

Temas Generales para la preparación de la Oposición al Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado.

Cuerpo Superior de Estadísticos del Estado
Especialidad de Estadística-Ciencia de Datos.

Almacenamiento y modelos de datos

Tema 1. Representación y almacenamiento de la información.
Introducción. Bits y bytes. Organización de la memoria principal. Representación de la información como cadena de bits. Inexistencia de “tipos” para los ficheros en disco

AUTOR: Alexandra Vlad

Creación: Junio 2021

ÍNDICE

1	INTRODUCCIÓN	3
2	REPRESENTACIÓN Y ALMACENAMIENTO DE LA INFORMACIÓN. BITS Y BYTES. REPRESENTACIÓN DE LA INFORMACIÓN COMO CADENA DE BITS. INEXISTENCIA DE “TIPOS” PARA LOS FICHEROS EN DISCO	4
3	ORGANIZACIÓN DE LA MEMORIA PRINCIPAL	10
4	RESUMEN ESQUEMÁTICO	14
5	GLOSARIO	15
6	BIBLIOGRAFÍA BÁSICA.....	16

1 Introducción

Antes de saber cómo se representa la información, vamos a entender bien el concepto de información. Coloquialmente, el término información es sinónimo de conocimiento, de noticia, de datos, etc. Podríamos pensar que la información existe cuando hay comunicación de datos, pero en realidad no es así. En un proceso de comunicación la información que se adquiere depende mucho del receptor. Por ejemplo, si mil personas escuchan una noticia en la radio, donde se transmite una gran cantidad de datos sobre lo ocurrido en el día de ayer, no todos los oyentes recibirán el mismo nivel de información. Solo aumentará la información de aquellos oyentes que no conocen ya la noticia, mientras que para los demás la información recibida será nula. Así mismo, tampoco recibirán la información aquellos oyentes que no entienden el idioma en el que se está transmitiendo.

Un sistema de comunicación está formado por los siguientes elementos básicos



Emisor, fuente o transmisor: es el que genera o transmite la información.

Receptor: es el que recibe la información.

Medio o canal: vía de transmisión de la información.

Podemos entender la transmisión de la información entre el ser humano y el ordenador como una comunicación en la que el emisor es una persona y el receptor el ordenador, o viceversa, y el medio o canal son los periféricos de entrada y salida del ordenador, que son los dispositivos que se conectan al ordenador y que van a permitir introducir datos para que el ordenador los procese y transforme en información.

Así pues, la relación establecida entre datos e información a través de un proceso de datos se denomina *sistema de información*.

Con todo esto, podemos definir **información** como la representación de hechos, objetos, valores, ideas, etc., que permite la comunicación entre emisor y receptor, y la adquisición de conocimiento de las cosas.

Así, la transmisión de la información entre el ser humano y la computadora puede hacerse de muchas formas:

- Mediante caracteres alfanuméricos (letras {a, b, ..., z} y números {0, 1, ..., 9}). Por ejemplo, los introducidos al ordenador mediante teclado.
- Mediante sonidos: como los introducidos al ordenador a través de un micrófono, o que salen del ordenador por los altavoces.
- Mediante vídeos: como las imágenes obtenidas a través de una cámara de vídeo.
- Mediante gráficos e imágenes: por ejemplo, una imagen introducida por un escáner, o fotografías descargadas de una cámara de fotos digital.
- En general, cualquier tipo de dato enviado por un periférico del ordenador capaz de tomar datos de cualquier tipo y enviarlo al ordenador, o a la inversa.

En cada caso, el canal es diferente, y para proceder a la comunicación de los datos es necesario cambiar la forma en que estos se representan. Por tanto, los datos deben ser traducidos o codificados.

En conclusión, para que exista información es necesario que el envío de los datos y el que lo reciba se entiendan, es decir, que utilicen el mismo código; de lo contrario, necesitarán un traductor de un código al otro.

2 Representación y almacenamiento de la información. Bits y Bytes. Representación de la información como cadena de bits. Inexistencia de “tipos” para los ficheros en disco

Como hemos mencionado anteriormente, debe distinguirse entre dato e información. Dato es cualquier número, imagen, audio, video, etc. que no permite tomar ninguna decisión, porque no tienen un significado concreto, los datos deben procesarse para dotarlos de significado y convertirlos en información. Eso es lo que ocurre cuando trabajamos con un sistema informático, los resultados que nos muestra el ordenador son para nosotros información que nos ayuda en alguna tarea.

Los ordenadores almacenan la información como una colección de bits (ceros y unos). Hay que tener en cuenta que los ordenadores no entienden el lenguaje humano. Cualquier dato, letra, símbolo, imagen, audio, video, etc., que se envíen a un ordenador debe convertirse primero al lenguaje máquina.

La tecnología actual permite construir todo tipo de dispositivos que distingan entre dos estados diferentes (0's y 1's). Así pues, nuestra tecnología digital gestiona entradas y salidas y las procesa admitiendo únicamente dos valores diferentes (aun así, el concepto genérico de digital se refiere a un número finito que puede ser más de dos). Desde el punto de vista teórico, el hecho de admitir más de dos estados en un sistema digital haría inmensamente más eficiente y potente dicho sistema (por ejemplo, los sistemas cuánticos se basan en el control de más de dos estados de cada uno de sus elementos, pero por el momento siguen siendo dispositivos teóricos en los que a nivel práctico, el control de varios estados resulta muy costoso), pero por el momento nos vemos limitados a trabajar con bits (binary digits) que sólo aceptan 2 estados (podemos pensar en ellos como “encendido” y “apagado”).

Por este motivo, toda la información que se almacena y maneja en los ordenadores es información codificada en formato binario. Podemos pensar en la información que tenemos almacenada en una memoria digital. Esa memoria está compuesta por un gran conjunto de elementos (podemos pensar en casilla o celdas) que sólo son capaces de distinguir dos estados diferentes. Cualquier información que queramos almacenar en ella habrá que codificarla de alguna manera a base de cadenas de dichos dos estados. Toda la información, números, texto, colores, sonido, imagen... hay que codificarla a una secuencia de dos estados distintos que identificamos con 0 y 1 (cadena de bits).

Generalmente, en el disco duro de un ordenador, la información se almacena en archivos. Un archivo es un conjunto de bits (y también de bytes) que tiene un nombre único, y que para el sistema operativo de nuestro ordenador se entiende como una unidad lógica. Es importante resaltar aquí la diferencia entre la representación del archivo en el disco duro, que ocupará un número determinado de bytes (o de unidades de memoria física) y la representación del mismo para el sistema operativo, para el cual el archivo es una unidad.

Estos archivos pueden ser de muy diversos tipos (imágenes, sonidos, vídeos, ejecutables, archivos de configuración, archivos propios del sistema operativo...). Lo que pueda significar un bit o un byte en un disco duro depende directamente del tipo de archivo en que nos encontremos. De este modo, un byte en un fichero de imagen puede hacer referencia al color de un píxel, mientras que ese mismo byte en una base de datos puede representar un número cualquiera, o en un fichero de un procesador de textos puede contener información sobre la tipografía utilizada (Arial, Times, Verdana...).

Bits y Bytes

Bit

Un bit (binary digit: dígito binario) es la unidad de información más pequeña que puede almacenarse en un ordenador; puede tomar el valor 0 o 1. Un bit representa el estado de un dispositivo que puede

tomar uno de los dos estados. Por ejemplo, un interruptor puede estar ya sea apagado o encendido. La convención es representar el estado de encendido como 1 y el estado de apagado como 0. Un interruptor electrónico puede representar un bit. En otras palabras, un interruptor puede almacenar un bit de información. Actualmente los ordenadores utilizan varios dispositivos binarios de dos estados para almacenar datos.

Un solo bit no puede resolver el problema de la representación de datos, si cada pieza de datos (carácter) pudiera representarse por un 1 o un 0, entonces solo se necesitaría un bit. Sin embargo, se hace necesario almacenar números más grandes, texto, gráficos y otros tipos de datos. Es aquí donde se hacen necesarios los PATRONES DE BITS.

Para representar diferentes tipos de datos se utiliza un PATRON DE BITS, una secuencia o como a veces se le llama, una cadena de bits.

Ejemplo:

1 0 0 0 1 0 1 0 1 1 1 1 1 1

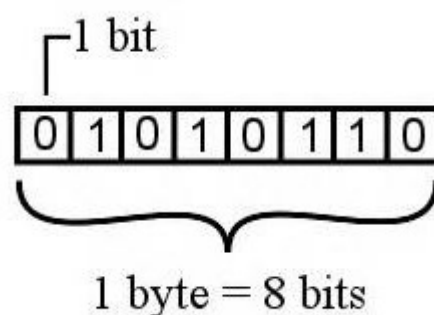
Esto significa que, si se quiere almacenar un patrón de bits formado por 16 bits, necesitan 16 interruptores electrónicos. Si quiere almacenar 1000 patrones de bits, cada uno de 16 bits, necesita 16 000 bits y así sucesivamente.

La memoria de los ordenadores no es capaz de identificar el tipo de datos. Ésta solo almacena los datos como PATRONES DE BITS. Es responsabilidad de los dispositivos de entrada/salida o de los programas interpretar un patrón de bits como un número, texto o algún otro tipo de datos. Los datos se codifican cuando entran a la computadora y se decodifican cuando se presentan al usuario.

Byte

Un patrón de bits con una longitud de 8 se llama BYTE. Este término es también utilizado para medir el tamaño de la memoria o de otros dispositivos de almacenamiento.

Por ejemplo, se dice que la memoria de un ordenador que puede almacenar 8 millones de bits de información es una memoria de 1 millón de bytes.



La siguiente tabla muestra la conversión de bits y bytes

Byte Value	Bit Value
1 Byte	8 bits
1024 Bytes	1 Kilobyte
1024 Kilobytes	1 Megabyte
1024 Megabytes	1 Gigabyte
1024 Gibabytes	1 Terabyte
1024 Terabytes	1Petabyte
1024 Petabytes	1 Exabyte
1024 Exabytes	1 Zettabyte
1024 Zettabytes	1 Yottabyte
1024 Yottabytes	1 Brontobyte
1024 Brontobytes	1 Geopbyte

Bits como números

La mayoría de los sistemas empleados para contar se basan en el sistema de numeración decimal. Eso quiere decir que empleamos 10 dígitos distintas (del 0 al 9) y cualquier número se construye por la combinación de n dígitos. El número de cifras es lo que se llama la base de numeración. El sistema decimal es un sistema en base 10.

- Si disponemos de 1 dígito decimal, podemos contar de 0 a 9, es decir, 10 combinaciones distintas.
- Si disponemos de 2 dígitos decimales, podemos contar de 0 a 99. Son 100 (10^2) combinaciones distintas.
- Si disponemos de 3 dígitos, podemos contar de 0 a 999, o sea, 1.000 (10^3) combinaciones distintas.
- En general, si disponemos de n dígitos, podremos diferenciar 10^n combinaciones distintas y podremos contar de 0 a 10^n-1 .

Los sistemas numéricos más conocidos son:

- Sistema numérico binario, consta de solo dos valores, 0 y 1.
- Sistema numérico octal, cuya base es 8 y utiliza los dígitos de 0 a 7.
- Sistema de numeración decimal, tiene como base el 10.
- Sistema numérico hexadecimal, tiene como base el 16.

Sistema numérico		
Sistema	Base	Dígitos
Binario	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

En la siguiente tabla se muestra la correspondencia entre las diferentes bases de numeración:

Decimal	Binario	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Bits como caracteres

Una secuencia de bits puede representar también los caracteres de algún alfabeto. Sin embargo, en este caso el código binario asignado a cada letra no es más que una convención, un acuerdo, y podría ser cualquier otra asignación. De hecho, como vamos a ver, existen varias maneras de representar los caracteres y todas ellas se consideran distintos estándares.

Los cuatro sistemas más populares son:

- **ASCII** Código Norteamericano de Estándares para Intercambio de Información (American Standard Code for Information Interchange). Este código utiliza 7 bits para cada símbolo. Esto significa 128 (2^7) símbolos distintos que pueden definirse mediante este código.
- **ASCII extendido**. Para hacer que el tamaño de cada patrón sea de 1 byte (8 bits), a los patrones de bits ASCII se les aumenta un cero más a la izquierda. Cada patrón cabe fácilmente en un byte de memoria.
- **EBCDIC**. Código Extendido de Intercambio decimal codificado en binario desarrollado por IBM al principio de la era de las computadoras. Utiliza patrones de ocho bits, de manera que puede representar hasta 256 símbolos. Solo es utilizado en computadoras Mainframes de IBM.
- **Unicode**. El estándar mundial de caracteres Unicode utiliza de 4 a 32 bits para representar letras, números y símbolos.
El número de caracteres que permite Unicode es tan elevado que el mismo ha dado lugar a distintas formas de codificación. Las tres más extendidas son:
 - UTF - 8. Codificación orientada a byte (de 8 bits u octeto) con símbolos de longitud variable (cada símbolo ocupará en memoria de 1 a 4 octetos). La codificación UTF - 8 permite representar no sólo 256 caracteres (un octeto) sino cualquier carácter que esté en Unicode. Esto es posible debido a la idea de codificación de longitud variable. Si queremos representar un carácter que esté entre los símbolos 0 y 127 de Unicode (que coinciden con los ASCII), la codificación utilizará en memoria un único octeto. Si queremos representar un carácter que esté fuera de dicho rango, su codificación utilizará de 2 a 4 octetos, 16 a 32 bits.
 - UTF - 16. Codificación de 16 bits de longitud variable (cada símbolo ocupará en memoria 1 ó 2 segmentos de 16 bits). Por tanto, desde el punto de vista de memoria, es bastante peor que UTF - 8 (por lo general, un texto que no contenga caracteres especiales, ocupará el doble que en UTF - 8). Sin embargo, permite codificar con dicha longitud de palabra $2^{16} = 65536$ caracteres, lo cual nos asegura que prácticamente cualquier carácter que queramos almacenar va a estar disponible. Gracias a que es de longitud variable, si pretendemos codificar algún carácter que no esté entre los 65536 disponibles, pero sí en las tablas Unicode (hasta la posición 10FFFFF), aumentará la longitud de palabra a 2 bytes.
 - UTF - 32. Codificación de longitud fija. Dispone de 232 (más de $4 * 10^9$) símbolos. Cada carácter ocupa en memoria 32 bits (4 octetos o bytes).

Bits como datos

Además de números o caracteres, cualquier información que procese el ordenador debe representarse en binario. Un fichero del procesador de textos, una presentación, una foto de las vacaciones, un archivo de audio de una canción en mp3, una película, un juego... todos ellos son ejemplos de información binaria.

En este caso, tenemos una gran variedad de formas de codificar y representar la información. Por ejemplo, una combinación de bits puede ser el color de un pixel en la pantalla del ordenador, un punto sobre el papel, una nota de una melodía o las coordenadas del puntero del ratón en la pantalla. Pero

incluso para un mismo tipo de archivo una lista de ceros y unos puede interpretarse de formas distintas. Una imagen puede estar almacenada como un bmp, jpg, gif, png, raw y muchos otros. Y cada uno de ellos, una misma combinación de bits se interpreta de distintas maneras.

Esta variedad en la representación de la información es la que provoca que, si tratamos de abrir un fichero con un programa distinto al que lo ha creado, éste nos indique que no puede abrir el fichero. O que si se ha producido un error en la transmisión de datos (por ejemplo, descargando un fichero), éste sea ilegible para el ordenador, ya que la secuencia binaria no tiene ningún sentido.

Bits como instrucciones

Pero los bits no se emplean para representar sólo los datos. El ordenador es una maquina digital todo lo que maneja está codificado en binario y que modifica esta información ejecutando las instrucciones de un programa. Y estas instrucciones también deben estar codificadas en binario.

En este caso, cómo se traducen las secuencias de bits en instrucciones es algo que se realiza dentro del procesador y que esta codificado directamente en sus circuitos.

3 Organización de la memoria principal

Antes de entrar en el detalle de lo que es la memoria principal de un dispositivo, vamos a ver los tipos de memoria que hay, así como su jerarquía dentro de una computadora.

En primer lugar, la memoria es cualquier dispositivo capaz de almacenar información (datos e instrucciones) en forma de bits (0's y 1's). Generalmente, la memoria se clasifica en dos categorías:

- Memoria volátil: necesitan de una fuente de energía para mantener la información. Pierde la información cuando se apaga la fuente de energía.
- Memoria no volátil: mantiene la información sin aporte de energía, cuando la fuente de energía se apaga no pierde ningún dato.

Clasificación de la memoria en función de la forma de acceso:

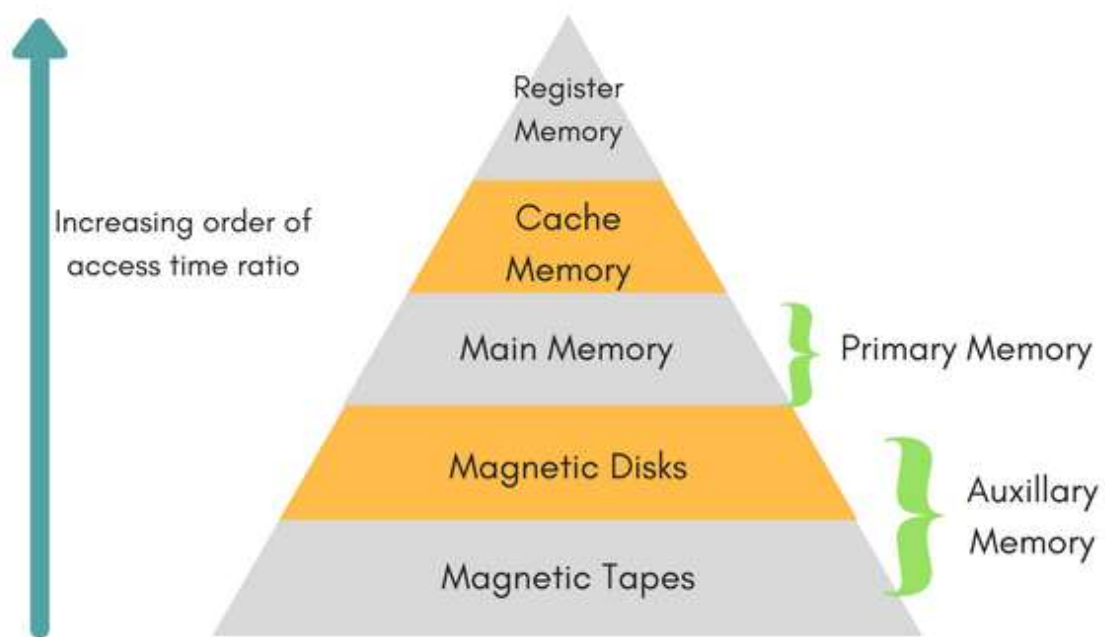
- Acceso aleatorio: La memoria se organiza como un vector, en el que cada elemento individual de memoria tiene una dirección única. Se accede a una posición determinada proporcionando la dirección. El tiempo de acceso es independiente de la posición a la que se ha accedido y es independiente de la última posición a la que se ha accedido. Este tipo de accesos se suele utilizar en memorias RAM (Random Access Memory) y ROM (Read Only Memory). La memoria principal del ordenador responde a este modo de acceso, y por eso generalmente se la conoce como RAM, y por tanto usar el término de memoria RAM para describir la memoria principal no es erróneo, tampoco es suficientemente preciso.
- Acceso secuencial: Se accede desde la última posición a la que se ha accedido, leyendo en orden todas las posiciones de memoria hasta llegar a la posición deseada. El tiempo de acceso depende de la posición a la que se quiere acceder y de la posición a la que se ha accedido anteriormente. Este tipo de acceso se utiliza básicamente en dispositivos de cinta magnética.
- Acceso directo: la memoria se organiza en bloques y cada bloque de memoria tiene una dirección única, se accede directamente al principio de un bloque y dentro de este se hace un acceso secuencial hasta llegar a la posición de memoria deseada. El tiempo de acceso depende de la posición a la que se quiere acceder y de la última posición a la que se ha accedido. El acceso directo es un método de acceso que se utiliza en discos magnéticos.
- Acceso asociativo: Se trata de un tipo de memoria de acceso aleatorio donde el acceso se hace basándose en el contenido y no en la dirección. Se especifica el valor que se quiere localizar y se compara este valor con una parte del contenido de cada posición de memoria; la comparación se lleva a cabo simultáneamente con todas las posiciones de la memoria. Este método de acceso se suele utilizar en las memorias caché.

Otra clasificación de la memoria es en función de las capacidades para leer y escribir en la misma:

- De solo lectura (ROM: Read Only Memory). Todas las memorias de este tipo no son volátiles, es decir mantienen la información de manera indefinida. Se parece más a un almacenamiento permanente de información. También almacena el programa cargador de arranque, para cargar e iniciar el sistema operativo cuando la computadora está encendida. Solo permiten lectura de su información, pero no su destrucción.
 - ROM (Memoria de solo lectura no programable): En el proceso de fabricación queda cargada con la información correspondiente.
 - PROM (Memoria de solo lectura programable): Se adquiere el dispositivo de memoria con una arquitectura general y es programable (se carga la información) mediante un sistema no reversible. Se puede escribir una única vez,

-
- De lectura/escritura (RWM: Read Write Memory)
 - No volátiles (mantienen la información de manera indefinida):
 - EPROM (Erasable PROM): Memorias que se pueden borrar usando luz ultravioleta y se reprograman eléctricamente.
 - EEPROM (Electrically Erasable PROM): Memorias que se borran y graban eléctricamente.
 - FLASH: permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Gracias a ello, la tecnología flash, mediante impulsos eléctricos, permite velocidades de funcionamiento superiores frente a la tecnología EEPROM.
 - Volátiles (necesitan una fuente de alimentación para mantener la información):
 - SRAM (RAM estática): los procesos de lectura son rápidos, pero admiten menos densidad de integración.
 - DRAM (RAM dinámica): los procesos de lectura son destructivos por lo que necesitan reescribir cada vez que se lee y esto las hace más lentas. Su capacidad de integración es mayor, hecho que permite obtener más capacidad en menos espacio.
 - SDRAM (RAM dinámica síncrona): generalmente las memorias DRAM son asíncronas, lo que quiere decir que los cambios en el estado de la memoria se producen tan rápido como se puede, pero sin sincronizarse con el resto de elementos de la placa. Sin embargo, las memorias síncronas (SDRAM en particular), se sincronizan con el reloj del sistema, y por tanto espera a las señales del reloj para responder a las peticiones de lectura y escritura de datos. Su comportamiento es más complejo, pero a la vez permite mayor velocidad.

A continuación, se muestra una imagen donde se refleja la jerarquía de memoria de una computadora que facilitará la visión sobre donde se ubica la memoria principal:



La capacidad total de memoria de una computadora se puede visualizar mediante la jerarquía de componentes. El sistema de jerarquía de memoria consta de todos los dispositivos de almacenamiento contenidos en un sistema informático, desde la memoria auxiliar lenta hasta la memoria principal rápida y la memoria caché más pequeña.

El tiempo de acceso a la memoria auxiliar es generalmente 1000 veces mayor que el de la memoria principal, por lo que se encuentra en la parte inferior de la jerarquía.

La memoria principal ocupa la posición central porque está equipada para comunicarse directamente con la CPU y con los dispositivos de memoria auxiliares a través del procesador de entrada / salida.

Cuando la CPU necesita un programa que no reside en la memoria principal, se incorporan desde la memoria auxiliar. Los programas que no se necesitan actualmente en la memoria principal se transfieren a la memoria auxiliar para proporcionar espacio en la memoria principal para otros programas que están actualmente en uso.

La memoria caché se utiliza para almacenar datos de programas que se están ejecutando actualmente en la CPU. La relación de tiempo de acceso aproximado entre la memoria caché y la memoria principal es de aproximadamente 1 a 7 ~ 10.

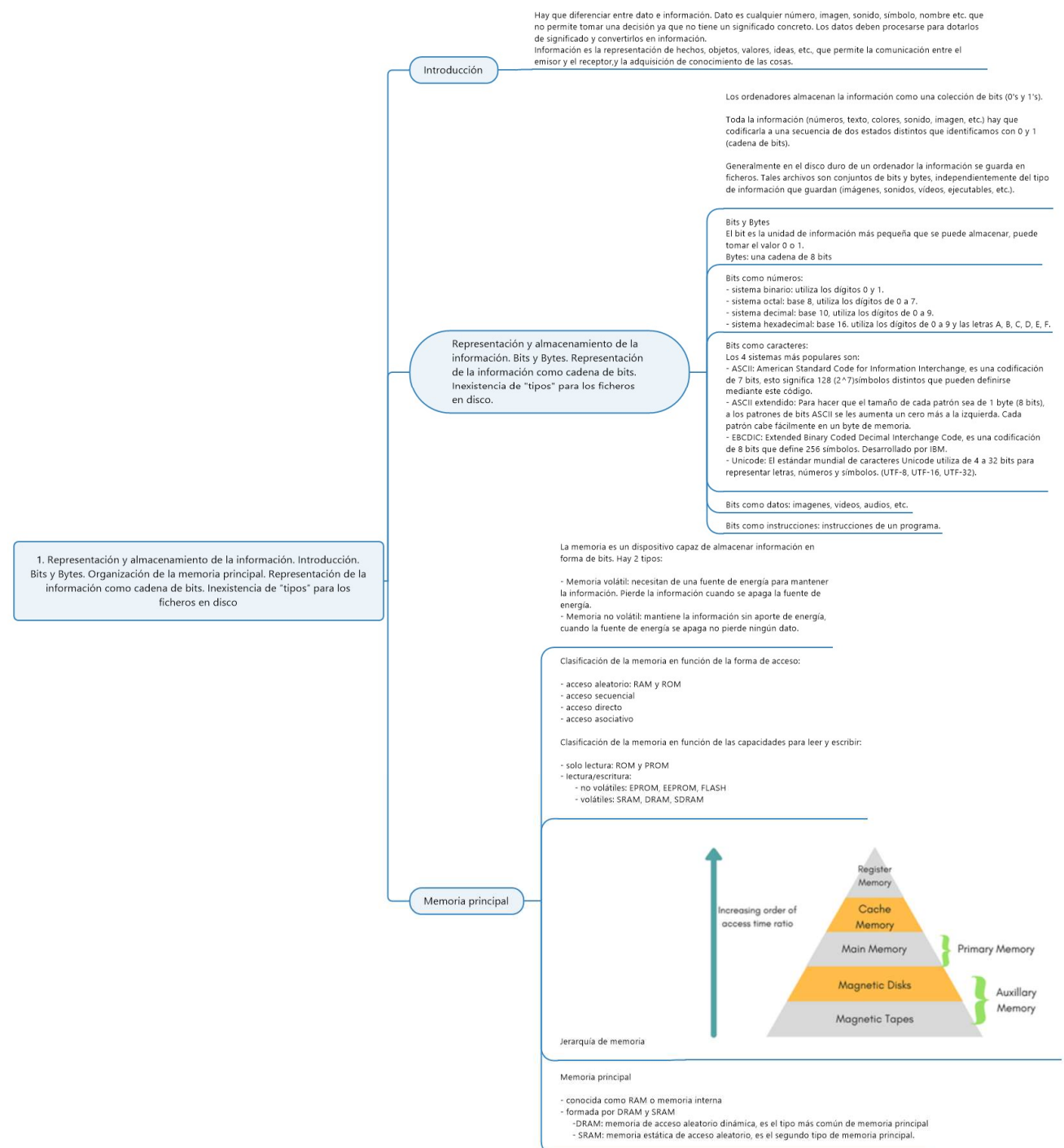
Como se ha mencionado anteriormente, la memoria principal se denomina de acceso aleatorio, también conocida como RAM. También se le conoce como memoria interna. Esta es la parte de la computadora que almacena el software del sistema operativo, aplicaciones de software y otra información para la unidad de procesamiento central (CPU) y así tener acceso rápido y directo cuando sea necesario para realizar las tareas correspondientes. Se llama "acceso aleatorio" porque la CPU puede acceder directamente a una sección de la memoria principal, y no debe emprender el proceso en un orden secuencial.

La memoria principal está disponible en dos tipos: la memoria dinámica de acceso aleatorio (DRAM) y la memoria estática de acceso aleatorio (SRAM).

DRAM La memoria de acceso aleatorio dinámico (DRAM) es el tipo más común de memoria principal en una computadora. Es una fuente de memoria prevalente en PCs, así como estaciones de trabajo. La memoria dinámica de acceso aleatorio está en constante restauración de toda la información que se está almacenando en la memoria. Refresca los datos mediante el envío de millones de pulsos por segundo a la celda de almacenamiento de memoria.

SRAM La memoria estática de acceso aleatorio (SRAM) es el segundo tipo de memoria principal en un ordenador. Se utiliza comúnmente como una fuente de memoria en los dispositivos integrados. Los datos contenidos en SRAM no tienen que estar continuamente renovados, ya que la información en esta memoria principal permanece como una "imagen estática" hasta que se sobrescribe o se elimina cuando la computadora está apagada. Ya que SRAM es menos densa y más eficiente de energía, cuando no está en uso, por lo tanto, es una elección mejor que la DRAM para ciertos usos como cachés de memoria ubicados en CPUs. Por el contrario, la densidad de DRAM hace que sea una mejor opción para la memoria principal.

4 RESUMEN ESQUEMÁTICO



5 GLOSARIO

ASCII American Standard Code for Information Interchange

EBCDIC Extended Binary Coded Decimal Interchange Code

UNICODE Unicode Standard

UTF Unicode Transformation Format

RAM Random Access Memory

ROM Read Only Memory

RWM Read Write Memory

PROM Programmable ROM

EPROM Erasable PROM

EEPROM Electrically Erasable PROM

SRAM RAM estática

DRAM RAM dinámica

SDRAM RAM dinámica síncrona

CPU Unidad central de procesamiento

6 BIBLIOGRAFÍA BÁSICA

- <https://www.danieldona.com/informatica%20basica/2%20representacion%20de%20la%20informacion.pdf>
- https://www.tutorialspoint.com/computer_concepts/computer_concepts_representation_data_information.htm
- https://computersciencewiki.org/index.php/Data_representation#:~:text=Computers%20use%20binary%20%2D%20the%20digits,and%201%20%2D%20to%20store%20data.&text=It%20is%20represented%20by%20a,up%20of%20billions%20of%20transistors.
- http://cv.uoc.edu/annotation/8255a8c320f60c2bfd6c9f2ce11b2e7f/619469/PID_00218275/PID_00218275.html
- https://www.unirioja.es/cu/jearansa/1112/ficheros/Tema_1.pdf
- <https://es.wikipedia.org/wiki/ASCII>
- <https://es.wikipedia.org/wiki/Unicode>
- https://m.riunet.upv.es/bitstream/handle/10251/10789/Representaci%C3%B3n_de_la_informaci%C3%B3n_digital_.pdf?sequence=1&isAllowed=y