

TEMA 1
**Representación y comunicación de la
información**

Contenidos:

- **Información: definición y elementos**
- **Representación de la información**
- **Comunicación de la información**

TEMA 1.- REPRESENTACIÓN Y COMUNICACIÓN DE LA INFORMACIÓN

CONTENIDOS

1. Información: definición y elementos

1.1. Elementos de un sistema de información

2. Representación de la información

2.1 Representación de números enteros

2.2 Representación de códigos intermedios

2.3 Representación de números reales

2.4 Representación de caracteres

2.5 Representación de imágenes

3. Comunicación de la información

3.1 Elementos de un sistema de comunicación digital

3.2 Encriptación de datos en las comunicaciones

3.3 Compresión de datos en las comunicaciones

3.4 Redundancia de datos

3.5 Clasificación de los sistemas de transmisión

3.6 Problemas en la transmisión

“El éxito no se logra solo con cualidades especiales. Es sobre todo, un trabajo de constancia, de método y de organización” - J.P. Sergent (Pintor)

1. Información: definición y elementos

Definimos un **dato** como una colección de hechos considerados de forma aislada, es decir, los datos tienen un significado, pero en general no son de utilidad por sí solos. En sentido general, la **información es un conjunto organizado de datos procesados que constituyen un mensaje** sobre un determinado fenómeno.

Actualmente el **procesamiento de la información**, gracias a la evolución de la informática y las nuevas tecnologías se ha convertido en **uno de los puntales más importantes de nuestra sociedad hasta el punto de hablar de Sociedad de la información** y comparar el cambio que ha experimentado nuestra sociedad con la incorporación de los ordenadores y la expansión de las redes de comunicaciones en todos los ámbitos de la vida. Las **TIC** (Tecnologías de la Información y las Comunicaciones) constituyen hoy el **mismo factor de transformación que en su momento** tuvieron la **imprenta o la máquina de vapor** durante la Revolución Industrial.

De hecho **una de las industrias actuales más importantes es la Industria de la comunicación de datos y todo lo asociado a las TIC es un sector muy pujante** en la actualidad.

1.1. Elementos de un sistema de Información

Para el **tratamiento de la información**, se requiere que sea transmitida de un lugar a otro, y para esto son necesarios los siguientes elementos:

- ↳ **El emisor**: fuente u origen de la información
- ↳ **El medio o canal de transmisión**: mediante el cual se realiza la transmisión de la **información**. Hay de muchos tipos siendo uno de los más importantes el aire pero también sistemas como cables de cobre, fibra óptica, canal IDE, canal SATA, buffer de memoria, canal de radiofrecuencia, etc ...
- ↳ **El receptor**: quien **recibe la información**
- ↳ **El mensaje**: la **información transmitida**. Entre el emisor y el receptor **deben utilizar un lenguaje y una forma de actuar que sea entendido por ambas partes**. Al lenguaje se le **llama código** y a la **forma de actuar en la transmisión de información** se **llama protocolo**.

2. Representación de la información

Un **ordenador** es una máquina que procesa información. La **ejecución de una tarea** implica la realización de unos **tratamientos**, según especifica **un conjunto ordenado de instrucciones** (es decir, un programa) **sobre unos datos**. Para que el ordenador ejecute un **programa** es necesario darle **información de dos tipos**:

- ↳ **Instrucciones que forman el programa**
- ↳ **Los datos con los que debe operar ese programa**

Entonces, ¿cómo se representa la información con la que trabaja el computador? Según la RAE, los **datos son la representación de la información de manera adecuada para su tratamiento por un ordenador**. De hecho, uno de los **aspectos más importantes** relacionado con la información, es **cómo representarla**. Normalmente se le da al ordenador en la forma usual escrita que utilizan los humanos, es decir, con ayuda de un alfabeto o conjunto de símbolos, los **caracteres**.



Podemos considerar que existen **dos niveles en la representación de la información**

- **Nivel de representación externa:** usada por las personas e inadecuada para el ordenador
- **Nivel de representación interna:** adecuada al ordenador y no inteligible directamente por el ser humano.

Los **caracteres** que se utilizan para la **representación externa** son:

- **Numéricos:** constituidos por las diez dígitos en el sistema decimal
- **Alfabéticos:** letras mayúsculas y minúsculas
- **Especiales:** símbolos no incluidos en los grupos anteriores, como:), (, *, /, +, -, [,]...
- **De control:** representan órdenes de control, como el carácter indicador de fin de línea. Muchos de ellos son generados e insertados por el propio ordenador.
- **Gráficos:** símbolos con los que se pueden representar figuras elementales.

Veremos, a continuación, cómo **estos caracteres usados en la representación externa son representables en los ordenadores (representación interna)**. Este paso de una representación a otra se denomina **codificación** y el proceso inverso **decodificación**.

Los **elementos básicos que constituyen un ordenador son de naturaleza binaria**, ya que solo pueden adoptar **dos valores, 0 y 1** (corresponden a dos niveles de tensión, dos valores de corriente, dos situaciones de una lámpara...). Los sistemas binarios son más fiables, por su **mayor inmunidad frente al ruido, y más sencillos de construir** al contemplar únicamente dos valores, de ahí su **idoneidad para la representar información en un ordenador**.

Al tener que traducir toda la información suministrada a ceros y unos, es **necesario establecer una correspondencia entre el conjunto de todos los caracteres: {A, B, C, D,...Z, a, b, c,...z, 0, 1,...9, /, +,...}** y el **conjunto binario: {0, 1}ⁿ** de forma que **a cada elemento del primero le corresponda un elemento distinto del segundo**. Por ejemplo, si queremos representar las cinco primeras letras mayúsculas del alfabeto, podemos establecer la siguiente correspondencia:

Carácter	Código
A	100
B	011
C	110
D	111
E	101

Como se ve en este ejemplo, se ha establecido una correspondencia entre el conjunto {A, B, C, D, E} y el conjunto {0, 1}3. Aquí hemos utilizado el número mínimo de bits necesarios para representar estas 5 letras, es decir, tres (en este caso, n debe ser al menos 3).

Para realizar las **operaciones aritméticas sobre datos numéricos, el propio ordenador efectúa una transformación de la representación en códigos de E/S a una representación basada en el sistema de numeración en base 2**, que resulta más adecuada.

A la hora de hablar de cómo representar la información, **hay que distinguir previamente qué tipo de información queremos representar:** números enteros, reales, caracteres,

imágenes... En función del tipo de información a representar, deberemos decantarnos por unos métodos de representación u otros:

- ↳ Representación de **números enteros**
- ↳ Representación de **códigos intermedios**
- ↳ Representación de **números reales**
- ↳ Representación de **caracteres**
- ↳ Representación de **imágenes**

2.1. Representación de números enteros

Para empezar a **comprender cómo procesa información un ordenador**, debemos **entender primero cómo representar las cantidades**. Para poder cuantificar la información **se desarrollaron los sistemas de numeración** que pueden definirse de la siguiente manera:

- ↳ Un **sistema de numeración** es el **conjunto de símbolos y reglas que se utilizan para la representación de los datos numéricos o cantidades**.
- ↳ Un sistema de numeración **se caracteriza fundamentalmente por su base**, que es el **número de símbolos distintos que utiliza**, y además es el coeficiente que determina cuál es el valor de cada símbolo de acuerdo a la posición que ocupa.
- ↳ Los sistemas de numeración actuales **son sistemas posicionales que representan cualquier número como una secuencia de cifras**, contribuyendo **cada una de ellas con un valor que depende de la cifra en sí y de su posición en la secuencia**. Un número N se representaría como una secuencia de cifras: $N \rightarrow n_2, n_1, n_0, n_{-1}, n_{-2}$

Siendo el valor numérico del número N interpretado en base b :

$$N \rightarrow n_2 * b^2 + n_1 * b^1 + n_0 * b^0 + n_{-1} * b^{-1} + n_{-2} * b^{-2}$$

Ejemplo:

Base = 10 \Rightarrow Conjunto de símbolos = {0, 1, 2, 3, 4, ..., 9}
 $3278,52 = 3000 + 200 + 70 + 8 + 0,5 + 0,02 = 3*10^3 + 2*10^2 + 7*10^1 + 8*10^0 + 5*10^{-1} + 2*10^{-2}$

El sistema que utilizamos **habitualmente** es el **sistema decimal**, aunque es posible usar cualquier base. **Nos centraremos en** las siguientes:

- ↳ Base 2 ($b = 2$): **sistema binario natural**. Alfabeto = {0, 1}.
- ↳ Base 8 ($b = 8$): **sistema octal**. Alfabeto = {0...7}.
- ↳ Base 10 ($b = 10$): **sistema decimal**. Alfabeto = {0...9}.
- ↳ Base 16 ($b = 16$): **sistema hexadecimal**. Alfabeto = {0...9, A...F}.

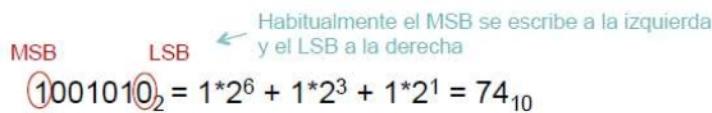
En un sistema con **base b** los **dígitos posibles son: 0, 1, ..., $b-1$** . Con n dígitos se pueden representar b^n números posibles, desde el 0 hasta el b_{n-1} .

Sistema binario

En este sistema la **base es 2**. Permiten **representar perfectamente la información** en los **sistemas digitales**. Los **dígitos posibles son 0 y 1**. Un dígito en sistema binario se denomina “bit”.

Con n bits se pueden representar 2^n números, donde el **bit de mayor peso** se denomina **bit más significativo** o **MSB** (“Most Significant Bit”), y el **bit de menor peso** se denomina **bit menos significativo** o **LSB** (“Least Significant Bit”). Ejemplo:



MSB LSB Habitualmente el MSB se escribe a la izquierda


$$1001010_2 = 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^1 = 74_{10}$$

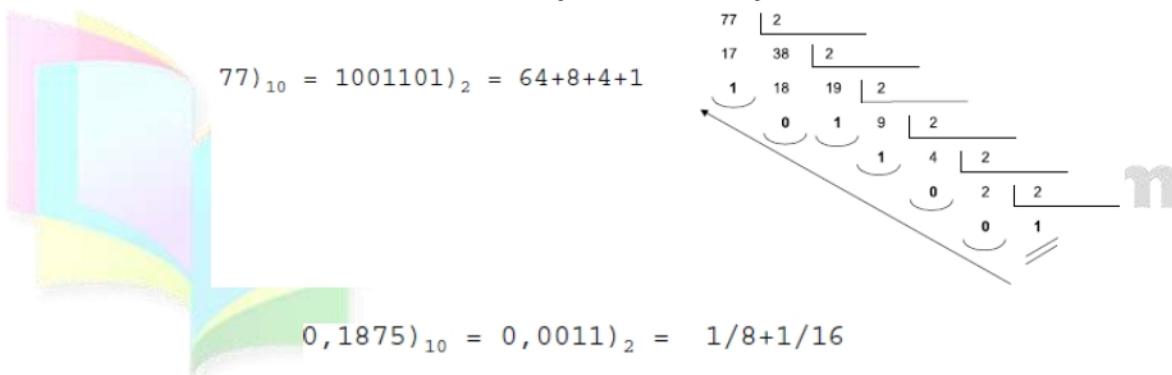
Transformaciones de base binaria a decimal

Para transformar un número representado en el **sistema binario** a su representación en **decimal**, únicamente hay que aplicar el **teorema fundamental de la numeración**, visto un poco más arriba.

Transformaciones de base decimal a binaria

Para transformar un número de decimal a binario, se transforman de forma **independiente** la **parte entera** y la **parte fraccionaria**, y **después se concatenan** los resultados obtenidos.

- Parte entera: ir dividiendo entre 2 la parte entera del número decimal original y, sucesivamente, los cocientes que se van obteniendo. Los restos de estas divisiones y el último cociente son las cifras binarias. El último cociente es el bit más significativo y el primer resto el menos significativo.
- Parte fraccionaria: ir multiplicando por 2 la parte fraccionaria del número decimal original y, sucesivamente, las partes fraccionarias de los números obtenidos. El número binario se forma con las partes enteras que se van obteniendo.



$$\begin{array}{r}
 0.1875 \quad 0.3750 \quad 0.75 \quad 0.5 \\
 \times 2 \quad \times 2 \quad \times 2 \quad \times 2 \\
 \hline
 0.375 \quad 0.75 \quad 1.5 \quad 1.0
 \end{array}$$

Operaciones aritméticas

Para la realización de **operaciones aritméticas** en el **sistema binario** se utilizan las siguientes tablas:

SUMA	RESTA	PRODUCTO	DIVISIÓN
$0 + 0 = 0$	$0 - 0 = 0$	$0 \cdot 0 = 0$	$0 / 0 = \text{Indetermin.}$
$0 + 1 = 1$	$0 - 1 = 1$ y debo 1	$0 \cdot 1 = 0$	$0 / 1 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$1 \cdot 0 = 0$	$1 / 0 = \infty$
$1 + 1 = 0$ y llevo 1	$1 - 1 = 0$	$1 \cdot 1 = 1$	$1 / 1 = 1$

Operaciones lógicas

Las **operaciones lógicas** en el sistema binario se realizan bit a bit y, para ello, se utilizan las siguientes tablas:

SUMA LÓGICA (OR)	PRODUCTO LÓGICO (AND)	COMPLEMENTO (NOT)	OPERACIÓN XOR
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\overline{0} = 1$	$0 \oplus 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\overline{1} = 0$	$0 \oplus 1 = 1$
$1 + 0 = 1$	$1 \cdot 0 = 0$		$1 \oplus 0 = 1$
$1 + 1 = 1$	$1 \cdot 1 = 1$		$1 \oplus 1 = 0$

2.2. Representación de códigos Intermedios

SISTEMA OCTAL

En este sistema la **base es 8**, por tanto, los dígitos posibles son $\{0, 1, 2, 3, 4, 5, 6, 7\}$ y con **n dígitos se pueden representar hasta 8^n números**. El objetivo es poder expresar de forma más abreviada secuencias de 0's y 1's.

Transformaciones de base binaria a octal y de octal a binaria

Las conversiones octal-binario y binario-octal pueden hacerse fácilmente. El **paso de binario a octal se efectúa formando grupos de tres cifras** ($b=8=2^3$) **a derecha e izquierda del punto decimal y efectuando la conversión a octal de cada grupo individual** (basta con memorizar el equivalente binario de cada cifra octal). Para pasar **de octal a binario, cada cifra octal se convierte individualmente a binario** manteniendo el orden del número original.

Ejemplo: pasar el número octal $17352,16_8$ a binario.

Se toma cada dígito octal y se sustituye por la expresión binaria correspondiente:

Octal:	1	7	3	5	2	,	1	6
Binario:	001	111	011	101	010	,	001	110

Con esto nos queda: $00111011101010,001110_2$ (los ceros a la izquierda y a la derecha se pueden eliminar).

Este código se utiliza cuando el número de bits a representar es múltiplo de 3, para aprovechar mejor la capacidad de almacenamiento.

Transformaciones de base octal a decimal y de decimal a octal

Para pasar **de octal a decimal**, basta con aplicar el teorema fundamental de la numeración. Para pasar **de decimal a octal**, se pasan **independientemente la parte entera** (realizando **divisiones sucesivas entre 8**) y **la parte fraccionaria** (realizando **sucesivas multiplicaciones por 8**) y después **se concatena el resultado** (de forma similar al caso binario).

$$137_8 = 1 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0 = 95_{10}$$

SISTEMA HEXADECIMAL

Se tiene que **la base es $b=16$ y, por tanto, es necesario disponer de un alfabeto de 16 símbolos: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$** . Algunas de las notaciones más habituales de los números en sistema hexadecimal son:

$$23AF16 = 23AF_{\text{hex}} = 23AF_{\text{h}} = 0x23AF = 0x23 0xAF$$



Un dígito hexadecimal permite representar lo mismo que 4 bits (ya que $2^4=16$). **Un dígito hexadecimal se denomina también “nibble”.** **Dos dígitos hexadecimales** equivalen por tanto a **8 bits**. El conjunto de 8 bits o dos dígitos hexadecimales, se denomina **“byte”**.

Transformaciones de base binaria a hexadecimal y de hexadecimal a binaria

Las conversiones **hexadecimal-binario** y **binario-hexadecimal** se realizan prácticamente **de la misma manera que en el caso octal**. La única diferencia estriba en que las **agrupaciones** deben ser en este caso de **4 bits**, en lugar de grupos de 3 bits.

Ejemplo: pasar a binario el número hexadecimal $7B3, F_{16}$.

Se toma cada dígito hexadecimal y se sustituye por la representación binaria correspondiente:

Hexadecimal:	7	B	3	,	F
Binario:	0111	1011	0011	,	1111

Con esto nos queda: $01110110011,1111_2$ (los ceros a la izquierda y a la derecha se pueden eliminar)

Transformaciones de base hexadecimal a decimal y de decimal a hexadecimal

Las **conversiones entre el sistema decimal y hexadecimal se realizan igual que en el sistema binario**, sin más diferencia que **tener en cuenta** que, en este caso, la **base es 16**. Para pasar de hexadecimal a decimal, basta con aplicar el teorema fundamental de la numeración. Para pasar de decimal a hexadecimal, se pasan independientemente la parte entera (realizando sucesivas divisiones entre 16) y la parte fraccionaria (realizando sucesivas multiplicaciones por 16) y después se concatena el resultado.

2.3. Representación de números reales

Para la **representación de números reales**, normalmente se utiliza la **notación en coma flotante** (también llamada **punto flotante o notación exponencial**) de la siguiente forma:

$$N = M \times B^E \quad \text{donde } M = \text{mantisa}, \quad B = \text{base}, \quad E = \text{exponente}$$

Es posible cambiar la representación de N, sin cambiar su valor, simplemente aumentando (disminuyendo) en una unidad E y dividiendo (multiplicando) M por B.

$$13257,35 = \left\{ \begin{array}{l} 13257,35 \times 10^0 \\ 1,325735 \times 10^4 \\ 0,1325735 \times 10^5 \\ 1325735 \times 10^{-2} \end{array} \right.$$

Habitualmente, se utiliza la **representación normalizada IEEE754**, cuyo esquema es el siguiente:

s	e	m
1 bit	n_e bits	n_m bits

donde el significado de cada campo es:

1. **Campo de signo:**

- ↳ $S = 0 \rightarrow$ N° positivo.
- ↳ $S = 1 \rightarrow$ N° negativo.

2. **Campo del exponente:**

- ↳ entero sesgado, donde $S = 2^{n_e-1} \rightarrow e = E + S$

3. **Campo de la mantisa:**



- When the **most significant bit of the mantissa is in position 0 (units)**, the number is said to be **normalized**.
 $101,111 \times 2^{-5} = 1,01111 \times 2^{-3}$; $0,0011 \times 2^7 = 1,1 \times 2^4$
- The field of the mantissa only stores the fractional part of the normalized number**, because the 1 is always present (it is said to be **implied**).
- When the processor reads the data from memory, it will add 1:
 - So $1 \leq M < 2$ $M = 1.m$
 - Example: $101,111 \times 2^{-5} = 1,01111 \times 2^{-3}$ $m = 01110...0$

Ejemplo:

Calculate the real number in base 10 corresponding to the 32-bit sequence $C0580000_{16}$ which is in IEEE 754 single precision (n=32 bits, with $n_e=8$, then $n_m=23$):

- $C0580000_{16} = 1100\ 0000\ 0101\ 1000\ 0000\ 0000\ 0000\ 0000_2$
 $1\ 10000000\ 101100000000000000000000_2$
- $s=1 \rightarrow$ negative (-)
- $e=10000000_2=128_{10}$; $e=E+S$; $S=2^7-1=127_{10} \rightarrow E=e-S=128-127=1_{10}$
- $m=101100000000000000000000_2 \rightarrow M=1.m=1.10110..0_2 \rightarrow M=1.1011_2$
- $N=-M \cdot 2^E = -1.1011_2 \cdot 2^1 = -11.011_2 \cdot 2^0 = -11.011_2 = -3.375_{10}$

2.4. Representación de caracteres

For the **representation of characters** are used the so-called **input/output codes** or **external codes**, codes that associate to each character (alphanumeric, numeric or special) a **determined combination of bits**. In other words, an **I/O code** is a correspondence between two sets:

$$\begin{aligned}\alpha &= \{0, 1, 2, \dots, 9, A, B, \dots, Y, Z, a, b, \dots, y, z, *,), (, /, %, $, \dots\} \\ \beta &= \{0, 1\}^n\end{aligned}$$

The number of elements m of α depends on the device or computer system that is being used. In general, with n bits we can encode $m = 2^n$ different symbols.

There are **normalized I/O codes** that are usually used by computer manufacturers:

- BCD interchange normalized code** (Standard Binary Coded Decimal Interchange Code): normally, this code uses 6 bits, so it can represent $2^6 = 64$ characters. Sometimes a bit is added to the left to verify errors in transmission or recording (parity bit), so each character is represented by 7 bits.
 The four positions on the right are called **position bits**. The two following bits to the left are called **zone bits**, which are 00 for numerical characters. The last bit is optional for verification.

Verificación	Bits de zona			Bits de posición		
6	5	4	3	2	1	0

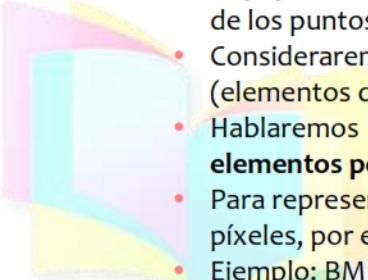
- EBCDIC** (Extended Binary Coded Decimal Interchange Code): uses 8 bits to represent each character, so it can encode up to 256 different symbols. This allows representing a large variety of characters: includes lowercase letters and many special symbols. It is also possible (using combinations that start with 00) to encode **control characters** that provide **commands or control signals**.



- **Código ASCII** (American Standard Code for Information Interchange): **utiliza 7 bits** (128 caracteres representables). Este código **es de los más utilizados**, se puede decir que la mayor parte de las transmisiones de datos entre dispositivos se realizan en esta codificación. **Se suele incluir un octavo bit a la izquierda para detectar posibles errores de transmisión o grabación** (bit de paridad). El código **ASCII de 8 bits** se denomina **código ASCII extendido**.
- **UNICODE**: **creado en 1991 para tener códigos alfanuméricos estándar, comunes en todas las regiones** (se utiliza el mismo código UNICODE para idiomas Chino, Árabe, etc.). Es un **código de 16 bits, el más moderno, desarrollado como estándar**. Es una **ampliación del código ASCII** de 8 bits, **para poder codificar distintos juegos de caracteres** (latino, griego, árabe, kanji, cirílico, etc). Algunas versiones son: UTF-8, UTF-16,...

2.5. Representación de Imágenes

A la hora de codificar una imagen podemos utilizar **diferentes procedimientos**:

- 
- **Mapas de bits**: las **imágenes están compuestas por infinitos puntos**, donde **cada punto tiene un nivel de gris o un color**.
 - Hay que determinar **cuántos puntos vamos a considerar** ya que no podemos almacenar y procesar infinitos puntos
 - Hay que **asociarles un nivel de color o de gris**: el nivel asociado será la media de los puntos representados
 - Consideraremos que **una imagen está formada por una matriz de píxeles** (elementos de imagen).
 - Hablaremos de **resolución de la imagen** en términos de **número de elementos por línea * Número de elementos por columna**
 - Para representar una imagen se almacenan sucesivamente los atributos de los píxeles, por ejemplo de izquierda a derecha y de arriba abajo
 - Ejemplo: BMP, JPEG,...
 - **Mapas de vectores**: las **imágenes se representan como colecciones de objetos** (líneas, polígonos, textos,...) y **los objetos se modelan mediante vectores y ecuaciones**.
 - Al **visualizar la imagen en pantalla** se **evalúan las ecuaciones y se escalan los vectores para generar la imagen**
 - Son **adecuadas para gráficos geométricos**, pero no para imágenes reales
 - Ocupan menos espacio que los mapas de bits, son más fáciles de procesar o escalar, pero la calidad de la imagen es peor.
 - Ejemplos: EPS, TrueType...

Las imágenes pueden representarse en escala de grises o en color, donde en la representación en escala de grises, cada píxel tiene asociado un valor medio de gris

Normalmente **cada píxel puede tomar 256 valores distintos (1 byte)** y las **representaciones de color más conocidas son**:

- **RGB**: la intensidad media de los colores primarios rojo (R), verde (G) y azul (B) se codifica por separado.
- **CYMK**: el color se representa mediante el cian (C), amarillo (Y), magenta (M) y negro (K).

3. Comunicación de la información

La **transmisión de información tiene carácter unidireccional y unilateral**, lo que altera la esencia interactiva del proceso comunicativo, o sea, **el proceso comunicativo deja de ser tal si no garantiza interacción social, intercambios, interinfluencias, retroalimentación**. Así, el **proceso humano a través del cual se transmiten ideas, opiniones, puntos de vista de un emisor a un receptor, sin que medie la interacción social, la interinfluencia y la retroalimentación se considera transmisión de información**.

El **proceso comunicativo incluye la transmisión de información, la que propicia o favorece la interacción social, el intercambio de sentimientos, opiniones, o cualquier otro tipo de información mediante el habla, la escritura u otros códigos**, y tiene carácter bilateral y bidireccional, posibilita la apertura y cierre de ciclos comunicativos y en todos los casos hay retroalimentación.

En el proceso de transmisión de información, el mensaje es un fin y en el proceso comunicativo el mensaje es un medio.

Así, gestionando las posibles **relaciones entre usuario y ordenador** tendremos:

- ↳ El usuario introduce datos en el ordenador para que sean procesados **y produzcan** unos resultados de **salida**.
- ↳ El **usuario debe codificar sus ideas** mediante el código de un lenguaje reconocido por el ordenador.
- ↳ El **usuario debe utilizar un dispositivo adecuado o canal de transmisión para transmitir el mensaje al ordenador**.
- ↳ Para que el **mensaje llegue íntegro** a la memoria del ordenador debe transmitirse **sin perturbaciones**. Esto hace necesario utilizar un adaptador periférico-ordenador adecuado al dispositivo transmisor empleado.
- ↳ Para poder “enterarse” de la idea del usuario, subyacente en el mensaje, **el ordenador debe traducirlo a lenguaje máquina**.

Por último, en la **comunicación usuario-ordenador** tenemos:

- ↳ **El sistema operativo del ordenador hace que se procese la información recibida** en el sentido esperado por el usuario, siempre que su arquitectura y filosofía le confieran capacidad para realizarlo, y que no hayan entrado elementos perturbadores junto con el mensaje.
- ↳ **El ordenador debe transmitir, utilizar o almacenar los resultados obtenidos en la forma que le haya sido indicada**, para lo que tendrá que utilizar un adaptador adecuado.

El proceso de comunicación de la información, se puede resumir con la siguiente fórmula:

Comunicación = intercambio de información = Emisor, Receptor y Canal o Sistema de transmisión



3.1. Elementos de un sistema de comunicación digital



1. El **emisor** es el encargado de generar la información a transmitir. Dependiendo del tipo de información, esta se **codifica**, mediante la transformación más adecuada. Luego, si es necesario, los datos se **encriptan** para que no puedan ser interceptados. Opcionalmente, se **comprimen** para reducir su tamaño y así aumentar la velocidad de transmisión. Antes de enviar los datos, se **realiza una nueva codificación** orientada a que el receptor pueda detectar y **corregir errores**. Para enviar los datos, estos se transforman en una señal analógica que pueda ser **enviada a través del canal de comunicación**.
2. En la **recepción** se realiza el proceso inverso. Se **decodifica** la señal analógica, se **detectan y corrigen** posibles fallos. Si procede se **descomprimen** los datos y se **desencriptan**. Y por último se **decodifica** la información de forma que esta pueda ser **comprendida por el receptor**.

NOTA: la codificación de la información la hemos visto en el punto anterior de representación de la información.

3.2. Encriptación de datos en las comunicaciones

La aparición de la Informática y el uso masivo de las comunicaciones digitales, han producido un número creciente de problemas de seguridad. Las **transacciones** que se **realizan a través de la red** pueden ser **interceptadas**, y por tanto, la **seguridad de esta información** debe **garantizarse**. Este desafío ha generalizado los objetivos de la **criptografía** para ser la parte de la **criptología** que se encarga del estudio de los **algoritmos, protocolos** (se les llama protocolos criptográficos), y **sistemas** que se **utilizan para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican**.

El tipo de propiedades de las que se ocupa la criptografía son por ejemplo:

- **Confidencialidad:** garantiza que la información está **accesible únicamente a personal autorizado**. Para conseguirlo **utiliza códigos y técnicas de cifrado**.

- ↳ **Integridad:** garantiza la **corrección y completitud de la información**. Para conseguirlo puede usar por ejemplo **funciones hash**.
- ↳ **No repudio:** permite **vincular un documento o transacción a una persona o un sistema de gestión criptográfico automatizado**. Cuando se trata de una persona, se trata de asegurar su conformidad respecto a esta vinculación (content commitment) de forma que pueda entenderse que la vinculación gestionada incluye el entendimiento de sus implicaciones por la persona.
- ↳ **Autenticación:** es decir **proporciona mecanismos que permiten verificar la identidad del comunicador**. Para conseguirlo puede usar por ejemplo función hash criptográfica MAC.

3.3. Compresión de datos en las comunicaciones

Opcionalmente, estos **símbolos codificados** (y encriptados si es necesario) **pueden ser comprimidos con el fin de conseguir una velocidad de transferencia mayor**.

Dentro de los diferentes algoritmos de compresión **tenemos dos variantes: con pérdida o sin pérdida de información**:

1. **Compresión sin pérdida (Lossless):** que, a su vez se dividen en dos tipos:
 - a. Compresores estadísticos: algunos ejemplos de este tipo de compresores son:
 - ↳ **Compresores del tipo Huffman o Shannon-Fano:** codificación inversamente proporcional a la probabilidad de aparición del mensaje.
 - ↳ **Compresores aritméticos:** compresor también basado en la probabilidad de aparición de un mensaje, pero en este caso para la representación se utiliza un número en coma flotante.
 - ↳ **Compresores predictivos:** pretenden predecir el siguiente mensaje a transmitir partiendo del conocimiento que se tiene de los mensajes enviados hasta el momento.
 - b. Compresores basados en diccionario o sustitucionales:
 - ↳ **Compresión RLE:** compresor de los menos eficaces pero de los más sencillos. Se basa en la sustitución de caracteres repetidos por un carácter más el número de repeticiones. De dicho carácter
 - ↳ **Compresores Lempel-Ziv (LZ):**
 - LZ78: esta técnica consiste en la elaboración dinámica de un diccionario donde se irán almacenando las cadenas aparecidas hasta el momento y a las que se le irán asignando identificadores, de forma que si aparece una cadena que ya está en el diccionario, se enviará su identificador.
 - LZ77: este método guarda un registro de caracteres enviados, pero no construyen un diccionario propiamente dicho, sino que mantienen una "historia (o ventana)" de la entrada y un "buffer de adelantamiento". Este método de compresión es utilizado en la mayoría de los compresores actuales, por ejemplo, hacen uso de él PK-ZIP, RAR, ARJ, etc.
 - c. Compresores híbridos o de dos fases: los compresores actuales **utilizan ambos métodos (estadísticos y sustitucionales)** para **comprimir de manera más óptima**.
2. **Compresión con pérdida de información:** se utiliza sobre todo para la **codificación de archivos multimedia**, dónde no es tan importante que la información recibida sea exacta y sí es muy importante la capacidad de comprensión.
 - a. Codificación diferencial: la **secuencia de valores son representados como la diferencia con respecto a un valor previo**.



- b. Compresión basada en transformadas: se basa en la **representación de la imagen muestreada en términos de contenido de frecuencias de la imagen**.
- c. Cuantización de vectores: **división de la imagen en rectángulos de tamaño fijo y se basa en la utilización de un "libro de códigos" que se crea previamente**, a raíz de las estadísticas de un grupo de imágenes de prueba que son similares en contenido a la imagen que se va a comprimir.
- d. Compresión fractal: similar a la anterior técnica, pero en vez de seleccionar el bloque del libro más parecido, **se puede aplicar una transformación geométrica a los bloques del libro de códigos para hacerlos coincidir mejor con el bloque de píxeles que está siendo evaluado**.
- e. Técnicas de compresión de imágenes en movimiento (interframe): la compresión se basa por la **eliminación de redundancia que se produce por las mínimas diferencias entre imágenes sucesivas**.

3.4. Redundancia de datos

Las **redes** deben ser capaces de **transferir datos de un dispositivo a otro con total exactitud**, si los datos recibidos no son idénticos a los emitidos, el sistema de comunicación es **inútil**. Sin embargo, siempre que se transmiten de un origen a un destino, se pueden corromper por el camino. Los **sistemas de comunicación** deben tener **mecanismos para detectar y corregir errores que alteren los datos recibidos** debido a múltiples factores de la transmisión. La detección y corrección de errores **se implementa bien en el nivel de enlace de datos o bien en el nivel de transporte del modelo OSI**.

Por lo tanto, **debemos asegurarnos** que si dicha transferencia de información **causa errores**, éstos puedan ser **detectados**. El **método para detectar y corregir errores** es **incluir en los bloques de datos transmitidos bits adicionales denominados redundancia**.

Se han desarrollado **dos estrategias básicas para manejar los errores**:

- ↳ **Incluir suficiente información redundante en cada bloque de datos para que se puedan detectar y corregir los bits erróneos.**
- ↳ **Incluir sólo la información redundante necesaria en cada bloque de datos para detectar los errores.** En este caso el número de bits de redundancia es menor.

Si consideramos un **bloque de datos formado por m bits de datos y r de redundancia**, la longitud final del bloque será **n**, donde **$n = m + r$** .

El **bit de paridad (par o impar)** es un ejemplo de redundancia para detectar errores. El **inconveniente que tiene es que no es capaz de detectar un cambio en más de un bit**. Hay **otros tipos de métodos** capaces de corregir dichos fallos, uno de estos fallos consiste en añadir un **bit de paridad vertical por cada grupo de bytes**, cada uno de los cuales lleva su propio bit de paridad. Si el error fuera múltiple este se podrá detectar pero no corregir.

Existen **métodos más efectivos** como son los **códigos de Hamming** que son **capaces de detectar errores múltiples y corregir errores sencillos** utilizando en proporción menos bits de paridad. Estos códigos **consisten en añadir varios bits de paridad colocados en las posiciones que son potencia de 2, de forma que cada uno proteja a varios bits del dato**. Estos códigos son cada vez más utilizados en las memorias.

Si lo que se pretende es la **detección de errores en las comunicaciones serie**, existen **códigos especiales** ya que los errores producidos suelen afectar a varios bits

consecutivos. Estos códigos llamados **polinomiales o de redundancia cíclica (CRC)** consisten en **añadir a cada bloque un residuo (módulo) con respecto a un valor concreto representado por un polinomio**.

Los **códigos de Huffman** presentan la particularidad de emplear códigos de longitud variable, de manera que el número de bits utilizados para codificar un carácter va a depender de la frecuencia de utilización.

3.5. Clasificación de los sistemas de transmisión

El sistema de transmisión puede ser clasificado según diferentes criterios:

- ↳ **Según sistema de transmisión:**
 - Por línea: cable de par trenzado, coaxial, fibra óptica.
 - Por radio: ondas terrestres y ondas vía satélite.
- ↳ **Según la direccionalidad de la transmisión:**
 - Simplex: la comunicación se efectúa en un solo sentido.
 - Semi-duplex: existe comunicación en ambos sentidos pero no simultáneamente.
 - Full-duplex: hay comunicación en ambos sentidos de manera simultánea.
- ↳ **Según la forma de sincronización:**
 - Asíncrona: emisor y receptor se sincronizan cuando se envía un carácter.
 - Síncrona: se puede enviar la señal de sincronismo a través de una línea independiente o bien utilizar una codificación auto sincronizada, de manera que el dato incluya una señal de temporización.
- ↳ **Según la naturaleza de la señal:**
 - Analógicos: señal que toma valores continuos en el tiempo.
 - Digitales: señal que toma un número finito de valores (si sólo toma 2 se llaman señal digital binaria).

3.6. Problemas en la transmisión

Los problemas asociados a la transmisión están **relacionados con la naturaleza del medio físico de transporte**, cuyo efecto en conjunto es la **degradación de la señal transmitida**, que llega alterada al punto de recepción.

Los **problemas asociados a la transmisión de señales**, se pueden enunciar como:

- ↳ **Distorsión**: consiste, en **alteraciones de la forma de la señal**, debidos las características del medio.
- ↳ **Atenuación o pérdidas de potencia**: consiste en la **pérdida del nivel de la señal**, debido al desgaste.
- ↳ **Alteración de la temporización**: causados por **retardos**, y tiempos de transmisión diferentes para distintas formas de señales.
- ↳ **Ruido**: son las **señales no deseadas que acompañan a la señal enviada**, y que aparecen en recepción debido a su paso por el medio, y que en ocasiones pueden ocultarla, y, en todo caso dificultar su detección.

