

Sistema BCD (Binary-Coded Decimal):

Es un código que se utiliza para representar números decimales en binario. Cada número del 0 al 9 es representado por su equivalente en binario en 4 bits.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Ejemplos:

- 25 representado en BCD sería: 00100101
- 99 sería: 10011001
- 44 sería: 01000100

Sistemas de numeración importantes de cara al examen

Decimal Son de base 10 y usan 10 dígitos para expresarse		0 1 2 3 4 5 6 7 8 9
Binario 0 1 Son de base 2 y usan 2 dígitos para expresarse	Hexadecimal 0 1 2 3 4 5 6 7 8 9 Son de base 16 y usan 16 dígitos para expresarse	A B C D E F 10 11 12 13 14 15
		Octal 0 1 2 3 4 5 6 7 Son de base 8 y usan 8 dígitos para expresarse

Pasar de binario, octal y hexadecimal a decimal:

Binario a decimal:

$$\begin{array}{r} \text{Binario: } 1011010101 \\ \text{Potencias: } 2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\ \text{Resultado: } 512 + 0 + 128 + 64 + 0 + 16 + 0 + 4 + 0 + 1 = 725 \\ 1011010101_{(2)} = 725_{(10)} \end{array}$$

Hexadecimal a decimal:

$$\begin{array}{r} \text{Hexadecimal: } F12A4 \\ \text{Equivalente decimal: } 15 \ 1 \ 2 \ 10 \ 4 \\ \text{Potencias: } 16^4 \ 16^3 \ 16^2 \ 16^1 \ 16^0 \\ \text{Resultado: } 983040 + 4096 + 512 + 160 + 4 \\ F12A4_{(16)} = 987812_{(10)} \end{array}$$

Octal a decimal

$$\begin{array}{r} \text{Octal: } 7743 \\ \text{Potencias: } 8^3 \ 8^2 \ 8^1 \ 8^0 \\ \text{Resultado decimal: } 3584 + 448 + 32 + 3 = 4067 \\ 7743_{(8)} = 4067_{(10)} \end{array}$$

ALMACENAMIENTO: HDD, SSD, M.2 Y NVMe

1. HDD (Hard Disk Drive)



- Usa grabación magnética para almacenar y recuperar datos.
- Tiene platos y cabezales mecánicos en movimiento.
- Ofrece más capacidad por euro que un SSD.
- Es más lento, más ruidoso y más sensible a golpes que un SSD.
- Es memoria no volátil.

2. SSD (Solid State Drive)



- Usa memoria flash y no tiene partes mecánicas.
- Acceso y transferencia mucho más rápidos que un HDD.
- Menor latencia, ruido y consumo.
- Mayor resistencia a golpes que un HDD.
- Mayor precio por GB que un HDD.
- Es memoria no volátil.

3. M.2

M.2 es un formato físico, no un protocolo.



- Formato compacto para unidades de almacenamiento.
- Puede emplear interfaz SATA o PCIe.
- No todos los SSD M.2 son NVMe.
- Se instala directamente en la placa base y normalmente no necesita cable de datos.

4. NVMe (Non-Volatile Memory Express)

NVMe es un protocolo para SSD sobre PCIe.



- Ofrece mayor rendimiento y menor latencia que SATA/AHCI.
- Suele encontrarse en SSD M.2, aunque también existe en tarjetas PCIe y otros formatos.
- Puede aprovechar varias generaciones de PCIe.

INTERFACES DE CONEXIÓN HDD/SSD



IDE / PATA
Interfaz paralela antigua.



SATA
Muy común en HDD y SSD de 2,5 pulgadas.



SCSI
Interfaz clásica de entornos profesionales.



SAS
Evolución serie de SCSI, habitual en servidores.



PCIe
Usada por SSD NVMe de alto rendimiento.



SATA y NVMe no son lo mismo: SATA es una interfaz; NVMe es un protocolo optimizado para SSD sobre PCIe.

Sistemas RAID: (Redundant Array of Independent/Inexpensive Disks)

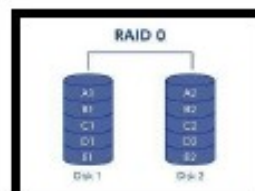
Son un conjunto de unidades físicas trabajando como una unidad lógica.

RAID 0 (volumen dividido/data striping):

No tiene redundancia de datos. Distribuye la información (tiras) entre 2 o más discos.

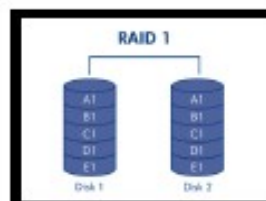
Velocidad de escritura y lectura por encima de todo. / Sin información de paridad.

No es tolerante a fallos (si un disco falla, se pierden todos los datos).



RAID 1 (espejo / disk mirroring):

Desventaja: Requiere el doble de espacio. Mínimo de 2 discos. Si un disco falla podemos recuperar toda la información



RAID 2: División nivel bits / Disco de paridad dedicado / rara vez utilizado

RAID 3: Divide por bytes, no por bloques / Mínimo de 3 discos / rara vez utilizado

RAID 4: (discos de paridad dedicada / IDA) Divide por bloques, no por bytes. Poco usado

Atributos:

Cada tipo de entidad ha de tener un conjunto mínimo de atributos que identifiquen unívocamente cada ocurrencia del tipo de entidad. Este atributo o atributos se denomina identificador principal.

Se pueden definir restricciones sobre los atributos, según las cuales un atributo puede ser:

Univaluado: Solo puede tomar un valor para todas y cada una de las ocurrencias del tipo entidad al que pertenece.

Obligatorio: Atributo que tiene que tomar al menos un valor para todas y cada una de las ocurrencias del tipo entidad al que pertenece.

Relación de exclusividad:

Se dice que una relación es exclusiva cuando la existencia de una relación entre dos tipos de entidades implica la no existencia de las otras relaciones.

En la representación de las relaciones exclusivas se incluye un arco sobre las líneas que conectan el tipo de entidad a los dos o más tipos de relación.

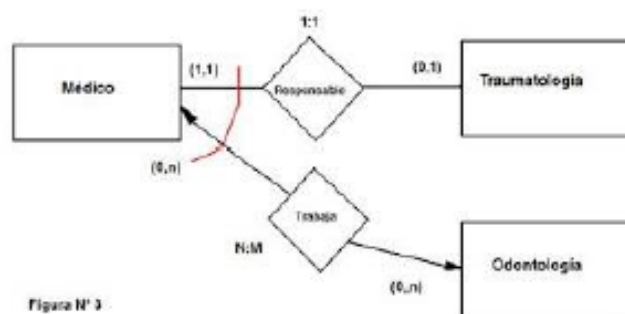


Figura N° 3

Si un médico es responsable de traumatología no puede trabajar en otra odontología (y viceversa).

Jerarquía (tipos):

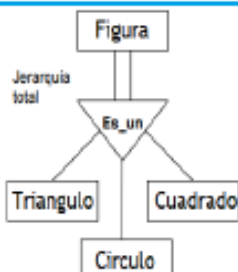
"o": Representa que los subtipos pueden solaparse, lo que significa que una entidad puede pertenecer a más de una de las subentidades en la jerarquía.



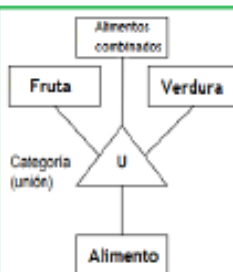
"d": Representa que los subtipos son disjuntos, lo que significa que una entidad solo puede pertenecer a una de las subentidades en la jerarquía.



"Es_un": Representa una jerarquía total, lo que implica que las entidades de la superclase deben pertenecer a al menos una de las subclases en la jerarquía.



"u": Representa uniones por categorías, lo que significa que una entidad puede ser clasificada como una entidad superior y también como una o varias entidades inferiores en la jerarquía.



FHS (Filesystem Hierarchy Standard)

/bin: Directorio estático, almacena los binarios para el funcionamiento del sistema(arranque).
/sbin: Binarios administración de sistema. Programas ejecutables, generalmente de administración
/boot: Configuración de arranque.
/dev: Incluye todos los dispositivos de almacenamiento.
/etc: Ficheros de configuración del sistema host.
/media: Puntos de montaje para dispositivos de medios como unidades de lectores de CD o USB.
/opt: Paquetes de aplicaciones opcionales (de terceros).
/proc: Sistema de ficheros virtual que documenta sucesos y estados de núcleo. Info procesos.
/tmp: Ficheros temporales.
/usr: Aplicaciones y archivos compartidos de solo lectura para múltiples usuarios(bibliotecas, documentación).

Linux maneja permisos para mayor seguridad.

Tipos de usuarios:

- **Root (superusuario):** Acceso total al sistema.
- **Usuarios estándar:** Acceso restringido, sin permisos administrativos.

Permisos en archivos y directorios:

Cada archivo y directorio tiene tres tipos de permisos para tres grupos de usuarios:

- **r (read):** Lectura.
- **w (write):** Escritura.
- **x (execute):** Ejecución.

Comando chmod (cambiar permisos de un archivo o directorio):

	OWNER	GROUP	OTHERS	
	U	G	O	
Permisos	r w x	r w x	r w x	
	↓			
b (dispositivo bloque)				Puede representarse como un bit, siendo 1 para activo y 0 para inactivo.
c (dispositivo de caracteres)				
d (directorio)				Lectura activa y escritura y ejecución inactivas: 100 (4 en decimal)
l (enlace)				Lectura y escritura activas, ejecución inactiva: 110 (6 en decimal)
p (archivos de tubería)				Lectura, escritura y ejecución activas: 111 (7 en decimal)
s (socket)				

`chmod 755 file1.txt` #lectura, escritura y ejecución para el propietario; lectura y ejecución para el grupo y otros
`chmod 644 file2.txt` #lectura y escritura para el propietario; sólo lectura para el grupo y otros
`chmod 700 file3.txt` #acceso total para el propietario; ningún permiso para el grupo y otros
`chmod 666 file4.txt` #lectura y escritura para el propietario, grupo y otros; ningún permiso de ejecución

Lectura (r): 4
Escritura (w): 2
Ejecución (x): 1

Comando Umask (establece los permisos predeterminados para nuevos archivos):

Cuando se usa Umask se definen los permisos que NO quieres otorgar por defecto a los archivos y directorios que se crean. Ejem: "Umask 022" esto dirá que los nuevos archivos no deberían tener permisos de escritura para el GROUP y para OTHERS.

Más sobre procesos UNIX:

Proceso: Programa en ejecución con su propio espacio de memoria y recursos.

Identificadores:

- **PID:** ID único del proceso.
- **PPID:** ID del proceso padre.

Usuarios:

- **UID:** ID del usuario propietario del proceso.
- **GID:** ID del grupo propietario del proceso.

Jerarquía de la memoria



Memoria caché

Una memoria caché es una capa de almacenamiento de datos de alta velocidad que almacena un subconjunto de datos, normalmente transitorios, de modo que las solicitudes futuras de dichos datos se atienden con mayor rapidez que si se debe acceder a los datos desde la ubicación de almacenamiento principal. El almacenamiento en caché permite reutilizar de forma eficaz los datos recuperados o procesados anteriormente.

Caché L1, L2 y L3

- **L1:** La más rápida y pequeña, integrada en el núcleo.
- **L2:** Más grande y algo más lenta.
- **L3:** Compartida entre núcleos, mayor tamaño y menor velocidad.

Ley de Moore

El cofundador de Intel, **Gordon Moore**, predijo que el número de transistores en un chip se duplicaría aproximadamente cada 2 años, con un aumento mínimo en el costo.

Taxonomía de Flynn (1996)

Clasifica arquitecturas de computación según el flujo de instrucciones y datos:

SISD (Single Instruction, Single Data)

- Computadoras tradicionales (PC, CPU).
- Un solo procesador ejecuta una instrucción sobre un solo dato.

SIMD (Single Instruction, Multiple Data)

- Procesamiento en paralelo (GPU, IA, gráficos).
- Una instrucción se aplica simultáneamente a múltiples datos.

MISD (Multiple Instruction, Single Data)

- Poco común, usado en sistemas críticos (control de vuelo).
- Varios procesadores ejecutan distintas instrucciones sobre el mismo dato.

MIMD (Multiple Instruction, Multiple Data)

- Supercomputadoras, servidores, IA.
- Varios procesadores autónomos que ejecutan las diferentes instrucciones sobre distintos datos.

	Una instrucción	Múltiples instrucciones
Un dato	SISD	MISD
Múltiples datos	SIMD	MIMD

Más usado hoy en día: MIMD y SIMD por su alto rendimiento en computación moderna.

Arrays en Java:

Los arrays se utilizan como contenedores para almacenar datos relacionados (en vez de declarar variables por separado para cada uno de los elementos del array)

Distintas formas de declararlo:

```
int [] numeros = new int [78];
int [] array1 = {1, 2, 3, 4, 5};
String [] array1 = {"prueba", "hola"};
int [] array2;
array2 = new int[5];
```

La primera posición del array numeros es la posición 0, la última posición sería la 77.
.length para saber la longitud del array.

Para añadir un valor a un array se hace con la siguiente sintaxis:

```
numeros[i] = 3;
```

Donde "i" es la posición donde queremos almacenar el valor.

Diferencia Vector vs Array:

Un vector es una clase en Java, mientras que un array es un objeto de tipo primitivo.

Los vectores son dinámicos, lo que significa que pueden crecer o reducirse en tamaño automáticamente mientras se usan. Los arrays, por otro lado, tienen un tamaño fijo que se establece al momento de su creación y no pueden ser modificados.

Declaración de vector de números enteros:

```
Vector<Integer> numeros = new Vector<Integer>();
```

También se puede especificar el tamaño inicial del vector al momento de crearlo:

```
Vector<Integer> numeros = new Vector<Integer>(5);
```

ArrayList:

- La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.
- Un ArrayList se trata de un arreglo en el cual se en listan referencias a objetos mediante índices.

Matriz en Java:

Estructura de datos que permite almacenar varios valores del mismo tipo, organizados en una tabla con filas y columnas. Es similar a un array, pero tiene una dimensión adicional para almacenar múltiples valores.

Pero en realidad una matriz en Java es un array de arrays. (Array bidimensional)

```
int[][] array = new int[5][5];
```

Operador ternario:

Es una forma compacta de escribir una sentencia if-else.

Sintaxis: `condición ? valorSiVerdadero : valorSiFalso;`

```
int resultado = (x % 2 == 0) ? 0 : 1;
```

En este caso, si x es par, la condición (x % 2 == 0) es verdadera y se asigna el valor 0 a resultado. Si x es impar, la condición es falsa y se asigna el valor 1 a resultado.

OPERADORES LÓGICOS

El operador **AND**: devuelve TRUE las expresiones son ciertas (ambas), y FALSE si no lo son.

Queremos saber que pokemons son de Fuego y voladores:

SELECT nombre FROM POKEMON WHERE (TIPO1 = 'FUEGO') AND (TIPO2 = 'VOLADOR') ;

Nº	Nombre	Tipo1	Tipo2	Salud	Ataque	Defensa	Velocidad	AtEsp	DefEsp	Total
6	Charizard	Fuego	Volador	78	84	78	100	109	85	534
146	Moltres	Fuego	Volador	90	100	90	90	125	85	580

El operador **OR**: si uno de los dos operandos es cierto el resultado es (TRUE) y se muestra.

SELECT * FROM pokemon WHERE tipo1 = 'Agua' OR tipo2 = 'Planta';

Nº	Nombre	Tipo1	Tipo2	Salud	Ataque	Defensa	Velocidad	AtEsp	DefEsp	Total
7	Squirtle	Agua		44	48	65	43	50	64	314
8	Wartortle	Agua		59	63	80	58	65	80	405
9	Blastoise	Agua		79	83	100	78	85	105	530
46	Paras	Bicho	Planta	35	70	55	25	45	55	285
47	Parasect	Bicho	Planta	60	95	80	30	60	80	405
54	Psyduck	Agua		50	52	48	55	65	50	320

El operador **NOT**: el resultado invierte la condición de la expresión.

SELECT * FROM pokemon

WHERE tipo1 = 'Bicho' AND NOT nombre = 'Paras' AND tipo2 = 'Planta';

Nº	Nombre	Tipo1	Tipo2	Salud	Ataque	Defensa	Velocidad	AtEsp	DefEsp	Total
47	Parasect	Bicho	Planta	60	95	80	30	60	80	405

ORDER BY (Ordenar los registros devueltos)

SELECT * FROM pokemons ORDER BY nombre DESC

Nº	Nombre	Tipo1	Tipo2	Salud	Ataque	Defensa	Velocidad	AtEsp	DefEsp	Total
41	Zubat	Veneno	Volador	40	45	35	55	30	40	245
145	Zapdos	Electrico	Volador	90	90	85	100	125	90	580
40	Wigglytuff	Normal	Hada	140	70	45	45	85	50	435

Si no indicamos **ASC** ni **DESC**, el comportamiento por defecto será (**ASC**).

AS (Asignación de un alias a un dato devuelto)

- El operador **AS** es opcional.

SELECT nombre AS N, tipo1 AS T FROM pokemons

N	T
Bulbasaur	Planta
Ivysaur	Planta
Venusaur	Planta

GROUP BY

Se usa para agrupar filas que tienen los mismos valores.

SELECT tipo1 AS tipos FROM pokemon GROUP BY tipo1

(Esta consulta devuelve todos los tipos de pokemon que hay en tipo1, sin el **GROUP BY** saldrían los tipos repetidos)

Hijacking	Proceso por el que se intenta secuestrar un elemento específico del entorno de Internet a través de rutas no autorizadas. Secuestros de urls, de dominios, dns, navegador o sesión...
Clickjacking	Técnica que engaña a los usuarios para que hagan clic en elementos invisibles o disfrazados en una página web, permitiendo realizar acciones no deseadas como permitir accesos o compartir información.
Man in TheMiddle	Ataque en el que el agresor intercepta una comunicación asumiendo el rol de intermediario. El atacante debe ser capaz de observar e interceptar mensajes entre las dos víctimas.
Rogue	Falsos programas de seguridad que no son realmente lo que dicen ser, sino que todo lo contrario. Bajo la promesa de solucionar falsas infecciones, cuando el usuario instala estos programas, su sistema es infectado. (antivirus falsos)
Keylogger	Hardware o software malicioso que, sin tu permiso o conocimiento, registra todas las teclas que pulsas para operar tu computadora o celular. Registra las pulsaciones del teclado.
ZoomBombing	Intrusión indeseada y perturbadora, generalmente por troles y hackers de Internet, en una videoconferencia.
Sunburst	Ataque a la cadena de suministro (ataque de terceros)
Eavesdropping Attack	Intercepción de comunicaciones de red no cifradas para obtener información sensible.
Botnet	Red de dispositivos infectados controlados de forma remota por un atacante para realizar ataques DDoS, envío de spam, minería de criptomonedas, etc
Rootkit	Malware diseñado para obtener acceso privilegiado (root/admin) en un sistema y ocultar su presencia, permitiendo el control total del sistema sin ser detectado.
HONEYPOT	Conjunto de ordenadores cuya intención es atraer a atacantes simulando ser sistemas vulnerables o débiles a los ataques. Distraen a los atacantes de las máquinas más importantes del sistema y recopilan información sobre su identidad y métodos con el fin último de mejorar la seguridad de las redes.
Zero Day	Vulnerabilidad desconocida para el fabricante y sin parche disponible en el momento de su explotación.

CPDS Y SUS TIERS:

Creada por el Uptime Institute en el estándar ANSI/TIA-942.

El concepto de Tier indica el nivel de fiabilidad de un CPD asociado a cuatro niveles de fiabilidad definidos. Cuanto más alto es el número de Tier, mayor disponibilidad y menor indisponibilidad.

Cuenta con 4 tiers: Tier I, Tier II, Tier III, Tier IV.

Tier	% disponibilidad
Tier I	99.671%
Tier II	99.741%
Tier III	99.982%
Tier IV	99.995%

Características de cada Tier (ANSI/TIA-942):

El estándar ANSI/TIA-942, desarrollado por el Uptime Institute, clasifica los Centros de Procesamiento de Datos (CPD) en cuatro niveles o Tiers según su grado de disponibilidad, redundancia y tolerancia a fallos. A mayor Tier, mayor fiabilidad y menor tiempo de inactividad anual.

Tier I (Básico): No dispone de redundancia en sus componentes. Cuenta con una única línea de alimentación y refrigeración. Requiere al menos una fuente de energía. Su disponibilidad aproximada es del 99,671%, con un tiempo de inactividad anual de hasta 28,8 horas. Está orientado a pequeñas empresas con tolerancia a interrupciones.

Arboles:

Un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho.

Nodos: Se le llama Nodo a cada elemento que contiene un Árbol.

Nodo Raíz: Se refiere al primer nodo de un Árbol, Solo un nodo del Árbol puede ser la Raíz.

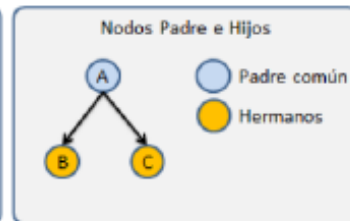
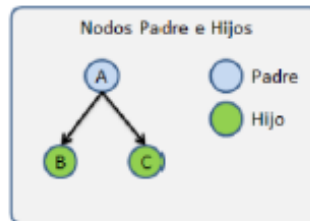
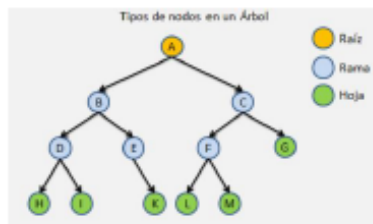
Nodo Padre: Aquellos nodos que tiene al menos un hijo.

Nodo Rama: Nodos que no son la raíz (es decir tienen padre) y además tiene al menos un hijo.

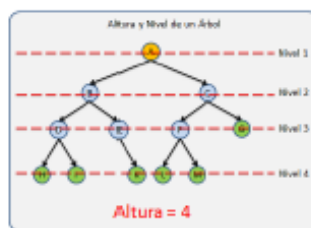
Nodo Hijo: Los hijos son todos aquellos nodos que tiene un padre.

Nodo Hermano: Nodos que comparte a un mismo padre en común dentro de la estructura.

Nodo Hoja: Aquellos nodos que no tienen hijos, se encuentran en los extremos de la estructura.



Altura: Le llamamos altura al número máximo de niveles de un Árbol.



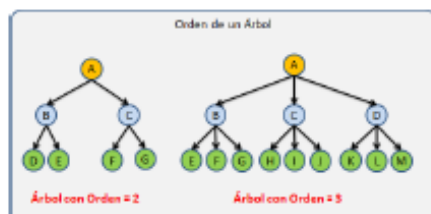
- si el árbol es vacío su altura es 0.

- si el árbol no está vacío su altura es 1 más que el máximo de las alturas de sus hijos.

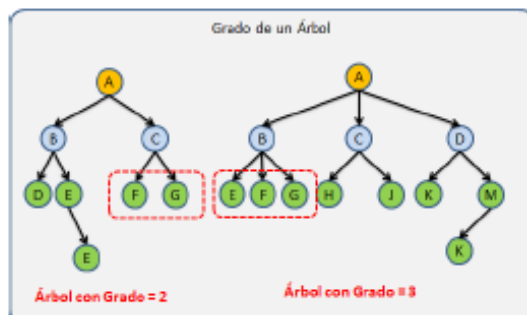
- La altura del nodo F sería 3.

Orden: El Orden de un árbol es el número máximo de hijos que puede tener un Nodo.

Un árbol binario podrá tener un máximo de 2 hijos, un ternario de 3 y así...



Grado: número mayor de hijos que tiene alguno de los nodos del Árbol, está limitado por el orden.



CSV (Comma Separated Values)



CSV es un formato de archivo de texto que se utiliza para **almacenar datos en forma de tabla**, en la que cada fila representa un registro y cada columna un campo del registro. No está estandarizado, pero es **ampliamente utilizado**.

Características principales

- ✓ Cada campo se separa por una **coma (,)** o por **punto y coma (;)** (según la configuración).
- ✓ Cada registro (fila) se separa por un salto de línea.
- ✓ Formato sencillo y ligero: fácil de leer, crear e intercambiar entre aplicaciones.
- ✓ No guarda formato (colores, estilos, fórmulas...). **Solo datos.**

Ejemplo de tabla (origen)

Año	Marca	Modelo	Descripción	Precio
1997	Ford	E350	ac, ABS, moon	3000.00
1999	Chevy	Venture	Extended Edition	4900.00
1999	Chevy	Venture	Extended Edition, Very Large	5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

Representación en CSV (texto)

CSV

```
Año,Marca,Modelo,Descripción,Precio
1997,Ford,E350,"ac, ABS, moon",3000.00
1999,Chevy,Venture,"Extended Edition",4900.00
1999,Chevy,Venture,"Extended Edition, Very Large",5000.00
1996,Jeep,Grand Cherokee,"MUST SELL! air, moon roof, loaded",4799.00
```

Ejemplos importantes

- El separador más común es la coma (,) pero puede ser ; (punto y coma).
- Si un campo contiene el separador, saltos de línea o comillas dobles, debe ir entre comillas dobles (").
- Las comillas dobles dentro de un campo se escriben duplicadas: "".
- No existe un estándar único (RFC 4180 es la referencia más usada, pero no obligatorio).

Ventajas e inconvenientes

✓ Ventajas

- Sencillo, ligero y legible.
- Fácil de generar e importar en muchas aplicaciones (Excel, bases de datos, scripts...).

✗ Inconvenientes

- No guarda estructura ni tipos de datos.
- Ambigüedades posibles (separadores, codificación, comillas...).

SGML (Standard Generalized Markup Language)



SGML es un **metamodelo** para definir lenguajes de marcado.

Permite **estandarizar y definir formalmente un documento electrónico**, independientemente del software, la aplicación o el sistema que se utilice. HTML y XML **provienen de él**.

Características clave

- ✓ Define la estructura y las reglas de documentos.
- ✓ Permite crear lenguajes de marcado propios (DTD de SGML).
- ✓ Es independiente de la plataforma y del software.
- ✓ Proporciona gran flexibilidad y capacidad de descripción.



Relación con otras tecnologías



HTML

Lenguaje de presentación en la web (derivado de SGML a través de HTML SGML).



XML

Lenguaje de marcado extensible (derivado directamente de SGML).



Otros

DocBook, MathML, etc. también se basan en SGML.

Conceptos asociados

- **DTD (Document Type Definition)**: define las etiquetas permitidas, su estructura y las reglas del lenguaje SGML.
- **Instancia**: documento concreto que cumple las reglas definidas en un DTD.
- SGML es **más complejo** que XML, por eso XML simplifica su sintaxis.
- Hoy en día SGML se usa menos directamente, pero sigue siendo la **base teórica** de muchos lenguajes de marcado.

BLOQUE 3 (Desarrollo de sistemas)

Contenido

TEMA1. DISEÑO DE BASES DE DATOS	3
Modelo entidad Relación Extendido	5
Paso del diagrama Entidad-Relación al modelo relacional (Diseño lógico) ...	7
Modelo relacional.....	8
Normalización	11
TEMA2. LENGUAJE JAVA	13
LOS 4 PILARES FUNDAMENTALES DE LA POO	13
Condicionales	17
Tipos de bucles	18
LENGUAJE JAVASCRIPT. ECMAscript 2025 (última versión).....	26
LENGUAJE C#.....	28
LENGUAJE C++	30
LENGUAJE C	30
LENGUAJE PYTHON, versión 3.13	31
LENGUAJE PHP (Hypertext Preprocessor)	34
GO (GOLANG)	38
Los ORM (Object-Relational Mapping) permiten interactuar con bases de datos sin escribir SQL manualmente.....	39
TEMA 3. SQL (Structured Query Language)	40
CONSULTAS SELECT	41
Oracle Database	54
Conexión con la base de datos(ejemplo de conexión JDBC):	55
Sintaxis de la conexión JDBC.....	55
Tipos de Drivers JDBC.....	55
TEMA 4. UML 2.5.1 Y PATRONES	56
TIPOS DE DIAGRAMAS UML.....	56
DIAGRAMA EXCLUSIVO DE METRICA V3.....	62
METRICA V3.....	63
Patrones de diseño GoF (Gang Of Four).....	63
TEMA 5. Arquitectura Jakarta EE 10 / Jakarta EE 11	65
Arquitectura .NET	67

Compilación en .NET	67
ADO.NET	68
Visual Basic .Net (VB)	69
ASP.NET	70
DESARROLLO DE INTERFACES	70
Novedades de .NET 9 (2024)	70
TEMA 6. SERVICIOS WEB	71
SOAP (Simple Object Access Protocol):	71
REST (Representational State Transfer).....	72
Arquitectura cliente-servidor.....	73
Arquitectura multicapas.....	73
Arquitectura SOA	74
TEMA 7. HTML 5 (Hypertext Markup Language).....	75
CSS (Cascading Style Sheets).....	80
DTD (DOCUMENT TYPE DEFINITION).....	83
XML(Extensible Markup Language)	84
Json (JavaScript Object Notation)	86
CSV (Comma Separated Values)	87
SGML (Standard Generalized Markup Language)	87
TEMA 8. ACCESIBILIDAD	88
Conceptos de seguridad en aplicaciones WEB.....	90
TEMA 9. REPOSITORIO GIT.....	91
Softwares de control de versiones	92
Plataformas de desarrollo colaborativo de Software.....	92
Metodologías de desarrollo de software	93
Pruebas.....	94

BLOQUE 4 (Sistemas y comunicaciones)

Contenido

TEMA 1. Administración del sistema operativo y software de base (Este tema también se toca en el Bloque 2) ..	2
1. Gestión del sistema operativo.....	2
2. Actualización del sistema	2
Métodos de actualización:	2
Tipos de actualizaciones:	2
3. Mantenimiento del Sistema	2
Optimización del sistema:	2
Gestión del almacenamiento:	2
4. Reparación del Sistema	3
5. Firmware del sistema: BIOS, UEFI, MBR y GPT	3
TEMA 2. ADMINISTRACIÓN BBDD, SISTEMAS DE ALMACENAMIENTO	3
TEMA 3. SERVICIOS DE CORREO ELECTRÓNICO	5
TEMA 4. ADMINISTRACIÓN DE REDES	8
TEMA 5. CONCEPTOS DE SEGURIDAD, AMENAZAS	10
TEMA 6. COMUNICACIONES	14
Modelo organizativo (Internet y Redes)	14
Tema 7. MODELO OSI (Open Systems Interconnection)	17
ARQUITECTURA TCP / IP	18
NIVEL 2 - CAPA DE ENLACE DE DATOS	18
NIVEL DE RED (IPV4)	19
DIRECCIONAMIENTO- CLASSFUL	21
SUBNETTING (SOBRE CLASES)	22
TÉCNICA (CIDR) "CLASS LESS INTER DOMAIN ROUTING"	22
TEMA 8. ARQUITECTURA DE RED. PROTOCOLOS HTTP, HTTPS y SSL/TLS	25
HTTP Status Codes	27
HERRAMIENTAS CNN	28
TEMA 9. SEGURIDAD EN REDES DE COMUNICACIÓN, VPN	29
TEMA 10. REDES LOCALES	31