

Representación interna de los datos.

TEMA 10

ABACUS NT

Oposiciones 2021

Índice

- 1. Introducción**
- 2. Representación interna de los datos**
 - 2.1. Definiciones.**
 - 2.2. Justificación de la existencia de diferentes sistemas de representación.**
- 3. Sistemas de numeración y codificación. Unidades y magnitudes informáticas.**
 - 3.1. Sistemas de numeración.**
 - 3.1.1. Sistemas de Numeración Posicionales.**
 - 3.1.2. Teorema Fundamental de la numeración.**
 - 3.1.3. El sistema decimal**
 - 3.1.4. El sistema binario**
 - 3.1.5. Códigos intermedios.**
 - 3.2. Representación interna de la información.**
 - 3.2.1. Representación numérica**
 - 3.2.2. Representación alfanumérica (Textos) : Códigos de E/S**
 - 3.2.3. Representación de instrucciones**
 - 3.2.4. Representación de Imagen y video**
 - 3.2.5. Representación de sonido**
 - 3.3. Unidades y magnitudes informáticas.**
 - 3.3.1. Unidades de medida para el almacenamiento de información.**
 - 3.3.2. Unidades de medida para la transmisión de información.**
 - 3.3.3. Unidades de medida para imagen**
 - 3.4. Redundancia en las codificaciones binarias. Bits de control.**
 - 3.5. Compresión de los datos**
 - 3.6. Criptografía. Confidencialidad en las comunicaciones.**
- 4. Conclusión.**
 - 4.1. Sistema educativo. Relación con el currículo.**
- 5. Bibliografía**

1. Introducción

En el presente tema, vamos a estudiar y analizar los **aspectos fundamentales** relacionados con la representación y comunicación de la información.

Es un tema de suma importancia ya que la misma palabra “informática” deriva del término francés “informatique”: **información automática**, y que según la RAE es el conjunto de conocimientos científicos y técnicas que permiten el tratamiento de la información de forma automática.

Un computador, computadora u ordenador, es una máquina electrónica capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas y proporcionar la información resultante a través de un medio de salida; todo ello de forma automática, es decir, sin necesidad de la intervención de un operador humano, y bajo el control de un programa previamente almacenado en el propio ordenador.

Es importante considerar que este flujo de información se realiza de forma **digital** por lo general. El término dígito (etimológicamente deriva de dedo en latín) nos da una idea aproximada de que la representación no va a tomar cualquier valor del mundo real (señales analógicas) sino sólo un número discreto de estos al igual que el número de dedos de nuestras manos es limitado, al igual que el sistema decimal (base diez).

Ligado íntimamente a la representación interna de la información está el **concepto de bit**. Por simplicidad y velocidad, los ordenadores internamente sólo van a poder almacenar dos estados (voltaje o ausencia de este, magnetizado o no, refleja un láser o no, etc.) que matemáticamente vamos a representar con **un cero o un uno**. De aquí aparece surge el concepto de dígito binario, Binary digit o Bit, ya que en inglés esa palabra tiene connotaciones de algo muy pequeño, esta será la unidad más pequeña de medida de la información.

2. Representación interna de los datos

2.1. Definiciones.

Para ser rigurosos, es necesario explicar los siguientes términos que vamos a utilizar:

- **Bit:** Unidad elemental de almacenamiento de información en los ordenadores, puede tomar 2 valores (1 ó 0).
- **Sistema de representación:** Conjunto de reglas y símbolos que determinan como se codifica una determinada información.

2.2. Justificación de la existencia de diferentes sistemas de representación.

Existen **diferentes sistemas** de representación, ya que estos nos van a condicionar:

- El coste de **almacenamiento** de la información

- El coste de **traducción** de la información
- El coste de **tratamiento** de la información

Además, para **seleccionar** un sistema es importante tener en cuenta la **naturaleza de la información** que se desea representar, ésta se clasifica según los autores Alberto Prieto, Antonio Lloris, y Juan Carlos Torres, en su obra Introducción a la Informática, en:

- Textos.
- Valores numéricos.
- Instrucciones.
- Sonidos.
- Imágenes y vídeos.

Dependiendo de estos factores, un determinado sistema resultará más **adecuado** que otro. A continuación, vamos a profundizar en los detalles de cada uno de los tipos de información.

3. Sistemas de numeración y codificación. Unidades y magnitudes informáticas.

3.1. Sistemas de numeración.

Podemos definir un sistema de numeración como un conjunto de símbolos y reglas: $N=(S,R)$

3.1.1. Sistemas de Numeración Posicionales.

Existen sistemas de numeración no posicionales, como por ejemplo el utilizado para representar los números romanos o el chino, y sistemas de numeración posicionales, en los que la **posición de un símbolo dentro del número** determina su valor.

3.1.2. Teorema Fundamental de la numeración.

Para conocer el valor que representa un número en un **sistema posicional**, se calcula aplicando el **teorema fundamental de la numeración**, el cual dice que dado un número $N=\{x_n \dots x_0, x_{-1} \dots x_{-d}\}$, su valor en base B es:

$$N = \sum_{i=-d}^n x_i B^i$$

3.1.3. El sistema decimal

El ser humano tiene 10 dedos. No es casualidad que el sistema de numeración más utilizado tenga diez símbolos, del 0 al 9.

Alguno de los **sistemas de numeración** más utilizados es:

- Binario (base 2). Es el sistema de representación de los ordenadores
- Decimal (base 10). Es el sistema de representación numérico del hombre.
- Hexadecimal (base 16)

3.1.4. El sistema binario

El ordenador utiliza el sistema binario por su sencillez de representación. La Base del sistema es $B=2$, y el conjunto de símbolos del lenguaje es $L=\{0,1\}$

Para transformar un **número de base binaria a decimal** se puede aplicar el teorema fundamental de la numeración, por ejemplo:

$$10)_2 = 1 * 2^1 + 0 * 2^0)_{10} = 2$$

A su vez, para transformar un **número entero de base decimal a base binaria**, se obtiene dividiendo por 2, y después se realizan sucesivamente divisiones por los cocientes obtenidos, hasta llegar a un cociente menor a 2. El resultado es el último cociente junto con los restos obtenidos en el orden inverso de aparición. Otra forma de calcularlo es mediante el **método de Ruffini**.

Para calcular la **parte decimal**, esta se multiplica por dos y vamos cogiendo la parte entera resultante en orden de aparición hasta obtener cero o tener la precisión requerida.

3.1.5. Códigos intermedios.

Otros códigos comúnmente utilizados relacionados con los ordenadores son los llamados códigos de representación intermedia, que nos permiten **representar de forma práctica** múltiples informaciones. Estos son el Octal ($B=8$, $L=\{0,\dots,7\}$) y el Hexadecimal ($B=16$, $L=\{0,\dots,9,A,\dots,F\}$)

Estos códigos tienen la particularidad de ser **muy fáciles de convertir en binario** simplemente usando la equivalencia de 1 símbolo a 3 o a 4, respectivamente de octal a binario o de hexadecimal a binario.

3.2. Representación interna de la información.

Existen diferentes sistemas de representación, ya que estos nos van a **condicionar**:

- El coste de **almacenamiento** de la información
- El coste de **traducción** de la información
- El coste de **tratamiento** de la información

Además, para **seleccionar** un sistema es importante tener en cuenta la **naturaleza de la información** que se desea representar, ésta se clasifica según los autores *Alberto Prieto, Antonio Lloris, y Juan Carlos Torres*, en su obra *"Introducción a la Informática"* (2006), en:

- Textos.

- Valores numéricos.
- Instrucciones.
- Sonidos.
- Imágenes y vídeos.

Dependiendo de estos factores, un determinado sistema resultará **más adecuado** que otro.

3.2.1. Representación numérica

Un primer enfoque al problema de representar un número signado de n-bits es la representación en **Signo y Magnitud**:

- Se utiliza el MSB (Bit más significativo) para representar el número que, por convención: un 0 denota un número positivo, y un 1 denota un número negativo;
- Los (n-1)-bits restantes para representar el significando que es la magnitud del número en valor absoluto.

Un problema de esta representación es la **necesidad de una operación condicional** al operar con ellos para determinar su signo y otro la **doble representación del cero** (1000000 y 0000000)

Mucho más utilizado es la representación en **Complementos**, que nos permite una aritmética más sencilla.

- **Complemento a 1** (a la base menos 1): Se invierten los ceros y unos del significado -equivale a restarle la base menos 1-, lo cual **nos permite operar aritméticamente**, aunque hay que tener en cuenta el acarreo y sigue teniendo doble representación para el cero. Es utilizada por los protocolos TCP, UDP, ICMP, y otros. p.eje: $1010_2 = -5_{10}$
- **Complemento a 2** (a la base): Se suma 1 al complemento a 1 del número, lo cual nos da una notación muy efectiva para operar aritméticamente. Es la notación más utilizada. p.eje: $1011_2 = -5_{10}$

Para representar números **reales o muy grandes** se suele utilizar la **notación punto flotante**, la cual se basa en la propiedad de que todo número n se puede representar por $n = m \cdot B^e$ donde m es la mantisa, B es la base (no es necesario explicitar este valor) y e es el exponente.

Actualmente, existe la **norma IEEE 754** de aceptación prácticamente universal, que representa número reales con 32 o con 64 bits.

3.2.2. Representación alfanumérica (Textos) : Códigos de E/S

Para la representación numérica se utiliza la denominada **Codificación de entrada/salida**, que asocia a cada **carácter** una determinada combinación de bits.

Según el autor *J. Glenn Brookshear, en su libro Introducción a la computación*, en la década de los 40 y 50 se empezaron a utilizar una variedad muy amplia de diferentes códigos de E/S, lo que generó

la existencia de problemas de comunicación, es por ello que actualmente se utilizan **códigos de E/S normalizados**, como, por ejemplo:

- **BCD** (Binary Coded Decimal) Para representar números en decimal dígito a dígito utilizamos 4 bits. Esto simplifica enormemente el hardware. Tiene varias variantes como el código Aiken o el XS3 - Exceso a 3.
- **EBCDIC** (Extended BCD Interchange Code): Utiliza 8 bits. Representa minúsculas, mayúsculas, símbolos especiales y caracteres de control.
- **ASCII** (American Standard Code for Information Interchange): Hay dos versiones, la primera Utiliza 7 bits de datos y 1 de paridad. La segunda es el código ASCII extendido que utiliza 8 bits para datos.
- **UNICODE**: los símbolos están organizados por páginas, las cuales corresponden a diferentes idiomas o a un conjunto de símbolos determinado. Utiliza un número variable de bits (8, 16 o 32) dependiendo del UTF -Unicode Transformation Format- utilizado, ya que de esta forma podemos operar con menos información para códigos conocidos sin estar limitados en la cantidad de símbolos potencialmente representables. Actualmente es el código más utilizado.
- **HTML**: Dada la naturaleza internacional de las codificaciones en la web, el lenguaje de marcas de hipertexto utiliza su propia notación para eliminar barreras internacionales, aunque también pueden ser salvadas mediante el uso adecuado de UNICODE. En HTML para representar un carácter especial como un letra acentuada, pongamos la “é” por ejemplo, se puede utilizar el código: “é”; para la “Ú”: “Ú”; para la “ñ”: “ñ”

3.2.3. Representación de instrucciones

Las instrucciones de un lenguaje en alto nivel o en lenguaje ensamblador se dan en forma de texto, representándose en consecuencia con un código de E/S, y el traductor correspondiente, ya sea un compilador o un intérprete, se encarga de transformarlas en instrucciones máquina que internamente se representarán como cadena de bits.

3.2.4. Representación de Imagen y video

La codificación de imágenes estáticas se puede realizar de formas: mediante **mapas de bits** (indicando para cada bit de la pantalla su color) o mediante **representación vectorial** (Indicando únicamente los vértices y color de cada polígono que queramos generar). Es frecuente comprimir las imágenes para que ocupen menos tamaño reduciendo la **profundidad de color**, la **resolución** de la imagen, o el número de pixeles a representar mediante la **aproximación de zonas de bits de igual color**.

Un video siempre se va a formar a partir de imágenes estáticas, siendo importante el número de imágenes (marcos) que se representan **FPS** y teniendo en cuenta que la compresión también se

puede realizar mediante **transiciones** entre imágenes, es decir, manteniendo los elementos que no cambian y guardando información de los que sí.

3.2.5. Representación de sonido

Para representar sonido es necesario digitalizarlo. Este proceso sólo puede tomar un número discreto de valores del mundo real, por lo que se realiza un **muestreo** que consiste en tomar un cierto número limitado de valores (*frecuencias en el tiempo*) de dicho sonido. En la calidad de sonido también va a influir la **amplitud** de la muestra, es decir los límites que consideremos a la misma.

3.3.Unidades y magnitudes informáticas.

3.3.1. Unidades de medida para el almacenamiento de información.

Como veíamos al comienzo del tema, el **bit** es la unidad más pequeña para medir cualquier información, pero igual que no medimos la distancia entre ciudades en centímetros, tampoco vamos a usar el bit, sino múltiplos suyos, como norma general.

De hecho, la mínima cantidad de información que un ordenador va a tratar será de **8 bits**, esto es **1 Byte**, y los múltiplos del mismo utilizando prefijos griegos: **Kilo, Mega, Giga, Tera, Peta, Exa, Zetta, Yotta**.

Tradicionalmente (desde los años 60) la escala de multiplicación ha sido siempre expresada en potencia de 2 elevado a 10 (1024), aunque han surgido intentos de normalizar el uso de la escala de 1000 por parte del IEC desde el año 1998, **introduciendo la nomenclatura Kibibyte, Mebibyte, etc.** para potencias de 2. Su uso es debatido aún hoy en día (año 2019) por la comunidad científica, de forma que unos adoptan el estándar en base 10 (Ubuntu, por ejemplo) y otros en base 2 (Microsoft sin ir más lejos).

3.3.2. Unidades de medida para la transmisión de información.

Tradicionalmente la envío de información en medios analógicos se efectuaba en **baudios**, que mide el número de cambios en la señal por segundo, sin embargo, actualmente se utiliza el **bit por segundo (bps o bit/s)** que es una medida mucho más precisa.

Para nombrar a los múltiplos del bit/s se suele debería la notación Kilobits/s, Megabits/s etc., sin embargo, es frecuente encontrarse estas medidas expresadas en **bytes por segundo** y también en **Megatransfers**.

Un **Megatransfer** (Millones de transferencias por segundo) es equivalente al Megabit por segundo excepto por el hecho de que mide todos los datos que se transfieren incluyendo información de sincronización y control de errores, lo cual permite utilizar unidades redondas a la hora de dar información sobre la tasa de transferencia de un bus de comunicaciones.

3.3.3. Unidades de medida para imagen

Las unidades básicas para medir una imagen es el **pixel** o punto de luz, cuyo tamaño, profundidad de color y distancia a otros puntos va a depender totalmente de la tecnología empleada y la precisión que se le quiera dar. Para sus múltiplos se utilizan los mismos prefijos que para el bit.

3.4. Redundancia en las codificaciones binarias. Bits de control.

Paridad simple

Para mejorar la fiabilidad de las comunicaciones, se puede añadir un bit de paridad para conseguir que el número de 1 de una cadena sea par o impar, de tal forma que el receptor considerará que se ha producido un error si la paridad no es la esperada.

Sin embargo, este método no es capaz de detectar cambios que se hayan producido en un número par de bits.

Paridad de Bloque

Dispone la información por filas y bloques y añade un bloque completo con la paridad (vertical). Si se combina con paridad simple (horizontal) también se pueden corregir errores.

Retransmisión

Es una forma muy eficaz si el coste de retransmisión es inferior al coste del envío de información redundante. Es un sistema muy utilizado actualmente, pero es necesario detectar que existe un error.

Códigos Hamming

Es un código detector y corrector de errores que lleva el nombre de su inventor, Richard Hamming. Utiliza las posiciones potencia de dos para transmitir bits de paridad redundantes, y un bit adicional en el código Hamming extendido que es capaz de detectar fallos en dos bits y corregir uno.

Redundancia cíclica

Desarrollados por *W. Wesley Peterson en 1961*, los CRC (Códigos de Redundancia Cíclica), también llamados códigos polinómicos, constituyen el método de detección de errores más empleado en comunicaciones. Su uso está muy extendido porque pueden implementarse en hardware con mucha facilidad.

Estos códigos se basan en el uso de un **polinomio generador** $G(X)$ de grado r , y en el principio de que n bits de datos binarios se pueden considerar como los coeficientes de un polinomio de orden $n-1$. Por ejemplo, los datos 1 0 1 1 1 pueden tratarse como el polinomio $x^4 + x^2 + x^1 + x^0$

A estos bits de datos se añaden r bits de redundancia de forma que el polinomio resultante sea divisible por el polinomio generador, **sin generar resto**. El receptor verificará si el polinomio recibido es divisible por $G(X)$. Si no lo es, habrá un error en la transmisión.

3.5. Compresión de los datos

Consiste en la reducción del número de bits a transmitir y existen distintas técnicas dependiendo del tipo de datos a transmitir. A modo de ejemplo podemos citar:

Códigos Huffman

Desarrollado por David A. Huffman, este es principalmente un algoritmo para la compresión de datos símbolo a símbolo según una distribución de probabilidad, que da lugar a un código prefijo (es decir, la cadena de bits que representa a un símbolo en particular nunca es prefijo de la cadena de bits de un símbolo distinto) y representa los caracteres más comunes usando las cadenas de bits más cortas. Es un sistema sin pérdida de información.

Codificación JPG

Es una compresión con pérdida para imágenes y está basado en las propiedades del ojo humano que es capaz de detectar mucho mejor los cambios de la luminiscencia que los cambios cromáticos, pudiendo de esta forma homogeneizar zonas cromáticas y ahorrar pequeños cambios de información en la variación cromática de los pixels contiguos.

Codificación MPEG

Codificación de video y audio con pérdida basado en el uso de marcos (zonas de pixels que no cambian entre fotogramas) y reducción del muestreo de audio.

Codificación MPEG Audio Layer III (MP3)

Sistema con pérdida para el sonido que reduce el número de bits por segundo para muestreo en su digitalización, así como el recorte de la onda a las frecuencias audibles por un oído humano medio.

3.6. Criptografía. Confidencialidad en las comunicaciones.

Para asegurar la confidencialidad en las comunicaciones, se utilizan sistemas criptográficos, con los que se obtiene un texto inteligible a priori a partir de un texto en claro.

Los algoritmos de cifrado se clasifican en:

Algoritmos de clave simétrica: Utilizan la misma clave para el cifrado y descifrado. Por ejemplo, AES.

Algoritmos de clave asimétrica: Existe una clave pública y privada, las cuales sirven para descifrar la información cifrada por la otra clave. Por ejemplo, RSA.

4. Conclusión.

La representación de la información es un pilar fundamental de nuestra sociedad actual. Es por esto que la representación utilizando métodos estandarizados y eficaces, así como garantizar la

seguridad y efectividad de su transmisión sean una prioridad en el desarrollo e implementación de redes informáticas.

Actualmente con el uso de internet, han cobrado especialmente importancia la transmisión de esos datos, haciendo necesaria un sistema de **representación internacional**, estandarizado y que se adapte fácilmente a las características regionales de representación alfanumérica.

Por otro lado, la confidencialidad de la información y la garantía de que la transmisión de los datos codificados se produzca sin error, puede llevar a que se añada cierto contenido de información **redundante**, y utilizar técnicas de **encriptación y compresión** de los datos.

4.1. Sistema educativo. Relación con el currículo.

Este tema es aplicado en el aula en los módulos profesionales siguientes, con las atribuciones docentes indicadas (PES/SAI):

Formación profesional básica

- Montaje y Mantenimiento de sistemas y componentes informáticos (TPB en Informática de Oficina/ TPB en informática y Comunicaciones) (PES/SAI)
- Instalación y mantenimiento de redes para transmisión de datos (TPB en Informática de Oficina/ TPB en informática y Comunicaciones) (PES/SAI)

Grado Medio

- Montaje y Mantenimiento de Equipos (1º GM de SMR) (PES/SAI)
- Redes locales (1º GM SMR) (PES)

Grado Superior

- Sistemas informáticos 1º GS de DAM (PES/SAI)
- Sistemas informáticos 1º GS de DAW (PES/SAI)
- Fundamentos de hardware (1º GS ASIR) (PES/SAI)
- Planificación y Administración de Redes. (1º de ASIR) (PES)

Bachillerato:

- Tecnologías de la Información y la Comunicación I: 1º de bachillerato (PES)
- Programación y computación: 2º de bachillerato (libre configuración autonómica) (PES)

5. Bibliografía

- De Anasagasti, Miguel. "Fundamentos de la Computadora" 9ªed 2004 Edt. Paraninfo
- Patterson D.A. y Hennessy JL. "Estructura y diseño de computadoras: la interfaz hardware/software" 4ª Ed. (2005) Edt McGraw-Hill
- Prieto A, Lloris A, Torres JC. "Introducción a la Informática" 4ªed. (2006) Edt. McGraw-Hill

- Stallings W. "Organización y Arquitectura de Computadoras" (2006) 5^a Ed. Edt. Prentice-Hall
- Ramos A, Ramos MJ y Viñas S "Montaje y Mantenimiento de Equipos" (2012). Edt McGraw-Hill
- Jiménez Cembreras, Isabel M^a "Sistemas Informáticos" 2^aEd (2018) Edt. Garceta