TEMA 1: Sistemas de Almacenamiento de Información

Módulo

Bases de Datos

para los ciclos

Desarrollo de Aplicaciones Web

Desarrollo de Aplicaciones Multiplataforma

1.1 Ficheros

La información que se almacena en los dispositivos de almacenamiento de un sistema informático se estructura en ficheros, también llamados archivos.

> Un ***fichero*** es un conjunto lógico de información relacionada (una canción, una película, un documento, ...), tratada como un todo y organizada de forma estructurada. Es una secuencia de dígitos binarios que organiza información relacionada con un mismo aspecto.

El ***Sistema Operativo*** será el encargado de gestionar estos ficheros agrupándolos en carpetas que a su vez pueden agruparse en otras carpetas.

Cada fichero tiene un formato, una estructura conocida que permite a una aplicación obtener la información que aloja. Por ejemplo un fichero que aloja una imagen podría contener los siguientes datos: dimensión X, dimensión Y, color del primer punto, color del segundo punto, ... El formato deberá ser conocido por la aplicación que deba interpretarlo.

Ejemplo: fichero para una imagen de tamaño X x Y:

X
Y
color primer punto
color segundo punto
.....
.....

Cada fichero tiene un nombre con una ***extensión***. La extensión es la parte del nombre que hay tras el último punto e indica el tipo de información que contiene el fichero. Los sistemas operativos suelen asociar a cada fichero ***un icono y una aplicación*** predeterminada en función de esa extensión.

Ni que decir tiene que cambiar la extensión de un fichero hará que el Sistema Operativo interprete que tiene otro formato (o sea, le asignará otro icono y otra aplicación) pero puesto que el contenido del fichero sigue siendo el mismo, el resultado será ilegible porque la aplicación espera un formato y se encuentra otro.

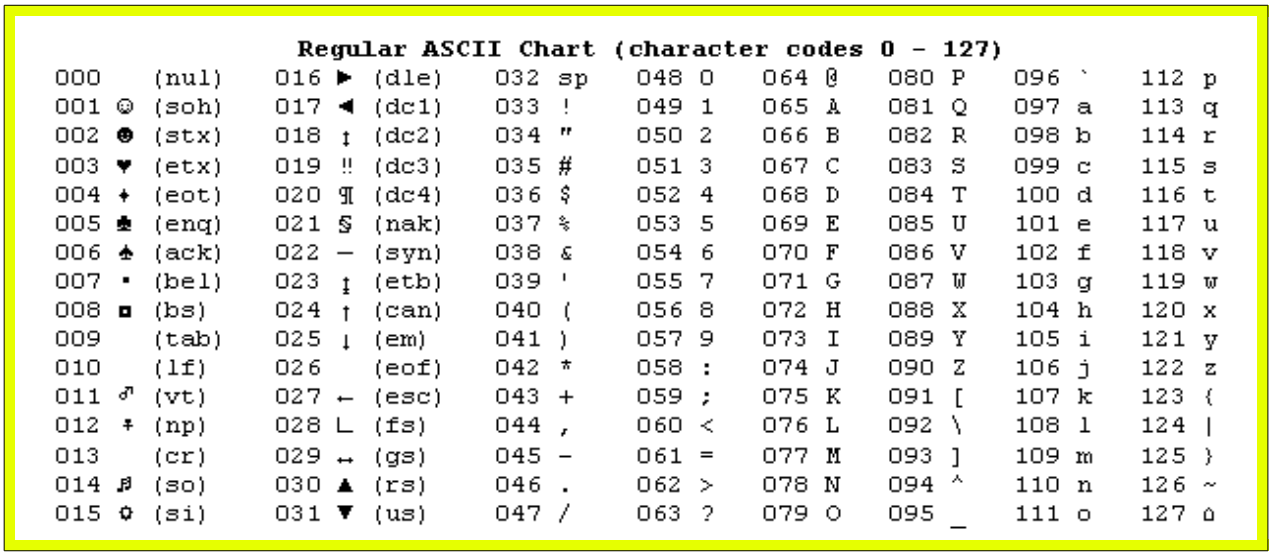
Igualmente, aunque no manipulemos la extensión de un fichero, si nos empeñamos en abrirlo con la aplicación equivocada, el resultado será imprevisible.

1.1.1 Tipos de ficheros

Se distingue fundamentalmente entre dos tipos de ficheros:

FICHEROS DE TEXTO

Los ***ficheros de texto plano*** almacenan una serie de caracteres imprimibles (letras, dígitos, signos de puntuación, ...), uno tras otro. No incluyen ninguna otra cosa, ni siquiera aspectos sobre como debe visualizarse. Cada uno de los caracteres se almacena con un código binario como ***ASCII*** (American Standard Code for Information International) o ***Unicode***.



Regular ASCII Chart (character codes 0 - 127)

000	(nul)	016 ►	(dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001 ☺	(soh)	017 ◄	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002 ☻	(stx)	018 ↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003 ♥	(etx)	019 ‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004 ♦	(eot)	020 ¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005 ♣	(enq)	021 §	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006 ♠	(ack)	022 ▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007 •	(bel)	023 ↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008 ◘	(bs)	024 ↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009	(tab)	025 ↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	(lf)	026 →	(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011 ♂	(vt)	027 ←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012 ♀	(np)	028 └	(fs)	044	,	060	<	076	L	092	\	108	l	124	|
013	(cr)	029 ↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014 ♫	(so)	030 ▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015 ☼	(si)	031 ▼	(us)	047	/	063	?	079	O	095	_	111	o	127	⌂

En la imagen vemos la asignación de códigos de la primera mitad de la tabla ***ASCII***, es decir, desde 0 hasta 127, conocida como ***ASCII Standard***. En este código, cada carácter se representa como un byte lo que hace que los ficheros ocupen muy poco espacio. La desventaja es que sólo permite 256 caracteres distintos y deja de ser útil en un entorno global como Internet donde se necesita poder usar símbolos de muchos alfabetos simultáneamente.

Actualmente se usan otros códigos que permiten muchos más caracteres, ocupando más espacio cada uno de ellos. Una versión de ***Unicode*** (***UTF-8***) usa un tamaño variable de códigos entre 1 y 4 bytes para cada carácter, de modo que los caracteres más habituales sólo ocupan un byte y respetan los valores del ***ASCII Standard***.

Los ficheros de texto pueden tener extensiones distintas. La más habitual es ***.txt***, pero dependiendo de su contenido tenemos ***.ini***, ***.inf***, ***.conf***, ***.c***, ***.java***, ***.html***, ***.css***, ***.php***, ***.xml***, …

Nota: Es importante no confundir ficheros de texto con documentos de texto. Un fichero de texto, por ejemplo, ***alfabeto.txt***, se genera y edita con un editor de textos (***Bloc de Notas*** o ***gedit***, ...) y contiene exclusivamente un código por cada carácter; no admite formato. Un documento de texto, por ejemplo, ***alfabeto.doc***, se genera y edita con un procesador de textos (***MS-Word*** o ***Writer*** de ***LibreOffice***, ...) y contiene todos los datos necesarios para representar el contenido en papel: tamaño del papel, color, tipo y tamaño de letra, alineación, etc, además naturalmente del contenido.

Puedes comprobar esta diferencia fácilmente. Crea un documento con un procesador de textos que incluya una frase y crea otro documento con un editor de textos que incluya la misma frase. Después compara el tamaño de ambos. También puedes probar a abrir cada uno de ellos con la aplicación intercambiada (el fichero de texto con el procesador y el documento con el editor) para ver que pasa.

FICHEROS BINARIOS

Los ***ficheros binarios*** tienen un formato que será reconocido por la aplicación adecuada. Vídeos, música, documentos de texto formateado (generados por procesadores de textos) se almacenan en ficheros binarios.

Generalmente, aunque hay excepciones, los datos de una base de datos se almacenan en ficheros binarios. Sus extensiones y formatos dependerán del ***SGBD*** con el que se hayan generado: ***.mdb*** para ***MS-Access***, ***.frm***, ***.myd***, o ***.myi*** para ***MySQL***, ***.db*** para ***SQLite***, etc.

1.2 Bases de Datos

Una ***Base de Datos*** es una colección de información relativa a un contexto o problema, organizada con una cierta estructura, que se almacena en ficheros relacionados entre si y que mantiene los datos de forma ordenada, coherente (sin contradicciones) y sin duplicidades.

Que una **base de datos sea plana**, básicamente significa que está compuesta de una sola tabla. En los casos reales esto no ocurre nunca porque cualquier organización tendrá muchos más datos que almacenar y hacerlo en una única tabla nos llevará a repetir muchas veces el mismo dato.

Repetir varias veces el mismo dato aumenta el trabajo y el espacio de almacenamiento, pero ocasiona problemas mayores, por ejemplo, podemos incluir errores al repetirlo y que la base de datos los tome por datos distintos.

Otro problema es que si el dato cambia (por ejemplo una persona cambia su número de teléfono), tendremos que hacer el cambio en todas las repeticiones.

Veamos un ejemplo de base de datos plana:

Paciente	Dirección	Diagnóstico	Doctor	Especialidad	Consulta
Pedro	Luna, 10	Sarampión	López	General	310
Luís	Arroyo, 15	Sinusitis	Alonso	Otorrino	402
Pedro	Luna, 10	Diabetes	Sanz	Digestivo	120
Luís	Arroyo, 15	Fractura	López	General	310
Ana	Estrella, 22	Colitis	Sanz	Digestivo	120

Sólo hay que echar un vistazo para ver que a los pacientes ***Pedro*** y ***Luís*** les han atendido en dos ocasiones y en las dos ocasiones hemos tenido que incluir su dirección. Del mismo modo, cada vez que el doctor ***López*** atiende a un paciente (lo que ocurrirá muchas veces al día), debemos anotar su especialidad y su consulta (que siempre son las mismas).

Este mismo ejemplo lo vamos a desarrollar en una **base de datos relacional**. En este caso la base de datos estará compuesta de tres tablas, pero nos encargaremos de hacer las conexiones adecuadas para que los datos no queden aislados y se puedan relacionar los datos de unas tablas con los datos de las otras:

TABLA 1		
Cod.	Paciente	Dirección
401	Pedro	Luna, 10
402	Luís	Arroyo, 15
403	Ana	Estrella, 22

TABLA 2			
Cod.	Doctor	Especialidad	Consulta
100	López	General	310
101	Alonso	Otorrino	402
102	Sanz	Digestivo	120

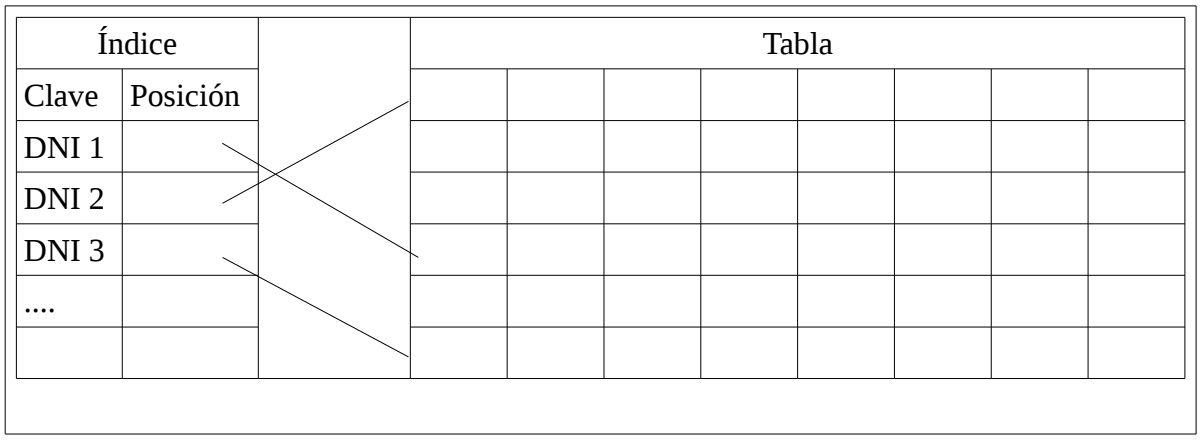
TABLA 3		
Pac.	Doctor	Diag.
401	100	Sarampión
402	101	Sinusitis
401	102	Diabetes
402	100	Fractura
403	102	Colitis

Cada vez que aparece un paciente nuevo incluimos sus datos en la tabla 1 y no tendremos que volver a hacerlo. Para cada doctor usamos la tabla 2, sólo una vez. Y por cada visita de un paciente

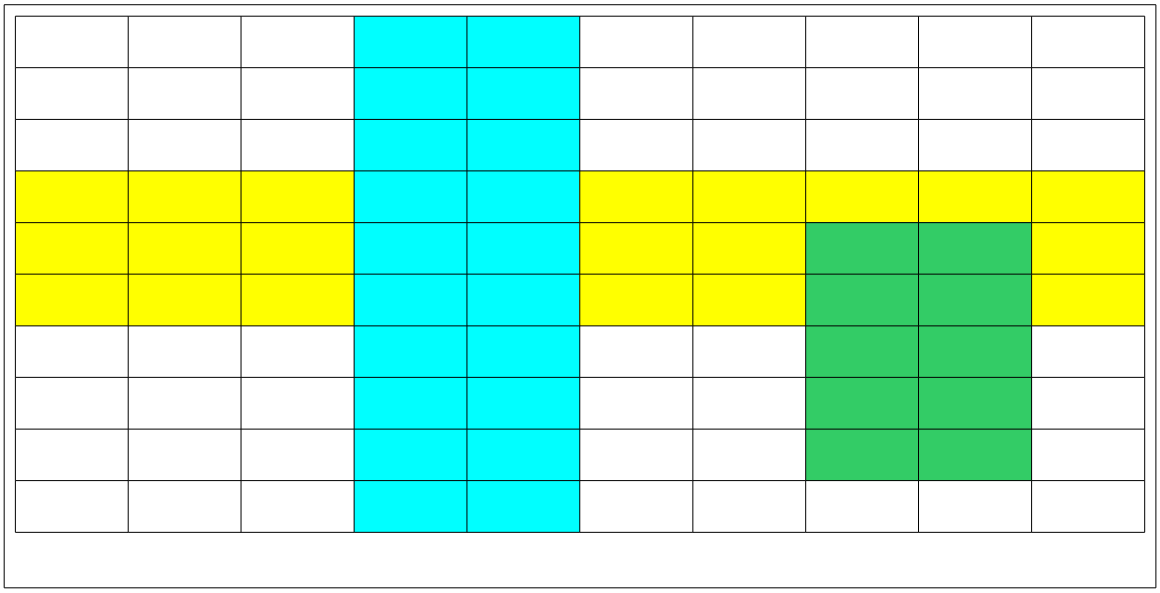
a un doctor anotamos un registro en la tabla 3. Sólo tendremos que incluir los datos propios de la visita, como el diagnóstico. Los datos del paciente y del doctor, sin tener que repetirlos, podrán ser accedidos gracias al código, que relaciona este registro con otro registro de la tabla 1 y de la tabla 2.

1.2.1 Conceptos

- ***Dato***: un trozo de información indivisible: 1960, López, etc. Cada dato será de un tipo.
- ***Tipo de dato***: indica la naturaleza del dato (carácter, numérico, booleano, etc)
- ***Campo***: (o columna) es un identificador de toda una familia de datos que además serán todos del mismo tipo. Por ejemplo, un campo es ***primer apellido***. Un dato de ese campo podría ser ***López.***
- ***Registro***: (o tupla o fila) Es el conjunto de todos los datos que definen a un elemento de nuestra base de datos: una ***persona***, un ***vehículo***, ...
- ***Campo clave***: Es un campo especial que identifica de forma única a cada registro, por ejemplo un ***NIF***.
- ***Tabla***: Es el conjunto de todos los registros que definen elementos del mismo tipo: ***Clientes***, ***Médicos***...
- ***Consulta***: Instrucción para hacer peticiones a la BD. El resultado será un dato o conjunto de datos que satisfacen un grupo de criterios. Viene del inglés ***query*** y quizá ***petición*** sería mejor traducción porque también existen consultas que no devuelven datos, sino que eliminan o modifican datos.
- ***Índice***: Estructura que almacena los campos clave de una tabla organizándolos para hacer más fácil la búsqueda. Manejar (ordenar) el índice es más fácil y rápido que manejar la tabla completa. Un índice guarda parejas ***clave-posición***.



- ***Vista***: Es una tabla virtual (no existe como tal en la BD, aunque a todos los efectos la podremos manejar como una tabla más) que contiene un subconjunto de filas y/o columnas de una o más tablas.



- ***Informe***: Es un conjunto de datos seleccionados que se presentan en el formato adecuado para un usuario: en pantalla, en A4,...

- ***Guiones***: (o ***scripts***) conjunto de instrucciones que actúan sobre la BD para realizar operaciones de mantenimiento.

- ***Procedimientos***: Script especial que está almacenado en la BD y forma parte de su esquema.

1.3 Sistemas Gestores de Bases de Datos

Un ***Sistema Gestor de Bases de Datos*** (***SGBD***) es un conjunto de herramientas que facilitan la consulta, uso y actualización de un BD, es decir, es la interfaz entre el usuario y la base de datos. Ejemplos: ***Oracle***, ***MySQL***, ***DB2***, ***MS-Access***, ***SQLite***.

1.3.1 Funciones de un SGBD

Las funciones de un ***SGBD*** son las siguientes:

- Permitir a los usuarios almacenar datos, acceder a ellos y actualizarlos, ocultando las características físicas.
- Garantizar la integridad de los datos, según las reglas que dicte el programador. O sea, no se permiten operaciones que dejen datos incompletos o incorrectos.
- Integrar, junto al SO, un sistema de seguridad en el acceso de usuarios.
- Proporcionar un diccionario de metadatos, con el esquema de la BD, o sea, la estructura de tablas, registros y campos, usuarios y permisos,.. Debería de ser tan accesible como el resto de datos.
- Permitir uso de transacciones, garantizando su corrección o deshaciendo los cambios cuando no sea posible.
- Ofrecer estadísticas del funcionamiento de la BD que incluyan cualquier incidencia.
- Permitir concurrencia de usuarios sobre un mismo conjunto de datos, arbitrando mecanismos para los conflictos que esto genere.
- Independizar los datos de la aplicación para facilitar la exportación.
- Ofrecer conectividad con el exterior.
- Incorporar herramientas para copias de seguridad y restauración en caso de desastre.

1.3.2 El Lenguaje SQL

El lenguaje ***SQL*** (***Structured Query Language***, estandarizado por la ***ISO***) es la interfaz de programación del ***SGBD*** . Todas las BD que soporten ***SQL*** deberán respetar el estándar. Se divide en 4 sublenguajes:

- ***DML***: (Data Manipulation Language, Lenguaje de Manipulación de Datos) incluye las instrucciones que permiten añadir o borrar datos, actualizarlos o consultarlos: ***SELECT***, ***INSERT***, ***UPDATE***, ***DELETE***.
- ***DDL***: (Data Definition Language, Lenguaje de Definición de Datos) incluye las instrucciones necesarias para crear o eliminar tablas: ***CREATE*** y ***DROP***.
- ***DCL***: (Data Control Language, Lenguaje de Control de Datos) para gestionar el acceso de los usuarios a la BD: ***GRANT*** y ***REVOKE***.
- ***TCL***: (Transaction Control Language, Lenguaje de Control de transacciones) para completar o deshacer transacciones: ***COMMIT*** y ***ROLLBACK***.

1.3.3 Tipos de SGBD

- ***SGBD ofimáticas***: para BD pequeñas de uso doméstico o pequeñas empresas. Ejemplos: ***MS-Access***, ***Base de Datos de LibreOffice***.
- ***SGBD corporativas***: tienen capacidad para gestionar BD enormes para medianas y grandes empresas. Requieren servidores de grandes prestaciones. Manejan grandes cantidades de datos rápida y eficientemente, atendiendo a muchos usuarios. Ejemplos: ***Oracle***, ***DB2*** que son software propietario y bastante caro. Una solución intermedia es ***MySQL***, gratuito.