

Tema 10

Representación interna de los datos

 Cómo se codifican y almacenan los distintos tipos de datos en un sistema digital.

◆ 1. Introducción

Todo dato digital se representa internamente en **binario** (base 2).

📌 A partir de esta base se definen:

- Estructuras
- Codificaciones
- Estándares

➡ Que permiten representar texto, números, estructuras complejas y multimedia.

2. Representación de caracteres

ASCII

- Estándar de 7 u 8 bits
- Capacidad: 128/256 símbolos
- Limitado a inglés y caracteres básicos

Unicode

- Codificaciones: UTF-8, UTF-16
- Soporta más de 140.000 símbolos
- Universal: alfabetos, emojis, símbolos técnicos

3. Representación de booleanos

- Codificación mínima: 1 bit
 - 0 → falso
 - 1 → verdadero

Usos:

- Condiciones (if , while)
- Lógica digital
- Bit de control en hardware

12 4. Representación de números enteros

Método	Descripción
Signo y magnitud	Bit de signo + valor absoluto
CA1 (Comp. a 1)	Bit de signo, invierte bits negativos
CA2 (Comp. a 2)	Estándar moderno: aritmética sencilla
Exceso-Z	Desplaza el cero, útil para exponentes

5. Representación de números reales

Coma fija

- Precisión limitada
- Rango muy acotado

Coma flotante (IEEE 754)

Precisión	Bits totales	Exponente	Mantisa
Simple	32	8	23
Doble	64	11	52
Cuádruple	128	15	113

Problemas comunes:

- Redondeo

6. Números complejos

 Se representan como **dos números reales**:

- Parte real
- Parte imaginaria

Aplicaciones:

- Procesamiento de señales (audio, radio)
- Simulación física
- Computación cuántica

7. Representación de estructuras

◆ 7.1 Estructuras lineales

Arrays

- Memoria contigua
- Acceso en tiempo constante $O(1)$
- Usos: vectores, matrices

Listas enlazadas

- Nodos con punteros
- Tipos: simple, doble, circular

Pilas (LIFO)

- Inserción/extracción solo en el tope
- Implementación con array o lista enlazada
- Usos: llamadas, deshacer, parsers

Colas (FIFO)

- Inserción por el final, extracción por el inicio
- Variantes: circular, con prioridad
- Usos: buffers, planificación

7.2 Jerárquicas y grafos

Árboles

- Nodos con punteros a hijos
- Tipos:
 - **BST**: ordenados
 - **AVL, R-B**: balanceados
 - **B+**: optimizados para disco/BBDD

Grafos

- Representación:
 - Matriz de adyacencia (denso)
 - Lista de adyacencia (disperso)
- Usos: redes, rutas, algoritmos IA

7.3 Tablas hash

- Clave → índice por función hash
- Acceso promedio: $O(1)$

Colisiones:

- Encadenamiento (listas enlazadas)
- Abierto (sondeo)

Usos:

- Diccionarios
- Cachés
- Compiladores

8. Multimedia y datos complejos

8.1 Imagen

- **Raster:** JPEG, PNG → píxeles
- **Vectorial:** SVG → fórmulas geométricas

 Raster = resolución fija. Vectorial = escalable sin pérdida.

8.2 Sonido y vídeo

- **Muestreo:** frecuencia en Hz (ej. 44.1 kHz)
- **Resolución:** bits por muestra

Formatos:

- Audio: MP3, FLAC
- Vídeo: MP4, H.265

Compresión:

- **Intraframe:** por fotograma
- **Interframe:** entre fotogramas

 Códecs: H.264, AV1, AAC...

8.3 Datos 3D

- Estructura: vértices + normales + mallas
- Formatos: OBJ, GLTF

Usos:

- Videojuegos
- Realidad aumentada / virtual
- Simulación y animación

9. Seguridad y compresión

9.1 Cifrado

Tipo	Ejemplos	Clave
Simétrico	AES	Misma clave
Asimétrico	RSA, ECC	Par público/privado

 Usos: HTTPS, VPN, correos seguros, discos cifrados

9.2 Compresión

- Sin pérdida: ZIP, PNG



Reversible. No pierde información.

- Con pérdida: JPEG, MP3



Reduce calidad. Tamaño mucho menor.



Mejora almacenamiento y velocidad de transmisión.

9.3 Hash


- Algoritmos: SHA, MD5
- Salida: huella digital del contenido

Usos:

- Verificación de integridad
- Autenticación
- Índices de búsqueda

Conclusión

La representación binaria es esencial para:

- ✓ Procesar información de cualquier tipo
 - ✓ Construir estructuras y formatos eficientes
 - ✓ Proteger y comprimir datos
 - ✓ Adaptarse a una gran variedad de dispositivos y contextos
-  Todo en informática comienza con **unos y ceros**.