TEMA 2. REPRESENTACIÓN Y COMUNICACIÓN

1. Sistemas Binario

El ser humano en su vida diaria almacena y recupera información utilizando distintos sistemas. El ordenador realiza esas mismas acciones de manera codificada en el sistema binario.

- **Código**: es un conjunto de normal y convenios que permiten transformar la información de una representación concreta a otra.

2. Sistemas de Numeración

Un sistema de numeración es un **conjunto de símbolos y reglas** que se utilizan para la representación de datos numéricos o cantidades. Se caracteriza por su **base**, que es el número de **elementos** de los que consta.

Los sistemas de numeración pueden ser:
- **Posicional**: aquellos en los que un mismo dígito tiene valor distinto según la posición que ocupe. Cualquier número.
- **No Posicional**: independientemente de la posición que ocupe, siempre toma el mismo valor. Sistema romano.

2.1 Sistema Decimal

El sistema decimal utiliza 10 símbolos del 0 al 9 por lo que tiene un sistema de base 10. Un mismo dígito tiene distinto valor según la posición que ocupe.

2.2 Sistema Binario

Es el sistema que utilizan internamente los ordenadores. Los símbolos que utiliza es el 0 y el 1. Pertenece al grupo de los posicionales ya que dependiendo de donde se sitúe el 0 y el 1, el numero cambia.

2.3 Sistema Octal

Es un sistema posicional en la que la base es 8 ya que utiliza los símbolos del 0 al 7.
Tiene una conversión a binaria muy sencilla ya que su potencia es de base 2.

2.4 Sistema Hexadecimal

Es otro sistema utilizado en informática cuando tenemos que trabajar con grandes cadenas de 0 y 1. Consta de 16 símbolos del 0 al 9 y de la A a la F, donde A=10, B=11, C=12, D=13, E=14 y F=15.

2.5 Conversión entre los Sistemas de Numeración

Existen algunos métodos clásicos para pasar de un sistema a otro:

2.5.1 Decimal a Binario, Octal y Hexadecimal

- **Decimal a Binario**: vamos dividiendo entre 2 y cogemos los restos y el último cociente.
- **Decimal a Octal**: vamos dividiendo entre 8 y cogemos los restos y el último cociente.
- **Decimal a Hexadecima**l: vamos dividiendo entre 16 y cogemos los restos y el último cociente.

2.5.2 Binario, Octal y Hexadecimal a Decimal

En los 3 casos tenemos que poner las potencias encima de cada dígito e ir multiplicando y luego sumar los totales.

2.5.3 Binario a Octal y Hexadecimal

- **Binario a Octal**: agrupamos bits de 3 en 3 empezando por la derecha y cada agrupación le asignamos un octal.
- **Binario a Hexadecimal**: agrupamos de 4 en 4 y a cada agrupación le asignamos un hexadecimal.

2.5.4 Octal y Hexadecimal a Binario

El proceso es el invertido del punto ***2.5.3***. Convertimos en binario cada cifra octal/hexadecimal y formamos grupos de 3 en 3 o 4 en 4 según el sistema que estemos convirtiendo.

3. Representación de Caracteres Alfanuméricos

Existen varios tipos de caracteres:
- **Alfabéticos**: de la “A” a la “Z” y de la “a” a la “z”.
- **Numéricos**: del 0 al 9.
- **Especiales**: * / ; : ñ …
- **Geométricos**: son caracteres que representan formas y figuras geométricas.
- **De Control**: NL (linea siguiente)

Es necesario unos **códigos** para convertir los caracteres a combinaciones de 0 y 1 para que el ordenador sepa lo que le estamos transmitiendo. A cada carácter se le **asocia** un número según un código, y ese número es el que pasamos a binario.

3.1 Códigos de Entrada y Salida (E/S)

Los códigos de E/S permitirán **traducir** la información a una **representación** que el ordenador pueda interpretar. Los más utilizados en informática son el **ASCII**, el **ASCII-Extendido** y el **Unicode**.

3.1.1 Código ASCII

Es el **código estándar americano** para el cambio de información. Es el **más extendido** y utilizado por los sistemas operativos. Utiliza grupos de **7 bits**, lo que supone **128 caracteres** diferentes. Del 0 al 31 son caracteres de control y del 32 al 127 son los imprimibles (letras y símbolos).

3.1.2 Código ASCII-Extendido

Para solventar la **carencia** de todos los países, el código ASCII-Extendido utiliza **8 bits** en vez de 7 para así poder representar **256 caracteres**. Los 128 primeros son los mismo que el ASCII, pero los 128 restantes son específicos de un país o región, aunque del 128 al 159 no son imprimibles.

Estos códigos están recogidos en una normal internacional aprobada por **ISO**. Estos son:
- **ISO 8859-1**: Latino 1 → Europa Occidental: catalán, danés, español, francés, ingles, noruego, etc.
- **ISO 8859-2**: Latino 2 → Europa Oriental: eslovaco, croata, rumano, polaco, etc.

3.1.3 Código Unicode

Cuando queremos comunicar ordenadores que se encuentran en diferentes regiones del planeta, surge el problema de que utilizan distintos códigos ASCII-Extendido. Aquí es donde aparece el código Unicode.
El **Unicode** proporciona un **número único a cada carácter**, independientemente de la plataforma, idioma o programa.

La norma **ISO 10646** es la que regula este código. Utiliza **16 bits**, el cual permite representar 2^{16} (**65536 caracteres**). Se representa con “**U+nº de código**”, siendo el nº de código el valor que se le asigna a cada carácter. El formato más utilizado es el UTF-8.

El **procedimiento** de conversión es:
1º Averiguamos a qué grupo pertenece el carácter.
2º Pasamos el numero de hexadecimal a binario.
3º Encajamos estos numero en el formato del grupo al que pertenezca.

Existen **4 grupos de rango** de código:
- **000000 – 00007F** → Símbolos de **1 byte** donde el bit más significativo es el 0
- **000080 – 0007FF** → Símbolos de **2 bytes**. El primero empieza por 110 y el segundo por 10.
- **000800 – 00FFFF** → Símbolos de **3 bytes**. El primero empieza por 1110 y el resto por 10.
- **010000 – 10FFFF** → Símbolos de **4 bytes**. El primero empieza por 11110 y el resto por 10.

4. Representación Interna de Gráficos

Las imágenes también pueden ser codificadas al lenguaje de un ordenador. Pueden ser **mapas de bits** (están compuestas por píxeles) o **vectoriales** (se trabaja con nodos y tensores).

4.1 Mapa de Bits

Un **pixel** es la unidad mínima de información gráfica que se puede mostrar en una pantalla. El pixel está formado por puntos que no se visualizan a simple vista.

- **Resolución**: nos da una idea de la calidad de la imagen. Expresa el total de pixeles que tiene la imagen en filas y columnas. Se **calcula** los pixeles de "**ancho x alto**" y lo **multiplicamos** por la **superficie** de la imagen.
- **Profundidad de color**: es el número de bits usados para cada pixel y nos determina la cantidad de colores que puede mostrar la imagen. Se calcula en base de 2^x.

Las **ventajas** son: **mayor realismo** en la imagen y es el utilizado en fotografía.
Los **inconvenientes** son: **ocupan** mucho **tamaño** y tiene una **menor escalabilidad**. La calidad de imagen varía al cambiar la resolución.

Los **formatos** que se utilizan en los mapas de bits son:
- **Para Internet**: ".jpg", ".gif", ".png".
- **Para Edición**: ".psd", ".raw".
- **Para Impresión**: ".tiff", ".jpg".

El **tamaño** del archivo dependerá de la resolución (**R**), las dimensiones de la imagen (**Largo x Ancho**) y la profundidad de color (**P**). Se utiliza la siguiente fórmula: **Tamaño = R^2 x L x A x P**.

Si el tamaño que estamos calculado está en bits y lo queremos en bytes, recordemos que tendremos que **dividir el resultado entre 8**.

4.2 Gráficos Vectoriales

Una **imagen vectorial** es una imagen digital formada por objetos geométricos independientes, cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.

Las **ventajas** son: **menor tamaño**, **gran escalabilidad** y **fácil control** de forma.
Los **inconvenientes** son: **menor realismo**.

Los **formatos** que se utilizan en las imágenes vectoriales son:
- **.ai**: se utiliza en Adobe Illustrator.
- **.svg**: se utiliza como lenguaje de programación.
- **.eps**: es el más común. Pertenece a Adobe.
- **.pdf**: formato de documento y como imagen vectorial.

Un **metafichero** permite guardar contenidos de varios tipos en un mismo fichero, como fotografías, textos, vectores e incluso sonido y video.

Hay que tener en cuenta, que la calidad de fotografía no se basa en la resolución, si no en la **calidad del sensor** que utilice esa cámara.

5. Representación Interna del Sonido

El sonido es una onda y como tal, está caracterizada por:
- **Longitud de onda**: es la **distancia entre dos crestas**. Se mide en **metros**.
- **Amplitud (A)**: es la **distancia entre una cresta y la posición de equilibrio**. Se mide en **decibelios**.
- **Frecuencia (f)**: es el **numero de veces que se repite la señal** en 1 segundo. Se mide en **hercios**.
- **Periodo (T)**: es el **tiempo que tarda en realizar una oscilación completa**. Se mide en **segundos**.

El **oído humano** percibe sonido cuando las frecuencias se encuentran en el rango entre **infrasonidos** (20 Hz) ay **ultrasonidos** (20 KHz).

Podemos diferenciar 3 tipos de tonos:
- **Graves**: Frecuencias bajas desde 16 Hz hasta 256 Hz.
- **Medios**: Frecuencias medias desde 256 Hz hasta 2 KHz.
- **Agudos**: Frecuencias altas desde 2 KHz hasta 16 KHz.

5.1 Digitalización de la Señal

La onda de señal continua es analógica, por lo que tenemos que convertirla en formato digital.
Para convertirla, seguimos los siguientes pasos:
1º Se toma muestras de la señal analógica en intervalos iguales (**frecuencia de muestreo**). Cuanto mayor sea la frecuencia de muestreo, mayor será la calidad del sonido.
2º Se asigna un valor digital a cada una de las muestras (**cuantificación**). Cuantos más bits utilicemos, más parecido será al sonido analógico.

Los **parámetros** básicos para describir la secuencia de muestras que representa un sonido son:
- **Número de Canales**: 1 para mono, 2 para estéreo, 4 para el sonido cuadrafónico, etc.
- **Tasa de Muestreo**: es el número de muestras en cada canal.
- **Número de bits por muestra**: suelen ser de 8 (2^8 valores), 16 bits (2^{16} valores → convencional) y 22 bits (2^{22} valores → alta fidelidad).

5.2 Tamaño de un Sonido Almacenado

El tamaño es igual al numero de canales multiplicado por la calidad de muestro (bits) por la frecuencia (Hz) por la duración (segundos).

5.3 Formatos

Tipos de Audio	WAV	AIFF	PCM	FLAC	ALAC
Tipo	Sin Comprimir	Sin Comprimir	Sin Comprimir	Comprimido Pérd	Comprimido Pérd
Extensiones	.wav .wave	.aiff .aif	.pcm .au	.flac	.m4a .caf
Fuente Abierta	Parcialmente	Parcialmente	Si	Si	Parcialmente
Reducción Tamaño	No	No	No	60%	60%
Popularidad	Medio	Bajo	Alto	Medio	Alto

Tipos de Audio	WMA	MP3	OGG	AAC	WMA (pérd)
Tipo	Comprimido Pérd	Comprimido Pérd	Comprimido Pérd	Comprimido Pérd	Comprimido Pérd
Extensiones	.wma .wmv	.mp3	.ogg .oga	.aac .m4r	.wma
Fuente Abierta	Si	Si	Si	Parcialmente	No
Reducción Tamaño	60%	90%	90%	90%	90%
Popularidad	Alto	Más Popular	Bajo	Alto	Alto

6. Representación de Video

Desde un punto de vista informático, un video es una sucesión de imágenes a una determinada frecuencia (fps), con sonido.

Un fotograma o frame es cada una de las imágenes que forman un video. Se expresan con las siglas fps y en hercios (Hz).

Los **formatos** más utilizados son **.mp4**, **.avi**, **.flv**, **.dvd**.

7. Byte y Múltiplos

- **Bit**: unidad mínima de información digital.
- **Byte**: Conjunto de 8 bits.

Unidades Básicas de Información (en bytes)						
Prefijos del Sistema Internacional				Prefijo Binario		
Múltiplo	**Símbolo**	**Estándar**	**Binario**	**Múltiplo**	**Símbolo**	**Valor**
Kilobyte	KB	10^3	2^{10}	Kibibyte	KiB	2^{10}
Megabyte	MB	10^6	2^{20}	Mebibyte	MiB	2^{20}
Gigabyte	GB	10^9	2^{30}	Gibibyte	GiB	2^{30}
Terabyte	TB	10^{12}	2^{40}	Tebibyte	TiB	2^{40}
Petabyte	PB	10^{15}	2^{50}	Pebibyte	PiB	2^{50}
Exabyte	EB	10^{18}	2^{60}	Exbibyte	EiB	2^{60}
Zettabyte	ZB	10^{21}	2^{70}	Zebibyte	ZiB	2^{70}
Yottabyte	YB	10^{24}	2^{80}	Yobibyte	YiB	2^{80}

- Cuando estamos en la tabla de byte, para pasar a un múltiplo superior o inferior, tendremos que multiplicar o dividir entre **1000**.

- Cuando estamos en la tabla de bits, para pasar a un múltiplo superior o inferior, tendremos que multiplicar o dividir entre **1024**.

- Para pasar de un prefijo a otro, solo tendremos que coger el numero que nos da el fabricante, que suele estar en bytes, y multiplicarlo por el estándar que le corresponda y dividirlo por el binario correspondiente.