El Byte Biológico (2)

"Creo que hay un mercado mundial para alrededor de cinco computadoras."
Thomas Watson (Presidente de IBM en 1968)

No existe ninguna duda al respecto. La computadora ha revolucionado nuestra forma de vida. El modo en que trabajamos, estudiamos, nos comunicamos, jugamos, etc. ha cambiado radicalmente durante los últimos 30 años y este cambio vemos que es exponencial. Cada vez afecta más áreas de nuestra vida y con mayor rapidez.
Hemos llegado a un momento en el que este planeta simplemente no podría soportar la cantidad de gente que vive en él si no tuviésemos la ayuda de la Informática.
¿Cuándo y cómo comienza esta revolución?
Por supuesto que lo primero que uno piensa es que comenzó a mitad del siglo XX. Pues la verdad es que si piensa eso ¡le erró por un siglo!
Al que se considera el padre de la informática es un matemático del siglo XIX llamado Charles Babbage.
Babbage vivió entre 1791 y 1871. Durante su vida incursionó, además de la matemática, en múltiples temas como por ejemplo criptografía, fue el creador del sistema postal que se utiliza hasta hoy en día, ayudó en el diseño de los trenes de trocha ancha, etc. Pero, sin ninguna duda, su mayor logro fue en la automatización del cálculo matemático.
En el año 1822 presenta su primer modelo: la Máquina Diferencial. Esta máquina era una calculadora especialmente diseñada para el cálculo de polinomios.
Al año siguiente el gobierno británico le otorga un subsidio para que la construya, pero nunca logró terminarla. Si bien los diseños eran correctos, se encontró con problemas que la tecnología de aquellos años no podía resolver.
Babbage necesitaba que su máquina, de alguna manera pudiese representar los números. Para ello decidió que cada dígito de un número estaría indicado en una rueda dentada que podía tener 10 posiciones. Cada una de esas posiciones representaba uno de los dígitos del sistema decimal, de 0 a 9.
Para simbolizar entonces un número más grande que 9 lo que hizo fue apilar más de estas ruedas dentadas. De esa forma, por ejemplo el nivel inferior simbolizaba las unidades, el segundo nivel las decenas, el tercer nivel las centenas y así sucesivamente.
Si de iba incrementando de a uno, por ejemplo en el nivel de las unidades, al llegar a 9 había una muesca que con el siguiente incremento, arrastraba la rueda que se encontraba en el nivel siguiente, con lo cual se incrementaba en 1 la decena y las unidades pasaban a estar en 0. Esto ocurría en todos los niveles.
La Figura 24 puede verse la Máquina Diferencial. En esta figura pueden apreciarse claramente las ruedas dentadas mencionadas y su apilamiento.

Decíamos previamente que la máquina nunca logró funcionar debido a problemas de tipo tecnológico. Babbage se encontró fundamentalmente con dos problemas:

- El primero fueron las fuerzas de rozamiento. La máquina necesitaba para poder moverse el vencer todos los rozamientos internos que tenía y en esa época no contaban con motores que fuesen lo suficientemente poderosos
- El segundo fue la precisión en la construcción. Tampoco se contaba por aquel entonces con una metalurgia lo suficientemente desarrollada como para producir piezas que se ajustasen perfectamente entre ellas. Esto generaba mucha vibración

En 1991 en Museo de Ciencias de Londres construyó la Máquina Diferencial basándose en los planos de Babbage pero con tecnología moderna. Funcionó perfectamente.

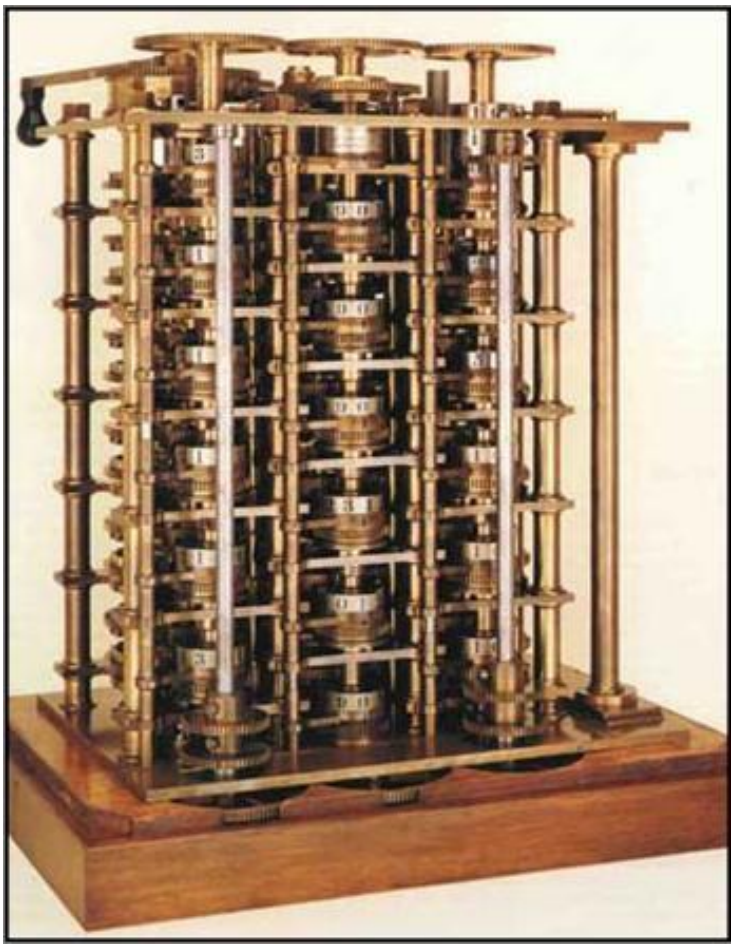

Figura 23

Este aparente fracasó no lo desanimó. En el año 1833 comenzó a trabajar en lo que sería su proyecto más ambicioso: la Máquina Analítica.
Esta máquina tenía una diferencia fundamental con la Máquina Diferencial. Era programable. El haber entendido en ese momento la versatilidad que un programa le daría a la Máquina Analítica es un adelanto conceptual asombroso. Es por esta razón que se la considera como la primera computadora.
Para el ingreso de la información utilizaba tarjetas perforadas. Este era un sistema que ya en ese momento estaba siendo utilizado en telares. Asimismo contenía un procesador aritmético (similar al de la Máquina Diferencial) e incluso una memoria que podía almacenar números.
Si bien Babbage no logró que sus máquinas funcionasen su labor fue fundamental para el desarrollo de la computación. Hubo dos herramientas que Babbage no utilizó. La primera porque no se debe haber dado cuenta que podría haberle simplificado mucho su diseño y la segunda porque sencillamente no existía en el siglo XIX. Estas dos herramientas son, respectivamente, el sistema binario y la electrónica.

El sistema binario

El sistema numérico decimal que utilizamos es el sistema arábigo. Este sistema consta de 10 símbolos diferentes, y sus combinaciones, para representar todos los números.
¿Alguna vez pensó por qué este sistema utiliza 10 símbolos diferentes?
La respuesta es muy sencilla: porque tenemos diez dedos. Es totalmente natural que hayamos creado toda nuestra aritmética basada en un sistema que tiene 10 símbolos diferentes.
Sin embargo no existe ninguna razón para que esto sea así. Toda la aritmética puede desarrollarse exactamente igual si trabajásemos en un sistema que tenga 4, 7 o 25 símbolos diferentes. Sencillamente no importa.
De todos estos sistemas existe un sistema que es particularmente importante y es el que utiliza sólo dos símbolos: el "0" y el "1". Este es el sistema binario.
Antes de comenzar a ver cómo funciona el sistema binario analicemos un poco el sistema decimal.
Como dijimos tenemos 10 símbolos diferentes: 0, 1, 2, 3, 4, 5, 6, 7, 8, y 9. Si quiero representar el número que sigue al 9 lo que deberé hacer es comenzar a combinarlos. De esta forma el número diez estará representado por la unión del Símbolo "1" y el "0". Es decir "10". A partir de acá puedo seguir haciendo combinaciones de dos símbolos hasta llegar al número 99. En este número se me acaban las combinaciones de esos 10 símbolos tomados de a dos. Nuevamente para seguir adelante deberé agregar un símbolo más y pasar a un número de 3 dígitos: el "100". Y así seguiremos sucesivamente hasta el infinito.

¿En qué números tuve que agregar un dígito más? Obviamente en el 10, el 100, el 1000, el 10000, etc. Es decir, en las potencias de 10: 10^1, 10^2, 10^3, 10^4, etc. donde 10, que es la cantidad de símbolos diferentes que tenemos, es la base del sistema.
Es importante notar que cada uno de estos números tiene una cantidad de ceros que es igual al exponente de la potencia. Asi por ejemplo el número un millón (1.000.000) contiene 6 ceros, por lo que lo podremos representar rápidamente por 10^6.
El sistema binario, y cualquier otro sistema que planteemos, funcionarán de la misma manera. Para entenderlo veamos las equivalencias entre los números decimales y esos mismos números representados en sistema binario:

Sistema decimal	Sistema binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Nuevamente… ¿en qué números deberemos aumentar en un dígito para poder seguir adelante? Ahora será en las potencias de 2: 2, 4, 8, 16, etc. pues 2 es ahora la base de nuestro sistema.
¿Qué ventaja podría haberle dado a Babbage trabajar en sistema binario?
Para representar cada número él tuvo que utilizar algo que se encontrase en 10 posiciones diferentes. Si hubiese utilizado sistema binario le habría alcanzado con utilizar algún elemento que estuviese sólo en dos posiciones: una para el "0" y otra para el "1". Esto hubiese simplificado mucho su diseño. Es mucho más fácil tener algo en 2 posiciones que en 10.
De hecho las computadoras modernas trabajan internamente en sistema binario. Todo lo que hacen lo logran trabajando con ceros y unos.
¿Cómo hacemos para saber cómo representar un determinado número en sistema binario?
Supongamos que queremos representar el número 83.
Lo que haremos es descomponerlo en sus potencias de 2 y luego sumar cada uno de los números generados.

Es decir:

Sistema decimal	Potencia de 2	Sistema binario
64	2^6	1000000
16	2^4	10000
2	2^1	10
1	2^0	1
83		1010011

Ya volveremos al tema de cómo hacen las computadoras modernas para almacenar la información.

La electrónica

Dijimos también que la electrónica fue una herramienta con la que Babbage no contó en su época. Esta herramienta le hubiera solucionado de raíz los problemas mecánicos con los que se encontró. La solución electrónica no presenta partes movibles por lo que no hubiera tenido problemas debidos a las fuerzas de rozamiento ni tampoco problemas debidos a la precisión metalúrgica requerida para la fabricación de las partes.
A mediados del siglo XIX las investigaciones en electricidad estaban recién comenzando y tuvimos que esperar hasta bien entrado el siglo XX para poder contar con la solución electrónica.
Veamos cómo se implementa esta solución.
Dijimos que Babbage utilizó ruedas que podían estar en 10 posiciones diferentes pero que la utilización del sistema binario nos permitiría utilizar algún elemento que pudiera estar sólo en dos posiciones. Ese elemento fueron unos pequeños anillos llamados "núcleos de ferrita".
La ferrita es un material que contiene hierro y que tiene la capacidad de magnetizarse. Si a este material le damos la forma de un anillo podremos magnetizarlo en dos diferentes sentidos, hacia la derecha o hacia la izquierda, que es lo mismo que decir que el polo norte del magneto esté de un lado o del otro del anillo. Esta característica podríamos utilizarla para codificar si se trata de un "0" o un "1" en sistema binario. Es decir: si está magnetizado hacia la derecha simboliza un "0" y si no, un "1".
Ahora bien, ¿cómo se hace para inducir en uno de estos anillos esa magnetización o para cambiarla cuando se desee pasar por ejemplo de un "0" a un "1"?
Los anillos de ferrita tienen la característica de que si se pasa un cable a través de ellos y se aplica una corriente continua, esa corriente inducirá la magnetización del anillo y el

sentido de esta magnetización dependerá del sentido que tuvo la corriente que pasó por el cable (Figura 24).

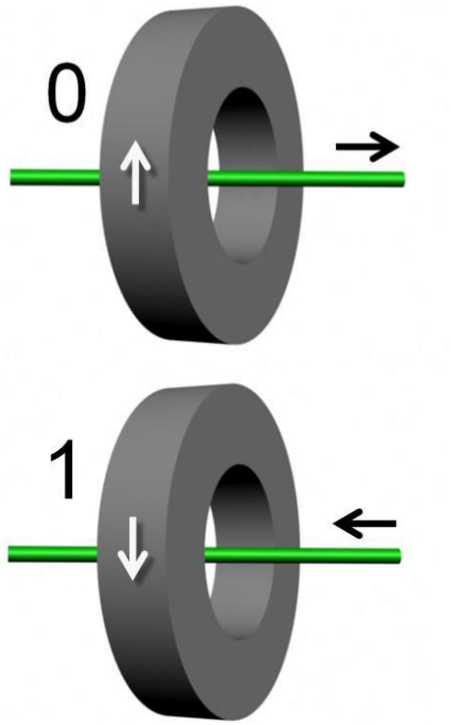

Figura 24

Esta característica es la que se utilizó en las primeras computadoras para construir las memorias que permitían almacenar los números.

El funcionamiento de la memoria

Supongamos que quisiéramos guardar el número 83, del cual antes calculamos su representación binaria. Ese número binario, como recordarán, era el "1010011".
Para poder almacenar este número se seguirá la misma estrategia que siguió Babbage. El apilaba las ruedas dentadas, en este caso apilaremos los anillos de ferrita. A ese núcleo de ferrita se lo llamo "bit" y a la pila de bits, que estaba en las primeras computadoras compuesta por 7 bits, se la llamo "byte". Esto permitiría representar como máximo el número "1111111" lo cual representa el número 127.
Por supuesto que esto parecería no alcanzar para trabajar con números más grandes que 127, lo cual no es lógico pues es obvio que se debería trabajar con números más grandes. Adicionalmente una computadora debería ser alfanumérica, es decir que debía poder representar tanto números como letras.

La solución a esto es que en realidad lo que se guarda allí no es el número en sí mismo sino un código que es el que representa cada uno de los símbolos que utilizamos habitualmente. Existe un código para la letra "a", otro para la "A", otro para el número "8", otro para el signo "@", y así con todos.
El número 83, por lo tanto no se guarda como tal sino que se utilizan 2 bytes: uno para guardar el "8" y otro para el "3". Es decir que lo que hacemos en cada byte es guardar caracteres. Por lo tanto, cuando decimos que una memoria tiene 512 Mbytes lo que estaremos diciendo es que es capaz de almacenar 512000 caracteres[1].
Creo que a todos nos ha ocurrido que alguna vez que quisimos escribir la letra "e" con acento ("é") y alguien nos dijo que debíamos apretar la tecla "ALT" y, sin soltarla, marcar en el teclado numérico el número 130. Al soltar, mágicamente, apareció la letra "é". Lo que hicimos en ese caso fue incluir ese código que indica cuál es el carácter a mostrar en la pantalla. Esto códigos son llamados "Códigos ASCII".
Dijimos que los dígitos ("0" y "1") se almacenaban en esos núcleo de ferrita llamados "bits" y de acuerdo a su sentido de magnetización. Esta magnetización está sujeta a ciertas leyes físicas que se encuentran descriptas por un gráfico llamado "ciclo de histéresis" (Figura 25).

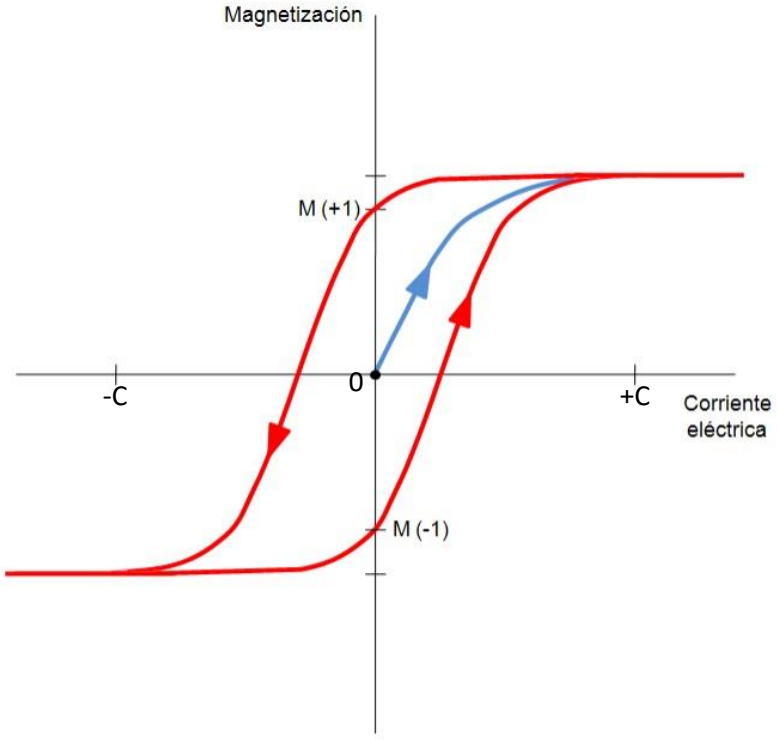

Figura 25

1 – En realidad cada Mbyte no contiene 1000 bytes sino 1024 bytes por lo que 512 Mbytes equivaldrían a 524288 bytes.

Este gráfico describe cómo se comporta la magnetización del anillo en función de la corriente que pase por el cable.
Supongamos que inicialmente el anillo no está magnetizado y no pasa ninguna corriente por el cable. Nos encontramos en el origen de coordenadas. Si comenzamos a pasar una corriente "positiva" es decir hacia el lado derecho del gráfico la magnetización del anillo irá creciendo, tal como lo indica la línea azul.
Una vez superado un valor límite de corriente, indicado como "+C", si se corta la corriente el camino que seguirá la magnetización es el indicado por la línea roja superior lo cual llevará la magnetización del anillo a la posición "M (+1)". Es decir que el anillo quedará magnetizado en un determinado sentido.
Si ahora aplicamos una corriente en el sentido inverso (hacia la izquierda), y si superamos el punto "-C", al cortar la corriente ocurrirá lo mismo y deberemos seguir el camino indicado en el gráfico por la línea roja inferior hasta el punto "M (-1)" en el que el anillo quedará magnetizado en el otro sentido.
Lo que tiene de interesante este gráfico es que para que se cambie de estado de magnetización debe superarse una cantidad de corriente límite. La curva roja siempre se deberá recorrer en el sentido anti-horario (el indicado por las flechas).
Supongamos que nos encontramos, tal como estábamos antes, en el punto de magnetización "M (+1)". Si aplicamos una corriente hacia la izquierda pero ésta no alcanza el punto "-C" lo que ocurrirá es que la magnetización retrocederá por la curva roja nuevamente hasta el punto "M (+1)". Sólo si superamos el punto "-C" se producirá el cambio de sentido en la magnetización.
Con todo esto en mente veamos cómo se almacenaba la información.
Dijimos que los bits estarían apilados en grupos de 7 (1 byte). Supongamos que en el medio de esa pila de bits tenemos uno que está en 0 y en el que queremos grabar un 1. Obviamente tendremos que hacer pasar una corriente por el cable que lo atraviesa de manera que se supere el límite y se grabe el 1 que deseamos. El problema con el que nos vamos a encontrar es que ese mismo cable está pasando por todos los bits de la pila, por lo que se cambiarán a 1 todos aquellos bits que estuviesen en 0. Los que estaban ya en 1 quedarán sin modificación. Es decir que, luego del paso de la corriente, toda la pila contendrá sólo valores "1".
¿Cómo solucionamos este problema?
La solución fue armar con ellos una estructura reticular de tipo cartesiano como la mostrada por le Figura 26.
Supongamos que deseamos cambiar sólo la información contenida sólo en el bit enmarcado en el rectángulo, el cual contiene un 0. Alcanzará con pasar una corriente menor a la corriente límite por los dos cables marcados con flechas. La cantidad de corriente será tal que en el bit en cuestión ambas se sumarán y serán suficientes para producir el cambio. Lo interesante de esto es que ese cambio se producirá sólo en ese

bit pues la corriente que atraviesa los otros anillos no llega a la corriente límite necesaria para producir un cambio de magnetización.

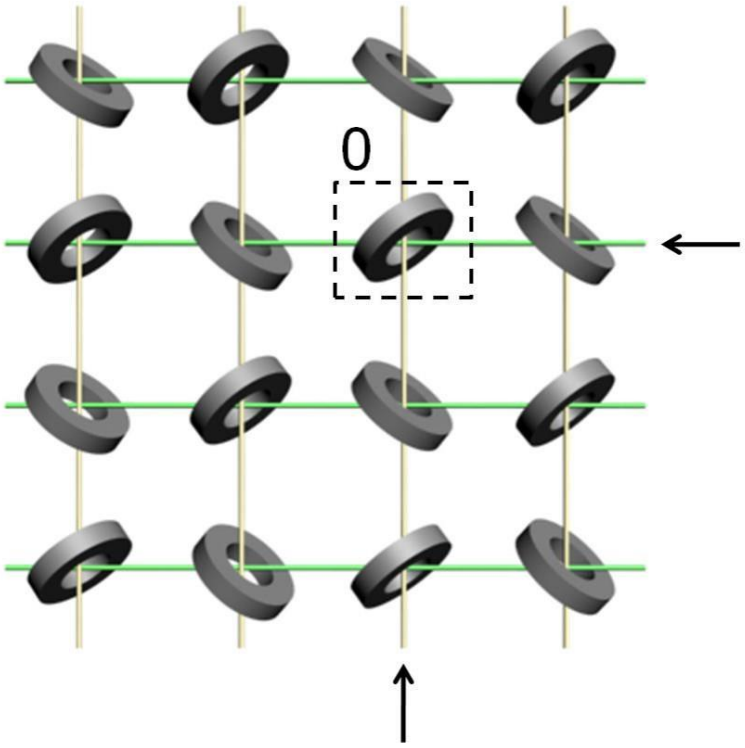

Figura 26

De esta forma podemos grabar información, pero… ¿cómo hacemos para leer información que ya esté grabada y sin modificarla?
Para poder analizar la información contenida en un determinado bit nos basaremos en otra propiedad física de los núcleos de ferrita (Figura 27).

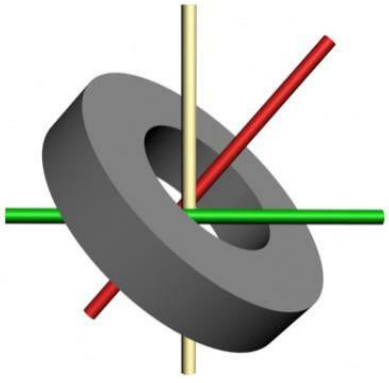

Figura 27

Si uno de estos núcleos es atravesado por cable y la magnetización del núcleo cambia, esto inducirá en ese cable una corriente eléctrica perfectamente medible. Es decir que alcanzará con pasar un tercer cable por cada núcleo. Si, en el ejemplo anterior, pasásemos corriente por los cables indicados con las flechas, cambiando la información contenida en el bit de 0 a 1, esto induciría una corriente en ese tercer cable la cual nos indicaría que ese cambio se produjo y, por lo tanto, sabremos cuál era la información grabada en el bit.
Ahora bien, esto acarrea otro problema más. La forma de lectura de la información es destructiva ya que ahora donde teníamos un 0 tendremos grabado un 1. La solución es que inmediatamente que fue leída la información se inducirá por los mismos cables una corriente contraria volviendo a grabar el 0 que habíamos borrado. De esta forma todo queda cómo estaba y habremos podido extraer la información deseada.
Esa segunda inducción de corriente se producía en el orden de los nanosegundos. Como para que no demos una idea de qué es un nanosegundo es la mil millonésima parte de un segundo. Hay tantos nanosegundos en 1 segundo ¡como segundos en 32 años!
La figura 28 nos muestra una memoria típica de las computadoras de los años 60. Normalmente por aquellos años estas memorias podían almacenar 32 Kbytes.
La cantidad de núcleos que contenían se puede calcular de la siguiente forma:

$$\text{Núcleos} = 32\ \text{Kbytes} \times 1024\ \frac{\text{bytes}}{\text{Kbyte}} \times 7\ \frac{\text{bits}}{\text{byte}}$$

Esto da 229.376 núcleos en cada memoria. ¡Y se armaban a mano!

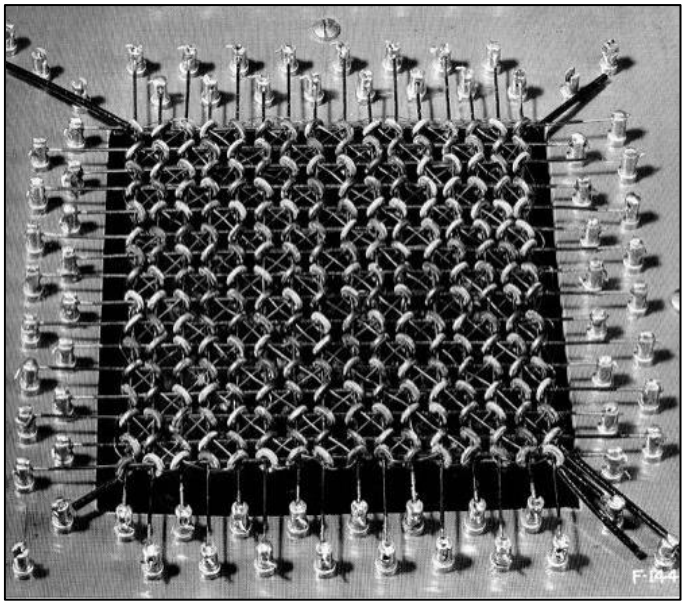

Figura 28

Hoy en día las memorias de las computadoras ya no funcionan de esta manera. Si bien siguen trabajando con sistema binario y armando bytes con unidades más básicas como los bits, estos ya no son núcleos de ferrita sino que son transistores los cuales están contenidos en chips electrónicos. Estos transistores son extremadamente pequeños y es así que una memoria pequeña puede contener varios Giga Bytes (1 Gbyte = 1.000.000.000 bytes).

Las inevitables comparaciones

Hasta acá hemos visto cómo se las han ingeniado la biología y el hombre para guardar la información.
A los que, por propios intereses y también por esas vueltas de la vida, hemos tenido que estar en contacto con estos dos temas (biología e informática) se nos hace inevitable el establecer los paralelismos.
Vamos a tratar de analizar la estructura de la información en ambos sistemas de manera de poder ver que similitudes y diferencias tenemos.
Para ello empezaremos por establecer una jerarquía en la organización de la información.
Vamos a dividirla en 4 niveles diferentes:

1) Nivel "Unidad Básica"
   Es la forma mínima de guardar información. Esta unidad de información no estaría formada por ningún nivel más bajo.
2) Nivel "Unidad de Codificación"
   Este nivel estará formado por unidades del nivel "Unidad Básica". Es la unidad que se encargará de codificar el siguiente nivel.
3) Nivel "Unidad Codificada"
   Este nivel indicará las unidades que fueron codificadas por el nivel anterior.
4) Nivel "Información Final".
   Este nivel indicará el último nivel y se logra cuando las unidades del nivel anterior se unen ente sí.

Hasta acá y explicado de esta manera parece un poco complicado, pero apenas los asociemos con los ejemplos que ya hemos visto se entenderá rápidamente.
Empecemos por la forma de codificación de la información en una célula.

La codificación de la información celular

Como vimos la información celular está guardada en el núcleo y codificada en el ADN.
El ADN estaba formado por dos largas cintas de desoxirribosas y fosfatos y, hacia el

interior se ubicaban las bases nitrogenadas, las cuales eran 4: Adenina, Timina, Citosina y Guanina.
Estas bases constituirían lo que llamamos el nivel "Unidad Básica" (Nivel 1). No existe una forma más primaria de almacenamiento de información.
Pasemos al siguiente nivel. ¿Qué se arma con las bases nitrogenadas?
Como también vimos agrupamientos de 3 bases dan origen a lo que llamamos un "codón". Este codón sería el nivel "Unidad de Codificación (Nivel 2).
Veamos ahora qué codificamos en ese codón. Como recordarán durante el proceso de traducción pasaremos del lenguaje de ADN (bases nitrogenadas y codones) al lenguaje de proteínas (aminoácidos). Es decir que lo que codificaremos con los codones serán los aminoácidos. Estos constituyen, por lo tanto el nivel "Unidad Codificada" (Nivel 3).
Por último, si unimos los aminoácidos llegaremos al nivel "Información Final" (Nivel 4): las proteínas.
Es decir que la organización de la información en una célula quedaría de la siguiente manera:

Nivel #	Nombre	Célula
1	Unidad Básica	Base nitrogenada
2	Unidad de Codificación	Codón
3	Unidad Codificada	Aminoácido
4	Información Final	Proteína

La codificación de la información en informática

En este caso vimos que la información se guardaba en sistema binario mediante la magnetización de núcleos de ferrita y que la unión de esta información codificaba los caracteres que necesitaríamos para llevar a cabo cualquier proceso.
Veamos cómo se organizarían de acuerdo a la jerarquización que planteamos anteriormente.
La "Unidad Básica" en este caso lo constituirían los unos y los ceros. No hay nada debajo de ellos. Y estos están, a su vez, codificados en los bits, es decir que son los bits los que constituyen la "Unidad Básica" (Nivel 1).
Con la unión de 7 bits conformamos un byte. Este sería el nivel "Unidad de Codificación" (nivel 2).
¿Qué codificamos con los bytes? Como vimos los bytes nos permiten, a través de los códigos ASCII, codificar cualquier carácter que necesitemos. Es decir que los caracteres codificados por los bytes constituyen el nivel "Unidad Codificada" (Nivel 3).
Por último ¿qué logramos uniendo los caracteres?

Esta pregunta puede tener varias respuestas dependiendo de cuáles sean los caracteres. Si lo que tenemos codificados son sólo caracteres de letras lo que estaremos formando serán palabras, pero si fuesen sólo números quizás (y no siempre) queramos simbolizar un número con el cual debamos realizar algún tipo de operación matemática, pero elijamos la primera opción. Es decir que las palabras serían el nivel "Información Final" (Nivel 4).
Es decir que la organización de la información en una computadora quedaría de la siguiente manera:

Nivel #	Nombre	Célula
1	Unidad Básica	Bit
2	Unidad de Codificación	Byte
3	Unidad Codificada	Carácter
4	Información Final	Palabra

Conclusiones

Las correlaciones son evidentes.
Una célula y una computadora utilizan exactamente la misma estrategia para guardar la información.
¿Cuáles serían las diferencias?
La primera diferencia está en el tipo de sistema utilizado para el almacenamiento de la información.
Las computadoras, como vimos, utilizan sistema binario, es decir sólo dos dígitos diferentes: 0 y 1. Las células utilizan sistema cuaternario. Poseen cuatro dígitos diferentes: Adenina, Timina, Citosina y Guanina.
La segunda diferencia importante es la cantidad de Unidades Básicas que conforman la Unidad de Codificación.
En el caso de las computadoras vimos que cada byte estaba formado por 7 bits, es decir que cada Unidad de Codificación contiene 7 Unidades Básicas. Esto implica que la cantidad máxima de posibilidades de codificación será 2^7, es decir. 128 caracteres diferentes.
En las células el codón está conformado por un grupo de 3 bases nitrogenadas, por lo que cada Unidad de Codificación contiene 3 Unidades Básicas. En este caso la cantidad de combinaciones posibles será, por lo tanto, 4^3, es decir: 64 posibilidades diferentes.

Un paso más allá…

Cuando dos computadoras deben comunicarse entre sí, es decir deben traspasarse información lo hacen enviando esta información en paquetes. El tamaño de estos paquetes de información puede variar mucho, pero supongamos que la cantidad de información que deben transmitirse el pequeña. Por ejemplo un grupo de símbolos (letras y números). Esto es lo que se conoce como un "string".
Para que las dos computadoras puedan entender la información que están intercambiando, la computadora emisora tiene que informarle a la computadora receptora dónde comienza y dónde termina cada string.
Esto lo hace mediante unos caracteres especiales, que tienen ya asignado un código ASCII específico para esta función, los cuáles indican el comienzo y el final del string.
¿Qué pasa en las células?
Durante la traducción, como dijimos, vamos a pasar de lenguaje ADN al lenguaje de las proteínas, y la molécula que se deberá leer para llevar a cabo la traducción es el ARNm. Cuando se comienza con la traducción siempre el ARNm comienza con el codón ATG el cuál codifica para el aminoácido Metionina (Met). Si nos fijamos en el cuadro del Código Genético (Figura 18), veremos que en general cada aminoácido está codificado por más de un codón, sin embargo ATG es el único codón que codifica a la Metionina. La Metionina es la indicación que el ARNm le da al ribosoma indicándole que allí tiene que comenzar el proceso de la traducción.
Cuando el proceso deba terminar existen otros codones (UAA, UAG y UGA) los cuales constituyen una señal STOP, es decir, indica que allí termina la codificación de la proteína. La analogía es asombrosa.

Una más…

La información contenida en una computadora puede dividirse claramente en dos tipos: programas y datos. Los programas constituyen la información que deberá utilizarse para poder procesar los datos y realizar con ellos algún tipo de tarea específica. Por ejemplo, en una computadora actual si deseo ver la información contenida en un archivo ".doc" voy a requerir tener instalado el programa Word. Si no lo tengo de nada me servirán los datos.
En el pasado (años '60) la forma de ingresar la información a la computadora era mediante tarjetas perforadas. Este sistema estaba derivado del sistema de tarjetas perforadas utilizado en telares al cual ya habíamos hecho alusión al hablar de la máquina Analítica de Babbage.
Para llevar a cabo un proceso se debían ingresar dos grupos de tarjetas. La primera contenía las instrucciones del programa. La segunda contenía los datos. El aspecto que tenían las tarjetas perforadas se muestra en la Figura 29.

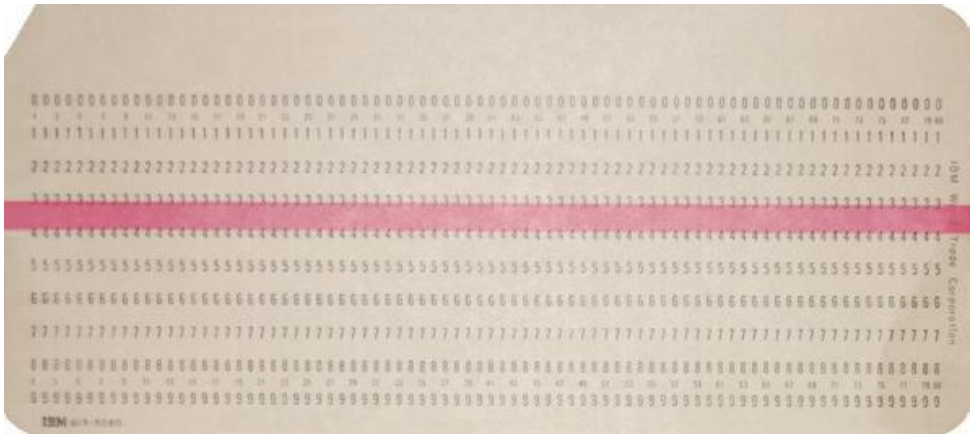

Figura 29

Es decir los programas son los que llevan a cabo las acciones y los datos son el paquete de información sobre el que actúan los programas.
Si lo pensamos un poco en las células ocurre algo similar.
Cuando analizábamos las funciones de las proteínas dijimos que las muchas de las proteínas que codificamos en el ADN son enzimas las cuales eran capaces de actuar sobre otras moléculas para producir reacciones específicas, y también dijimos que esas moléculas sobre las que actúan las proteínas se llaman sustratos.
Desde este punto de vista las enzimas serían las que llevan a cabo las acciones y los sustratos serían el paquete de moléculas sobre el que actúan las enzimas.
Es decir: las enzimas son los programas y los sustratos los datos.

La última…

Quizás una de las conclusiones más sorprendentes provenga de calcular, en términos de información, cuál es el tamaño del genoma humano.
En la primera década del siglo XXI se terminó finalmente de saber cuál es la secuencia de bases nitrogenadas en el genoma humano. Por supuesto esto difiere de persona a persona. No todos compartimos una secuencia exactamente igual, pero las diferencias son, porcentualmente, muy pequeñas.
Actualmente sabemos que el genoma humano contiene aproximadamente unos 3000 millones de pares de bases.
Si recordamos que cada codón contiene 3 bases esto significa que el ADN de un humano contendría aproximadamente 1000 millones de codones.
Por otro lado recodemos que desde el punto de vista de la información un codón sería equivalente a un byte, es decir que el genoma contendría 1.000.000.000 de bytes lo que no es otra cosa que 1 Gbyte. Es decir todo lo necesario para construir un sistema tan complejo como un ser humano está contenido en tan sólo 1 Gbyte.

Dr. Jorge Santiago