 1490
290 FOR y=1 TO sb+1:jug=y:FOR i=1 TO 5:GOSUB 1920:m(jug,i)=c
arta:NEXT:GOSUB 1820:NEXT
300 IF sb<>0 THEN GOSUB 1530
310 jug=1:GOSUB 530
320 IF mano<>0 THEN GOSUB 890:GOSUB 1620
330 IF mano=0 THEN GOSUB 1620:GOSUB 890
340 FOR y=1 TO sb+1:jug=y
350 FOR i=1 TO 5
360 IF d(y,i)=1 THEN GOSUB 1920:m(y,i)=carta
370 NEXT:GOSUB 1820:NEXT
380 IF sb<>0 THEN jug=2:GOSUB 1530
390 jug=1:GOSUB 530:GOSUB 1430:GOSUB 1080
400 jug=1:GOSUB 1530
410 GOSUB 1470:IF mano=0 THEN mano=1 ELSE mano=0
420 IF cap1>0 AND cap2>0 THEN PRINT FN locate$(24,1)'PULSA C
UALQUIER TECLA'::WHILE INKEY$='':WEND:GOTO 250
430 IF cap1<=0 THEN PRINT kls$:PRINT FN locate$(13,14):PRIN
T '! E N H O R A B U E N A ! H A S G A N A D O':PRINT FN l
ocate$(21,1);act$:END
440 PRINT kls$;FN locate$(13,23)'L O S I E N T O . H E G A
N A D O':PRINT FN locate$(21,1);act$:END
450 REM - SUBR. PRESENTA 1 CARTA -

```

```

460 cc$=MID$("CDTP",c,1)
470 IF a=11 THEN ac$="J":GOTO 520
480 IF a=12 THEN ac$="Q":GOTO 520
490 IF a=13 THEN ac$="K":GOTO 520
500 IF a=1 THEN ac$="A":GOTO 520
510 ac$=MID$(STR$(a),2)
520 PRINT ac$."cc$,
530 REM - EVALUADOR -
540 FOR i=1 TO 5:e(i)=0:t(i)=1:NEXT
550 FOR i=minimo TO 14:r(i)=0:NEXT
560 FOR i=1 TO 4:p(i)=0:NEXT
570 f(jug)=0:esc=0:col=0:estado=8:final=0:figuras=0
580 FOR i=1 TO 5
590 r(INT(m(jug,i)))=r(INT(m(jug,i)))+1
600 p(FN d(m(jug,i)))=p(FN d(m(jug,i)))+1
610 IF i=5 THEN 670
620 entr=1+(INT(m(jug,i)))-INT(m(jug,i+1))
630 IF entr=3 AND i=4 AND (estado=5 OR estado=6) THEN 660
640 IF entr=4 THEN entr=1:IF i<>1 AND i<>4 THEN estado=1:GO
TO 660
650 estado=a(estado,entr)
660 IF entr=1 THEN e(i)=1
670 NEXT
680 IF estado=5 OR estado=6 THEN e(5)=1:e(4)=0
690 esc=INT(estado/2)
700 FOR i=1 TO 4
710 IF p(i)>3 THEN col=p(i):FOR j=1 TO 5:z(j)=i<>FN d(m(jug
,j)):z(j)=-1*z(j):NEXT
720 NEXT
730 IF estado=2 THEN f(jug)=5+m(jug,1)/100-4*(col=5):RETURN
740 IF col=5 THEN f(jug)=7+(FN d(m(jug,1))/10):RETURN
750 FOR i=14 TO minimo STEP -1
760 IF r(i)=3 THEN GOSUB 1810:f(jug)=4+i/100:FOR j=14 TO min
imo STEP -1:f(jug)=f(jug)-(2*(r(j)=2)):NEXT:final=1
770 IF r(i)=4 THEN GOSUB 1810:f(jug)=8+i/100:final=1
780 NEXT:IF final<>0 THEN RETURN
790 FOR i=14 TO minimo STEP -1
800 IF r(i)<>2 THEN 860
810 f(jug)=1+i/100:GOSUB 1810
820 FOR k=i-1 TO minimo STEP -1
830 IF r(k)=2 THEN f(jug)=f(jug)+1:FOR j=1 TO 5:t(j)=t(j) AN
D (k<>INT(m(jug,j))):NEXT
840 NEXT
850 i=minimo
860 NEXT
870 FOR i=1 TO 5:figuras=figuras AND (11<=INT(m(jug,i))) AND
(14<>INT(m(jug,i))):NEXT:IF figuras<>0 THEN f(jug)=3
880 RETURN
890 REM - DESCARTES -
900 fz=INT(f(1)):farol=0
910 IF fz=5 OR fz=6 OR fz=7 OR fz=9 THEN 1040
920 IF fz=8 OR fz=2 OR fz=4 THEN 1050
930 IF col=4 THEN 1070
940 IF fz=1 AND ((10*FN d(f(1)))>12) THEN 1050
950 IF esc=3 THEN 1060
960 IF fz=1 AND ((10*FN d(f(1)))>(7+(minimo/2))) THEN 1050
970 IF esc=3 THEN 1060
980 IF fz=1 AND col=3 THEN IF RND>0.5 THEN 1070
990 IF fz=1 THEN 1050
1000 IF col=3 THEN 1070
1010 FOR i=1 TO 5:d(1,i)=1:NEXT
1020 IF RND>0.6 THEN FOR i=1 TO (3+INT(RND*3)):d(1,i)=0:NEXT
:farol=1
1030 des2=0:FOR i=1 TO 5:des2=des2+d(1,i):NEXT:PRINT FN loca

```

```

te$(16,1)'YO ME DESCARTO DE' des2:PRINT CHR$(7);:RETURN
1040 FOR i=1 TO 5:d(1,i)=0:NEXT:GOTO 1030
1050 FOR i=1 TO 5:d(1,i)=t(i):NEXT:GOTO 1030
1060 FOR i=1 TO 5:d(1,i)=e(i):NEXT:GOTO 1030
1070 FOR i=1 TO 5:d(1,i)=z(i):NEXT:GOTO 1030
1080 REM - SUBASTA -
1090 ap=1:PRINT FN locate$(8,31);2*ap' ':PRINT FN locate$(2,
28);cap1-1' ':PRINT FN locate$(5,28);cap2-1' ':IF prob<0.9 T
HEN tope=(1.25*prob-0.625)*cap1
1100 IF prob>=0.9 THEN tope=(5*prob-4)*cap1
1110 IF cap1<=10 AND prob>=0.8 THEN tope=cap1
1120 IF farol THEN tope=cap1*0.5*(1+RND)
1130 IF tope<0 THEN tope=0
1140 tope=INT(tope)
1150 IF tope>cap2 THEN tope=cap2
1160 IF mano<>0 THEN 1210
1170 GOSUB 1740:PRINT act$;FN locate$(13,1);:INPUT '?CUANTO
APUESTAS';iap2:PRINT in$;:IF iap2>cap1 OR iap2>cap2 OR iap2<
0 OR iap2>amax THEN PRINT FN locate$(13,1)'NO ES POSIBLE
':PRINT CHR$(7);:FOR i=1 TO 499:NEXT:GOTO 1170
1180 PRINT FN locate$(5,28);:PRINT cap2-iap2' ':jug=2:GOSUB
1740
1190 IF iap2<>0 THEN 1230
1200 IF tope<=1 THEN PRINT FN locate$(13,1):PRINT 'YO PASO T
AMBIEN ':GOTO 1390
1210 IF tope>1 THEN IF mano=0 OR RND>0.3 THEN 1270
1220 iap1=0:GOTO 1300
1230 IF ap+iap2<tope THEN PRINT FN locate$(13,1)'QUIERO TUS'
iap2:ap=ap+iap2:PRINT FN locate$(2,28);cap1-ap' ':PRINT FN l
ocate$(8,31);2*ap' ':GOTO 1270
1240 x=ap+iap2-tope
1250 IF x<=0.3*ap OR (x<=0.2*cap1 AND prob>0.3) THEN PRINT F
N locate$(13,1)'QUIERO TUS'iap2:ap=ap+iap2:PRINT FN locate$(
2,28);cap1-ap' ':PRINT FN locate$(8,31);2*ap:GOTO 1390
1260 PRINT FN locate$(13,1)'NO LAS VEO':cap1=cap1-ap:cap2=ca
p2+ap:RETURN
1270 x=tope-ap
1280 IF x>amax THEN x=amax
1290 iap1=INT(1+RND*x)
1300 PRINT FN locate$(16,1)'APUESTO' iap1' ':PRINT F
N locate$(2,28);cap1-ap-iap1' ':PRINT FN locate$(8,31);iap1+
2*ap'
1310 PRINT FN locate$(19,1)'?LAS VES (S/N)? ':GOSU
B 1770
1320 IF UPPER$(r$)='N' THEN cap1=cap1+ap:cap2=cap2-ap:RETURN
1330 jug=1:GOSUB 1740:PRINT FN locate$(16,1)'APUESTA VISTA':
PRINT CHR$(7);:ap=ap+iap1:PRINT FN locate$(8,31);2*ap' ':PRI
NT FN locate$(5,28);cap2-ap'
1340 PRINT FN locate$(19,1)'?APUESTAS MAS (S/N)?':GOSUB 1770
:IF UPPER$(r$)='N' THEN 1390
1350 IF cap2-ap=0 OR cap1-ap=0 THEN 1390
1360 GOSUB 1740:PRINT act$;FN locate$(19,1);:INPUT '?CUANTO
MAS';iap2:PRINT in$;:IF ap+iap2>cap1 OR iap2<=0 OR ap+iap2>c
ap2 OR iap2>amax THEN PRINT FN locate$(19,1)'NO ES POSIBLE
':PRINT CHR$(7);:FOR i=1 TO 499:NEXT:GOTO 1350
1370 PRINT FN locate$(5,28);cap2-ap-iap2' ':PRINT FN locate$
(8,31);iap2+2*ap'
1380 jug=2:GOSUB 1740:GOTO 1230
1390 IF sb<>0 THEN jug=2:GOSUB 530
1400 IF sb=0 THEN PRINT act$: INPUT '?QUE JUGADA TIENES';f(2
):PRINT in$
1410 IF f(1)>f(2) THEN cap1=cap1+ap:cap2=cap2-ap:RETURN
1420 cap1=cap1-ap:cap2=cap2+ap:RETURN
1430 REM - PROBABILIDADES -

```

```

1440 prob=0
1450 FOR i=1 TO INT(f(1)):prob=prob+b(desc,i):NEXT
1460 prob=prob+((f(1)-INT(f(1)))*100-minimo-1)*b(desc,1+INT(
f(1)))/(14-minimo):RETURN
1470 REM - VARIACION DE PROBS. -
1480 FOR i=1 TO 10:b(desc,i)=(b(desc,i)*10-(i+(1+INT(f(2))))
)/11:NEXT:RETURN
1490 REM - PANTALLA -
1500 PRINT kls$
1510 PRINT FN locate$(2,15)'T E N G O: 'cap1':PRINT FN lo
cate$(5,15)'T I E N E S:'cap2':PRINT FN locate$(8,15)'A P
O S T A D O:'ap'
1520 RETURN
1530 REM - DIBUJA CARTAS -
1540 PRINT FN locate$(10+(10 AND jug=1),1)
1550 d=1+(11 AND jug=1)
1560 FOR w=1 TO 5
1570 c=FN d(m(jug,6-w))
1580 a=INT(m(jug,6-w)):IF a=14 THEN a=1
1590 GOSUB 450
1600 NEXT
1610 RETURN
1620 REM - DESCARTE=EL -
1630 IF sb=0 THEN PRINT FN locate$(11,1);act$;:INPUT "?DE CU
ANTAS TE DESCARTAS";desc:PRINT in$;:desc=desc+1:PRINT FN loc
ate$(11,1);:PRINT SPACE$(30):RETURN
1640 desc=1
1650 FOR i=1 TO 5
1660 PRINT FN locate$(12,1+7*(i-1));
1670 PRINT '^'
1680 PRINT FN locate$(14,1)"?TE DESCARTAS DE ESTA (S/N)?:G
OSUB 1770
1690 d(2,6-i)=-1*(UPPER$(r$)="S"):desc=desc+d(2,6-i):IF UPPE
R$(r$)="N" THEN PRINT FN locate$(12,1+7*(i-1));:PRINT ' '
1700 NEXT
1710 PRINT FN locate$(14,1)"?ESTA BIEN ASI (S/N)? ' :G
OSUB 1770:IF UPPER$(r$)="N" THEN 1640
1720 GOSUB 1490:RETURN
1730 REM - SUBR. BORRAR -
1740 FOR i=13 TO 19 STEP 3:PRINT FN locate$(i,1)SPACE$(30):N
EXT
1750 RETURN
1760 REM - SUBR. TECLA -
1770 r$=INKEY$:IF UPPER$(r$)<>"S" AND UPPER$(r$)<>"N" AND UP
PER$(r$)<>"G" THEN GOTO 1770
1780 IF UPPER$(r$)="G" THEN GOSUB 1990:GOTO 1760
1790 PRINT CHR$(7);:RETURN
1800 REM - SUBR. AUX. -
1810 FOR j=1 TO 5:t(j)=(i<>INT(m(1,j))):t(j)=-1*t(j):NEXT:RE
TURN
1820 REM - ORDENADOR -
1830 FOR i=1 TO 5
1840 FOR j=i+1 TO 5
1850 IF m(jug,i)>m(jug,j) THEN 1890
1860 v1=m(jug,i)
1870 m(jug,i)=m(jug,j)
1880 m(jug,j)=v1
1890 NEXT
1900 NEXT
1910 RETURN
1920 REM - REPARTIDOR DE CARTAS -
1930 IF sb=0 THEN PRINT FN locate$(10,1);:PRINT CHR$(7);act$
;:INPUT "DAME UNA CARTA: ",carta:PRINT in$;:PRINT FN locate$
(10,1)SPACE$(30):RETURN

```

```

1940 num=55-INT(RND*n)
1950 IF c(num)<>0 THEN 1940
1960 c(num)=1
1970 carta=1.1+INT(num/4)+FN m(num)
1980 RETURN
1990 REM - GRABACION -
2000 PRINT:INPUT "?SALVAR (S) O CARGAR (C)?",r$
2010 IF UPPER$(r$)="S" THEN OPEN"O",1,"PROBS":FOR k=1 TO 6:F
OR l=1 TO 10:PRINT #1,b(k,l):NEXT l,k:CLOSE:SWAP i,k:GOSUB 1
730:SWAP i,k:RETURN
2020 IF UPPER$(r$)="C" THEN OPEN"I",1,"PROBS":FOR k=1 TO 6:F
OR l=1 TO 10:INPUT #1,b(k,l):NEXT l,k:CLOSE:SWAP i,k:GOSUB 1
730:SWAP i,k:RETURN
2030 GOTO 1990

```

## CALIFICACIÓN DE LAS JUGADAS

La máquina sigue un estricto juego limpio, y para decidir su descarte no incurre en la trampa fácil de mirar la baraja. Por si algún jugador desconfía de su honradez, el programa realiza una pregunta antes de decidir quién es «mano»: ¿vamos a jugar con una baraja de verdad? Si el jugador responde que no, se seguirá el camino habitual y el ordenador preparará su «baraja electrónica». De no ser así —vaya recordando al gran Tirso en «El condenado por desconfiado»—, que se atenga a las consecuencias: le espera, nada más y nada menos, que ir comunicando al ordenador las cartas que van saliendo de la baraja real, lo cual puede resultar algo tedioso.

En todo caso, y por si alguien no tiene en cuenta este consejo, indicaremos la notación con la que se darán a conocer las cartas al ordenador. En primer lugar, debe figurar el número de la carta (11 para J, 12 para Q, 13 para K y 14 para A) y separado de éste mediante un punto decimal, el palo correspondiente (1 para corazones, 2 para diamantes, 3 para tréboles y 4 para picas). Por ejemplo, el as de corazones se identificaría como 14.1; el nueve de trébol como 9.3, etc.

El juego se desarrollará conversacionalmente. A su término, el ordenador preguntará por la jugada obtenida, jugada que se introducirá en la máquina de acuerdo con la siguiente tabla de codificación:

TIPO	CALIDAD
1 Pareja	8 Ocho
2 Dobles parejas	9 Nueve
3 Figuras	10 Diez
4 Trío	11 Jack
5 Escalera	12 Dama
6 Full	13 Rey
7 Color	14 As
8 Póker	
9 Escalera de color	Jugada = Tipo.Calidad

Tal como se observa, además del «tipo» de jugada (pareja, trío, escalera...), hay que indicar la «calidad» de la misma mediante números, separados del tipo de jugada con un punto decimal. Por ejemplo, un full de ases se expresará como 6.14, una pareja de reyes como 1.13, etc.

Una vez que ya tenemos las cartas en la mano, o en la pantalla, ha llegado el momento de explicar el objetivo del juego. Este se concreta, sencillamente, en formar con las cinco cartas que poseemos la mejor jugada posible. Las combinaciones son juzgadas como mejores o peores de acuerdo con la dificultad que supone obtenerlas. En caso de que tanto el Amstrad como el jugador humano construyan la misma jugada, existe un elemento valorativo más: la calidad de la jugada, determinada por la carta más alta que tengamos en mayor número de palos. La tabla de jugadas que hemos visto anteriormente, las detalla en orden ascendente, del mismo modo que sus calidades.

Por supuesto, al haber omitido la presencia de los comodines, no cabe hablar de «repoker»: cuatro cartas iguales más un comodín. Y ya únicamente quedan por añadir algunas puntualizaciones. Por ejemplo, sólo se examinará la calidad de las jugadas cuando se produzca un empate en el tipo de las mismas. De existir esta igualdad, tanto en jugada como en calidad, ganará el jugador que sea «mano».

Al referirnos a las escaleras se indicó que debían ser cartas seguidas, en orden ascendente. Ello significa que los palos de la baraja no conforman una cadena cerrada, ya que el as y el ocho no son utilizables como eslabones de unión; realmente, forman una escalera que se interrumpe al llegar al as. Por lo tanto, la configuración Q, K, A, 8, 9 no constituye una escalera válida. Sí la conformarán las secuencias ascendentes y ordenadas que empiecen por ocho, nueve o diez.

Para terminar con el tema, cabe añadir que algunos jugadores admiten el hecho de que el as pueda figurar tanto al comienzo como al

final de la escalera. En tal caso, serán escaleras válidas tanto A, 8, 9, 10, J, como 10, J, Q, K, A. Nuestro programa no admite esta licencia, y siempre que aparezca un as en la escalera, debe estar emplazado al final de la misma. Para el juego del póker no tiene importancia el color de las cartas, sino su palo. Por lo tanto, la jugada «color» no coincidirá con la presencia de cinco cartas del mismo color, sino por cinco cartas del mismo palo. De forma análoga, la «escalera de color» ha de estar compuesta por cartas del mismo palo.

## DESARROLLO DEL JUEGO

El juego transcurrirá a base de sucesivas partidas, cada una de las cuales tiene siempre un ganador. En ellas se producirá, por lo tanto, un trasvase de capital, según lo apostado en cada jugada. El juego sólo se dará por terminado, cuando uno de los dos jugadores (la máquina o el usuario), se hunda en una desesperante bancarrota. Las atribuciones de «mano» irán pasando alternativamente de uno a otro jugador; éstas se resumen en los siguientes puntos:

- El «mano» es el primero en hablar. Tanto a la hora del descarte como de la subasta (apuestas).
- En caso de empate en tipo de jugada y calidad, gana el que va de «mano».

Una vez anunciado quién es el «mano», se procederá al reparto de cartas. Acto seguido se efectúan los descartes de uno y otro jugador, comenzando el «mano». Los jugadores se pueden descartar del número de cartas que deseen (de cero a cinco); si bien, sólo es posible un único descarte por baza. Cuando llega el turno de descarte del jugador humano, el ordenador emite el mensaje «Te descartas de esta (s/n)?», señalando a la carta que se refiere mediante una flecha; dicha operación comenzará por el naipe de la izquierda, y proseguirá con los restantes hasta terminar con las cartas en juego. Respondiendo con «s» o con «n», el jugador irá indicando las cartas de las que desea desprenderse.

Al terminar el descarte, el programa facilitará la corrección de errores mediante el mensaje: «Está bien así (s/n)?» El ordenador, como ocurre en una partida normal de póker, no ve las cartas que abandona el otro jugador, pero sí el número de ellas. Del mismo modo, él en su descarte, indicará cuántos naipes desecha.

El siguiente paso es sustituir los naipes descartados por otros nuevos. Un dato destacable es que para facilitar la localización de la mejor de nuestras jugadas, el programa siempre muestra las cartas ordenadas.

Inmediatamente a continuación, comienza la subasta; en ella, los jugadores (comenzando por el «mano»), realizan sus apuestas. De entrada, cada uno de los jugadores pone un punto, con el fin de que siempre se apueste algo (ello lo realiza automáticamente el programa). Por cada embite, el jugador puede apostar entre 0 y 11 puntos. Su oponente dispone entonces de dos opciones: aceptar la cantidad apostada, igualándola (ver la apuesta), o no aceptarla. En el último caso, perderá lo apostado sin que se lleguen a ver las cartas.

Si se opta por ver la apuesta, caben aún dos posibilidades más: aumentar la apuesta con un embite adicional o dejar las cantidades igualadas. En el último caso se descubren los naipes y ganará lo apostado el jugador con mejores cartas. De no ser así, se pasa la palabra al otro jugador; de nuevo, éste puede elegir entre no ver la apuesta o verla, e incluso volverla a subir.

## EL PROGRAMA

El programa se ha confeccionado de forma modular. El ciclo principal ocupa apenas unas líneas; su cometido se reduce a encadenar las llamadas a subrutinas. Entre éstas, cabe destacar las subrutinas de descarte y evaluador, en las que reside parte de la inteligencia del programa.

Algunas condiciones iniciales del juego, tal es el caso del número de cartas de la baraja, o el valor de la apuesta máxima, pueden ser alteradas fácilmente en la línea 140, cambiando los valores de las variables MINIMO y APMAX, respectivamente. De realizar algún cambio, hay que tener en cuenta que el programa tardará algún tiempo en acondicionarse a las nuevas normas. El propio ordenador irá acostumbrándose paulatinamente a su estilo de juego.

Por lo tanto, no cometamos el error de pensar que la máquina no va a ser capaz de descubrir nuestros «faroles»: si abusamos de ellos, el programa acabará por desconfiar, y puede darnos un buen disgusto en el momento más inesperado. Por otra parte, no pensemos que el ordenador siempre juega sin «faroles», ya que éstos forman parte del juego del póker, y él lo sabe bien. Se dice que una de las mejores

cualidades del buen jugador de poker es su capacidad para apuntarse «faroles» sin inmutarse; en este aspecto, podemos asegurar que el ordenador es un excelente jugador.

Cuando la máquina se desenchufa, olvida todo lo que ha aprendido durante la partida; no obstante, si deseamos que conserve los conocimientos adquiridos para otro día, podemos lograrlo mediante la subrutina 1890-1930. Respondiendo con la letra G (de grabar) a alguna de las preguntas que el programa realiza habitualmente durante el juego, se accederá automáticamente a dicha subrutina. Esta sirve tanto para la grabación de la matriz de aprendizaje actual, como para la carga de un aprendizaje anterior. Si durante la grabación o la carga, se produjese algún error, deberemos entrar de nuevo en el programa mediante GOTO 1890.

Si deseamos acortar la duración de la partida, podemos alterar los valores del capital inicial de cada jugador; valores que son otorgados por medio de las variables CAP1 y CAP2 en la línea 140.

## LA INTELIGENCIA DEL PROGRAMA

Dentro del cometido del programa, cabe distinguir dos misiones claramente diferenciadas. Por un lado, gestionar de forma correcta el desarrollo de la partida.

Tema aparte ha sido lograr que el programa se convierta en un jugador de poker. Su inteligencia puede sintetizarse en tres puntos fundamentales: deducción de los naipes a descartar, gestión de la su-basta de apuestas y evolución de la forma de juego.

A continuación, analizaremos cada uno de estos puntos, aunque procurando no desvelar el secreto de su manifiesta aptitud para el juego.

## DEDUCCIÓN DE LOS NAIPES A DESCARTAR

Uno de los momentos cruciales en toda partida de poker llega cuando el jugador, en base a las cinco cartas que tiene a la vista y, a veces (cuando no es «mano»), con el dato adicional que aporta el descarte del oponente, decide cuál va a ser su jugada y qué cartas va a desechar para ello.

Este punto está muy relacionado con la evolución de la forma de juego, ya que para el descarte no sólo se tiene en cuenta las posibilidades de obtener una determinada jugada, sino también el hecho de que ésta sea mejor que la de su oponente.

La posibilidad de obtener una cierta jugada obedece a un cálculo relativamente complejo. En todo caso, el mayor de los problemas se encuentra en el intento de evaluar las posibilidades de construir una «escalera». Veámoslo a través de un ejemplo.

Si el ordenador se encuentra con tres reyes en su mano, sabe que sus posibilidades de llegar a poker de reyes se reducen en el descarte el rey que colmaría su satisfacción. Observemos ahora el caso de la «escalera». En el supuesto de que disponga de las cartas 8, 9, 10 y J, más una carta inservible, está claro que la posibilidad de conseguir «escalera» es tan fácilmente evaluable como la de «poker» de reyes: es absolutamente necesario extraer una dama. Suponiendo que las cartas fueran 9, 10 J y Q (más una inservible), el problema se compliaría ligeramente. De nuevo, se dispondría de una base de cuatro cartas seguidas, aunque en esta ocasión, la quinta en discordia puede pertenecer a dos números diferentes: los ochos y los reyes.

Veámos un último ejemplo. Disponemos de las cartas 9, 10, Q y K (más una inservible). Las posibilidades de escalera, analizadas torpemente, serían bien pocas, ya que ante la necesidad de obtener cinco cartas seguidas sólo tenemos la base de dos seguidas: el 9 y el 10, o la Q y la K. Sin embargo, nos encontramos ante las mismas probabilidades que el primero de los ejemplos, ya que sólo nos hace falta una. En definitiva, el análisis de las posibilidades de obtener una escalera, no hay que limitarlo al número de «peldaños» seguidos de los que se dispone, sino que también hay que considerar el naípe de comienzo y fin de la «escalera», y cómo no, el número de huecos que existen en la misma.

De esta misión se encarga la subrutina denominada EVALUADOR. Aunque el problema a resolver haya quedado bastante delimitado, lo que no ha quedado nada claro es el algoritmo para conseguir su resolución y, menos aún, la forma de plasmarlo en el lenguaje BASIC. ¿Cuál es la solución adoptada?... ¿Estructuras IF-THEN? Ello habría complicado enormemente el programa y sin duda lo habría alargado aún más. En su defecto, nos hemos decidido a utilizar una técnica aplicada con gran éxito a la informática, regida por la denominada teoría de autómatas.

Concretamente, el método empleado obedece al nombre de autómata finito. Desde luego, el método resultará útil en otros programas

de esta índole, aunque cabe advertir de antemano que se trata de una técnica de relativa sofisticación, cuya complejidad puede escapar a personas no introducidas aún de lleno en la programación.

## LA SUBASTA DE APUESTAS

Otro punto importante en el juego del póker es saber llevar a cabo una buena subasta. ¡Con seguridad y confianza en la propia jugada, incluso cuando sea un «farol»!

El programa se conduce en la subasta en base a unas supuestas probabilidades de ganar la baza. A partir de ellas y del desarrollo de la subasta, decide si verá o no la apuesta del contrario, e incluso puede optar por subirla o plantarse. Para guiarse en la subasta, adopta un comportamiento cuya representación genérica aparece en el gráfico 4.5.8.; éste depende del porcentaje de probabilidades de victoria. Como se puede observar, existe una discontinuidad en la mencionada función de apuestas, de forma que hasta que no se alcanzan ciertas probabilidades, el ordenador no se decide a «ver» las apuestas, o a apostar cuando es mano.

A partir de este punto de discontinuidad, comienzan las apuestas, que siguen lógicamente, una rampa creciente según las probabilidades de victoria.

Por último, en un determinado punto, se produce una inflexión que incrementa notoriamente la pendiente de la rampa.

¿Cómo descubrir cuáles son los porcentajes exactos en los que se produce discontinuidad, inflexión...? Revelarlo tal vez fuera brindar un arma excesivamente potente para ganar a la máquina.

Al margen de otras consideraciones, hay que tener siempre en cuenta que para la evaluación de las probabilidades no sólo se detiene en la propia jugada, sino también en la que se le supone al contrario. Ello lo calcula el ordenador en base al descarte del jugador humano y a una matriz de aprendizaje; en ésta última, el programa intenta sintetizar el historial de jugadas acaecidas y el talante de su adversario.

Por supuesto, en el caso de anotarse un «farol», su gestión en la subasta difiere del cauce normal hasta aquí explicado.

## EVOLUCIÓN DE LA FORMA DE JUEGO

El núcleo de este sistema lo constituye una matriz de aprendizaje;

en ella, el programa registra la tendencia de las jugadas elegidas por el contrincante humano. Gracias a esto, y al dato que aporta el número de cartas desechadas, podrá calcular más tarde las probabilidades de victoria.

La evolución de la forma de juego se produce en el ordenador acorde con el juego del oponente. Por ejemplo, si un jugador es muy «farolero», el programa acabará desconfiando de él. Si es muy arriesgado, y suele ir a por todas las jugadas altas, el programa le seguirá, procurando ir a por jugadas mejores que le permitan vencer.

Para evitar cambios bruscos en las tendencias reflejadas en la matriz de aprendizaje, motivados por repentinos golpes de suerte, la actualización de ésta se realiza con la dilación adecuada.

## EL AUTÓMATA

Sin entrar en las intrincadas profundidades de la teoría matemática, podemos definir a los autómatas como instrumentos capaces de representar determinadas circunstancias que, lógicamente, cambian según ciertas condiciones iniciales y de acuerdo con unos datos de entrada.

En definitiva, el autómata es un mecanismo lógico que, partiendo de unas entradas y según unas normas previamente establecidas, entrega una solución final. Cada una de las distintas circunstancias del problema que representa el autómata se denomina ESTADO. En nuestro caso, debido a que el autómata utilizado en el programa no posee un número infinito de estados (sólo tiene 8), éste recibe el nombre de autómata finito. El ejemplo práctico incluido en el programa POKER, ayudará a comprender la teoría general de los autómatas.

Como ya hemos visto, la resolución del problema del descarte se encuentra con un considerable escollo al contemplar la posibilidad de realizar una escalera. No sólo es necesario atender al número de cartas seguidas, sino también a los posibles vanos que existan entre grupos de cartas; en cierto modo, hay que tener conciencia de la posición de las cartas con respecto a las anteriores.

Este tipo de problema se resuelve habitualmente mediante estructuras IF-THEN; sin embargo, en el caso concreto que nos ocupa, ello se traduciría en un considerable abuso de espacio de memoria, y en una no menor complicación del programa. En consecuencia, se ha

optado por construir un autómata capaz de solucionar el problema con admirable elegancia.

Nuestro autómata utiliza como entrada la diferencia entre la carta que se analiza y la anterior (téngase en cuenta, que previamente han sido ordenadas). Inicialmente, el estado del autómata se puede considerar como el ideal (escalera); si la próxima carta es una unidad mayor, el estado seguirá siendo el mismo, el ideal, ya que nos encontramos en camino hacia la escalera. Si la entrada fuera dos en lugar de uno, el autómata cambiaría de estado, manifestando que se ha producido un vano en la escalera; y, desde luego, también conmutará a un estado distinto del inicial cuando la entrada sea cero (la nueva carta es igual a la anterior).

La característica más clara de este autómata es que cuando las entradas son uno (naipes seguidos), no altera su estado anterior. Al finalizar las cinco entradas, una por cada carta, el estado del autómata (estado final) reflejará nuestra situación para obtener una escalera. El hecho de conservar el estado inicial, reflejaría, en nuestro caso, que ya se tiene una escalera.

Dentro del programa, el autómata está auxiliado por algunos IF-THEN, que permiten su simplificación y eliminan los denominados estados equivalentes a otros ya representados.

En la figura aparece la estructura del autómata diseñado para el programa. Sus filas reflejan los nuevos estados, a los que conmutarán los estados anteriores, según las entradas. A su vez, las columnas representan las entradas, que quedan reducidas a tres. Estas se determinan a partir del valor de la carta actual menos el valor de la carta anterior más uno; generando así entradas del tipo: 1=carta repetida, 2=carta seguida y 3=carta con una separación.

Por otra parte, el hecho de no representar más entradas, deriva de considerar que cuando una carta tiene una diferencia de dos con la anterior, la escalera es casi imposible; los estados que generaría contemplar una nueva entrada complicarían en exceso el autómata, no justificando su utilidad, de ahí que se hayan sustituido por medio de una estructura IF-THEN.

Nuestro autómata, residente en la tabla A(8,3), opera de la siguiente forma: la variable ESTADO indica en cada momento el estado del autómata y ENTR la entrada del naipe. El valor de la tabla, según la entrada correspondiente al estado actual —A (ESTADO,ENTR)— indicará el nuevo estado del autómata. En operación, se parte del estado 8 y el ideal es el 2. Veamos un ejemplo: la primera de las entradas es 2 (cartas seguidas); por lo tanto, partiendo del estado 8 y

con una entrada 2, el autómata pasa al 2 ( $A(8,2)=2$ ). Si la siguiente entrada fuera también 2, el nuevo estado sería 2 ( $A(2,2)=2$ ). De esta forma, se entra en un bucle en el que las entradas 2 al estado 2, dejan siempre al autómata en la misma situación. Como se ha indicado, el hecho de poseer todas las cartas seguidas conduce a un estado final 2.

# EL RAZONAMIENTO

## ABSTRACTO

**L**a función principal de la inteligencia abstracta es la razón, de ahí que se diga que el hombre es un animal racional. Como ya sabemos, somos capaces de partir, de dos conceptos formar un juicio, y en base a estos últimos extraer una conclusión o razonamiento, cumbre del proceso inteligente.

En nuestro camino para la emulación de las funciones inteligentes humanas en el Amstrad, llegamos pues al punto culminante: el razonamiento, es decir, la extracción por parte del ordenador, en base a dos verdades conocidas, de una tercera en principio ignorada. Para conseguirlo, previamente deberemos enseñar a nuestro ordenador una serie de reglas, que le lleven a razonamientos correctos. Veámos estas normas.

### CONCEPTOS

Podemos definir el concepto como la representación intelectual y abstracta de cualquier objeto. Así pues, debemos partir de la base de

que los conceptos pueden ser erróneos, dado que nadie está en posesión de la verdad, y que al ser el fundamento de los razonamientos, estos últimos pueden estar equivocados si parten de conceptos falsos; ello no implica que el camino seguido para obtener el razonamiento sea incorrecto, sino simplemente que un error de base genera una conclusión equivocada. Así por ejemplo, partiendo del concepto «Los pigmeos son una etnia inferior», no podemos llegar a ninguna conclusión acertada.

Por otra parte, en los conceptos debemos distinguir dos vertientes, denominadas *comprensión* y *extensión*. El conjunto de características que definen un concepto se denominan *notas lógicas* y son las que señalan su comprensión. Así por ejemplo, el concepto HOMBRE comprende entre otras notas, ser animal, mamífero, vertebrado, inteligente, etc... La extensión es el conjunto de individuos a los cuales se les puede aplicar un concepto determinado por su comprensión. Así pues, la extensión del concepto hombre es la de todos los hombres, la del concepto HOMBRE BLANCO, es sólo el conjunto de hombres de raza blanca, etc... Rápidamente sacamos la conclusión, que cuanta mayor sea la comprensión de un concepto, menor será su extensión y viceversa.

Desde el punto de vista de la relación de los conceptos, que como veremos es la síntesis de un juicio, pueden ser considerados como *coordinados* o *subordinados*, según tengan igual extensión o no, en cuyo caso serán subordinados aquellos de inferior extensión respecto a otros. Así, siguiendo el ejemplo anterior, los conceptos HOMBRE BLANCO y HOMBRE NEGRO son coordinados entre sí, mientras que ambos son subordinados del concepto HOMBRE.

Por último, existe una tercera clasificación de los conceptos, fundamental como veremos más adelante para la correcta formulación de juicios y razonamientos: según su extensión. En este caso, los conceptos se dividen en *universales*, cuando se hace referencia a toda su extensión, *individuales*, cuando la extensión es un solo individuo y *particulares*, cuando corresponde a una parte de la extensión total. No obstante, la segunda clasificación, individual, se suele englobar dentro de la universal, dado que en cierto modo un solo individuo es la totalidad de su extensión.

## JUICIOS

Un juicio se define como la relación de dos conceptos, que los une o los separa; de esta definición se desprende inmediatamente, que los

juicios pueden ser *positivos*, cuando unen los conceptos, o *negativos*, cuando los separan. Por otra parte, el juicio es algo que los filósofos consideran intramental, por lo que nosotros no nos ocuparemos de los juicios, sino de su expresión, denominada *proposición*. No obstante, podemos decir que si expresamos un juicio es porque existe en nuestra mente, y por extensión, si el ordenador formula una proposición ¿es que tiene una mente para albergar el juicio correspondiente?

Por otra parte, según la forma de las proposiciones, se pueden catalogar en *categorías e hipotéticas*. Las primeras de mayor interés para el experimento práctico que llevaremos a cabo al final de este capítulo, son también llamadas de *cópula verbal*; un ejemplo puede ser EL HOMBRE ES INTELIGENTE, CARLOS CORRE, etc...

Las proposiciones hipotéticas o de *cópula conjuncional* unen los conceptos mediante conjunciones, y sus formas más importantes son: *copulativas* (y, ni), *condicionales* (si... entonces) y *disyuntivas* (o... o).

Profundizando más en las proposiciones categóricas, efectuaremos una división de las mismas fundamental para la correcta emisión de juicios. Por una parte, se consideran *afirmativas* o *negativas*, según afirmen o nieguen el predicado del sujeto. Como vimos al comienzo de este epígrafe, sus correspondientes juicios (recordemos que la proposición es la expresión del juicio) pueden ser *positivos* o *negativos*.

Por último, las proposiciones se pueden considerar *universales* o *particulares*, según sea la extensión del concepto sujeto. Así por ejemplo, la proposición EL HOMBRE ES INTELIGENTE es universal, mientras que ALGUNOS HOMBRES SON LADRONES es particular.

Estas características, hicieron que los antiguos lógicos catalogaran las proposiciones categóricas mediante cuatro letras, sistema que seguirá también nuestro programa de ejemplo:

Proposiciones tipo A → Universales afirmativas.

Proposiciones tipo E → Universales negativas.

Proposiciones tipo I → Particulares afirmativas.

Proposiciones tipo O → Particulares negativas.

## RAZONAMIENTO

El razonamiento se define como la unión de dos o más juicios por parte de la inteligencia abstracta, lo cual propicia la conclusión de una verdad desconocida, llamada *consecuente*, a partir de dos verdades conocidas, denominadas *antecedente*.

No obstante, es evidente que no es posible obtener cualquier consecuente a partir de ciertos antecedentes. Del antecedente, JOSE ABASCAL ES DEL EQUIPO ESPAÑOL y EL EQUIPO ESPAÑOL ESTARA EN LAS OLIMPIADAS, no podemos extraer el consecuente LUEGO JOSE ABASCAL GANARA LA MEDALLA DE ORO EN LAS OLIMPIADAS.

Por último, existen dos clases de razonamientos, conocidos como *inductivos* y *deductivos*. Los primeros, extraen una conclusión universal a partir de un antecedente particular. Imaginemos que nos pudiéramos desplazar a un lejano planeta y examináramos un cierto número de los extraterrestres que lo pueblan. Tras comprobar que todos los observados son de color verde, podríamos extraer por inducción la conclusión LUEGO TODOS LOS HABITANTES DE ESTE PLANETA SON VERDES.

Como podemos ver, el razonamiento inductivo es extraordinariamente peligroso, en cuanto al alto riesgo de equivocación que implica, si no partimos de un número suficiente y fiable de antecedentes.

Por el contrario, los razonamientos deductivos, también conocidos como *silogismos*, suelen ser menos osados, ya que de un antecedente universal extraen un consecuente particular. No obstante, no por ello dejan de ser erróneos en ocasiones. Por ejemplo, en las versiones iniciales de nuestro programa SILOGISMOS, que figura al final de este capítulo, el ordenador llegaba a conclusiones tan cómicas como la siguiente: Carlos es programador, Fernando es programador, luego Carlos es Fernando.

Para evitar la formulación de silogismos erróneos, existen determinadas reglas que a continuación estudiaremos, con el fin de aplicarlas posteriormente a la programación. Concretamente, revisaremos los silogismos más comunes: los categóricos, es decir, aquellos que están formados por proposiciones categóricas, dejando a un lado los hipotéticos (los que empiezan por una proposición hipotética).

## LOS SILOGISMOS CATEGÓRICOS

Estos silogismos constan de tres proposiciones. Las dos primeras, constituyen el antecedente y son también conocidas como *premisas*, y la tercera es la *conclusión* o consecuente. Pues bien, los elementos (sujetos y predicados) que constituyen las proposiciones se denominan *términos*, y para que el razonamiento sea correcto, deben ser únicamente tres: *término mayor* (predicado de la conclusión), *término*

*menor* (sujeto de la conclusión) y *término medio* (el término restante). Por otra parte, los términos mayor y menor también se denominan *extremos*, debiendo ser el *mayor* de más amplia extensión que el *menor*.

El mecanismo del silogismo, explicado según la terminología recién aprendida, es la comparación de los extremos con el término medio. Para su mejor comprensión, podemos asignar unos valores simbólicos A, B y C a los términos menor, medio y mayor, respectivamente, lo cual nos permitirá establecer una «ecuación lógica»:

Los españoles (A) son hombres (B)  
 Los hombres (B) son inteligentes (C)  
 Luego los españoles (A) son inteligentes (C)

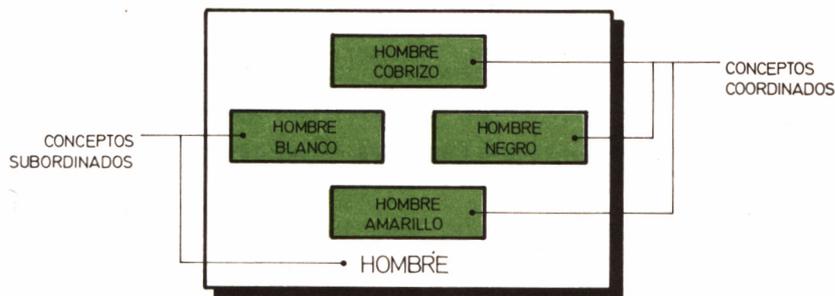
o lo que es lo mismo...

$A = B$   
 $C = B$   
 Luego  $A = C$ .

Esta estructura responde al llamado principio de la conveniencia y la discrepancia, cuyos enunciados son los siguientes:

— Si dos elementos son iguales a un tercero, implica que son iguales entre sí. O expresado simbólicamente:

$A = B$   
 $C = B$   
 Luego  $A = C$



4.6.1. Los conceptos se pueden dividir en subordinados o coordinados, según la relación de sus extensiones.

- Si el primer elemento es igual al tercero y el segundo distinto de éste, el primero y el segundo son también diferentes:

A = B  
 C # B  
 Luego A # C

o bien

A # B  
 C = B  
 Luego A # C

- Si dos elementos son distintos respecto al tercero, no es posible extraer ninguna conclusión.

A # B  
 C # B  
 ?????

Finalmente, existe un principio esencial en la corrección de los silogismos categóricos: lo que se afirma o niega de un elemento de forma universal, debe igualmente afirmarse o negarse de todo aquello que esté comprendido en él. Así por ejemplo, si yo afirmo universalmente que los humanos son inteligentes, y añado que los españoles son humanos, comprendiendo por tanto el elemento español dentro del humano, he de concluir que los españoles son inteligentes.

#### 4.6.2. Programa SILOGISMOS versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /--- SILOGISMOS (c)1986 GRAN BIBLIOTECA AMSTRAD ---
40 /----- CPC 464/472/664/6128 -----
50 /-----
60 /
70 l m$='AEIO':DIM m$(4,6),p$(3,2),e$(3,2),b$(100,100)
80 FOR i=1 TO 4
90 p=0
100 READ x$
110 IF x$='*' THEN 150
120 m$(i,p)=x$
130 p=p+1
140 GOTO 100
150 NEXT
160 DATA ABBC,AAA,AII,EAE,EIO,*
170 DATA ABCB,AEE,A00,EAE,EIO,*
180 DATA BABC,AAI,AII,EA0,EIO,IAI,0A0,*
190 DATA BACB,AAI,AEE,EA0,EIO,IAI,*

```

```

200 MODE 2
210 p=1:PRINT:INPUT "INTRODUZCA LA PREMISA MAYOR: ",my$:my$=
UPPER$(my$):l$(1)=my$
220 x$=my$:GOSUB 1120
230 IF e=1 THEN 210
240 p=2:PRINT:INPUT "INTRODUZCA LA PREMISA MENOR: ",mn$:mn$=
UPPER$(mn$):l$(2)=mn$
250 x$=mn$:GOSUB 1120
260 IF e=1 THEN 240
270 p=3:PRINT:INPUT "INTRODUZCA LA CONCLUSION: ",c1$:c1$=UPP
ER$(c1$):l$(3)=c1$
280 x$=c1$:GOSUB 1120
290 IF e=1 THEN 270
300 e=0:RESTORE 350
310 FOR i=1 TO 3
320 READ x$:IF e$(i,1)=e$(i,2) THEN PRINT:PRINT "LA 'x$' ES
TRIVIAL":e=1
330 NEXT
340 IF e=1 THEN GOSUB 1440:GOTO 200
350 DATA PREMISA MAYOR,PREMISA MENOR,CONCLUSION
360 ma$=e$(3,2):me$=e$(3,1)
370 PRINT:PRINT "EL TERMINO MENOR ES "me$""
380 PRINT:PRINT "EL TERMINO MAYOR ES "ma$""
390 ma=0:pma=0:me=0:pme=0:md=0
400 FOR i=1 TO 2
410 FOR j=1 TO 2
420 IF e$(i,j)=ma$ THEN ma=ma+1:pma=i:GOTO 450
430 IF e$(i,j)=me$ THEN me=me+1:pme=i:GOTO 450
440 md=md+1
450 NEXT j,i
460 IF ma<>1 OR me<>1 OR md<>2 OR pma=pme THEN PRINT:PRINT "
EL SILOGISMO NO ESTA BIEN FORMULADO":GOSUB 1440:GOTO 200
470 IF e$(1,1)<>ma$ AND e$(1,1)<>me$ THEN md$=e$(1,1) ELSE m
d$=e$(1,2)
480 PRINT:PRINT "EL TERMINO MEDIO ES "md$""
490 IF pma=1 THEN 570
500 PRINT:PRINT "LAS PREMISAS ESTAN CAMBIADAS DE ORDEN"
510 PRINT:PRINT mn$:x$=mn$:PRINT:PRINT my$:mn$=my$:PRINT:PRI
NT c1$:my$=x$
520 FOR i=0 TO 2
530 x$=p$(1,i):p$(1,i)=p$(2,i):p$(2,i)=x$
540 x$=e$(1,i):e$(1,i)=e$(2,i):e$(2,i)=x$
550 NEXT
560 x$=l$(1):l$(1)=l$(2):l$(2)=x$:x=pma:pma=pme:pme=x
570 f$=""
580 FOR i=1 TO 2
590 FOR j=1 TO 2
600 x$=e$(i,j):GOSUB 1400
610 f$=f$+x$
620 NEXT j,i
630 x$=RIGHT$(f$,2)+LEFT$(f$,2):f$=x$
640 FOR i=1 TO 4
650 IF m$(i,0)=f$ THEN fig=i:i=4
660 NEXT
670 PRINT:PRINT "EL SILOGISMO SIGUE LA FIGURA'fig': "LEFT$(f
$,1)"="MID$(f$,2,1)" ; "MID$(f$,3,1)"="RIGHT$(f$,1)
680 modo$=""
690 FOR i=1 TO 3
700 PRINT:PRINT "?PUEDA CONSIDERARSE UNIVERSAL LA PROPOSICIO
N "l$(i)"?"
710 x$=INKEY$:IF x$="" THEN 710 ELSE x$=UPPER$(x$)
720 IF x$<>"S" AND x$<>"N" THEN 710
730 x=-2*(x$="N")+VAL(e$(i,0))+1
740 modo$=modo$+MID$(lm$,x,1)
750 NEXT
760 PRINT:PRINT "EL SILOGISMO PERTENECE A LA FIGURA'fig'MODO
'modo$

```

```

770 FOR i=1 TO 6
780 IF m$(fig,i)=modo$ THEN 810
790 NEXT
800 PRINT:PRINT 'EL SILOGISMO NO ES CORRECTO. VUELVA A FORMU
LARLO':GOSUB 1440:GOTO 200
810 PRINT:PRINT 'EL SILOGISMO ES CORRECTO'
820 FOR i=1 TO 3
830 x$=MID$(modo$,i,1)+e$(i,2)
840 e1=1:FOR j=1 TO 100
850 IF b$(0,j)=" THEN b$(0,j)=e$(i,1):GOTO 870
860 IF b$(0,j)>e$(i,1) THEN 930
870 e2=1:FOR k=1 TO 100
880 IF b$(k,j)=x$ THEN e2=0:k=100:GOTO 900
890 IF b$(k,j)=" THEN b$(k,j)=x$:e2=0:k=100
900 NEXT
910 IF e2=1 THEN PRINT:PRINT 'NO ES POSIBLE A\ADIR NINGUNA N
OTA MAS A 'b$(0,j)''
920 e1=0:j=100
930 NEXT
940 IF e1=1 THEN PRINT:PRINT 'NO ES POSIBLE A\ADIR A LA BASE
EL CONCEPTO 'e$(i,1)''
950 NEXT
960 PRINT:INPUT '?SOBRE QUE CONCEPTO DESEA INFORMACION (!EN
SINGULAR!)':c$:c$=UPPER$(c$):IF c$="" THEN 200
970 e=1:FOR i=1 TO 100
980 IF b$(0,i)=" THEN i=100:GOTO 1090
990 IF b$(0,i)>c$ THEN GOTO 1090
1000 e=0:PRINT
1010 FOR j=1 TO 100
1020 IF b$(j,i)=" THEN j=100:i=100:GOTO 1080
1030 x$=b$(j,i)
1040 IF LEFT$(x$,1)="A" THEN PRINT 'TODO 'c$' ES 'MID$(x$,2)
':GOTO 1080
1050 IF LEFT$(x$,1)="E" THEN PRINT 'NINGUN 'c$' ES 'MID$(x$,
2):GOTO 1080
1060 IF LEFT$(x$,1)="I" THEN PRINT 'ALGUN 'c$' ES 'MID$(x$,2
)':GOTO 1080
1070 IF LEFT$(x$,1)="O" THEN PRINT 'ALGUN 'c$' NO ES 'MID$(x
$,2)
1080 NEXT
1090 NEXT
1100 IF e=1 THEN PRINT:PRINT 'NO TENGO INFORMACION SOBRE EL
CONCEPTO 'c$''
1110 GOTO 960
1120 e=0:ng=0:pl=0
1130 cm$=' NO ES ':l=INSTR(x$,cm$)
1140 IF l>0 THEN ng=1:GOTO 1240
1150 cm$=' ES ':l=INSTR(x$,cm$)
1160 IF l>0 THEN 1240
1170 pl=1
1180 cm$=' NO SON ':l=INSTR(x$,cm$)
1190 IF l>0 THEN ng=1:GOTO 1240
1200 cm$=' SON ':l=INSTR(x$,cm$)
1210 IF l>0 THEN 1240
1220 PRINT:PRINT 'NO HE LOCALIZADO EL NEXO':PRINT 'RECUERDE
INCLUIR EL VERBO 'SER'
1230 GOSUB 1440:e=1:RETURN
1240 p$(p,1)=LEFT$(x$,l-1)
1250 p$(p,2)=MID$(x$,l+LEN(cm$))
1260 FOR i=1 TO 2
1270 RESTORE 1330
1280 e$(p,i)=p$(p,i)
1290 FOR j=1 TO 10
1300 READ cm$
1310 IF LEFT$(e$(p,i),LEN(cm$))=cm$ THEN e$(p,i)=MID$(e$(p,i
),LEN(cm$)+2):j=10

```

```

1320 NEXT j,i
1330 DATA UNOS,UNAS,UNO,UNA,UN,LOS,LAS,LO,LA,EL
1340 IF p1=0 THEN 1380
1350 FOR i=1 TO 2
1360 PRINT:PRINT '?CUAL ES EL SINGULAR DE ''e$(p,i)''?':INPU
T e$(p,i):e$(p,i)=UPPER$(e$(p,i))
1370 NEXT
1380 e$(p,0)=STR$(ng)
1390 RETURN
1400 IF x$=ma$ THEN x$='C':GOTO 1430
1410 IF x$=me$ THEN x$='A':GOTO 1430
1420 IF x$=md$ THEN x$='B'
1430 RETURN
1440 PRINT:PRINT 'PULSE UNA TECLA'
1450 WHILE INKEY$='':WEND
1460 RETURN

```

#### 4.6.3. Programa SILOGISMOS versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- SILOGISMOS (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 in$=CHR$(27)+'f':ac$=CHR$(27)+'e':PRINT in$:k1s$=CHR$(27)
+'E'+CHR$(27)+'H':pq$=CHR$(27)+'x':gr$=CHR$(27)+'y'
90 PRINT pq$:k1s$
100 lm$='AEIO':DIM m$(4,6),p$(3,2),e$(3,2),b$(50,50)
110 FOR i=1 TO 4
120 p=0
130 READ x$
140 IF x$='*' THEN 180
150 m$(i,p)=x$
160 p=p+1
170 GOTO 130
180 NEXT
190 DATA ABBC,AAA,AII,EAE,EIO,*
200 DATA ABCB,AEE,A00,EAE,EIO,*
210 DATA BABC,AAI,AII,EA0,EIO,IAI,0A0,*
220 DATA BACB,AAI,AEE,EA0,EIO,IAI,*
230 PRINT k1s$
240 p=1:PRINT ac$:INPUT 'INTRODUZCA LA PREMISA MAYOR: ',my$:
PRINT in$:my$=UPPER$(my$):l$(1)=my$
250 x$=my$:GOSUB 1150
260 IF e=1 THEN 240
270 p=2:PRINT ac$:INPUT 'INTRODUZCA LA PREMISA MENOR: ',mn$:
PRINT in$:mn$=UPPER$(mn$):l$(2)=mn$
280 x$=mn$:GOSUB 1150
290 IF e=1 THEN 270
300 p=3:PRINT ac$:INPUT 'INTRODUZCA LA CONCLUSION: ',c1$:PRI
NT in$:c1$=UPPER$(c1$):l$(3)=c1$
310 x$=c1$:GOSUB 1150
320 IF e=1 THEN 300
330 e=0:RESTORE 380
340 FOR i=1 TO 3
350 READ x$:IF e$(i,1)=e$(i,2) THEN PRINT:PRINT 'LA 'x$' ES
TRIVIAL':e=1
360 NEXT
370 IF e=1 THEN GOSUB 1470:GOTO 230

```

```

380 DATA PREMISA MAYOR,PREMISA MENOR,CONCLUSION
390 ma$=e$(3,2):me$=e$(3,1)
400 PRINT:PRINT "EL TERMINO MENOR ES "me$""
410 PRINT:PRINT "EL TERMINO MAYOR ES "ma$""
420 ma=0:pma=0:me=0:pme=0:md=0
430 FOR i=1 TO 2
440 FOR j=1 TO 2
450 IF e$(i,j)=ma$ THEN ma=ma+1:pma=i:GOTO 480
460 IF e$(i,j)=me$ THEN me=me+1:pme=i:GOTO 480
470 md=md+1
480 NEXT j,i
490 IF ma<>1 OR me<>1 OR md<>2 OR pma=pme THEN PRINT:PRINT "
EL SILOGISMO NO ESTA BIEN FORMULADO":GOSUB 1470:GOTO 230
500 IF e$(1,1)<>ma$ AND e$(1,1)<>me$ THEN md$=e$(1,1) ELSE m
d$=e$(1,2)
510 PRINT:PRINT "EL TERMINO MEDIO ES "md$""
520 IF pma=1 THEN 600
530 PRINT:PRINT "LAS PREMISAS ESTAN CAMBIADAS DE ORDEN"
540 PRINT:PRINT mn$:x$=mn$:PRINT:PRINT my$:mn$=my$:PRINT:PRI
NT c$:my$=x$
550 FOR i=0 TO 2
560 x$=p$(1,i):p$(1,i)=p$(2,i):p$(2,i)=x$
570 x$=e$(1,i):e$(1,i)=e$(2,i):e$(2,i)=x$
580 NEXT
590 x$=l$(1):l$(1)=l$(2):l$(2)=x$:x=pma:pma=pme:pme=x
600 f$=""
610 FOR i=1 TO 2
620 FOR j=1 TO 2
630 x$=e$(i,j):GOSUB 1430
640 f$=f$+x$
650 NEXT j,i
660 x$=RIGHT$(f$,2)+LEFT$(f$,2):f$=x$
670 FOR i=1 TO 4
680 IF m$(i,0)=f$ THEN fig=i:i=4
690 NEXT
700 PRINT:PRINT "EL SILOGISMO SIGUE LA FIGURA"fig": "LEFT$(f
$,1)"="MID$(f$,2,1)" ; "MID$(f$,3,1)"="RIGHT$(f$,1)
710 modo$=""
720 FOR i=1 TO 3
730 PRINT:PRINT "?PUEDA CONSIDERARSE UNIVERSAL LA PROPOSICIO
N "l$(i)"?"
740 x$=INKEY$:IF x$="" THEN 740 ELSE x$=UPPER$(x$)
750 IF x$<>"S" AND x$<>"N" THEN 740
760 x=-2*(x$="N")+VAL(e$(i,0))+1
770 modo$=modo$+MID$(l$,x,1)
780 NEXT
790 PRINT:PRINT "EL SILOGISMO PERTENECE A LA FIGURA"fig"MOD0
"modo$
800 FOR i=1 TO 6
810 IF m$(fig,i)=modo$ THEN 840
820 NEXT
830 PRINT:PRINT "EL SILOGISMO NO ES CORRECTO. VUELVA A FORMU
LARLO":GOSUB 1470:GOTO 230
840 PRINT:PRINT "EL SILOGISMO ES CORRECTO"
850 FOR i=1 TO 3
860 x$=MID$(modo$,i,1)+e$(i,2)
870 e1=1:FOR j=1 TO 50
880 IF b$(0,j)="" THEN b$(0,j)=e$(i,1):GOTO 900
890 IF b$(0,j)<>e$(i,1) THEN 960
900 e2=1:FOR k=1 TO 50
910 IF b$(k,j)=x$ THEN e2=0:k=50:GOTO 930
920 IF b$(k,j)="" THEN b$(k,j)=x$:e2=0:k=50
930 NEXT
940 IF e2=1 THEN PRINT:PRINT "NO ES POSIBLE A!ADIR NINGUNA N
OTA MAS A "b$(0,j)""
950 e1=0:j=50

```

```

960 NEXT
970 IF e1=1 THEN PRINT:PRINT 'NO ES POSIBLE ANADIR A LA BASE
    EL CONCEPTO ``e$(i,1)``'
980 NEXT
990 PRINT ac$:INPUT '?SOBRE QUE CONCEPTO DESEA INFORMACION (
    'EN SINGULAR!');c$:PRINT in$;:c$=UPPER$(c$):IF c$="" THEN 23
0
1000 e=1:FOR i=1 TO 50
1010 IF b$(0,i)="" THEN i=50:GOTO 1120
1020 IF b$(0,i)<>c$ THEN GOTO 1120
1030 e=0:PRINT
1040 FOR j=1 TO 50
1050 IF b$(j,i)="" THEN j=50:i=50:GOTO 1110
1060 x$b$(j,i)
1070 IF LEFT$(x$,1)="A" THEN PRINT 'TODO 'c$' ES 'MID$(x$,2)
:GOTO 1110
1080 IF LEFT$(x$,1)="E" THEN PRINT 'NINGUN 'c$' ES 'MID$(x$,
2):GOTO 1110
1090 IF LEFT$(x$,1)="I" THEN PRINT 'ALGUN 'c$' ES 'MID$(x$,2
):GOTO 1110
1100 IF LEFT$(x$,1)="O" THEN PRINT 'ALGUN 'c$' NO ES 'MID$(x
$,2)
1110 NEXT
1120 NEXT
1130 IF e=1 THEN PRINT:PRINT 'NO TENGO INFORMACION SOBRE EL
CONCEPTO ``c$``'
1140 GOTO 990
1150 e=0:ng=0:p1=0
1160 cm$="" NO ES ':l=INSTR(x$,cm$)
1170 IF l>0 THEN ng=1:GOTO 1270
1180 cm$="" ES ':l=INSTR(x$,cm$)
1190 IF l>0 THEN 1270
1200 p1=1
1210 cm$="" NO SON ':l=INSTR(x$,cm$)
1220 IF l>0 THEN ng=1:GOTO 1270
1230 cm$="" SON ':l=INSTR(x$,cm$)
1240 IF l>0 THEN 1270
1250 PRINT:PRINT 'NO HE LOCALIZADO EL NEXO':PRINT 'RECUERDE
INCLUIR EL VERBO 'SER'
1260 GOSUB 1470:e=1:RETURN
1270 p$(p,1)=LEFT$(x$,l-1)
1280 p$(p,2)=MID$(x$,l+LEN(cm$))
1290 FOR i=1 TO 2
1300 RESTORE 1360
1310 e$(p,i)=p$(p,i)
1320 FOR j=1 TO 10
1330 READ cm$
1340 IF LEFT$(e$(p,i),LEN(cm$))=cm$ THEN e$(p,i)=MID$(e$(p,i
),LEN(cm$)+2):j=10
1350 NEXT j,i
1360 DATA UNOS,UNAS,UNO,UNA,UN,LOS,LAS,LO,LA,EL
1370 IF p1=0 THEN 1410
1380 FOR i=1 TO 2
1390 PRINT ac$:PRINT '?CUAL ES EL SINGULAR DE ``e$(p,i)``':;
INPUT e$(p,i):PRINT in$;:e$(p,i)=UPPER$(e$(p,i))
1400 NEXT
1410 e$(p,0)=STR$(ng)
1420 RETURN
1430 IF x$=ma$ THEN x$='C':GOTO 1460
1440 IF x$=me$ THEN x$='A':GOTO 1460
1450 IF x$=md$ THEN x$='B'
1460 RETURN
1470 PRINT:PRINT 'PULSE UNA TECLA'
1480 WHILE INKEY$="" :WEND
1490 RETURN

```

## REGLAS DE LOS SILOGISMOS CATEGÓRICOS

Son cuatro las formas que pueden adoptar un silogismo categórico, según el lugar que ocupe el término medio en las premisas, y se conocen bajo la denominación general de *figuras*:

— Figura 1: el término medio es el sujeto de la premisa en la que aparece el término mayor (premisa mayor) y el predicado en la que aparece el término menor (premisa menor).

Siguiendo nuestra notación «matemática», la ecuación lógica se representaría así:

$$\begin{aligned}A &= B \\B &= C\end{aligned}$$

— Figura 2: el término medio es el predicado de las dos premisas.

$$\begin{aligned}A &= B \\C &= B\end{aligned}$$

— Figura 3: el término medio hace de sujeto en las dos premisas (figura inversa a la segunda).

$$\begin{aligned}B &= A \\B &= C\end{aligned}$$

— Figura 4: el término medio hace de predicado en la premisa mayor y de sujeto en la menor (figura inversa a la primera).

$$\begin{aligned}B &= A \\C &= B\end{aligned}$$

Además de las figuras, los silogismos categóricos pueden presentar diferentes variedades en lo referente a la calidad y cualidad de las proposiciones que los componen. Para refrescarnos la memoria, repitiremos que estas variedades se designan mediante las primeras cuatro vocales:

Proposiciones tipo A → Universales afirmativas.

Proposiciones tipo E → Universales negativas.

Proposiciones tipo I → Particulares afirmativas.

Proposiciones tipo O → Particulares negativas.

En un rápido cálculo, descubriremos que cuatro elementos tomados de tres en tres, pues existen tres proposiciones en cada silogismo, generan 64 variaciones posibles, que multiplicadas por 4, el número de figuras, dan un total de 256 silogismos diferentes:  $2 \uparrow 8$ , un byte, ¡un número que le viene como anillo al dedo a un ordenador! Parece que

el destino hubiera decidido que el razonamiento y la Informática fueran dos conceptos que un día trabajaran juntos.

No obstante, de estos 256 modos posibles de construir un silogismo no todos son válidos: tan sólo 19.

Para la figura 1 → AAA, AII, EAE, EIO

Para la figura 2 → AEE, AOO, EAE, EIO

Para la figura 3 → AAI, AII, EAO, EIO, IAI, OAO

Para la figura 4 → AAI, AEE, EAO, EIO, IAI

Donde las vocales representan, respectivamente, la premisa mayor, la menor y la conclusión.

## SILOGISMOS

El funcionamiento interno del programa SILOGISMO ha sido prácticamente explicado en la sección principal de este capítulo, dado que sigue al pie de la letra las clasificaciones y reglas que han sido explicadas con anterioridad.

Así pues, daremos a continuación unas normas para su utilización, pudiendo el lector comprobar a partir de este momento, cómo el ordenador es capaz de extraer sus propias conclusiones, generando una información que no le ha sido previamente suministrada, almacenándola en su memoria, de forma que puede ser interrogado acerca de las conclusiones que ha extraído, e incluso actuando como corrector de algunos razonamientos humanos bastante dudosos.

Desde luego que siempre se mueve dentro del campo de los silogismos categóricos y por supuesto, del mismo modo que los humanos, no puede asegurar que extraiga una conclusión cierta si las premisas proporcionadas no lo eran. No obstante, este último punto no resta mérito a su funcionamiento, dado que lo que se puede asegurar es que la consecuencia deducida del antecedente suministrado es totalmente lógica.

## FUNCIONAMIENTO DE SILOGISMOS

El programa se desarrolla en dos etapas fundamentales: admisión de un silogismo e inclusión de los datos proporcionados por éste en una base de datos.

Para el cumplimiento de la primera fase, el usuario debe efectuar

la entrada de la premisa mayor, la menor y la conclusión. Dichas proposiciones deben emplear el verbo ser en sus formas ES o SON. Por otra parte, es recomendable que los conceptos se suministren en singular y masculino, aunque esto último no es de gran importancia. En caso de que SILOGISMOS detecte la presencia de un plural, nos interrogará sobre la forma singular del concepto, dejando que corra de nuestra parte el control del género del mismo.

Es posible la utilización de los artículos determinados e indeterminados, pero es preferible evitar los adverbios que cuantifiquen los conceptos, como ALGUNOS o TODOS, ya que, como veremos más adelante, el programa nos interrogará sobre la universalización de las proposiciones enunciadas.

Una vez cumplimentada esta primera entrada de datos, el ordenador efectúa las operaciones tendentes a calificar el silogismo: averiguar la posición de los términos mayor, menor y medio, llegando a incluso alterar el orden de las premisas en caso de que éste hubiera sido erróneamente indicado por el usuario. Tras un primer análisis, el programa ya se encuentra en disposición de emitir un primer dictamen sobre la corrección del silogismo, indicando, de obtener un resultado positivo, la figura a la que pertenece.

A continuación, se procede a preguntar acerca de la universalidad de las proposiciones que componen el razonamiento, ocupándose el programa de registrar si son positivas o negativas, atendiendo a la presencia de la negación ante el verbo SER. A partir de estos datos, el programa puede conocer el modo del silogismo, y junto con el tipo de figura, indicar definitivamente la corrección del mismo.

La segunda fase, comienza con la inclusión en la base de datos de la nueva información obtenida y finaliza preguntando si el usuario desea conocer los datos almacenados sobre algún concepto. En caso negativo, debe pulsarse simplemente RETURN.

En definitiva un programa simple; tan simple como la razón...

## POR BUEN CAMINO

**E**n el procedimiento de búsqueda de una solución mediante un sistema de IA, existe una premisa ineludible: la posibilidad de obtener al menos una solución en un tiempo razonable. Desde luego, si el programa ELIZA tardara una hora en darnos una respuesta a una pregunta, por inteligente que fuera ésta, se nos quitarían de golpe las ganas de hablar con él.

Por otra parte, en la resolución del problema se presentan tres fases:

- Representación del problema.
- Elección de un método adecuado para su solución.
- Consecución de la solución del mismo.

La representación adecuada del problema es un punto esencial para poder localizar la solución con prontitud, gracias a un método de búsqueda adecuado. En el caso concreto de la inteligencia artificial, en el que se plantean con frecuencia problemas que se pueden ir desgajando en multitud de subproblemas, la representación de los mismos suele adoptar la forma de un árbol de decisión.

Se trata de problemas de relativa complejidad, para cuya solución

es necesaria la toma de varias decisiones conjuntas, ya que sus efectos están ligados mutuamente. Conforman una especie de reacción en cadena, en las que cada decisión genera una nueva situación, en muchos casos diferente a la que se llegaría habiendo tomado otra decisión previa distinta. Enfrentados con este nuevo subproblema, debemos entonces optar por otra decisión que ramificará nuevamente el problema principal.

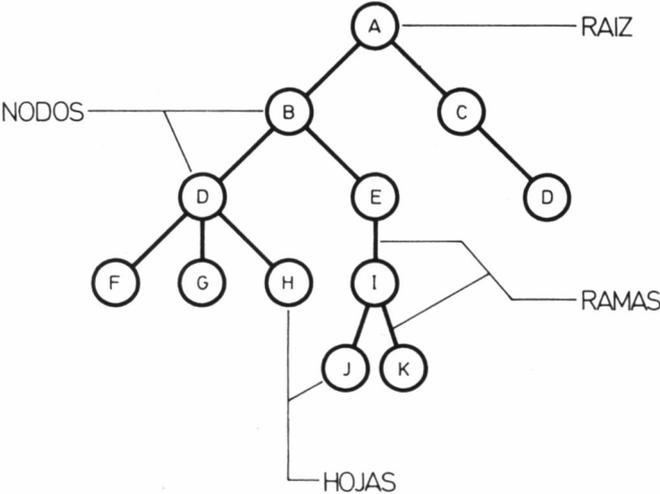
Este tipo de estructuras representadas esquemáticamente adoptan la forma de un árbol inverso, es decir, con las ramas creciendo hacia abajo.

Así por ejemplo, se iban dividiendo las diferentes situaciones ante las que se podían ver las cajas de cerillas de Michie según avanzaban en el juego, pero siempre condicionadas a las decisiones anteriores, es decir, al camino escogido para llegar hasta ellas.

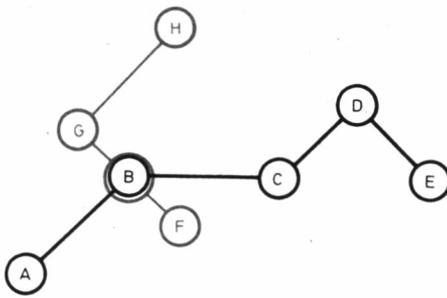
## LOS ARBOLES

Para que de ahora en adelante, comprendamos la terminología utilizada generalmente en los libros de IA, para tratar el análisis de árboles de búsqueda, comenzaremos con unas breves definiciones.

Se denomina *salida* o raíz a la situación de partida, desde la

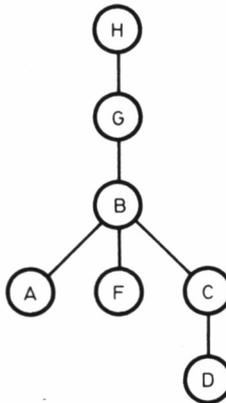


4.7.1. Gracias a las estructuras de árbol es posible resolver multitud de problemas de I.A.

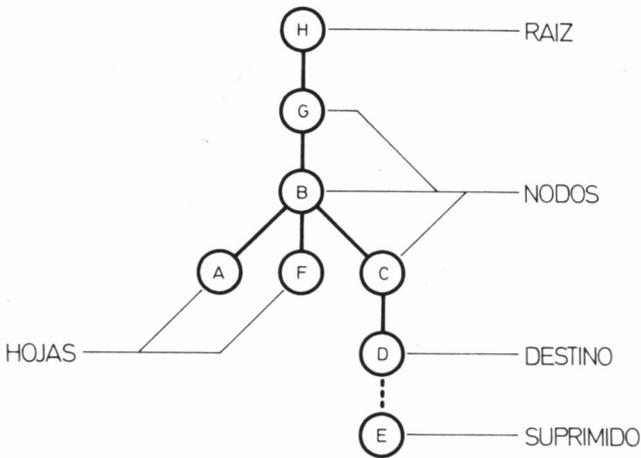


4.7.2. Esta red de metro ficticia nos servirá para ilustrar los problemas de transportes.

cual se genera el árbol, y *nodo* a cada uno de los estados diferentes que se pueden dar a partir de la salida y que a su vez dan lugar a alguna otra situación. Las situaciones finales, podríamos decir, los nodos terminales, es decir, aquellas posiciones que no generan otra diferente, se denominan *hojas*. Por otra parte, aquella o aquellas hojas que representan la solución se denominan *destino*. Finalmente, los segmentos carentes de orientación (ésta es siempre, por defecto, descendente) que unen todas las situaciones se denominan *arcos* o *ramas*.



4.7.3.



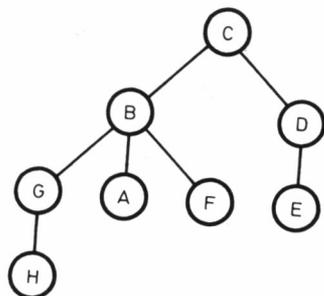
■ 4.7.4. La hoja E se suprime dado que se genera a partir de la solución.

Generalmente, se utilizan los problemas de transportes para ilustrar las estructuras de árbol. Como nosotros no vamos a ser menos, fijémonos en el mapa de una ficticia red de metro de considerablemente reducidas dimensiones. Supongamos que queremos llegar de la estación H a la D. El árbol que definiría el problema sería el gráfico 4.7.2.

En él podemos apreciar la raíz (H), el destino (D), el resto de los nodos (G, B y C) y las correspondientes hojas, que evidentemente son finales de línea (A y F). Por otra parte, suprimiremos la posible hoja E, dado que se genera a partir del destino, por lo que su representación es de manifiesta inutilidad, ya que en nuestra posterior búsqueda de la solución no nos vamos a pasar ésta de largo. Evidentemente, no existiría ningún problema en escoger cualquier otro recorrido, como por ejemplo el de C a H, en el cual no se suprime ningún paso, al coincidir el destino con el final de trayecto (fig. 4.7.5).

De forma similar, en la estructura de árboles tampoco se produce ningún problema si se generan por distintos caminos dos estados iguales. Así por ejemplo, si el Ayuntamiento tuviera la brillante idea inaugurar la estación I (fig. 4.7.6), prolongación de la línea negra, para servir de enlace y transbordo con la línea verde en D, nuestro primer problema quedaría resuelto con mayor facilidad.

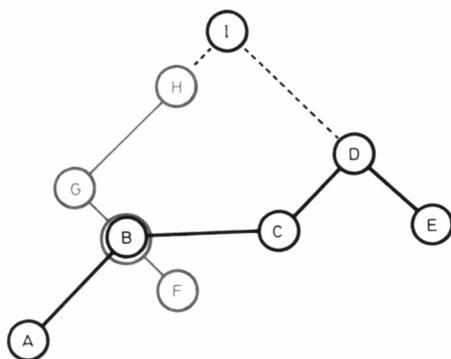
En el árbol correspondiente a esta nueva situación (fig. 4.7.7),



4.7.5.

podemos observar que a la solución D es posible acceder por dos caminos diferentes. En estos casos, representaremos por los dos recorridos el mismo estado, pero no trazaremos un arco de unión entre el punto I y el D proveniente de C, ya que hemos de respetar la norma según la cual a cada nodo u hoja sólo debe llegar un arco (fig. 4.7.8.).

Si D hubiera sido, además de una situación múltiple, un nodo, es decir, un generador de otros arcos, éstos sólo se habrían representado en uno de los casos, para evitar una complicación innecesaria, ya no sólo de la representación del problema (gráfico), sino de la posterior búsqueda de la solución. En la figura 4.7.9 tenemos una ilustración práctica de este caso, representada por un gráfico ficticio ajeno al problema de transporte que hemos estado viendo hasta el momen-



4.7.6.

to. En la parte izquierda del gráfico, apreciamos el árbol tal y como quedaría si efectuáramos un desarrollo completo del mismo, incluso cuando se repitieran situaciones iguales, y a su derecha, el árbol convenientemente podado (fig. 4.7.10).

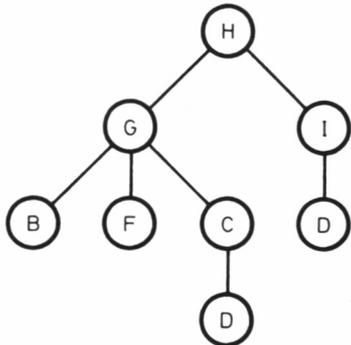
No debemos pensar que esta técnica de representación mediante árboles es inútil, y de hecho es empleada con éxito para la resolución de multitud de problemas típicos de IA. Sin ir más lejos, el autómata finito cuya efectividad pudimos comprobar en el programa de póker del capítulo quinto, puede ser representado igualmente mediante un árbol para clarificar su cometido, dado que cada estado es una consecuencia de los estados anteriores (fig. 4.7.11).

## MÉTODOS DE BÚSQUEDA

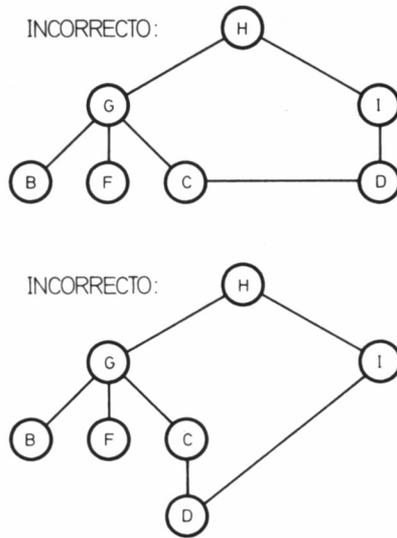
Muchos son los métodos de búsqueda diseñados para la obtención de soluciones en árboles. Entre ellos dos son los más destacados: por *niveles* y en *profundidad*. Más adelante, trataremos otros sistemas como la búsqueda *ordenada*, combinación del tratamiento por niveles y en profundidad, u otros con aplicaciones bastante concretas, como son el sistema *minimax* y el algoritmo de evaluación *alfa-beta*.

## BÚSQUEDA POR NIVELES

En un árbol, podemos apreciar diversos estratos según vamos profundizando hacia las hojas del mismo. El sistema de búsqueda por



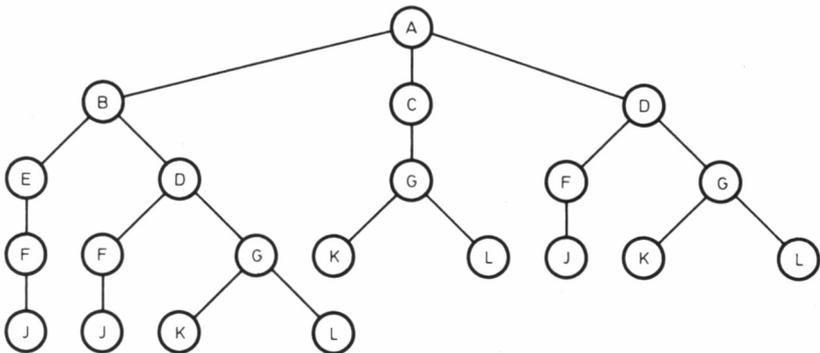
4.7.7. En un árbol se puede presentar una misma situación (D) en zonas diferentes (desde C e I).



■ 4.7.8. Los gráficos son incorrectos porque no respetan la norma según la cual a cada nodo u hoja debe llegar una sola rama.

niveles propone ir investigando si se ha llegado al destino, analizando todos los nodos de cada nivel antes de pasar al siguiente.

Así por ejemplo, retomando nuestro problema de comunicaciones, en el árbol construido para encontrar el camino entre C y H, analizaríamos tras la raíz, B y D, posteriormente, G, A, F y E, y por último H (fig. 4.7.12).



■ 4.7.9.

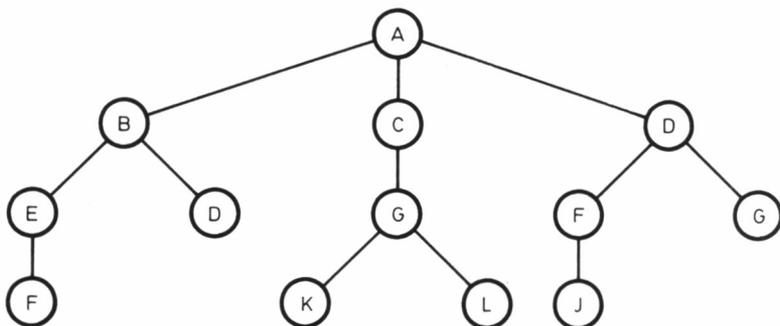
Para la codificación en forma de programa de este tipo de análisis, se emplea una estructura denominada cola o *queue*, en la cual la entrada y salida de elementos se gestiona mediante un sistema FIFO (*First In, First Out*, el primero en entrar es el primero en salir).

En nuestro ejemplo concreto, la cola estaría conformada por C, B, D, G, A, F, E, H, procediéndose a su evaluación de izquierda a derecha.

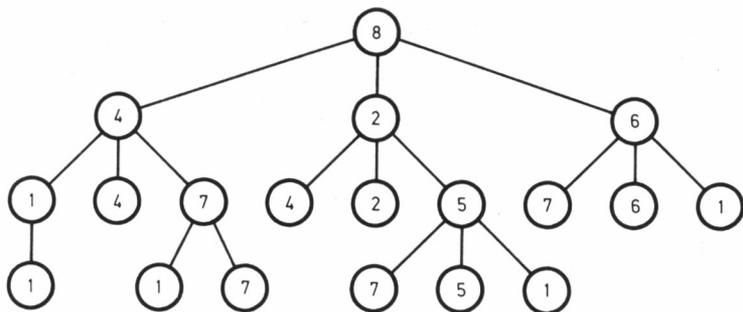
En problemas con multitud de ramas que quizá no se lleguen a analizar, dado que es probable encontrar la solución sin necesidad de tener que revisar todo el árbol, para evitar construir una cola innecesariamente larga de principio, procederemos a añadir a ésta los nodos generados por cada análisis. Lo veremos más claramente en el ejemplo gráfico; tengamos en cuenta que en la representación de los árboles, aparecen cobreados los nodos que ya han sido analizados (nodos cerrados) y en su estado normal los que restan por explorar (nodos abiertos).

De partida, la cola estará constituida sólo por el nodo raíz (C); una vez cerrado éste, se conformará la cola con sus ramificaciones (B, D). Tras analizar el nodo B, se añadirán a la estructura las ramificaciones de éste, quedando como D, G, A, F. Al cerrar el nivel completo (nodo D), se adosará el nodo E como ramificación del recién examinado (D), pasando la cola a tener la forma G, A, F, E. El proceso continuará con el análisis del nodo G, añadiéndose a la cola la hoja H (A, F, E, H). Finalmente, se efectúa el análisis de las hojas A, F y E, que como tales no generan ningún elemento adicional en la cola, localizándose la solución tras cerrar la hoja E, pasándose a analizar la H.

En este caso la búsqueda por niveles no nos ha reportado una excesiva ventaja, dado que la casualidad ha llevado la solución preci-



— 4.7.10.

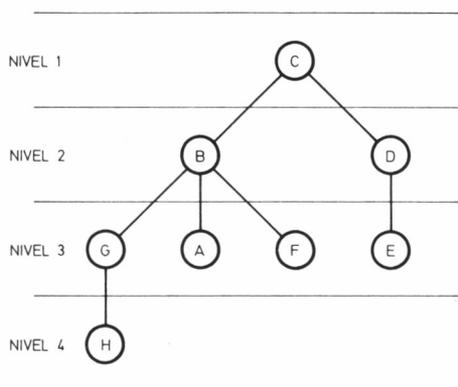


■ 4.7.11. Representación en forma de árbol del autómata finito empleado por el programa POKER.

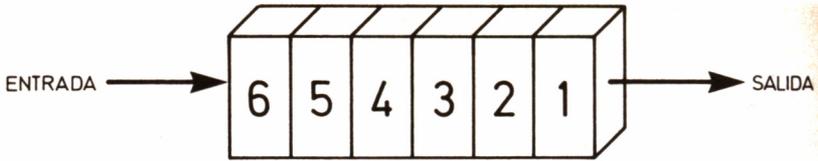
samente a la última hoja analizada, no obstante, la ventaja de este tipo de estructura se habría hecho evidente si la solución se hubiera encontrado, por ejemplo, en el nodo D (nos habríamos ahorrado el análisis de 5 ramas); y vaya, ¡no siempre vamos a tener tan mala suerte!

## BÚSQUEDA EN PROFUNDIDAD

En este método, la búsqueda se efectúa siguiendo las ramificaciones que de nivel a nivel generan un determinado nodo, hasta llegar a



■ 4.7.12. Distribución de un árbol para la búsqueda por niveles.

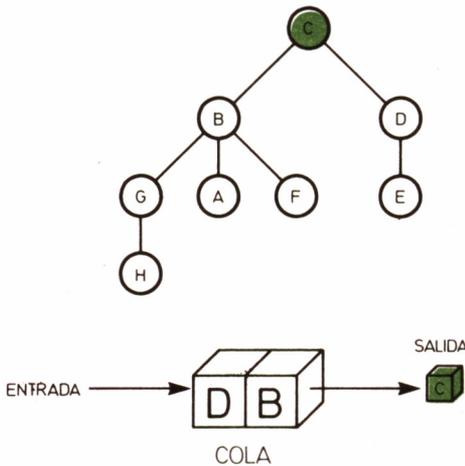


4.7.13. La cola es la estructura que gestiona los sistemas FIFO (First In, First out, primero en entrar, primero en salir).

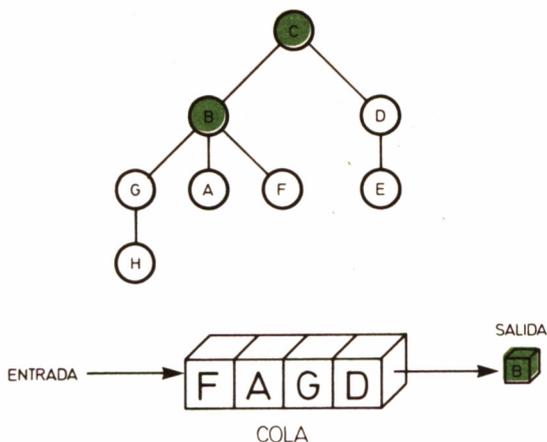
una hoja, o por supuesto, a la solución. Supongamos para la ilustración del problema, un nuevo caso, en el cual quisiéramos llegar de C a E. El orden en que se cerrarían los nodos sería el siguiente: C, D, G, H, A, F, D, E.

La gestión de este tipo de estructura se consigue gracias a una pila o *stack*, en el cual las entradas y salidas de elementos se efectúan siguiendo un sistema LIFO (*Last In, First Out*, el último en entrar es el primero en salir). Para una mayor comprensión del método, analicemos gráficamente el nuevo problema para pasar de C a E.

Cerrando la raíz, la pila estará constituida por B, D. Una vez examinado el nodo B, antes de continuar con el mismo nivel (D) se profundizará, anteponiendo las ramas de B (G, A, F) al resto de la pila, con lo que ésta tomará la forma G, A, F, D. Una vez cerrado el nodo C, se procederá de forma similar, primando en el orden de análisis su hoja H, con lo que la pila adoptará la forma H, A, F, D.

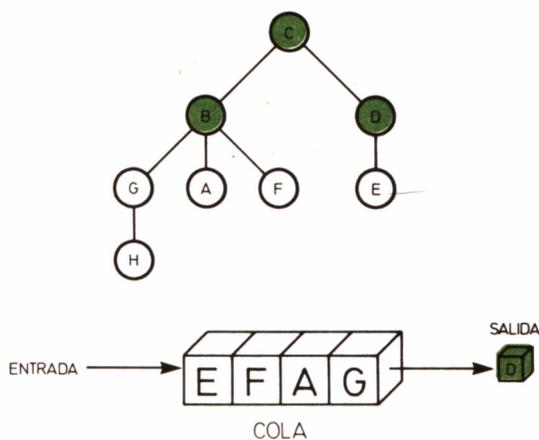


4.7.14. Al cerrar la raíz, se añaden a la cola los nodos B y D.

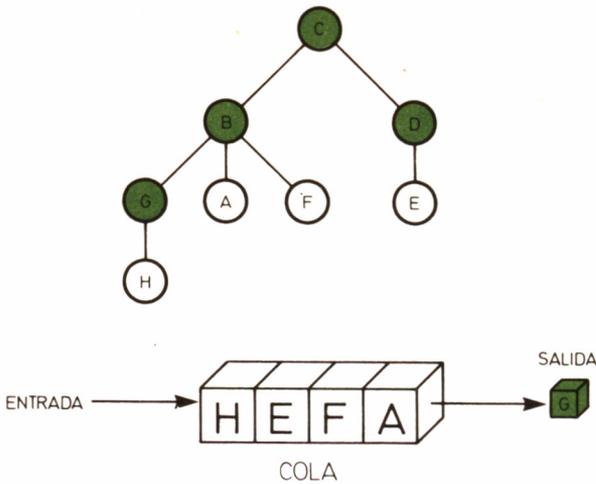


4.7.15. Al cerrarse el nodo B, se añaden a la cola el nodo G y las hojas A y F.

Como en el caso de los análisis por niveles, las hojas no producen ninguna entrada en la pila, de forma que H, A, F irán cerrándose y decrementando la longitud de ésta: A, F, D - F, D - D. Finalmente, al cerrar el nodo D, se añadirá su hoja H, que ahora será el único elemento de la pila, coincidiendo además con la solución (figs. 4.7.23 a 4.7.30).

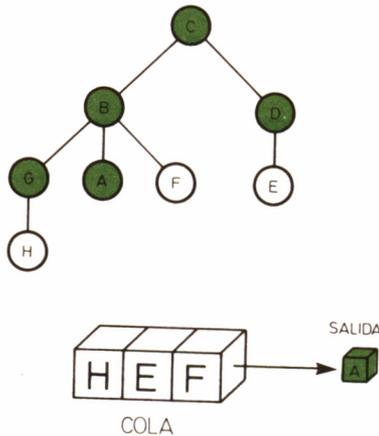


4.7.16. Cuando se cierra el nodo D, pasa a la cola su hoja E.

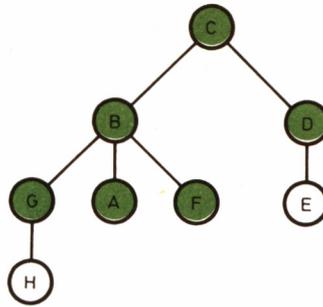


4.7.17. El cierre del nodo G, supone la adición a la cola de su hoja H: la solución.

De nuevo, en el ejemplo hemos tenido que investigar todo el árbol para llegar a la solución, pero esto ha sido de manera intencionada y por puras razones pedagógicas, dado que si hubiéramos empleado el sistema de búsqueda en profundidad sobre el ejemplo original (de C a H), la solución habría sido localizada en un tiempo récord: C, B, G,

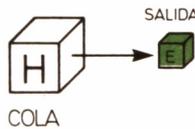
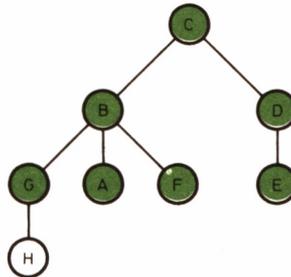


4.7.18. El cierre de A, por su condición de hoja, no implica la suma a la cola de ningún nuevo estado.

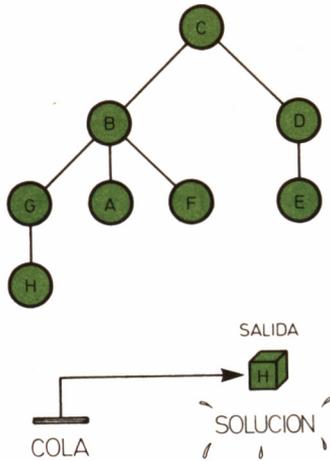


4.7.19. La hoja F decremanta nuevamente la longitud de la cola de búsqueda.

H, al ser ésta precisamente la primera ramificación en analizarse. En ocasiones, la profundidad que puede alcanzar un árbol, como por ejemplo, en un juego de ajedrez, es tan grande, que en la búsqueda de la solución por este método, el programador se ve obligado a estipular una determinada profundidad máxima, para conseguir llegar a una posi-



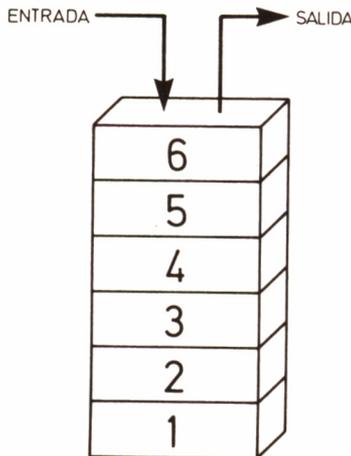
4.7.20. La hoja E finaliza el análisis del tercer nivel.



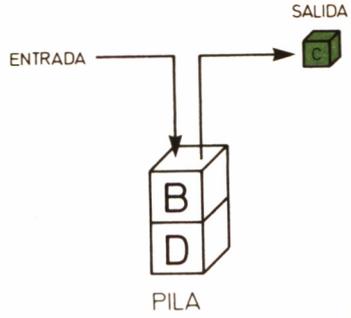
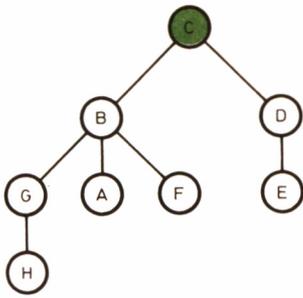
4.7.21. Por último, el análisis de H finaliza la búsqueda; en primer lugar, por ser la hoja-destino, y en segundo lugar, por agotamiento de la cola de búsqueda.

ble solución, o más correctamente, a la más válida de las posibles actuaciones en un tiempo razonable. Este tipo de restricción propicia que en bastantes casos la solución quede descartada por encontrarse más abajo de la profundidad máxima admitida.

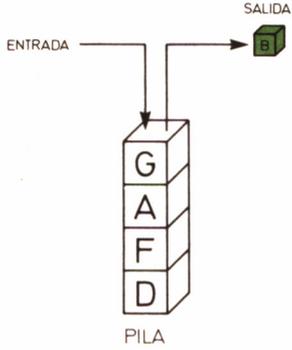
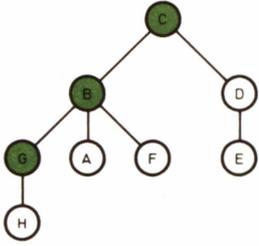
Así por ejemplo, en el gráfico correspondiente podemos observar



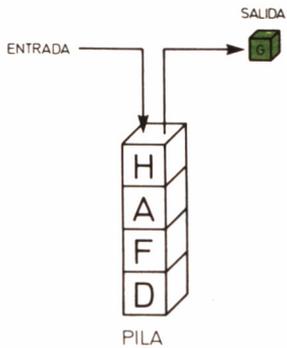
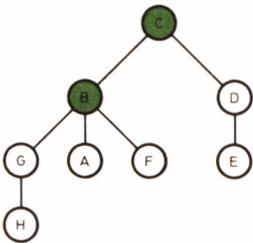
4.7.22. La pila es la estructura que gestiona los sistemas LIFO (Last In, First out, último en entrar, primero en salir).



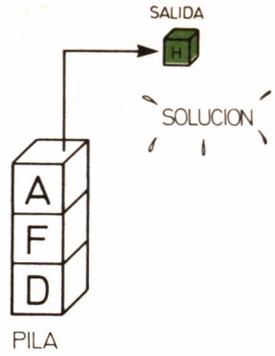
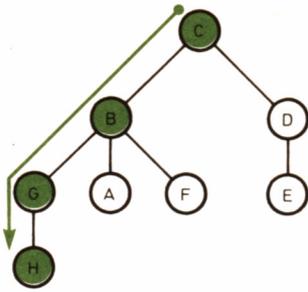
4.7.23.



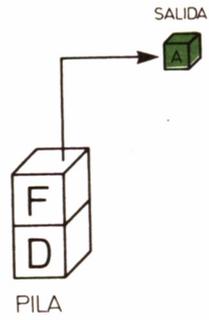
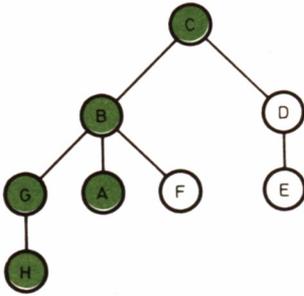
4.7.24.



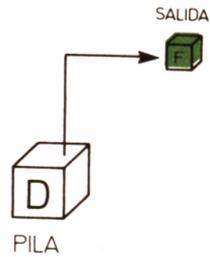
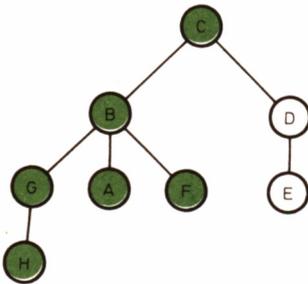
4.7.25.



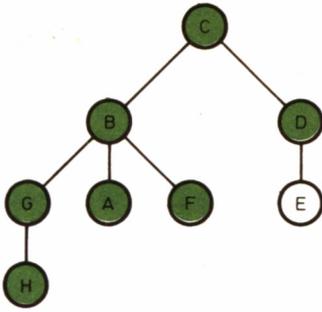
4.7.26.



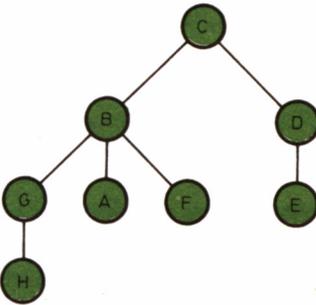
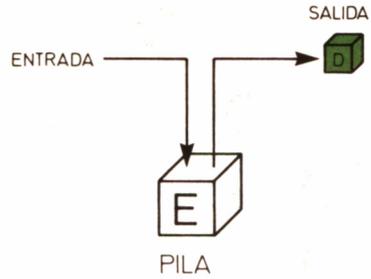
4.7.27.



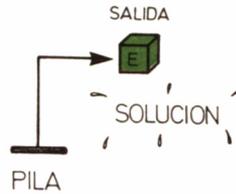
4.7.28.



4.7.29.



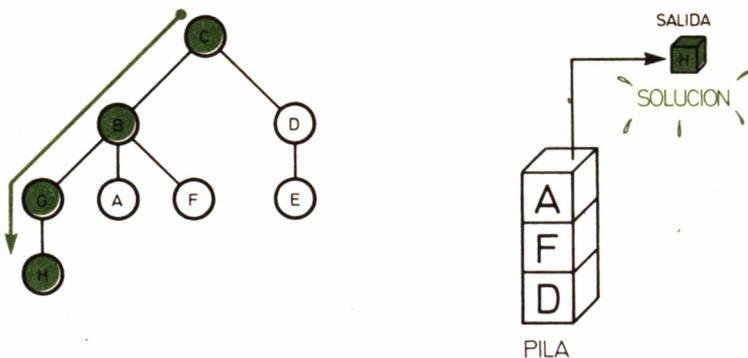
4.7.30.



como si en el problema de transporte C a H, se hubiera marcado una profundidad máxima de tres niveles, la solución no hubiera podido ser encontrada (fig. 4.7.32). No obstante, este método de «poda» de ramas se manifiesta de utilidad cuando la solución es múltiple. Así por ejemplo, en el juego de ajedrez se puede explorar por el sistema de profundidad los valores de las posibles jugadas hasta un cierto número de movimientos, para entonces elegir aquella de mayor valoración, lo cual no implica que sea la mejor, pero sí la «menos mala» de las analizadas hasta esa profundidad.

## METRO

Para que el estudio de los árboles no quede sólo en la teoría, vamos a aplicar nuestros conocimientos sobre un ejemplo práctico. El siguiente programa (METRO), nos permite crear una base de datos

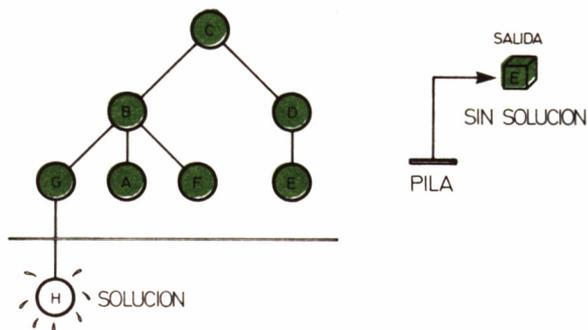


4.7.31. El sistema en profundidad sería el más rápido en la solución del problema de transporte de C a H.

sobre una red de comunicaciones; en el ejemplo, hemos escogido la red metropolitana de Madrid.

En el menú principal, podremos optar entre acceder a la gestión de la base de datos o efectuar una consulta, para lo cual deberá hallarse previamente en memoria alguna red.

Las redes serán introducidas a través de la opción CREAR del menú de gestión, debiendo especificar el número de líneas que las componen, el número de puntos de la línea más larga, el nombre otorgado a cada una de ellas y su longitud y, finalmente, la denominación de todos y cada uno de los puntos que la componen. Posteriormente, los datos son almacenados en cinta o disco en forma de fiche-



4.7.32. El sistema de poda por limitación de profundidad, puede llevarnos a descartar la solución del problema.

ro secuencial, mediante la opción GRABAR del mismo menú, y retornados a la memoria al ejecutar el programa, gracias a la opción CARGAR.

Como podemos comprobar por su estructura, el programa ha sido inicialmente concebido para albergar una red tipo metropolitano, dado que no tiene en cuenta la distancia de separación entre cada punto de la red, sino el número de ellos y el cambio de líneas (transbordos). No obstante, podría ser adaptado sin gran esfuerzo a cualquier otro problema de transporte.

Por otra parte, sigue un sistema de búsqueda en paralelo, algo lento, pero eficaz. A la «poda» del árbol contribuye un dato aportado por el usuario: el número máximo de transbordos que se desean realizar. Cuanto menor sea éste, menor será el tiempo invertido por el programa en encontrar la solución, si es que existe.

Es necesario reseñar, que el programa carece del conocimiento de la situación de los puntos de la red, sabiendo únicamente la línea a la que pertenecen. Así pues, si tenemos en cuenta que busca el camino «a ciegas», podemos asegurar que su funcionamiento es bastante eficaz.

Como datos para una prueba, podemos utilizar la red del metro de Madrid, cuyos datos se aportan en el cuadro adjunto, pudiendo realizar posteriormente las adaptaciones pertinentes para nuestro problema específico.

#### 4.7.33. RED DE LA COMPAÑIA METROPOLITANA DE MADRID.

LÍNEA: 1 PLAZA DE CASTILLA - PORTAZGO

PLAZA DE CASTILLA  
VALDEACEDERAS  
TETUAN  
ESTRECHO  
ALVARADO  
CUATRO CAMINOS  
RIOS ROSAS  
IGLESIA  
BILBAO  
TRIBUNAL  
GRAN VIA  
SOL  
TIRSO DE MOLINA  
ANTON MARTIN  
ATOCHA  
MENENDEZ PELAYO  
PACIFICO  
PUENTE DE VALLECAS

NUEVA NUMANCIA  
PORTAZGO

LINEA: 2 VENTAS - CUATRO CAMINOS

VENTAS  
MANUEL BECERRA  
GOYA  
PRINCIPE DE VERGARA  
RETIRO  
BANCO DE ESPAÑA  
SEVILLA  
SOL  
OPERA  
SANTO DOMINGO  
PLAZA DE ESPAÑA - NOVICIADO  
SAN BERNARDO  
QUEVEDO  
CUATRO CAMINOS

LINEA: 3 MONCLOA - LEGAZPI

MONCLOA  
ARGUELLES  
VENTURA RODRIGUEZ  
PLAZA DE ESPAÑA - NOVICIADO  
CALLAO  
SOL  
LAVAPIES  
ACACIAS - EMBAJADORES  
PALOS DE LA FRONTERA  
DELICIAS  
LEGAZPI

LINEA: 4 ESPERANZA - ARGUELLES

ESPERANZA  
ARTURO SORIA  
AVENIDA DE LA PAZ  
ALFONSO XIII  
PROSPERIDAD  
AVENIDA DE AMERICA  
DIEGO DE LEON  
LISTA  
GOYA  
VELAZQUEZ  
SERRANO  
COLON  
ALONSO MARTINEZ  
BILBAO  
SAN BERNARDO  
ARGUELLES

LINEA: 5 CANILLEJAS - ALUCHE

CANILLEJAS  
TORRE ARIAS  
SUANZES  
CIUDAD LINEAL  
PUEBLO NUEVO  
QUINTANA  
EL CARMEN  
VENTAS  
DIEGO DE LEON  
NÚEZ DE BALBOA  
RUBEN DARIO  
ALONSO MARTINEZ  
CHUECA  
GRAN VIA  
CALLAO

OPERA  
LA LATINA  
PUERTA DE TOLEDO  
ACACIAS - EMBAJADORES  
PIRAMIDES  
MARQUES DE VADILLO  
URGEL  
OPORTO  
VISTA ALEGRE  
CARABANCHEL  
ALUCHE

LINEA: 6 LAGUNA - CUATRO CAMINOS

LAGUNA  
CARPETANA  
OPORTO  
OPAVEL  
PLAZA ELIPTICA  
USERA  
LEGAZPI  
MENDEZ ALVARO  
PACIFICO  
CONDE DE CASAL  
SAINZ DE BARANDA  
O'DONNELL  
MANUEL BECERRA  
DIEGO DE LEON  
AVENIDA DE AMERICA  
REPUBLICA ARGENTINA  
NUEVOS MINISTERIOS  
CUATRO CAMINOS

LINEA: 5 CANILLEJAS - ALUCHE

CANILLEJAS  
TORRE ARIAS  
SUANZES  
CIUDAD LINEAL  
PUEBLO NUEVO  
QUINTANA  
EL CARMEN  
VENTAS  
DIEGO DE LEON  
NUEZ DE BALBOA  
RUBEN DARIO  
ALONSO MARTINEZ  
CHUECA  
GRAN VIA  
CALLAO  
OPERA  
LA LATINA  
PUERTA DE TOLEDO  
ACACIAS - EMBAJADORES  
PIRAMIDES  
MARQUES DE VADILLO  
URGEL  
OPORTO  
VISTA ALEGRE  
CARABANCHEL  
ALUCHE

LINEA: 6 LAGUNA - CUATRO CAMINOS

LAGUNA  
CARPETANA  
OPORTO  
OPAVEL  
PLAZA ELIPTICA  
USERA

LEGAZPI  
MENDEZ ALVARO  
PACIFICO  
CONDE DE CASAL  
SAINZ DE BARANDA  
O'DONNELL  
MANUEL BECERRA  
DIEGO DE LEON  
AVENIDA DE AMERICA  
REPUBLICA ARGENTINA  
NUEVOS MINISTERIOS  
CUATRO CAMINOS

LINEA: 7 AVENIDA DE AMERICA - LAS MUSAS

AVENIDA DE AMERICA  
CARTAGENA  
PARQUE DE LAS AVENIDAS  
BARRIO DE LA CONCEPCION  
PUEBLO NUEVO  
ASCAO  
GARCIA NOBLEJAS  
SIMANCAS  
SAN BLAS  
LAS MUSAS

LINEA: 8 FUENCARRAL - NUEVOS MINISTERIOS

FUENCARRAL  
BEGOVA  
CHAMARTIN  
PLAZA DE CASTILLA  
CUZCO  
LIMA  
NUEVOS MINISTERIOS

LINEA: 9 PAVONES - HERRERA ORIA

PAVONES  
ARTILLEROS  
VINATEROS  
ESTRELLA  
SAINZ DE BARANDA  
IBIZA  
PRINCIPE DE VERGARA  
NUÑEZ DE BALBOA  
AVENIDA DE AMERICA  
CRUZ DEL RAYO  
CONCHA ESPINA  
COLOMBIA  
PIO XII  
DUQUE DE PASTRANA  
PLAZA DE CASTILLA  
VENTILLA  
BARRIO DEL PILAR  
HERRERA ORIA

LINEA: 10 ALUCHE - ALONSO MARTINEZ (SUBURBANO)

ALUCHE  
EMPALME  
CAMPAMENTO  
BATAN  
LAGO  
PLAZA DE ESPAÑA - NOVICIADO  
TRIBUNAL  
ALONSO MARTINEZ

LINEA: R OPERA - NORTE (RAMAL NORTE)

OPERA  
NORTE

4.7.34. Programa METRO versión CPC.

```
10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- METRO (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- CPC 464/472/664/6128 -----
50 /-----
60 /
70 MODE 1
80 LOCATE 13,8:PRINT "MENU PRINCIPAL"
90 PRINT:PRINT
100 PRINT TAB(4)'1.- GESTION DE LA BASE DE DATOS'
110 PRINT
120 PRINT TAB(4)'2.- CONSULTA DE UN PROBLEMA'
130 PRINT
140 PRINT TAB(4)'3.- SALIDA DE PROGRAMA'
150 x$=INKEY$
160 IF x$<'1' OR x$>'3' THEN 150
170 ON VAL(x$) GOTO 180,1170,1770
180 MODE 1
190 LOCATE 6,7:PRINT "GESTION DE LA BASE DE DATOS"
200 PRINT:PRINT
210 PRINT TAB(6)'1.- CREACION'
220 PRINT
230 PRINT TAB(6)'2.- CARGA'
240 PRINT
250 PRINT TAB(6)'3.- GRABACION'
260 PRINT
270 PRINT TAB(6)'4.- RETORNO AL MENU PRINCIPAL'
280 x$=INKEY$
290 IF x$<'1' OR x$>'4' THEN 280
300 ON VAL(x$) GOTO 310,880,1070,70
310 MODE 2
320 IF db=0 THEN 400
330 PRINT "ESTA EN MEMORIA LA RED "nombre$
340 PRINT
350 PRINT "?DESEA BORRAR ESTA RED?"
360 x$=UPPER$(INKEY$):IF x$<>'S' AND x$<>'N' THEN 360
370 IF x$='N' THEN 180
380 ERASE red$,lin$:db=0
390 PRINT
400 INPUT "?NOMBRE PARA LA BASE DE DATOS";nombre$
410 IF LEN(nombre$)=0 OR LEN(nombre$)>12 THEN PRINT "NOMBRE
INCORRECTO"CHR$(7):GOTO 400
420 PRINT
430 INPUT "?NUMERO DE LINEAS";lineas
440 IF lineas<1 THEN 180
450 lineas=INT(lineas)
460 PRINT
470 INPUT "?MAXIMA LONGITUD DE LINEA";longmax
480 IF longmax<1 THEN 180
490 longmax=INT(longmax)
```

```

500 PRINT
510 DIM red$(lineas,longmax)
520 PRINT STRING$(80,'*')
530 DIM lin$(lineas)
540 FOR i=1 TO lineas
550 PRINT "?NOMBRE DE LA LINEA'i;:INPUT lin$(i)
560 PRINT
570 NEXT
580 PRINT STRING$(80,'*')
590 FOR i=1 TO lineas
600 PRINT "?LONGITUD DE 'lin$(i)' ";:INPUT a:red$(i,0)=STR$(
a)
610 PRINT
620 NEXT
630 PRINT STRING$(80,'*')
640 FOR i=1 TO lineas
650 PRINT "LINEA: 'lin$(i)
660 PRINT:PRINT
670 FOR j=1 TO VAL(red$(i,0))
680 PRINT "?Nombre de la estacion'j;:INPUT red$(i,j)
690 PRINT
700 NEXT
710 PRINT:PRINT
720 NEXT
730 CLS
740 PRINT "RED: 'nombre$
750 PRINT:PRINT
760 FOR i=1 TO lineas
770 PRINT "LINEA: 'lin$(i)
780 PRINT
790 FOR j=1 TO VAL(red$(i,0))
800 PRINT red$(i,j)
810 NEXT
820 PRINT:PRINT
830 NEXT
840 PRINT "PULSA UNA TECLA PARA CONTINUAR"
850 WHILE INKEY$="" :WEND
860 db=1
870 GOTO 180
880 MODE 2
890 IF db=0 THEN 960
900 PRINT "ESTA EN MEMORIA LA RED 'nombre$
910 PRINT
920 PRINT "?DESEA BORRAR ESTA RED?"
930 x$=UPPER$(INKEY$):IF x$<>'S' AND x$<>'N' THEN 930
940 IF x$="N" THEN 180
950 ERASE red$,lin$:db=0
960 PRINT:INPUT"?NOMBRE DE LA RED";nombre$:IF LEN(nombre$)=0
OR LEN(nombre$)>12 THEN 950
970 PRINT:PRINT:OPENIN nombre$:INPUT#9,lineas:INPUT#9,longmax
x:DIM red$(lineas,longmax),lin$(lineas)
980 FOR i=1 TO lineas:INPUT#9,lin$(i):NEXT
990 FOR i=1 TO lineas
1000 PRINT "LINEA: 'lin$(i):PRINT
1010 INPUT#9,red$(i,0)
1020 FOR j=1 TO VAL(red$(i,0))
1030 INPUT#9,red$(i,j)
1040 PRINT red$(i,j)
1040 PRINT red$(i,j)
1050 NEXT:PRINT:NEXT
1060 CLOSEIN:db=1:GOTO 180
1070 MODE 2

```

```

1080 PRINT 'GRABANDO LA RED 'nombre$:PRINT:PRINT
1090 OPENOUT nombre$
1100 PRINT#9, lineas:PRINT#9, longmax
1110 FOR i=1 TO lineas:PRINT#9, lin$(i):NEXT
1120 FOR i=1 TO lineas
1130 PRINT 'LINEA: 'lin$(i):PRINT:PRINT#9, red$(i,0)
1140 FOR j=1 TO VAL(red$(i,0)):PRINT#9, red$(i,j):PRINT red$(
i,j)
1150 NEXT:PRINT:NEXT
1160 CLOSEOUT:GOTO 180
1170 CLS
1180 IF db=0 THEN PRINT 'NO HAY BASE DE DATOS EN MEMORIA':GO
TO 1740
1190 INPUT '?ESTACION DE SALIDA';s$:s$=UPPER$(s$)
1200 FOR i=1 TO lineas
1210 FOR j=1 TO VAL(red$(i,0))
1220 IF red$(i,j)=s$ THEN ls=i:es=j:GOTO 1250
1230 NEXT j,i
1240 PRINT:PRINT 'NO CONOZCO ESA ESTACION':PRINT CHR$(7):GOT
O 1190
1250 PRINT
1260 INPUT '?ESTACION DE DESTINO';d$:d$=UPPER$(d$)
1270 FOR i=1 TO lineas
1280 FOR j=1 TO VAL(red$(i,0))
1290 IF red$(i,j)=d$ THEN ld=i:ed=j:GOTO 1320
1300 NEXT j,i
1310 PRINT:PRINT 'NO CONOZCO ESA ESTACION':PRINT CHR$(7):GOT
O 1260
1320 PRINT
1330 INPUT '?MAXIMO TRANSBORDO';trans
1340 PRINT
1350 IF s$=d$ THEN PRINT 'NO SE PRODUCE MOVIMIENTO':GOTO 174
0
1360 DIM c$(500,4)
1370 puntero=0:long=1
1380 c$(1,0)=STR$(0):c$(1,1)=STR$(ls):c$(1,2)=STR$(es):c$(1,
3)=STR$(0):c$(1,4)=' '
1390 puntero=puntero+1
1400 la=VAL(c$(puntero,1)):ea=VAL(c$(puntero,2))
1410 IF red$(la,ea)=d$ THEN 1620
1420 add=0
1430 mi=1:mf=lineas:IF VAL(c$(puntero,0))=trans AND puntero<
>1 THEN mi=la:mf=la
1440 FOR i=mi TO mf
1450 FOR j=1 TO VAL(red$(i,0))
1460 IF red$(i,j)<>red$(la,ea) THEN 1590
1470 IF i=la AND VAL(c$(puntero,3))=-1 THEN 1530
1480 IF j>=VAL(red$(i,0)) THEN 1530
1490 inc=1:FOR k=1 TO long:IF i=VAL(c$(k,1)) AND j+1=VAL(c$(
k,2)) THEN k=long:inc=0
1500 NEXT:IF inc=0 THEN 1530
1510 long=long+1:add=add+1:c$(long,0)=c$(puntero,0):c$(long,
1)=STR$(i):c$(long,2)=STR$(j+1):c$(long,3)=STR$(+1):c$(long,
4)=c$(puntero,4)+STR$(add)
1520 IF i<>la AND puntero<>1 THEN c$(long,0)=STR$(VAL(c$(pun
tero,0))+1)
1530 IF i=la AND VAL(c$(puntero,3))=+1 THEN 1590
1540 IF j<=1 THEN 1590
1550 inc=1:FOR k=1 TO long:IF i=VAL(c$(k,1)) AND j-1=VAL(c$(
k,2)) THEN k=long:inc=0
1560 NEXT:IF inc=0 THEN 1590
1570 long=long+1:add=add+1:c$(long,0)=c$(puntero,0):c$(long,

```

```

1)=STR$(i):c$(long,2)=STR$(j-1):c$(long,3)=STR$(-1):c$(long,
4)=c$(puntero,4)+STR$(add)
1580 IF i<>1a AND puntero<>1 THEN c$(long,0)=STR$(VAL(c$(pun
terero,0))+1)
1590 NEXT j,i
1600 IF puntero<long THEN 1390
1610 PRINT:PRINT "NO HA SIDO POSIBLE ENCONTRAR LA SOLUCION":
GOTO 1730
1620 MODE 2
1630 PRINT "EL TRAYECTO ES:":PRINT
1640 PRINT red$(VAL(c$(1,1)),VAL(c$(1,2)))
1650 mi=1:mf=2:mfb=1
1660 mfb=INSTR(mf,c$(puntero,4),' ')
1670 IF mfb=0 THEN PRINT d$:GOTO 1730
1680 FOR j=mi TO long
1690 IF LEFT$(c$(j,4),mfb)+' '=LEFT$(c$(puntero,4),mfb) THEN
PRINT red$(VAL(c$(j,1)),VAL(c$(j,2))):mi=j+1:j=long
1700 NEXT
1710 mf=mfb+1
1720 GOTO 1660
1730 ERASE c$
1740 PRINT:PRINT "PULSA UNA TECLA PARA CONTINUAR"
1750 WHILE INKEY$="" :WEND
1760 GOTO 70
1770 MODE 2
1780 END

```

#### 4.7.35. Programa METRO versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- METRO (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 in$=CHR$(27)+'f':ac$=CHR$(27)+'e':PRINT in$:kls$=CHR$(27)
+'E'+CHR$(27)+'H':DEF FN locate$(x,y)=CHR$(27)+'Y'+CHR$(32+x
)+CHR$(32+y):pq$=CHR$(27)+'x':gr$=CHR$(27)+'y'
90 PRINT pq$:kls$
100 PRINT FN locate$(8,31)*MENU PRINCIPAL*
110 PRINT:PRINT
120 PRINT TAB(24)*1.- GESTION DE LA BASE DE DATOS*
130 PRINT
140 PRINT TAB(24)*2.- CONSULTA DE UN PROBLEMA*
150 PRINT
160 PRINT TAB(24)*3.- SALIDA DE PROGRAMA*
170 x$=INKEY$
180 IF x$<'1' OR x$>'3' THEN 170
190 ON VAL(x$) GOTO 200,1190,1790
200 PRINT kls$
210 PRINT FN locate$(8,25)*GESTION DE LA BASE DE DATOS*
220 PRINT:PRINT
230 PRINT TAB(28)*1.- CREACION*
240 PRINT
250 PRINT TAB(28)*2.- CARGA*

```

```

260 PRINT
270 PRINT TAB(28)*3.- GRABACION*
280 PRINT
290 PRINT TAB(28)*4.- RETORNO AL MENU PRINCIPAL*
300 x$=INKEY$
310 IF x$<'1' OR x$>'4' THEN 300
320 ON VAL(x$) GOTO 330,900,1090,90
330 PRINT kls$
340 IF db=0 THEN 420
350 PRINT'ESTA EN MEMORIA LA RED 'nombre$
360 PRINT
370 PRINT'?DESEA BORRAR ESTA RED?'
380 x$=UPPER$(INKEY$):IF x$<>'S' AND x$<>'N' THEN 380
390 IF x$='N' THEN 200
400 ERASE red$,lin$:db=0
410 PRINT
420 PRINT ac$;;INPUT '?NOMBRE PARA LA BASE DE DATOS';nombre$
:PRINT in$;
430 IF LEN(nombre$)=0 OR LEN(nombre$)>12 THEN PRINT 'NOMBRE
INCORRECTO'CHR$(7):GOTO 420
440 PRINT
450 PRINT ac$;;INPUT '?NUMERO DE LINEAS';lineas:PRINT in$;
460 IF lineas<1 THEN 200
470 lineas=INT(lineas)
480 PRINT
490 PRINT ac$;;INPUT '?MAXIMA LONGITUD DE LINEA';longmax:PRI
NT in$;
500 IF longmax<1 THEN 200
510 longmax=INT(longmax)
520 PRINT
530 DIM red$(lineas,longmax)
540 PRINT STRING$(80,'*')
550 DIM lin$(lineas)
560 FOR i=1 TO lineas
570 PRINT ac$;;PRINT '?NOMBRE DE LA LINEA'i;;INPUT lin$(i):P
RINT in$;
580 PRINT
590 NEXT
600 PRINT STRING$(80,'*')
610 FOR i=1 TO lineas
620 PRINT '?LONGITUD DE 'lin$(i)' ';;PRINT ac$;;INPUT a:PRIN
T in$;;red$(i,0)=STR$(a)
630 PRINT
640 NEXT
650 PRINT STRING$(80,'*')
660 FOR i=1 TO lineas
670 PRINT 'LINEA: 'lin$(i)
680 PRINT:PRINT
690 FOR j=1 TO VAL(red$(i,0))
700 PRINT ac$;;PRINT '?Nombre de la estacion'j;;INPUT red$(i
,j):PRINT in$;
710 PRINT
720 NEXT
730 PRINT:PRINT
740 NEXT
750 PRINT kls$
760 PRINT 'RED: 'nombre$
770 PRINT:PRINT
780 FOR i=1 TO lineas
790 PRINT 'LINEA: 'lin$(i)
800 PRINT

```

```

810 FOR j=1 TO VAL(red$(i,0))
820 PRINT red$(i,j)
830 NEXT
840 PRINT:PRINT
850 NEXT
860 PRINT 'PULSA UNA TECLA PARA CONTINUAR'
870 WHILE INKEY$="":WEND
880 db=1
890 GOTO 200
900 PRINT kls$
910 IF db=0 THEN 980
920 PRINT 'ESTA EN MEMORIA LA RED 'nombre$
930 PRINT
940 PRINT "?DESEA BORRAR ESTA RED? (S/N)"
950 x$=UPPER$(INKEY$):IF x$<>'S' AND x$<>'N' THEN 950
960 IF x$="N" THEN 200
970 ERASE red$,lin$:db=0
980 PRINT ac$;:PRINT:INPUT"?NOMBRE DE LA RED";nombre$:PRINT
in$;:IF LEN(nombre$)=0 OR LEN(nombre$)>12 THEN 970
990 PRINT:PRINT:OPEN"I",1,nombre$:INPUT #1,lineas:INPUT #1,
ongmax:DIM red$(lineas,longmax),lin$(lineas)
1000 FOR i=1 TO lineas:INPUT #1,lin$(i):NEXT
1010 FOR i=1 TO lineas
1020 PRINT "LINEA: "lin$(i):PRINT
1030 INPUT #1,red$(i,0)
1040 FOR j=1 TO VAL(red$(i,0))
1050 INPUT #1,red$(i,j)
1060 PRINT red$(i,j)
1070 NEXT:PRINT:NEXT
1080 CLOSE:db=1:GOTO 200
1090 PRINT kls$
1100 PRINT "GRABANDO LA RED "nombre$:PRINT:PRINT
1110 OPEN"O",1,nombre$
1120 PRINT #1,lineas:PRINT #1,longmax
1130 FOR i=1 TO lineas:PRINT #1,lin$(i):NEXT
1140 FOR i=1 TO lineas
1150 PRINT "LINEA: "lin$(i):PRINT:PRINT #1,red$(i,0)
1160 FOR j=1 TO VAL(red$(i,0)):PRINT #1,red$(i,j):PRINT red$
(i,j)
1170 NEXT:PRINT:NEXT
1180 CLOSE:GOTO 200
1190 PRINT kls$
1200 IF db=0 THEN PRINT "NO HAY BASE DE DATOS EN MEMORIA":GO
TO 1760
1210 PRINT ac$;:INPUT "?ESTACION DE SALIDA";s$:PRINT in$;:s$
=UPPER$(s$)
1220 FOR i=1 TO lineas
1230 FOR j=1 TO VAL(red$(i,0))
1240 IF red$(i,j)=s$ THEN ls=i:es=j:GOTO 1270
1250 NEXT j,i
1260 PRINT:PRINT "NO CONOZCO ESA ESTACION":PRINT CHR$(7):GOT
O 1210
1270 PRINT
1280 PRINT ac$;:INPUT "?ESTACION DE DESTINO";d$:PRINT in$;:d
$=UPPER$(d$)
1290 FOR i=1 TO lineas
1300 FOR j=1 TO VAL(red$(i,0))
1310 IF red$(i,j)=d$ THEN ld=i:ed=j:GOTO 1340
1320 NEXT j,i
1330 PRINT:PRINT "NO CONOZCO ESA ESTACION":PRINT CHR$(7):GOT
O 1280

```

```

1340 PRINT
1350 PRINT ac$;:INPUT "?MAXIMO TRANSBORDO";trans:PRINT in$;
1360 PRINT
1370 IF s$=d$ THEN PRINT "NO SE PRODUCE MOVIMIENTO":GOTO 176
0
1380 DIM c$(500,4)
1390 puntero=0:long=1
1400 c$(1,0)=STR$(0):c$(1,1)=STR$(1s):c$(1,2)=STR$(es):c$(1,
3)=STR$(0):c$(1,4)=" "
1410 puntero=puntero+1
1420 la=VAL(c$(puntero,1)):ea=VAL(c$(puntero,2))
1430 IF red$(la,ea)=d$ THEN 1640
1440 add=0
1450 mi=1:mf=lines:IF VAL(c$(puntero,0))=trans AND puntero<
>1 THEN mi=la:mf=la
1460 FOR i=mi TO mf
1470 FOR j=1 TO VAL(red$(i,0))
1480 IF red$(i,j)<>red$(la,ea) THEN 1610
1490 IF i=la AND VAL(c$(puntero,3))=-1 THEN 1550
1500 IF j>=VAL(red$(i,0)) THEN 1550
1510 inc=1:FOR k=1 TO long:IF i=VAL(c$(k,1)) AND j+1=VAL(c$(
k,2)) THEN k=long:inc=0
1520 NEXT:IF inc=0 THEN 1550
1530 long=long+1:add=add+1:c$(long,0)=c$(puntero,0):c$(long,
1)=STR$(i):c$(long,2)=STR$(j+1):c$(long,3)=STR$(+1):c$(long,
4)=c$(puntero,4)+STR$(add)
1540 IF i<>la AND puntero<>1 THEN c$(long,0)=STR$(VAL(c$(pun
tero,0))+1)
1550 IF i=la AND VAL(c$(puntero,3))=+1 THEN 1610
1560 IF j<=1 THEN 1610
1570 inc=1:FOR k=1 TO long:IF i=VAL(c$(k,1)) AND j-1=VAL(c$(
k,2)) THEN k=long:inc=0
1580 NEXT:IF inc=0 THEN 1610
1590 long=long+1:add=add+1:c$(long,0)=c$(puntero,0):c$(long,
1)=STR$(i):c$(long,2)=STR$(j-1):c$(long,3)=STR$(-1):c$(long,
4)=c$(puntero,4)+STR$(add)
1600 IF i<>la AND puntero<>1 THEN c$(long,0)=STR$(VAL(c$(pun
tero,0))+1)
1610 NEXT j,i
1620 IF puntero<long THEN 1410
1630 PRINT:PRINT "NO HA SIDO POSIBLE ENCONTRAR LA SOLUCION":
GOTO 1750
1640 PRINT k1s$
1650 PRINT "EL TRAYECTO ES:":PRINT
1660 PRINT red$(VAL(c$(1,1)),VAL(c$(1,2)))
1670 mi=1:mf=2:mfb=1
1680 mfb=INSTR(mf,c$(puntero,4)," ")
1690 IF mfb=0 THEN PRINT d$:GOTO 1750
1700 FOR j=mi TO long
1710 IF LEFT$(c$(j,4),mfb)+ " " =LEFT$(c$(puntero,4),mfb) THEN
PRINT red$(VAL(c$(j,1)),VAL(c$(j,2))):mi=j+1:j=long
1720 NEXT
1730 mf=mfb+1
1740 GOTO 1680
1750 ERASE c$
1760 PRINT:PRINT "PULSA UNA TECLA PARA CONTINUAR"
1770 WHILE INKEY$="" :WEND
1780 GOTO 90
1790 PRINT k1s$
1800 PRINT ac$:PRINT gr$:END

```

## OTROS MÉTODOS DE BÚSQUEDA

Al comienzo de este capítulo, mencionamos algunos otros métodos de búsqueda; éstos, dada su complejidad, serán tratados sólo superficialmente.

El primero de ellos es el de búsqueda ordenada, en el cual se sigue básicamente el sistema de análisis en profundidad, aunque en combinación con una determinada regla heurística (nacida de la experiencia), que permite optimizar el orden en que se analizan las diferentes generaciones. Estas reglas pueden guiarse por la teoría *Hill climbing* o escalado de colinas, según la cual el ordenamiento de los nodos a analizar se decide en base a la distancia que resta. Su nombre proviene del símil que podemos encontrar en un escalador de montañas: éste elige el camino a seguir desde un determinado punto de la colina, en base a un cálculo de la distancia restante hasta la cumbre, escogiendo así el camino más corto.

Este tipo de regla también puede estar orientada por otros criterios inherentes a los nodos. Así, por ejemplo, nuestro escalador puede ser que considere, además de la distancia que resta, la comodidad del camino a seguir.

Finalmente, trataremos dos algoritmos de búsqueda utilizados con gran frecuencia en los juegos de tablero, como el ajedrez, las damas, las tres en raya o el othelo: minimax y alfa-beta.

Minimax se basa en asignar un cierto valor a los diferentes nodos que va analizando, lo cual se consigue a partir de una función de evaluación. Dichas funciones de evaluación asignan un cierto valor a una situación o movimiento según determinados criterios. Por ejemplo, el juego del Othelo que finaliza este capítulo evalúa en base a dos circunstancias: el valor estratégico de una posición y el número de piezas que se capturan en el movimiento ( $V_t = 63 * V_p + N_p$ ). Finalmente, el algoritmo minimax opta por aquella situación en que, obteniendo el máximo rendimiento, reduzcamos al mínimo el posible provecho del adversario.

Así pues, el valor definitivo de una jugada, vendría dado por la función de evaluación aplicada a nuestra jugada, menos el valor del movimiento que este movimiento podría propiciar al adversario.

Cuanta mayor sea la profundidad a la que apliquemos el sistema minimax, mejor será el resultado obtenido; ahora bien, esto llevado a su último extremo, se puede convertir en un trabajo tan exhaustivo que tarde una eternidad en determinados problemas de relativa com-

plejidad. Este es el momento de aplicar el algoritmo alfa-beta, para «podar» el árbol.

Llamaremos alfa al mejor valor que encontremos para nuestra jugada y beta al mejor de los valores de los posibles movimientos por los que puede optar el humano. Según investiguemos, abandonaremos todas aquellas ramas que se generen de un nodo cuyo alfa sea menor que el actual. Y del mismo modo, considerando que el jugador humano operará de forma similar, abandonando aquellas posibilidades de escaso valor, dejaremos de investigar las ramas que disminuyan el valor de beta, cuando se esté evaluando la posible respuesta que el jugador pueda dar a nuestra jugada.

## OTHELLO

Bajo esta denominación o la de REVERSI se conoce uno de los grandes juegos de mesa de relativamente reciente aparición, al menos comparado con el ajedrez o las damas. Aunque su complejidad no llega a alcanzar la del «juego de los juegos», el ajedrez, en practicantes con no demasiada experiencia, el othello se puede convertir en uno de esos ejercicios mentales que crea adicción.

Aunque sus normas son bastante simples, y serán conocidas por muchos de nosotros, no estará de más que les demos un pequeño repaso.

Se trata de un juego de dos contendientes, aunque en nuestro caso concreto uno de ellos será un Amstrad, dirigiendo uno las fichas blancas y otro las negras, que estarán representadas en la pantalla mediante un círculo (O) y un aspa (X), respectivamente. Antes de comenzar la partida, se efectuará un sorteo de qué color corresponderá a cada jugador, siendo el representante de las negras el que realizará el primer movimiento.

El tablero de juego está constituido por una red de  $8 \times 8$  casillas, apareciendo cuatro fichas en el cuadrado central del tablero (dos blancas y dos negras, en diagonal). A partir de este momento, ambos jugadores van turnándose en la colocación de fichas, excepto en el caso que uno de ellos no pueda situar ninguna ficha que, al menos, capture una del rival: en esta situación el jugador pierde su turno, que no recuperará hasta que le sea posible realizar una captura en su jugada.

Cabe recordar, que las fichas sólo pueden colocarse en casillas vacías y que cuando un jugador sitúa una ficha, captura todas las del

oponente que se encuentren en línea recta, entre la ficha colocada y otro de su mismo color. Capturar una ficha, equivale a cambiar su color convirtiéndola en propia.

La jugada se comunica al ordenador introduciéndole las coordenadas (letra y número) de la posición en la que se coloca la ficha. Al igual que en su equivalente de sobremesa, el juego puede darse por terminado en varias situaciones: cuando el tablero se llena, cuando a uno de los jugadores no le queda ninguna ficha sobre el tablero, o bien cuando ambos jugadores se ven imposibilitados para realizar una jugada. Estos dos últimos extremos son realmente hartamente improbables. En tal caso, se cuenta el número de fichas y resultará vencedor el contendiente con más fichas de su color en el tablero.

#### 4.7.36. Programa OTHELLO versión CPC.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- OTHELLO (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- CPC 464/472/664/6128 -----
50 /-----
60 /
70 DIM v$(9,9),f$(8,8),t$(9,9),p(8,8)
80 FOR i=1 TO 8
90 FOR j=1 TO 8
100 READ x
110 p(i,j)=x
120 NEXT j,i
130 DATA 9,6,8,7,7,8,6,9
140 DATA 6,3,4,5,5,4,3,6
150 DATA 8,4,2,1,1,2,4,8
160 DATA 7,5,1,0,0,1,5,7
170 DATA 7,5,1,0,0,1,5,7
180 DATA 8,4,2,1,1,2,4,8
190 DATA 6,3,4,5,5,4,3,6
200 DATA 9,6,8,7,7,8,6,9
210 v$(4,4)="0":v$(5,5)="0"
220 v$(4,5)="X":v$(5,4)="X"
230 fh=2:fa=2
240 MODE 0
250 LOCATE 10,16:PRINT "O X":LOCATE 10,18:PRINT "X O"
260 LOCATE 3,8:FOR i=1 TO 8:PRINT STR$(i):NEXT
270 LOCATE 1,10:FOR i=1 TO 8:PRINT TAB(2)CHR$(64+i):PRINT:NE
XT
280 FOR i=0 TO 8:MOVE 80,8+i*32:DRAWR 512,0:NEXT
290 FOR i=0 TO 8:MOVE 80+i*64,8:DRAWR 2,256:NEXT
300 RANDOMIZE TIME
310 IF RND>0.5 THEN am$="X":hu$="0" ELSE am$="0":hu$="X"
320 am=ASC(am$):hu=ASC(hu$)
330 LOCATE 4,1:PRINT am$: 2':LOCATE 14,1:PRINT hu$: 2'
340 IF hu$="X" THEN 350 ELSE 570

```

```

350 m$=hu$:GOSUB 1010:GOSUB 2220:IF e=1 THEN LOCATE 4,4:PRIN
T 'T U P A S A S'CHR$(?):FOR i=0 TO 1499:NEXT:LOCATE 4,4:P
RINT SPACE$(15)CHR$(?):GOTO 570
360 LOCATE 14,4:PRINT '- 'CHR$(?)
370 x$=INKEY$:IF x$='' THEN 370 ELSE x$=UPPER$(x$)
380 LOCATE 14,3:PRINT x$
390 IF x$<'A' OR x$>'H' THEN 360
400 f=ASC(x$)-64
410 LOCATE 14,4:PRINT '- 'CHR$(?)
420 x$=INKEY$:IF x$='' THEN 420 ELSE x$=UPPER$(x$)
430 LOCATE 15,3:PRINT x$
440 IF x$<'1' OR x$>'8' THEN LOCATE 14,3:PRINT ' ':GOTO 360
450 c=VAL(x$)
460 LOCATE 15,4:PRINT ' '
470 GOSUB 1010
480 IF v$(f,c)<>' ' THEN 360
490 ff=f:cf=c:m$=hu$:vis=1
500 GOSUB 1210
510 IF fic=0 THEN 360
520 GOSUB 1060
530 LOCATE 14,3:PRINT ' ':LOCATE 14,4:PRINT ' '
540 fh=fh+fic+1:fa=fa-fic
550 LOCATE 6,1:PRINT fa':LOCATE 16,1:PRINT fh'
560 IF fa+fh=64 OR fa=0 THEN 960
570 m$=am$:GOSUB 1010:GOSUB 2220:IF e=1 THEN LOCATE 5,4:PRIN
T 'Y O P A S O'CHR$(?):FOR i=0 TO 1499:NEXT:LOCATE 5,4:PRI
NT SPACE$(13)CHR$(?):GOTO 350
580 vis=0
590 vo=-1E+17
600 FOR i=1 TO 64
610 f=INT((i-1)/8)+1:c=i-8*(f-1)
620 LOCATE 4,3:PRINT CHR$(64+f)MID$(STR$(c),2)
630 IF v$(f,c)<>' ' THEN 860
640 m$=am$:GOSUB 2300:IF e=1 THEN 860
650 GOSUB 1010
660 ff=f:cf=c
670 GOSUB 1210
680 IF fic=0 THEN 860
690 vp=63*p(f,c)+fic
700 GOSUB 1160
710 vf=0
720 FOR j=1 TO 64
730 ff=INT((j-1)/8)+1:cf=j-8*(ff-1)
740 LOCATE 14,3:PRINT CHR$(64+ff)MID$(STR$(cf),2)
750 IF t$(ff,cf)<>' ' THEN 820
760 m$=hu$:GOSUB 2300:IF e=1 THEN 820
770 GOSUB 1110
780 GOSUB 1210
790 IF fic=0 THEN 820
800 v=63*p(ff,cf)+fic
810 IF v>vf THEN vf=v
820 NEXT
830 vt=vp-vf
840 IF vt>vo THEN vo=vt:fo=f:co=c:GOTO 860
850 IF vt=vo AND RND>0.5 THEN vo=vt:fo=f:co=c
860 NEXT
870 LOCATE 4,3:PRINT CHR$(64+fo)MID$(STR$(co),2)CHR$(?):LOCA
TE 14,3:PRINT ' '
880 ff=fo:cf=co
890 m$=am$:vis=1
900 GOSUB 1010
910 GOSUB 1210

```

```

920 GOSUB 1060
930 fh=fh-fic:fa=fa+fic+1
940 LOCATE 6,1:PRINT fa' ':LOCATE 16,1:PRINT fh' '
950 IF fa+fh<64 AND fh>0 THEN 350
960 IF fa=fh THEN LOCATE 2,4:PRINT 'E M P A T A D O S'
970 IF fa>fh THEN LOCATE 2,4:PRINT 'H E G A N A D O'
980 IF fh>fa THEN LOCATE 2,4:PRINT 'V E N C I S T E !'
990 WHILE INKEY$='':WEND
1000 MODE 2:END
1010 FOR k=1 TO 8
1020 FOR l=1 TO 8
1030 f$(k,l)=v$(k,l)
1040 NEXT l,k
1050 RETURN
1060 FOR k=1 TO 8
1070 FOR l=1 TO 8
1080 v$(k,l)=f$(k,l)
1090 NEXT l,k
1100 RETURN
1110 FOR k=1 TO 8
1120 FOR l=1 TO 8
1130 f$(k,l)=t$(k,l)
1140 NEXT l,k
1150 RETURN
1160 FOR k=1 TO 8
1170 FOR l=1 TO 8
1180 t$(k,l)=f$(k,l)
1190 NEXT l,k
1200 RETURN
1210 fic=0:IF m$='X' THEN c$='0' ELSE c$='X'
1220 p=ff-1
1230 WHILE p>0
1240 IF f$(p,cf)=m$ THEN 1290
1250 IF f$(p,cf)='' THEN 1280
1260 p=p-1
1270 WEND
1280 GOTO 1340
1290 FOR k=ff-1 TO p+1 STEP -1
1300 f$(k,cf)=m$
1310 fic=fic+1
1320 IF vis=1 THEN LOCATE 2+2*cf,8+2*k:PRINT m$
1330 NEXT
1340 p=ff+1
1350 WHILE p<9
1360 IF f$(p,cf)=m$ THEN 1410
1370 IF f$(p,cf)='' THEN 1400
1380 p=p+1
1390 WEND
1400 GOTO 1460
1410 FOR k=ff+1 TO p-1
1420 f$(k,cf)=m$
1430 fic=fic+1
1440 IF vis=1 THEN LOCATE 2+2*cf,8+2*k:PRINT m$
1450 NEXT
1460 p=cf-1
1470 WHILE p>0
1480 IF f$(ff,p)=m$ THEN 1530
1490 IF f$(ff,p)='' THEN 1520
1500 p=p-1
1510 WEND
1520 GOTO 1580
1530 FOR k=cf-1 TO p+1 STEP -1

```

```

1540 f$(ff,k)=m$
1550 fic=fic+1
1560 IF vis=1 THEN LOCATE 2+2*k,8+2*ff:PRINT m$
1570 NEXT
1580 p=cf+1
1590 WHILE p<9
1600 IF f$(ff,p)=m$ THEN 1650
1610 IF f$(ff,p)=" THEN 1640
1620 p=p+1
1630 WEND
1640 GOTO 1700
1650 FOR k=cf+1 TO p-1
1660 f$(ff,k)=m$
1670 fic=fic+1
1680 IF vis=1 THEN LOCATE 2+2*k,8+2*ff:PRINT m$
1690 NEXT
1700 pf=ff-1:pc=cf-1
1710 WHILE pf>0 AND pc>0
1720 IF f$(pf,pc)=m$ THEN 1770
1730 IF f$(pf,pc)=" THEN 1760
1740 pf=pf-1:pc=pc-1
1750 WEND
1760 GOTO 1820
1770 FOR k=1 TO ff-pf-1
1780 f$(ff-k,cf-k)=m$
1790 fic=fic+1
1800 IF vis=1 THEN LOCATE 2+2*(cf-k),8+2*(ff-k):PRINT m$
1810 NEXT
1820 pf=ff-1:pc=cf+1
1830 WHILE pf>0 AND pc<9
1840 IF f$(pf,pc)=m$ THEN 1890
1850 IF f$(pf,pc)=" THEN 1880
1860 pf=pf-1:pc=pc+1
1870 WEND
1880 GOTO 1940
1890 FOR k=1 TO ff-pf-1
1900 f$(ff-k,cf+k)=m$
1910 fic=fic+1
1920 IF vis=1 THEN LOCATE 2+2*(cf+k),8+2*(ff-k):PRINT m$
1930 NEXT
1940 pf=ff+1:pc=cf-1
1950 WHILE pf<9 AND pc>0
1960 IF f$(pf,pc)=m$ THEN 2010
1970 IF f$(pf,pc)=" THEN 2000
1980 pf=pf+1:pc=pc-1
1990 WEND
2000 GOTO 2060
2010 FOR k=1 TO pf-ff-1
2020 f$(ff+k,cf-k)=m$
2030 fic=fic+1
2040 IF vis=1 THEN LOCATE 2+2*(cf-k),8+2*(ff+k):PRINT m$
2050 NEXT
2060 pf=ff+1:pc=cf+1
2070 WHILE pf<9 AND pc<9
2080 IF f$(pf,pc)=m$ THEN 2130
2090 IF f$(pf,pc)=" THEN 2120
2100 pf=pf+1:pc=pc+1
2110 WEND
2120 GOTO 2180
2130 FOR k=1 TO pf-ff-1
2140 f$(ff+k,cf+k)=m$
2150 fic=fic+1

```

```

2160 IF vis=1 THEN LOCATE 2+2*(cf+k),8+2*(ff+k):PRINT m$
2170 NEXT
2180 IF fic=0 THEN 2210
2190 f$(ff,cf)=m$
2200 IF vis=1 THEN LOCATE 2+2*cf,8+2*ff:PRINT m$
2210 RETURN
2220 e=1:vis=0
2230 FOR i=1 TO 8
2240 FOR j=1 TO 8
2250 IF f$(i,j)<>' ' THEN 2280
2260 ff=i:cf=j:GOSUB 1210
2270 IF fic>0 THEN e=0:j=8:i=8
2280 NEXT j,i
2290 RETURN
2300 e=1
2310 FOR k=-1 TO 1
2320 FOR l=-1 TO 1
2330 IF k=0 AND l=0 THEN 2380
2340 IF m$=hu$ THEN 2370
2350 IF v$(f+k,c+l)<>m$ AND v$(f+k,c+l)<>' ' THEN e=0:l=1:k=1
2360 GOTO 2380
2370 IF t$(ff+k,cf+l)<>m$ AND t$(ff+k,cf+l)<>' ' THEN e=0:l=1
:k=1
2380 NEXT l,k
2390 RETURN

```

#### 4.7.37. Programa OTHELLO versión PCW.

```

10 /-----
20 /----- FERNANDO LOPEZ MARTINEZ -----
30 /----- OTHELLO (c)1986 GRAN BIBLIOTECA AMSTRAD -----
40 /----- PCW 8256/8512 -----
50 /----- ADAPT. PCW CARLOS DE LA OSSA VILLACANAS -----
60 /-----
70 /
80 in$=CHR$(27)+'f':ac$=CHR$(27)+'e':PRINT in$:k1s$=CHR$(27)
+'E'+CHR$(27)+'H':DEF FN locate$(x,y)=CHR$(27)+'Y'+CHR$(32+x
)+CHR$(32+y):pq$=CHR$(27)+'x':gr$=CHR$(27)+'y'
90 PRINT k1s$:FN locate$(15,26)*PULSE UNA TECLA PARA COMENZA
R*:WHILE INKEY$='':time=1+(time MOD 1000):WEND
100 DIM v$(9,9),f$(8,8),t$(9,9),p(8,8),tb(4,33)
110 FOR i=1 TO 8
120 FOR j=1 TO 8
130 READ x
140 p(i,j)=x
150 NEXT j,i
160 DATA 9,6,8,7,7,8,6,9
170 DATA 6,3,4,5,5,4,3,6
180 DATA 8,4,2,1,1,2,4,8
190 DATA 7,5,1,0,0,1,5,7
200 DATA 7,5,1,0,0,1,5,7
210 DATA 8,4,2,1,1,2,4,8
220 DATA 6,3,4,5,5,4,3,6
230 DATA 9,6,8,7,7,8,6,9

```

```

240 v$(4,4)="0":v$(5,5)="0"
250 v$(4,5)="X":v$(5,4)="X"
260 fh=2:fa=2
270 PRINT pq$:kl$:GOSUB 2450
280 PRINT FN locate$(13,39)"0"FN locate$(13,43)"X"FN locate$(
(15,39)"X"FN locate$(15,43)"0"
290 PRINT FN locate$(5,26);:FOR i=1 TO 8:PRINT STR$(i);" ";
:NEXT
300 PRINT FN locate$(6,20);:FOR i=1 TO 8:PRINT TAB(23)CHR$(6
4+i):PRINT:NEXT
310 RANDOMIZE TIME
320 IF RND>0.5 THEN am$="X":hu$="0" ELSE am$="0":hu$="X"
330 am=ASC(am$):hu=ASC(hu$)
340 PRINT FN locate$(1,26);am$: 2*FN locate$(1,52)hu$: 2*
350 IF hu$="X" THEN 360 ELSE 580
360 m$=hu$:GOSUB 1020:GOSUB 2230:IF e=1 THEN PRINT FN locate
$(14,59)"T U P A S A S"CHR$(7):FOR i=0 TO 1499:NEXT:PRINT
FN locate$(14,59);SPACE$(15)CHR$(7):GOTO 580
370 PRINT FN locate$(3,52);" - "CHR$(7)
380 x$=INKEY$:IF x$="" THEN 380 ELSE x$=UPPER$(x$)
390 PRINT FN locate$(2,52);x$" "
400 IF x$<"A" OR x$>"H" THEN 370
410 f=ASC(x$)-64
420 PRINT FN locate$(3,52)" - "CHR$(7)
430 x$=INKEY$:IF x$="" THEN 430 ELSE x$=UPPER$(x$)
440 PRINT FN locate$(2,53);x$
450 IF x$<"1" OR x$>"8" THEN PRINT FN locate$(2,52)" ":GOTO
370
460 c=VAL(x$)
470 PRINT FN locate$(3,53)" "
480 GOSUB 1020
490 IF v$(f,c)<>" THEN 370
500 ff=f:cf=c:m$=hu$:vis=1
510 GOSUB 1220
520 IF fic=0 THEN 370
530 GOSUB 1070
540 PRINT FN locate$(2,52)" ":PRINT FN locate$(3,52)" "
550 fh=fh+fic+1:fa=fa-fic
560 PRINT FN locate$(1,28);fa" ":PRINT FN locate$(1,54);fh"
.
570 IF fa+fh=64 OR fa=0 THEN 970
580 m$=am$:GOSUB 1020:GOSUB 2230:IF e=1 THEN PRINT FN locate
$(14,59)"Y O P A S O"CHR$(7):FOR i=0 TO 1499:NEXT:PRINT FN
locate$(14,59);SPACE$(13)CHR$(7):GOTO 360
590 vis=0
600 vo=-1E+17
610 FOR i=1 TO 64
620 f=INT((i-1)/8)+1:c=i-8*(f-1)
630 PRINT FN locate$(2,26);CHR$(64+f)MID$(STR$(c),2)
640 IF v$(f,c)<>" THEN 870
650 m$=am$:GOSUB 2310:IF e=1 THEN 870
660 GOSUB 1020
670 ff=f:cf=c
680 GOSUB 1220
690 IF fic=0 THEN 870
700 vp=63*p(f,c)+fic
710 GOSUB 1170
720 vf=0

```

```

730 FOR j=1 TO 64
740 ff=INT((j-1)/8)+1:cf=j-8*(ff-1)
750 PRINT FN locate$(2,52);CHR$(64+ff)MID$(STR$(cf),2)
760 IF t$(ff,cf)<>' THEN 830
770 m$=hu$:GOSUB 2310:IF e=1 THEN 830
780 GOSUB 1120
790 GOSUB 1220
800 IF fic=0 THEN 830
810 v=63*p(ff,cf)+fic
820 IF v>vf THEN vf=v
830 NEXT
840 vt=vp-vf
850 IF vt>vo THEN vo=vt:fo=f:co=c:GOTO 870
860 IF vt=vo AND RND>0.5 THEN vo=vt:fo=f:co=c
870 NEXT
880 PRINT FN locate$(2,26);CHR$(64+fo)MID$(STR$(co),2)CHR$
):PRINT FN locate$(2,52)
890 ff=fo:cf=co
900 m$=am$:vis=1
910 GOSUB 1020
920 GOSUB 1220
930 GOSUB 1070
940 fh=fh-fic:fa=fa+fic+1
950 PRINT FN locate$(1,28);fa' ':PRINT FN locate$(1,54);fh
.
960 IF fa+fh<64 AND fh>0 THEN 360
970 IF fa=fh THEN PRINT FN locate$(14,5)'E M P A T A D O S
980 IF fa>fh THEN PRINT FN locate$(14,5)'H E G A N A D O
990 IF fh>fa THEN PRINT FN locate$(14,5)'V E N C I S T E !
1000 WHILE INKEY$="" :WEND
1010 PRINT ac$;gr$:END
1020 FOR k=1 TO 8
1030 FOR l=1 TO 8
1040 f$(k,l)=v$(k,l)
1050 NEXT l,k
1060 RETURN
1070 FOR k=1 TO 8
1080 FOR l=1 TO 8
1090 v$(k,l)=f$(k,l)
1100 NEXT l,k
1110 RETURN
1120 FOR k=1 TO 8
1130 FOR l=1 TO 8
1140 f$(k,l)=t$(k,l)
1150 NEXT l,k
1160 RETURN
1170 FOR k=1 TO 8
1180 FOR l=1 TO 8
1190 t$(k,l)=f$(k,l)
1200 NEXT l,k
1210 RETURN
1220 fic=0:IF m$='X' THEN c$='O' ELSE c$='X'
1230 p=ff-1
1240 WHILE p>0
1250 IF f$(p,cf)=m$ THEN 1300
1260 IF f$(p,cf)="" THEN 1290
1270 p=p-1
1280 WEND

```

```

1290 GOTO 1350
1300 FOR k=ff-1 TO p+1 STEP -1
1310 f$(k,cf)=m$
1320 fic=fic+1
1330 IF vis=1 THEN PRINT FN locate$(5+2*k,23+4*cf);m$
1340 NEXT
1350 p=ff+1
1360 WHILE p<9
1370 IF f$(p,cf)=m$ THEN 1420
1380 IF f$(p,cf)=" " THEN 1410
1390 p=p+1
1400 WEND
1410 GOTO 1470
1420 FOR k=ff+1 TO p-1
1430 f$(k,cf)=m$
1440 fic=fic+1
1450 IF vis=1 THEN PRINT FN locate$(5+2*k,23+4*cf);m$
1460 NEXT
1470 p=cf-1
1480 WHILE p>0
1490 IF f$(ff,p)=m$ THEN 1540
1500 IF f$(ff,p)=" " THEN 1530
1510 p=p-1
1520 WEND
1530 GOTO 1590
1540 FOR k=cf-1 TO p+1 STEP -1
1550 f$(ff,k)=m$
1560 fic=fic+1
1570 IF vis=1 THEN PRINT FN locate$(5+2*ff,23+4*k);m$
1580 NEXT
1590 p=cf+1
1600 WHILE p<9
1610 IF f$(ff,p)=m$ THEN 1660
1620 IF f$(ff,p)=" " THEN 1650
1630 p=p+1
1640 WEND
1650 GOTO 1710
1660 FOR k=cf+1 TO p-1
1670 f$(ff,k)=m$
1680 fic=fic+1
1690 IF vis=1 THEN PRINT FN locate$(5+2*ff,23+4*k);m$
1700 NEXT
1710 pf=ff-1;pc=cf-1
1720 WHILE pf>0 AND pc>0
1730 IF f$(pf,pc)=m$ THEN 1780
1740 IF f$(pf,pc)=" " THEN 1770
1750 pf=pf-1;pc=pc-1
1760 WEND
1770 GOTO 1830
1780 FOR k=1 TO ff-pf-1
1790 f$(ff-k,cf-k)=m$
1800 fic=fic+1
1810 IF vis=1 THEN PRINT FN locate$(5+2*(ff-k),23+4*(cf-k));
m$
1820 NEXT
1830 pf=ff-1;pc=cf+1
1840 WHILE pf>0 AND pc<9
1850 IF f$(pf,pc)=m$ THEN 1900

```

```

1860 IF f$(pf,pc)=" THEN 1890
1870 pf=pf-1:pc=pc+1
1880 WEND
1890 GOTO 1950
1900 FOR k=1 TO ff-pf-1
1910 f$(ff-k,cf+k)=m$
1920 fic=fic+1
1930 IF vis=1 THEN PRINT FN locate$(5+2*(ff-k),23+4*(cf+k));
m$
1940 NEXT
1950 pf=ff+1:pc=cf-1
1960 WHILE pf<9 AND pc>0
1970 IF f$(pf,pc)=m$ THEN 2020
1980 IF f$(pf,pc)=" THEN 2010
1990 pf=pf+1:pc=pc-1
2000 WEND
2010 GOTO 2070
2020 FOR k=1 TO pf-ff-1
2030 f$(ff+k,cf-k)=m$
2040 fic=fic+1
2050 IF vis=1 THEN PRINT FN locate$(5+2*(ff+k),23+4*(cf-k));
m$
2060 NEXT
2070 pf=ff+1:pc=cf+1
2080 WHILE pf<9 AND pc<9
2090 IF f$(pf,pc)=m$ THEN 2140
2100 IF f$(pf,pc)=" THEN 2130
2110 pf=pf+1:pc=pc+1
2120 WEND
2130 GOTO 2190
2140 FOR k=1 TO pf-ff-1
2150 f$(ff+k,cf+k)=m$
2160 fic=fic+1
2170 IF vis=1 THEN PRINT FN locate$(5+2*(ff+k),23+4*(cf+k));
m$
2180 NEXT
2190 IF fic=0 THEN 2220
2200 f$(ff,cf)=m$
2210 IF vis=1 THEN PRINT FN locate$(5+2*ff,23+4*cf);m$
2220 RETURN
2230 e=1:vis=0
2240 FOR i=1 TO 8
2250 FOR j=1 TO 8
2260 IF f$(i,j)<>" THEN 2290
2270 ff=i:cf=j:GOSUB 1220
2280 IF fic>0 THEN e=0:j=8:i=8
2290 NEXT j,i
2300 RETURN
2310 e=1
2320 FOR k=-1 TO 1
2330 FOR l=-1 TO 1
2340 IF k=0 AND l=0 THEN 2390
2350 IF m$=hu$ THEN 2380
2360 IF v$(f+k,c+l)<>m$ AND v$(f+k,c+l)<>" THEN e=0:l=1:k=1
2370 GOTO 2390
2380 IF t$(ff+k,cf+l)<>m$ AND t$(ff+k,cf+l)<>" THEN e=0:l=1
:k=1
2390 NEXT l,k

```

```

2400 RETURN
2410 DATA 150,154,154,154,158,154,154,154,158,154,154,154,15
8,154,154,154,158,154,154,154,158,154,154,154,154,158,154,154,15
4,158,154,154,154,156
2420 DATA 149,32,32,32,149,32,32,32,149,32,32,32,149,32,32,3
2,149,32,32,149,32,32,32,149,32,32,149,32,32,149
2430 DATA 151,154,154,154,159,154,154,154,159,154,154,154,15
9,154,154,154,159,154,154,154,159,154,154,154,159,154,154,15
4,159,154,154,154,157
2440 DATA 147,154,154,154,155,154,154,154,155,154,154,154,15
5,154,154,154,155,154,154,154,155,154,154,154,155,154,154,15
4,155,154,154,154,153
2450 RESTORE 2410:FOR i=1 TO 4
2460 FOR j=1 TO 33
2470 READ tb(i,j)
2480 NEXT j,i
2490 PRINT FN locate$(6,25);:FOR j=1 TO 33:PRINT CHR$(tb(1,j
));:NEXT:PRINT
2500 FOR f=3 TO 17
2510 IF f/2=INT(f/2) THEN i=3 ELSE i=2
2520 PRINT FN locate$(4+f,25);:FOR j=1 TO 33:PRINT CHR$(tb(i
,j));:NEXT:PRINT
2530 NEXT
2540 PRINT FN locate$(22,25);:FOR j=1 TO 33:PRINT CHR$(tb(4,
j));:NEXT
2550 RETURN

```

## CÓMO JUEGA NUESTRO OTHELLO

La base de los movimientos del ordenador es un tablero, representado en la matriz numérica P, en el cual se indica un valor figurado para cada una de las posibles posiciones que oscila entre 1 y 9 (el valor cero corresponde a las posiciones ocupadas por las fichas iniciales). A partir de estos valores, y el número de fichas que se capturan en cada movimiento, el programa otorga un valor determinado a cada jugada, dado por una función de evaluación del tipo  $V=63 \cdot P + F$ , donde P es el valor posicional y F la cantidad de fichas capturadas.

De esta forma el ordenador va analizando todo el tablero. Por cada posición evaluada, procede a ponerse en el lugar del contrario, y siguiendo la misma función de evaluación antes expresada, calcular cual puede ser la mejor de las respuestas del humano.

Por último, el valor definitivo de una posición es el obtenido por la máquina menos el que presumiblemente el humano podrá obtener a continuación. Así pues, el programa ilustra la aplicación del algoritmo

minimax, siguiéndose una poda de profundidad dos, ya que el ordenador se limita a considerar su movimiento y la posible respuesta del contrario.

No obstante, OTHELLO obtiene resultados de consideración, y se precisa de una cierta experiencia en el juego para vencerle, aunque desde luego esto no es ni mucho menos imposible.

## SU EXCELENCIA EL S.E.

**E**n determinados problemas de extraordinaria complejidad, se descubrió que la utilización de los métodos de búsqueda de soluciones que vimos en el capítulo anterior, no daban los resultados esperados, sobre todo debido a las dimensiones que llegaban a tomar los árboles de estados, aun después de haber sido convenientemente podados.

Tales problemas eran aquellos en que la concurrencia de un hombre se hacía absolutamente necesaria, más exactamente de humanos expertos en un campo determinado, como por ejemplo el diagnóstico de enfermedades, etc...

Inmediatamente se acometió el diseño de programas, que posibilitaran la aplicación de ordenadores también a estas tareas. Estos programas son los denominados Sistemas Expertos (SE). Así pues, estos sistemas son la aplicación práctica más palpable de los estudios sobre inteligencia artificial.

### CARACTERÍSTICAS DE LOS SISTEMAS EXPERTOS

En general, podríamos decir que estos programas se caracterizan por reunir unas determinadas estrategias de búsqueda de las solucio-

nes, acompañadas de una heurística o experiencia que colaboran en esta tarea, más una base general de conocimientos sobre el tema que capacitan al programa para actuar como un experto.

Las diferencias fundamentales entre un S.E. y un sistema informático convencional, se pueden sintetizar en los siguientes puntos:

— Los S.E. almacenan principalmente información simbólica, mientras que los convencionales operan sobre informaciones numéricas.

— Los S.E. pueden incrementar su base inicial de conocimientos, mientras que los sistemas convencionales no son capaces de «aprender».

— Los S.E. son capaces de explicar el procedimiento seguido para llegar a un determinado resultado.

Según estas características, llegaremos a la conclusión que la mayoría de nosotros estamos en condiciones de construir un sistema experto, si bien no de una calidad abrumadora, sí lo suficientemente eficaz como para resolver determinados problemas.

De hecho, ya hemos tenido la oportunidad de disfrutar en este libro de un sistema experto en la corrección de errores ortográficos, con una base de datos capaz de «aprender» nuevos conocimientos. Por otra parte, 'VOBDIR' podía informarnos de todos los movimientos que tenía que realizar previamente, para dejar los cubos en un determinado estado.

## UN SIMPLE PROGRAMA

Desde que cualquier programador, profesional o aficionado, construye su primer programa, se convierte en una obsesión llegar a realizar otros programas que superen a los anteriores en complejidad, efectividad, «inteligencia», etc... Por fin, sólo algún programador, generalmente profesional, llega a construir un día un sistema que puede llamar «inteligente», que generalmente se aplica a una determinada tarea: construye su propio Sistema Experto. La alegría por este logro es muy grande, pero no eterna.

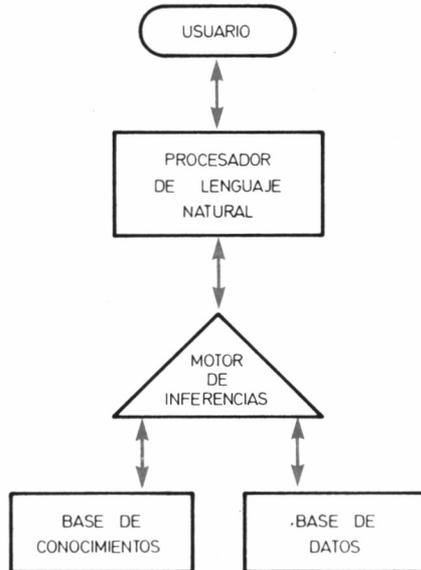
Un buen día, se da cuenta que su experto no sabe resolver cierto problema, y entonces acomete la tarea de modificar o construir un nuevo S.E. que supere al anterior. Lo consigue; y una vez logrado esto, el sistema experto anterior pasa a ser un simple programa.

Así pues, sistema experto no es más que una definición temporal para un programa de extraordinaria calidad, pero que dejará de serlo

cuando su efectividad ya no sea «sobresaliente», precisamente por abundar creaciones de sus características en el mundo de la programación.

Pensemos en hace muchos años, cuando los ordenadores eran máquinas misteriosas, únicamente controlables por humanos altamente especializados. Un buen día, uno de ellos diseñó un programa de contabilidad, con una capacidad para almacenar, modificar y reprocesar los datos contables de varios años, lo suficientemente hábil como para facilitar la entrada de información por parte del humano al máximo, que nos brinda en unos minutos todos aquellos informes en los que el jefe del departamento de contabilidad de nuestra empresa tarda días, que no comete un sólo error en el proceso de los datos... en definitiva un auténtico experto en contabilidad. Hoy en día, cuando un programa de estas características se puede adquirir por menos de 20.000 pesetas ¿quién dice ahora que un programa de contabilidad sea un sistema experto? Nadie.

Así pues, aquel que se sienta con ánimo, puede emprender hoy mismo la ruda tarea de crear un S.E. en cualquier campo; pero tenga cuidado en que sea un programa lo suficientemente bueno para que, al menos durante un tiempo, su creación pueda disfrutar del orgullo y



— 4.8.1. Esquema general de un Sistema Experto.

beneficio de ser llamado Sistema Experto, antes que alguien lo mejore y se convierta en un vulgar programa.

## CONSTRUCCIÓN Y OBJETIVOS A CUBRIR POR UN S.E.

A la hora de plantearnos la construcción de un sistema experto, hemos de tener en cuenta la restricción que nos viene directamente dada por la capacidad física del equipo que disponemos. Por tanto, debemos restringir suficientemente el dominio de nuestro S.E. como para que tenga cabida en la memoria del ordenador.

Por otra parte, será precisa la concurrencia de uno de los llamados Ingenieros del Conocimiento, expertos en la extracción de información del tema determinado sobre el que versará el S.E. de otros expertos humanos, libros, experiencia propia o ajena, u otras fuentes de información. En nuestro caso concreto, lo mejor que podemos hacer es constituirnos en Ingenieros del Conocimiento, para lo cual será necesario confeccionar un experto sobre un tema que conozcamos bien.

No debemos incurrir en el error de intentar competir con los grandes S.E. conocidos, porque mucho nos tememos que sin contar con un sistema de capacidad comparable y una ayuda similar a la de los grandes equipos de investigación en I.A., no vamos a obtener otro resultado que el desaliento ante nuestra impotencia. Para evitar caer en el error, he aquí una lista de los S.E. más difundidos en estos momentos:

AGE	Construcción de S.E.
AL/X	Construcción de S.E.
CADUCEUS	Versión modernizada de INTERNIST.
CASNET	Diagnóstico médico.
DENDRAL	Química.
EL	Diseño electrónico.
EMYCIN	Adquisición de conocimientos.
EXPERT	Adquisición de conocimientos.
GUIDON	Medicina.
HEARSAY III	Construcción de S.E.
INTERNIST	Diagnóstico médico.
KAS	Adquisición de conocimientos.
MECHO	Mecánica.
MICRO-EXPERT	Construcción de S.E.
MOLGEN	Genética.

MYCIN	Diagnóstico médico.
PECOS	Programación.
PIP	Diagnóstico médico.
PROSPECTOR	Geología.
PUFF	Diagnóstico médico.
R1	Configuración de ordenadores.
ROSIE	Construcción de S.E.
SACON	Ingeniería.
SAGE	Construcción de S.E.
SECS	Química.
SOPHIE	Electrónica.
SYNCHEM	Química.
SYSTRAN	Traducción de lenguaje natural.
TEIRESIAS	Adquisición de conocimientos.
VM	Medicina.
XCON	Versión modernizada de R.

Tampoco debemos amilanarnos, y hemos de acometer con ánimo la tarea a la que nos enfrentamos, intentando conseguir, aunque sea en un dominio muy restringido, un auténtico sistema experto y no un simple programa. Tengamos en cuenta que la última misión del S.E. es llegar a superar al experto humano, ya que de no ser así, no tendría sentido su existencia.

Finalmente, veamos las partes que suelen configurar un S.E.:

— Interface de comunicación con el usuario: Esta comunicación deberá establecerse, preferentemente, en lenguaje natural.

— Base de conocimientos: Almacena los conocimientos del S.E. sobre su dominio concreto, en forma de reglas. Estas reglas pueden tener la orma IF/THEN.

— Base de datos: Contiene los datos para el trabajo del S.E. sobre un determinado problema. Es en este espacio donde se pueden almacenar también datos diversos, como por ejemplo los pasos seguidos por el sistema para la solución del problema, de forma que puedan serle comunicados al usuario en caso de que éste así lo solicite.

— Motor de inferencias: Aquí reside la «inteligencia» del S.E. Es el mecanismo de control que relaciona todas las reglas de la base de conocimientos con los datos concretos del problema para conseguir una solución óptima del mismo.

Una vez respetadas las normas que hemos visto en este capítulo y aprendidos los conocimientos que se proporcionan a lo largo del volumen, sólo resta para el diseño de un S.E., la fuerza de voluntad suficiente, ejercicio que, evidentemente, depende completamente del lector.



# NOTAS

A large, stylized letter 'D' in a dark purple color with a white outline, positioned at the start of the main text block.

Desde antaño, el hombre ha intentado imitar mediante artilugios diversos el comportamiento humano, inicialmente el mecánico; desde las bailarinas de las antiguas cajas de música, hasta los complejos robots de las cadenas de montaje actuales. Con el avance tecnológico, sus aspiraciones en este sentido se han ido sintetizando en la obtención de mecanismos que simulen la más preciosa cualidad del hombre, lo que le distingue del resto de los animales que pueblan la Tierra: la inteligencia.

GRAN BIBLIOTECA  
AMSTRAD

450 ptas.  
(incluido IVA)

Precio en Canarias, Ceuta y Melilla: 435 ptas.