

Contenidos: semana 1

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)

Curso: Administración de Bases de Datos 1° A

Libro: Contenidos: semana 1

Imprimido por: Christian Pappolla

Día: martes, 9 de agosto de 2022, 22:08

Tabla de contenidos

1. Datos e información

1.1. Campos, Registros y arreglos

2. Sistemas de archivos

2.1. Organización de los archivos

2.2. Archivos de Textos

2.3. Archivos XML

2.4. Archivos JSON

1. Datos e información



¿Qué es un dato?

Los datos son Valores almacenados que representan hechos o realidades del mundo real.

¿Qué es información?

Se refiere a los datos que han sido procesados y comunicados de tal manera que pueden ser entendidos e interpretados por el receptor.

Información es el significado de los datos

¿Por qué la necesidad de información?

Información también puede decirse, con más precisión, que Información se refiere al conocimiento derivado del análisis o tratamiento de los datos. Este conocimiento o información se utiliza para tomar decisiones con vistas a un accionar concreto. Esta es la importancia que tiene la Informática en la actualidad, permite obtener información confiable, precisa y oportuna para tomar mejores decisiones. Esto permite a las empresas y organizaciones el logro eficiente de sus objetivos.

¿Qué es un sistema de información?

Un sistema de información es una colección de datos debidamente recopilados y estructurados, que proporcionan información sobre la realidad.

- Los sistemas de información existen desde las primeras civilizaciones (censo romano).
- Revolución tecnológica: Desarrollo de la Informática.
- En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.
- Se destacan por su diseño, facilidad de uso, flexibilidad, mantenimiento automático de los registros, apoyo en toma de decisiones críticas y mantener el anonimato en informaciones irrelevantes.
- Primeros sistemas de información informatizados: sistemas de ficheros.

Codificación de los datos

Los datos que manejan las computadoras es digital.

Se construye a partir de las unidades contables llamadas dígitos. Las unidades de una computadora están constituidas por circuitos formados por componentes electrónicos denominados puertas, que manejan señales eléctricas que no varían de modo continuo sino que solo pueden tomar dos estados discretos (dos voltajes). Cerrado y abierto, bajo y alto, 0 y 1. De este modo la memoria de una computadora esta formada por millones de componentes de naturaleza digital que almacenan uno de dos estados posibles

¿Cómo se manejan todos estos tipos de dato?

Representación uniforme al ser utilizados y almacenados por la PC. (PATRÓN DE BITS.)

¿Qué es un BIT?

Binary Digit = Dígito Binario

Unidad más pequeña de datos que puede almacenarse en una computadora (0 o 1).

¿Qué es un patrón de BIT?

Número sucesivos de BIT que representa un dato

- Ejemplo un patrón de bit de 16 bits:

1 0 0 0 1 0 1 0 1 1 1 1 1 1

- Ejemplo de patrón de 8 bits (Llamado byte)

B Y T E

1000010 1011001 1010100 1000101

Este sistema de 8 bit para representar los caracteres corresponde al código ASCII.

B = 66 en base 10.

¿Unidades de Medida de la Información?

La capacidad de almacenamiento de una computadora o de un soporte de información, (tal como disco, disquete o cinta magnética) se mide en bytes.

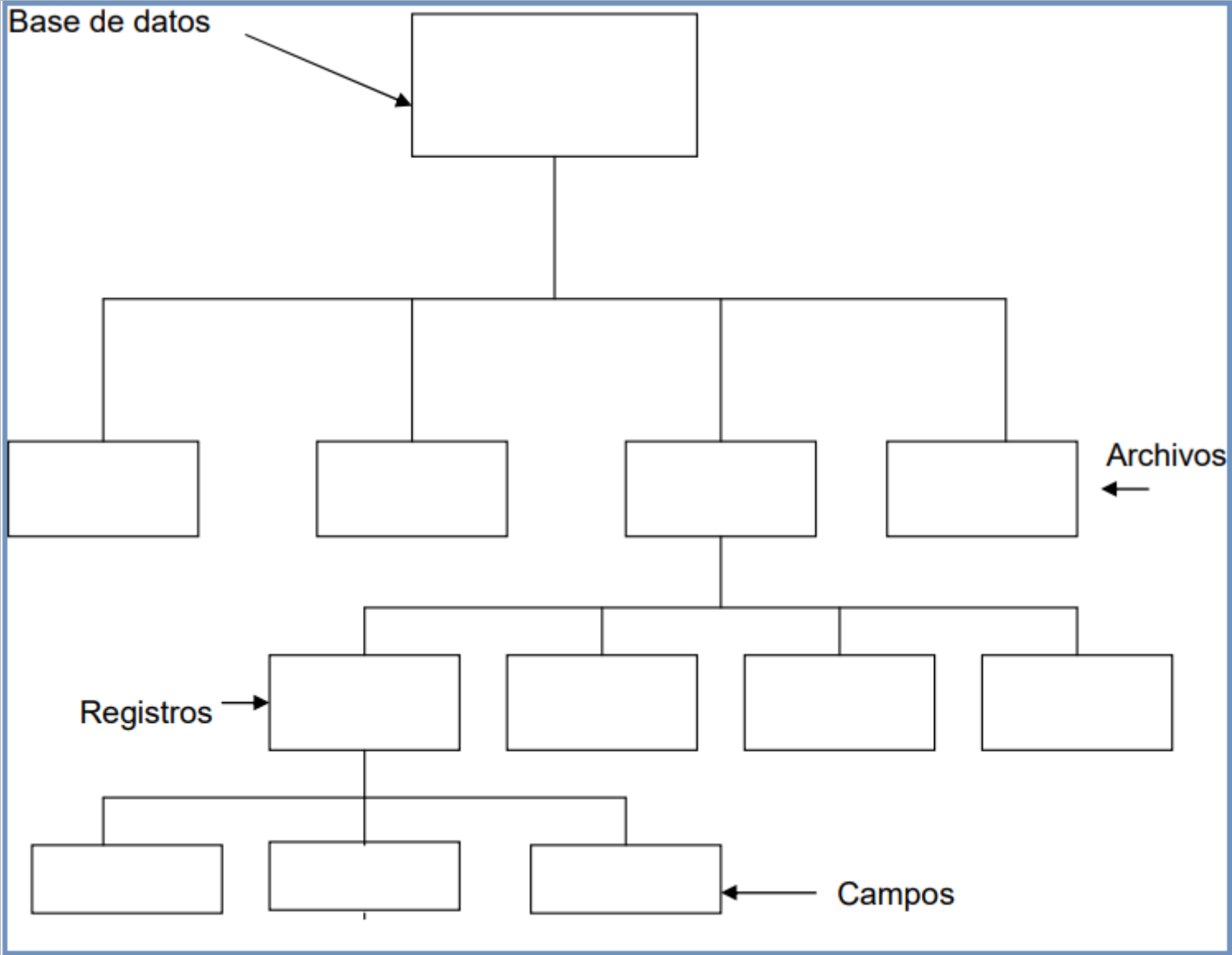
- 1 Byte 8 bits
- 1 Kilobyte (KB) 1024 bytes
- 1 Megabyte (MB) 1024 KB = 1.048.576 bytes
- 1 Gigabyte (GB) 1024 MB = 1.073.741.824 bytes
- 1 Terabyte (TB) 1024 GB
- 1 Petabyte (PB)
- 1024 TB
- 1 Exabyte (EB)
- 1024 PB

La razón por la que se utiliza el factor multiplicador 1024 en lugar de 1000 es por ser la potencia de 2 más próxima a 1000, cuestión importante desde el punto de vista electrónico.

¿Cómo identifica la memoria de la computadora el tipo de datos?

No los identifica, solo almacena los datos como PATRONES DE BITS.
Los dispositivos de entrada/salida o programas interpretan un patrón de bit.

Estructura de la jerarquía de los datos



1.1. Campos, Registros y arreglos

Campo

- Un campo es un espacio de almacenamiento para un dato en particular.
- Cada campo tiene un nombre.
- Por ejemplo, algunos campos son numéricos mientras otros almacenan texto, también varía el tamaño de estos.
- Al conjunto de ellos forman un registro
- En las bases de datos, un campo es la mínima unidad de información a la que se puede acceder.

Registro

- Es un conjunto de campos referentes a una entidad en particular y constituyen una unidad para su proceso.
- Un ejemplo de un registro puede ser la información de un determinado Empleado o, que contiene los campos: Legajo, nombre, apellido, dni, etc.
- Un registro (también llamado fila o tupla) representa un ítem único de datos implícitamente estructurados en una tabla.

Ejemplo de un registro de un empleado:

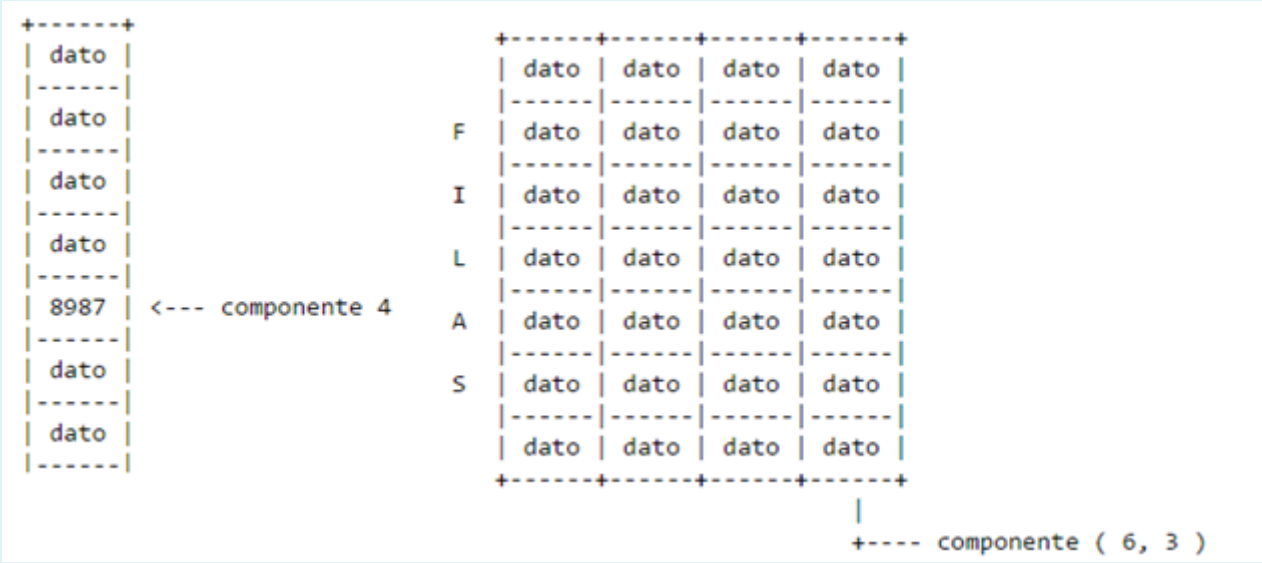
Legajo	Nombre	Apellido	DNI	FechaNacimiento
--------	--------	----------	-----	-----------------

Arreglos

- Es un tipo de dato cuya función es tener una lista de campos en memoria (números, textos, registros, objetos, etc).
- Todos los elementos son del mismo tipo de datos. A diferencia de los registros que estos tienen una estructura que soportan distintos tipos de datos.
- Tienen dimensión (1 - Vector , 2 – matriz, N - Multidimensional)
- Los vectores se almacenan en memoria central de la computadora en un orden adyacente.

Así, un vector con 10 elementos se almacena físicamente en 10 posiciones de memoria sucesivas.

Ejemplo de un vector y una matriz

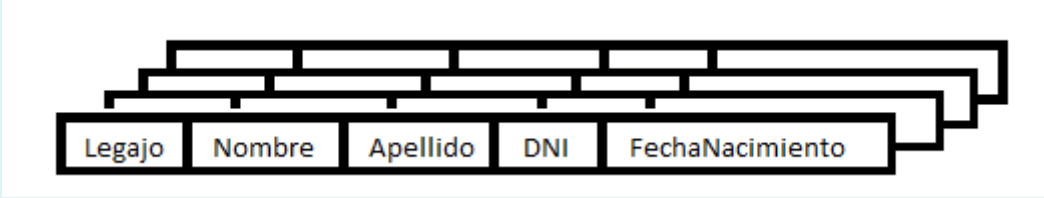


2. Sistemas de archivos

Archivos

Es el conjunto de registros homogéneos referidos a objetos de la misma naturaleza o del mismo tipo, almacenados en un soporte externo, que presenta entre sí una relación lógica y que pueden ser consultados individualmente de forma iterativa o sistemática.

La representación de un archivo donde se muestra como representa un conjunto de registros.



Características

- Posee nombre. Único en la misma ubicación.
- Acciones posibles: Apertura, Cierre, fin de archivo, fin de línea, Modificación y lectura.
- Tamaño.
- Formato.

Acciones sobre los archivos

- Apertura: En caso no existir crea un archivo vacío en caso de existir este queda abierto, pudiendo borrar el contenido o no, dependiendo el uso.
- Cierre: El archivo una vez manipulado más que nada en la alteraciones de e contenido debe cerrarse para marcar el fin del mismo.
- Fin de archivo (EOF): Es una marca que poseen los archivos para indicar donde es su fin.
- Fin de línea (EOFL): Usado en archivos de textos indica el fin de línea del renglón.
- Modificación: Entre las acciones de modificaciones tenemos el Alta, baja y modificación de registros.
- Lectura: Es la operación que nos permite acceder al archivo de datos para conocer la información alojada en uno, varios o todos los registros del archivo.

Sistemas de archivos

Un Sistema de Archivos (SF) es un conjunto de programas informáticos que permiten al usuario almacenar, consultar y modificar datos. Dichos datos se almacenan en ficheros diseñados para una determinada aplicación. Cada programa define y maneja sus propios datos.

Desventaja

- Separación y aislamiento de datos.
- Duplicación de datos (se desperdicia capacidad de almacenamiento y se puede perder la coherencia de datos).
- Dependencia de los datos (la estructura física de los datos se encuentra codificada en cada programa de aplicación, dificultad para realizar cambios en dicha estructura).
- Formatos de ficheros incompatibles (la estructura de cada fichero es completamente dependiente de cada aplicación particular).
- Consultas fijas (los SF dependen del programador de las aplicaciones, poco adecuado para recuperar posteriormente la información según criterios cambiantes).

Los archivos también denominados ficheros (file); es una colección de información (datos relacionados entre sí), localizada o almacenada como una unidad en alguna parte de la computadora.

2.1. Organización de los archivos

Soporte de almacenamiento Archivos

Los soportes secuenciales son aquéllos en los que los registros informaciones están escritos unos a continuación de otros y para acceder a un determinado registro, n se necesita pasar por los n-1 registros anteriores.



Los soportes direccionables se estructuran de modo que las informaciones registradas se pueden localizar directamente por su dirección y no se requiere pasar por los registros precedentes. En estos soportes los registros deben poseer un campo clave los diferencie del resto de los registros del archivo. Una dirección en un soporte direccionable puede ser número de pista y número de sector en un disco.

Los soportes direccionables son los discos magnéticos, aunque pueden actuar como soporte secuencial.

Según las características del soporte empleado y el modo en que se han organizado los registros, se consideran dos tipos de acceso a los registros de un archivo:



La secuencia puede corresponder al orden físico de los registros en el archivo (organización secuencial) o bien al orden de claves (ascendente o descendente) de los registros (organización indexada).

Tipos de archivos

Podemos dividir los archivos en dos grandes grupos. Éstos son los ejecutables y los no ejecutables o archivos de datos. La diferencia fundamental entre ellos es que los primeros están creados para funcionar por si mismos y los segundos almacenan información que tendrá que ser utilizada con ayuda de algún programa.

Dentro de los tipos de archivos nos vamos a centrar en aquellos que permiten guardar información que luego podemos consultar.

Estos archivos pueden ser archivos *binarios* o de *texto*. Los dos tipos de archivos pueden tener el mismo aspecto en la superficie, pero codifican datos de forma diferente.

Mientras que tanto los archivos binarios como los de texto contienen datos almacenados como una serie de bits (valores binarios de 1s y 0s), los bits de los archivos de texto representan caracteres, mientras que los bits en los archivos binarios representan datos personalizados.

Mientras que los archivos de texto contienen solamente datos de texto, los archivos binarios pueden contener tanto datos binarios personalizados y de texto.

Los archivos de datos binarios se diferencian en dos grupos de acuerdo al tipo de acceso. Pudiendo ser de acceso secuencial o indexado.

Archivos Secuenciales

- Se refiere al procesamiento de los registros.
- Los registros están organizados en forma de una lista.

- Son recuperados y procesados uno por uno de principio a fin.
- Para ir a registro N-1 hay que pasar por cada uno de los registros que lo anteceden.



Archivos indexados

- Consiste en incluir índices en el almacenamiento de los archivos.
- Facilita la búsqueda de algún registro.
- Un índice en un archivo consiste en un listado Clave - valor
- La búsqueda se hace primero sobre el índice, el contiene el puntero al registro del archivo, de forma tal que no hace falta leer todos los registros para un buscar un registro por clave.

Puntero	Legajo	nombre	apellido	dni
5	123	Miguel	Russo	4255252
6	11	Miguel	Brindizi	4255252
7	122	chino	benitez	4255252
8	22	Guillermo	scheletto	4255252
9	58	Carlos	bianchi	4255252
10	858	Carlos	Harberger	4255252

Clave	Dirección (Puntero)
Benitez	7
bianchi	9
brindizi	6
Harberger	10
russo	5
scheleto	8

2.2. Archivos de Textos

- Son utilizados para almacenar documentos que consisten en texto.
- La lectura de estos archivos es secuencial.
- Cada renglón se divide por un Fin de línea (EOLn).
- El fin de este archivo es con End on File (EOF).
- Dependen del formato de codificación de caracteres (ASCII, UTF, EBCDIC).

Qué es la codificación de caracteres

La codificación de caracteres (character encoding) es el método que permite convertir un carácter de un lenguaje natural (alfabeto) en un símbolo de otro sistema de representación, como un número, un símbolo o una secuencia de pulsos eléctricos en un sistema electrónico, aplicando una serie de normas o reglas de codificación. En este artículo vamos a centrarnos en cómo los sistemas informáticos utilizan la codificación de caracteres para convertir y representar los diferentes lenguajes escritos del mundo a una representación que dicho sistema entienda, y así poder almacenar o transmitir dicha información.

La codificación de caracteres se basa en definir tablas que indiquen el carácter en el lenguaje natural y su correspondencia en el lenguaje del sistema informático.

Estas tablas se denominan conjunto de caracteres (charset o character set), mapa de caracteres (character map) o página de códigos (page code).

Ejemplo Tabla ASCII

Decimal			Hex			Char		
0	0	[NULL]	32	20	[SPACE]	64	40	@
1	1	[START OF HEADING]	33	21	!	65	41	A
2	2	[START OF TEXT]	34	22	"	66	42	B
3	3	[END OF TEXT]	35	23	#	67	43	C
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D
5	5	[ENQUIRY]	37	25	%	69	45	E
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F
7	7	[BELL]	39	27	'	71	47	G
8	8	[BACKSPACE]	40	28	(72	48	H
9	9	[HORIZONTAL TAB]	41	29)	73	49	I
10	A	[LINE FEED]	42	2A	*	74	4A	J
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K
12	C	[FORM FEED]	44	2C	,	76	4C	L
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M
14	E	[SHIFT OUT]	46	2E	.	78	4E	N
15	F	[SHIFT IN]	47	2F	/	79	4F	O
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W
24	18	[CANCEL]	56	38	8	88	58	X
25	19	[END OF MEDIUM]	57	39	9	89	59	Y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z
27	1B	[ESCAPE]	59	3B	;	91	5B	[
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_
96	60	`	100	64	d	104	68	h
97	61	a	101	65	e	105	69	i
98	62	b	102	66	f	106	6A	j
99	63	c	103	67	g	107	6B	k
			104	68	h	108	6C	l
			105	69	i	109	6D	m
			110	6E	n	111	6F	o
			111	6F	o	112	70	p
			112	70	p	113	71	q
			113	71	q	114	72	r
			114	72	r	115	73	s
			115	73	s	116	74	t
			116	74	t	117	75	u
			117	75	u	118	76	v
			118	76	v	119	77	w
			119	77	w	120	78	x
			120	78	x	121	79	y
			121	79	y	122	7A	z
			122	7A	z	123	7B	{
			123	7B	{	124	7C	
			124	7C		125	7D	}
			125	7D	}	126	7E	~
			126	7E	~	127	7F	[DEL]

2.3. Archivos XML

- XML, siglas en inglés de eXtensible Markup Language, traducido como "Lenguaje de Marcado Extensible" o "Lenguaje de Marcas Extensible“.
- Es un meta-lenguaje que permite definir lenguajes de marcas.
- Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil

Estructura de un documento XML

- La estructura de un archivo XML consta de etiquetas para así poder almacenar información que nos interesa para después lograr un acceso a esta.
- Un documento XML consta de un prólogo y un elemento raíz. El prólogo contiene la información (meta información) sobre el resto del documento.
- También en él se incluye la descripción de estructura del resto del documento, generalmente recogida en un DTD (Document Type Definition) o en un Schema XML.
- El documento debe tener un solo elemento raíz.
- No se pueden repetir atributos en un mismo elemento.
- Los valores de los atributos deben ir entre comillas simples o dobles.
- Todos los elementos deben estar correctamente anidados.
- XML distingue mayúsculas de minúsculas por lo que todos los elementos y atributos deben seguir la definición.
- Todas las etiquetas (tags) abiertas deben tener su respectivas etiquetas de cierre.

ejemplo de archivo xml:

```
<?xml versión="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE agenda SYSTEM "agenda.dtd">
<agenda>
  <entrada>
    <nombre-completo>Marta Elena Zorrilla
    </nombre-completo>
    <direccion>
      <calle> Avda Los Castros</calle>
      <ciudad>Santander</ciudad>
      <codigo-postal>39005</codigo-postal>
      <region>Cantabria</region>
      <pais>España</pais>
    </direccion>
    <tel preferente="true"> 942201363</tel>
    <correo-e href="zorrillm@unican.es"></correo-e>
  </entrada>
</agenda>
```

2.4. Archivos JSON

- JSON (acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.
- Si bien es frecuente ver JSON posicionado contra XML, también es frecuente el uso de JSON y XML en la misma aplicación. Por ejemplo, una aplicación de cliente que integra datos de Google Maps con datos meteorológicos en SOAP hacen necesario soportar ambos formatos.
- JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc, que atienden a millones de usuarios) cuando la fuente de datos no es importante el no disponer de procesamiento seguro para manipular los datos en el cliente.
- En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o de sus rendimientos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de eval() y el auge del procesamiento nativo de XML incorporado en los navegadores modernos.
- Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función eval(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

Tipo de datos JSON

- **Números:** Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos.

Ejemplo: 123.456

- **Cadenas:** Representan secuencias de cero o más caracteres. Se ponen entre doble comilla.

Ejemplo: "Hola"

- **Booleanos:** Representan valores booleanos y pueden tener dos valores: true y false

- **null:** Representan el valor nulo.

- **Array:** Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes.

Ejemplo ["juan","pedro","jacinto"]

- **Objetos:** Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas. El valor puede ser de cualquier tipo.

Ejemplo:

```
{
  "departamento":8,
  "nombredepto":"Ventas",
  "empleados":[{"nombre":"Pedro", "apellido":"Fernandez"} ]
}
```