AUTORES

MARIBEL CAMPOS MONGE

Ingeniera Técnica en Informática de Gestión por la Universidad Politécnica de Valencia (UPV). Profesora de Secundaria de Informática desde 2004. Especialista en Redes Informáticas y Servicios en Red. Coordinadora e Instructora de la academia CISCO.

Experiencia como miembro del Tribunal de Oposiciones de Secundaria de Informática de Valencia. Preparadora de Oposiciones de Secundaria de Informática.

JORGE LÓPEZ QUEROL

Ingeniero Técnico en Informática de Gestión por la Universidad Politécnica de Valencia (UPV). Maestro de primaria y profesor de secundaria de Informática desde 2008. Especialista en Redes Informáticas y programación. Coordinador e instructor de academia CISCO. Preparador de Oposiciones de Secundaria de Informática.

EVA MARÍA CAMPOS MONGE

Ingeniera Superior en Informática por la Universidad Politécnica de Valencia (UPV) superando la intensificación en Gestión. Diploma de Estudios Avanzados y Suficiencia Investigadora en Programación Declarativa e Ingeniería de la Programación por la Universidad Politécnica de Valencia (UPV).

Experiencia como investigadora en el área de Ingeniería del Software en el Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia (UPV).

Profesora de Secundaria de Informática desde 2002 especializada en Bases de Datos, Redes Informáticas y Servicios en Red. Miembro del Equipo Directivo. Coordinadora e instructora de Academia CISCO. Preparadora de Oposiciones de Secundaria de Informática.

INTRODUCCIÓN BLOQUE I

Este es el primero de un total de cuatro libros que forman el temario actualizado de Oposiciones de Secundaria de Informática. Su objetivo fundamental es el de facilitar al opositor la preparación de la prueba escrita.

El contenido de este libro está desarrollado basándose en la legislación actual que regula el contenido de estas pruebas.

Este volumen contiene los 20 primeros temas de los 74 que componen el temario de Informática de Secundaria. En estos temas se desarrollan los bloques de Representación de la información, Hardware y Sistemas Operativos ofreciendo un contenido totalmente actualizado recogiendo las últimas novedades en las disciplinas que se presentan.

Cada uno de los temas consta de un índice que presenta el esquema general del tema, la introducción, el desarrollo del tema en cuestión, una conclusión y bibliografía/webgrafía.

En la prueba escrita es recomendable que se introduzca el punto de bibliografía/webgrafía al final del tema. En este libro se presenta la bibliografía agrupada por bloques con el fin de facilitar al opositor la tarea de recordarla.

Los temas se presentan de forma acotada para que el opositor sea capaz de desarrollarlo en el tiempo estipulado, asegurando que se tratan todos los puntos de interés con la profundidad adecuada.

Los temas pertenecientes al mismo bloque tienen contenidos en común, lo que permitirá al opositor rentabilizar tiempo de estudio y poder amortizar píldoras de conocimiento aplicables a distintos temas.

Además, este volumen viene acompañado de material adicional en el que el lector puede encontrar trucos sobre cómo afrontar el examen, ejemplos para añadir a los temas, contextualización en los ciclos formativos y otros recursos de interés.

Tema 1

REPRESENTACIÓN Y COMUNICACIÓN DE LA INFORMACIÓN

1.1 INTRODUCCIÓN

Hasta hace unas décadas, los computadores eran los únicos dispositivos que almacenaban información de forma digital. Con la digitalización que está viviendo la sociedad en los últimos tiempos, ahora mismo se puede encontrar casi cualquier cosa conectado a Internet y almacenando información a mayor o menor escala. Con el IoT se pueden encontrar interconectados relojes, electrodomésticos, teléfonos móviles, casas domóticas, etc.

Si algo tienen en común todos estos dispositivos es que almacenan la información de forma digital. La unidad mínima de información es un bit, que solo puede tener dos valores: 0 y 1. Por tanto, independientemente del tipo de información con el que se trabaje (número, texto, una imagen, un vídeo, etc.) todo debe traducirse finalmente a una secuencia de ceros y unos.

Los tipos de datos simples como los numéricos o el texto se representarán directamente en binario. Sin embargo, los datos más complejos como una imagen pueden contener, además de la propia representación de la imagen, metadatos que ofrezca información extra sobre cómo debe mostrarse esa imagen. De esta manera se conseguirá que la representación interna de la imagen genere una representación externa (como la verá el usuario final) adecuada a su tipo de datos y de la mayor calidad posible.

En la sociedad digital actual, los datos no se quedan estáticos almacenados en un dispositivo, sino que viajan de un dispositivo a otro continuamente. Por tanto, es importante estudiar también cómo se realiza la comunicación de esta información.

1.2 REPRESENTACIÓN DE DATOS NUMÉRICOS

Un sistema de numeración es un convenio sobre cómo agrupar y manipular unos determinados signos mediante un conjunto de reglas de manera que permitan representar cantidades. El sistema de numeración recogerá también las operaciones permitidas.

El sistema de numeración más utilizado por las personas es el sistema decimal que tiene diez dígitos del 0 al 9, pero también se utilizan de forma cotidiana otros sistemas de numeración como el sexagesimal que es el encargado de medir el tiempo.

1.2.1 Sistemas de numeración posicionales

Los sistemas de numeración posicionales son los que más se utilizan. La base de cada sistema es el número de dígitos que contiene. El hecho de ser posicionales indica que un dígito tendrá un valor diferente en función de la posición que ocupe:

	Decimal	**Binario**	**Octal**	**Hexadecimal**
Base	10	2	8	16
Dígitos	Del 0 al 9	0 y 1	Del 0 al 7	Del 0 al 9 y de la A a la F

1.2.2 Operaciones básicas

Los números binarios permiten operaciones aritméticas y operaciones lógicas. Las siguientes tablas presentan el resultado de las operaciones básicas en notación binaria.

Suma			
a	b	a+b	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Resta			
a	b	a-b	Acarreo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

AND		
a	b	a AND b
0	0	0
0	1	0
1	0	0

OR		
a	b	a AND b
0	0	0
0	1	1
1	0	1
1	1	0

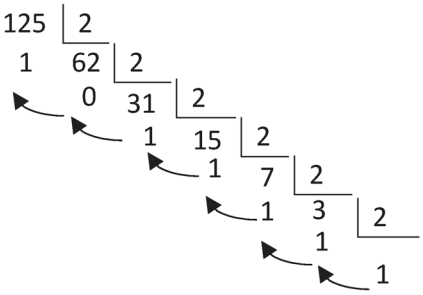
NOT	
a	NOT a
0	1
1	0

Estas operaciones pueden combinarse entre sí para realizar operaciones más complejas.

1.2.3 Cambio de base

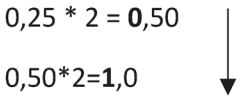
- **Para convertir un número de decimal a binario** (Este método puede utilizarse para convertir un número decimal a cualquier base sustituyendo el 2 por la base a la que se quiere convertir)

- Parte entera:
  - se divide la cantidad entre la base destino (en binario 2). Si el cociente es mayor que la base (2) se vuelve a dividir.
  - Se obtendrá el número en binario concatenando el cociente de la última operación y el resto desde la última división hasta la primera.
    Ejemplo:



Por tanto $125_{10}=1111101_2$

- Para la parte fraccionaria:
  - se multiplica la parte fraccionaria de la cifra decimal por la base destino (en binario 2). La parte entera serán los dígitos de la parte fraccionara del número resultante. Se continuará hasta que la parte fraccionaria sea 0.
    Ejemplo:

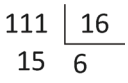


105,0510= 1111101,012

Para convertir un número de decimal a hexadecimal se repetirá el mismo proceso, pero teniendo en cuenta que la base destino es 16.

Ejemplo:

$111_{10}= 6F_{16}$



Decimal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- **Para convertir un número de binario a decimal**

  Se hace utilizando potencias de 2. Por ejemplo, el número 11001 en binario es equivalente al número 25 en decimal.

  $11001_2 = 1*2^4+1*2^3+1*2^0=16+8+1=25_{10}$

1.2.4 Enteros

Los números enteros son aquellos que contienen además de su cifra, el signo positivo o negativo. El signo puede representarse utilizando varias técnicas.

- **Signo/magnitud:** se añade un nuevo bit que indica el signo. Normalmente es el bit de mayor peso (el de la izquierda). Si el número es positivo el bit de signo será 0 y si es negativo será 1.

  Ejemplo:

  $$7_{10}=0111_2 \qquad -7_{10}=1111_2$$

  Hay que señalar dos aspectos de esta técnica de representación que la hacen poco interesante:

  - Existen dos representaciones para el valor cero.
  - En las operaciones de suma y resta; para obtener el resultado de A – B, primero hay que mirar el signo, si los dos operandos tienen el mismo signo, se restan, en caso contrario se suman, lo mismo ocurre con A + B. Un procedimiento engorroso y complejo que genera circuitería adicional.

- **Complemento a 1:** los números positivos se representan con la notación binaria natural y los negativos se representan en complemento a 1 (Ca1). Simplemente invertirá el valor de los bits. En este método el 0 también tiene doble representación.

  Ejemplo:

  $$7_{10}=0111_2 \qquad -7_{10}=1000_2$$

- **Complemento a 2:** los números positivos se representan con la notación binaria natural y los negativos se representan en complemento a 2 (Ca2). En este caso se invertirá el valor de los bits y al resultado se le sumará 1.

  Este método es un poco más eficiente, el 0 solo tiene una representación y además tiene un rango de presentación un poco mayor que el Ca1.

  Ejemplo:

  $$7_{10}=0111_2 \qquad -7_{10}=1001_2$$

- **Exceso a Z:** esta codificación se basa en la idea de elegir un valor binario que represente el valor 0 (será Z). Todos los valores mayores que Z serán números positivos y todos los menores serán números negativos.

  Tiene la ventaja de que se pueden comparar números entre sí directamente, cosa que no era posible en las codificaciones Ca1 y Ca2. Cada número x se representará con el número binario que corresponda a x+Z.

1.2.5 Reales

Un número real es aquel que consta de una parte entera y una parte fraccionaria separadas por la coma decimal.

Existen dos métodos para la representación de los números reales: coma fija y coma flotante.

Coma fija

En los métodos de representación de coma fija se establece un número de bits para la parte entera y número de bits para la parte fraccionaria. Esto limita mucho el rango de números que se pueden representar.

Por ejemplo, se puede decidir que se dediquen 4 bits a la parte entera y 4 para la parte flotante. En este caso no podría representarse el número 10011 ni tampoco el 0,10011.

Coma flotante

Los métodos de coma flotante permiten mover la posición de la coma dentro del número total de bits que se tiene para representar el número. Esta flexibilidad favorece que el rango de números que se pueden representar sea mucho mayor.

Tienen tres componentes:

- Signo: es un bit que indica si el número es positivo o negativo.
- Mantisa: que tiene el número completo sin la coma.
- Exponente: número entero con signo

El número F se calculará siguiendo la fórmula: $F=(\text{signo})\ \text{Mantisa}*2^{\text{exponente}}$

El estándar de representación de coma flotante es el IEEE 754. Este estándar cuenta con la representación en precisión simple y doble precisión.

Precisión simple (32 bits):

Signo	Exponente	Mantisa
1 bit	8 bits	23 bits

Precisión doble:

Signo	Exponente	Mantisa
1 bit	11 bits	52 bits

1.3 REPRESENTACIÓN DE DATOS ALFANUMÉRICOS

1.3.1 ASCII

El código ASCII se caracteriza por asociar cada uno de sus caracteres a un byte (8 bits). De ese byte se reserva el último bit como carácter de control indicando la paridad, por lo que el número total de valores que pueden representarse es de 2^7 que hace un total de 128 caracteres diferentes.

Estos 128 valores son suficientes para representar todas las letras del alfabeto inglés en mayúsculas y minúsculas, los números del 0 al 9 y algunos caracteres especiales. Sin embargo, no tienen cabida las vocales acentuadas, o algunos caracteres como "¿" muchos símbolos matemáticos imprescindibles en algunos contextos.

Para eliminar esta carencia se incorporó el octavo bit del byte que se utilizaba para paridad para duplicar el número de caracteres que pueden representarse. De esta manera se tiene desde el código 0 al 255.

ASCII extendido: Cuando se utilizan los 8 bits para la representación de los caracteres y símbolos y se prescinde del bit de control.

El problema del ASCII extendido es que no está estandarizado y cada fabricante lo codifica de una manera.

1.3.2 EBCDIC

EBCDIC se trata de una codificación basada en 8 bits (un byte) desarrollada por IBM. No llegó a estandarizarse y solo se utiliza en dispositivos de la marca.

1.3.3 Unicode

Unicode es un están dar que permite la codificación de cualquier carácter en cualquier idioma. Utiliza 32 bits (4 bytes) para la codificación de un carácter. Es la codificación ideal para intercambios de texto de manera internacional. Está regulado por el estándar ISO/ IEC 10646. Realmente el estándar ISO/IEC 10646 recoge la codificación de Unicode y alguna otra extensión. Unicode cuenta con tres formatos de codificación y el estándar ISO con cuatro.

Dentro de la codificación Unicode hay varios formatos de codificación de caracteres. El más utilizado actualmente es el UTF8.

- **UTF8:** formato de codificación de caracteres de Unicode que utiliza símbolos de extensión variable, es decir, la longitud de la codificación puede ir de 1 a 4 bytes, siendo los contenidos en ASCII los que se codifican con un byte y siendo los símbolos menos utilizados los que se codifican con cuatro. Tiene correspondencia con ASCII.
- **UTF16:** la mayoría de los caracteres se codifican con dos bytes. No existe correspondencia con ASCII. Es recomendable para el texto asiático..
- **UTF32:** todos los caracteres tienen 4 bytes. Ocupa mucho, pero es muy rápido. No suele utilizarse.

1.4 REPRESENTACIÓN DE INFORMACIÓN MULTIMEDIA

1.4.1 Imágenes

El almacenamiento de datos de las imágenes dependerá del tipo de archivo en el que se encuentre la imagen, así como de si se encuentra en blanco y negro o en color. El tipo de archivo determinará si existe compresión, en caso de haberla si es con o sin pérdida de información.

- **Formato TIFF:** es un tipo de imagen sin compresión, por tanto, las imágenes almacenadas en este formato tienen mucho peso.

- **Formato GIF:** es el tipo de imagen que suele utilizarse en la web. Reduce el número de colores a 256, y suele concatenar varias imágenes para hacer una animación. Sobre la imagen a la que se le ha reducido el número de colores a 256 se aplica una compresión sin pérdida. Por tanto, si la imagen inicial tenía como máximo 256 colores no se perderá calidad, sin embargo, si se almacena en formato GIF una imagen con 16 millones de colores se perderá más del 99% de la información de la imagen inicial.

- **Formato PNG**: Apareció como una mejora del formato GIF porque permite hasta 16 millones de colores. Utiliza una compresión sin pérdida y no permite realizar animaciones.

- **Formato JPG**: Este formato utiliza un método de compresión llamado también JPG. Es el formato idóneo para fotografías o imágenes que tienen tonos continuos similares. A pesar de que su método de compresión tiene pérdida se consigue un gran ahorro de almacenamiento con una muy buena calidad. Este formato da la posibilidad de elegir el grado de compresión que se desea en la imagen (con herramientas tipo Photoshop puede modificarse este grado de compresión).

  La compresión JPG sigue los siguientes pasos:

  - La información RGB de cada píxel se traduce a una información en la que se almacena luminosidad y color. Para el ojo humano la luminosidad es muy importante.
  - Se divide la imagen en bloques de 8x8 y se aplica un algoritmo llamado Transformada Directa del Coseno.
  - Se realiza la cuantificación (compresión con pérdida).
  - Se codifica esta información utilizando un algoritmo que no tiene pérdida de resolución.

1.4.2 Sonido

El sonido se representa fielmente con una señal analógica igual a la señal sonora. Sin embargo, el almacenamiento en un equipo informático de un archivo de sonido debe realizarse en digital. Para ello debe realizarse la transformación de analógico a digital.

Esta transformación se realiza en tres fases:

- **Muestreo**: se toman un número determinado de muestras en un pequeño intervalo de tiempo. Cuanto menor sea el intervalo mejor calidad tendrá el resultado final.
- **Cuantificación**: La muestra se representará con un valor numérico, y se redondeará al valor más próximo dentro del conjunto de valores posibles (este mecanismo provocará cierta pérdida)
- **Codificación**: Esa cuantificación se codifica al número de bits determinado.

Los tipos más utilizados para almacenar ficheros de audio son:

- **MP3:** su algoritmo de compresión de datos tiene pérdida. Es el más conocido porque genera ficheros muy reducidos para la calidad que tiene.
- **WAV:** se trata de ficheros que contienen audio sin comprimir. Su calidad es excelente pero sus archivos tienen gran peso.

1.4.3 Vídeo

Un fichero de vídeo deberá almacenar una serie de imágenes junto con el sonido asociado. Además, puede almacenar cierta información complementaria como subtítulos o metadatos.

Es importante conocer algunos aspectos del almacenamiento de un fichero de vídeo:

- Tamaño de la imagen
- Relación de aspecto (4:3, 16:9)
- Número de imágenes por segundo
- Frecuencia de muestreo de audio
- Bitrate: la cantidad de datos total que se va a reproducir por segundo.

La codificación de vídeo aúna la codificación de imágenes y de sonido.

Los formatos más utilizados son:

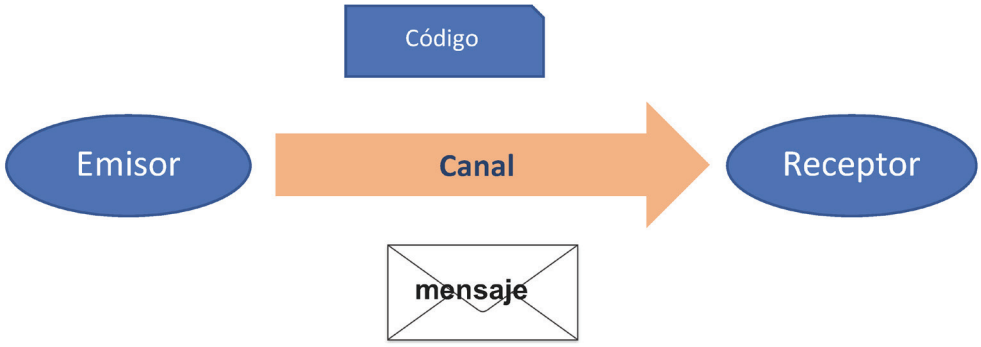
- **MP4**: es un formato que suele utilizarse en redes sociales como YouTube, Facebook o Instagram. Tiene bastante buena definición con un almacenamiento no muy grande.

- **AVI**: tiene un mecanismo de compresión no muy elevada, incluso puede almacenar vídeos sin ningún tipo de compresión. Suele generar archivos bastante pesados. Ideal para grabaciones caseras.
- **MOV**: comprime mucho mejor que el AVI. A pesar de esto suelen ser vídeos de alta calidad por lo que los ficheros resultantes suelen ser pesados. Desarrollado por APPLE.
- **MKW**: permite alta definición con unos mecanismos de compresión bastante eficientes. Ideal para reproducción de películas en alta definición.
- **WMW**: creado por Microsoft. Comprime mejor que MP4.

1.5 COMUNICACIÓN DE LA INFORMACIÓN

Un sistema de comunicación cuenta con una serie de elementos básicos:

1. **Emisor**: es el agente que inicia la comunicación.
2. **Receptor**: es el agente destino de la comunicación.
3. **Mensaje**: información que el emisor quiere transmitir al receptor.
4. **Canal**: medio por el que viaja el mensaje.
5. **Código**: es el lenguaje que se utiliza para la comunicación.



Cuando el mensaje se lanza al canal para su envío, el canal suele alterar de alguna forma el contenido. En función del tipo de canal habrá más variaciones o menos.

Por ejemplo, si el canal es el aire el mensaje puede sufrir más alteraciones que si se trata de un envío digital que utiliza fibra óptica donde la información va más protegida.

En cualquier caso, deben minimizarse estas alteraciones ya que el objetivo es que el receptor obtenga el mensaje original, y para esto se utilizan algunas técnicas:

- **Cifrado**: si el canal puede ser interferido por algún otro agente que no es el receptor y puede obtener el contenido del mensaje o incluso modificarlo, se puede utilizar el cifrado de datos. Esto hace que el mensaje no viaje en el lenguaje natural que utilizan emisor y receptor, si no que se aplica un algoritmo de cifrado que hace que, aunque el mensaje sea interceptado, no pueda conocerse su contenido.

- **Compresión**: en ocasiones la información que debe comunicarse entre emisor y receptor tiene un tamaño bastante grande y eso hace que la comunicación tarde más tiempo del esperado. Si se aplica un algoritmo de compresión al mensaje se puede conseguir reducir el tamaño del mensaje. Hay que tener en cuenta que algunos algoritmos de compresión (normalmente los que más reducen el tamaño del mensaje original) aportan pérdida de la calidad.

- **Redundancia**: para minimizar la pérdida de información ya sea por deficiencias del canal o porque se ha aplicado compresión a los datos, se pueden utilizar técnicas de redundancia que ayuden al receptor a detectar posibles fallos e incluso, en algunas ocasiones, será capaz de corregirlos.

Los códigos que se utilizan en la transmisión suelen tener el mismo tamaño, estos códigos estarán formados por los bits del mensaje original más otros que se añadirán para codificar la fuente y llevar un control del posible error. Se puede utilizar, por ejemplo, el código Hamming para añadir bits de paridad a los bits de datos.

Otra opción es utilizar códigos cíclicos (CRC)

Los códigos cíclicos incorporan como bits de control el resultado de un cálculo basado en la operación módulo o resto de la división entre dos números. El algoritmo usado es el siguiente:

- Se considera que los datos que se van a codificar son coeficientes de un polinomio (ejemplo 101101 → $1x^5 +0x^4 +1x^3 +1x^2 +0x^1 +1x^0$ → $x^5 +x^3 +x^2 +1$). A este polinomio se le denomina D(x) y tendrá d bits (tamaño del paquete de datos) luego su grado será d-1.

- Emisor y receptor se deben ponerse de acuerdo en el uso de un polinomio generador G(x) de grado r.
- Se añaden r bits puestos a cero al extremo inferior del paquete de datos, de forma que ahora el tamaño de éste será d + r. El hecho de añadir los r bits por la derecha equivale a realizar el producto $P(x)= x^r * D(x)$.
- Se divide la serie de bits correspondientes a P(x) por la serie G(x) empleando la técnica de división módulo 2 (división binaria sin considerar los préstamos que se realizan en las restas intermedias y considerando que el dividendo "cabe" en el divisor cuando al menos tenga tantos bits significativos como este.).
- El resto R(x) se obtiene empleando la sustracción en módulo 2 (equivale a XOR o suma lógica) de P(x) con lo que el resultado se hace divisible por G(x). Este resultado es la palabra de código C(x).
- El receptor intenta dividir C(x) por G(x), la condición de error es que ese cociente produzca un resto distinto de cero, si esto es así se solicita la retransmisión.

La eficiencia del sistema depende del polinomio generador elegido. Por ejemplo, los códigos de 16 bits capturan todos los errores simples y dobles, todos los errores en que el número de bits afectados es impar, todos los errores en ráfaga con tamaño de ráfaga menor o igual a 16, y el 99,997% de los errores de ráfaga de 17 bits.

1.6 CONCLUSIÓN

Como se ha visto en este tema, es indispensable conocer cómo se representa la información de manera digital. Lo que en un principio puede resultar trivial considerando que únicamente es necesario realizar un cambio de base para que cualquier dato se represente con solo dos valores 0 y 1, se complica en cuanto se añaden distintos tipos de datos.

Desde el inicio de la informática se ha trabajado para conseguir la representación óptima para cada tipo de datos, y aún en nuestros días se sigue trabajando en nuevos formatos que permitan representar en el menor espacio posible, la mayor cantidad de información.

Con respecto a la comunicación de los datos, la globalización actual y la interconexión mundial que existe hacen que conseguir una buena transmisión de datos en el menor tiempo posible y manteniendo su integridad y seguridad sea otro de los principales campos de estudio en el campo de la informática en nuestros días.