Bases de Datos para el Big Data

Métodos de captura de información

Índice

MÉTODOS DE CAPTURA DE INFORMACIÓN

Origen y calidad de los datos

Evaluación de calidad:
- **Completitud**: grado en el que los valores se encuentran en un conjunto de datos.
- **Credibilidad**: nivel de fiabilidad del organismo que proporciona el conjunto de datos.
- **Consistencia**: grado en el que los datos carecen de contradicciones.
- **Interpretabilidad**: grado en el que los datos deben ser interpretados por una persona.
- Precisión: nivel de exactitud del valor.

Niveles de abstracción:
- **Datos**: conjunto de hechos discretos y objetivos sobre un evento.
- **Información**: datos con significado.
- **Conocimiento**: combinación de información contextualizada, experiencias, valores e intuición.

Organización de los datos

Ficheros planos:
- **CSV** *(comma separated values):* RFC 4180. Define un registro por línea y separa los campos por comas.
- **JSON**: RFC 7159 y ECMA-404. Describe objetos encapsulados por llaves y listas por corchetes.
- **XML**: descrito en el estándar XML 1.0 de W3C. Permite almacenar información de forma legible utilizando etiquetas.

Bases de datos:
- Conjunto de datos persistentes, utilizados por sistemas de aplicación.
- En el modelo Entidad-Relación (E/R) una entidad es cualquier objeto repensado en la BBDD y un vínculo representa relaciones entre ellos.
- La unidad básica de almacenamiento es el campo, agrupados en registros y estos en ficheros almacenados.

Casos de estudio

Procesamiento de sitio web sobre cursos *online*.

Procesamiento de *logs* de servidor web.

API de acceso a transacciones bancarias.

Almacenamiento de información sobre productos en un fichero CSV.

Representación de información geolocalizada en formato JSON.

Almacenamiento de información sobre clientes de una base de datos relacional.

Fuentes de datos:
- Captura manual: encuestas y observaciones.
- Análisis de documentos estructurados: estructurados (HTML) y sin formato (lenguaje natural).
- Salida de aplicaciones: *logs* o bases de datos.
- Sensores: dispositivos de medición.
- Datos de acceso público: gubernamentales y servicios web públicos.

Bases de datos relacionales y SQL:
- Los ficheros almacenados se representan en forma de tablas (relaciones), con columnas (campos) y filas (registros).
- El estándar SQL define un lenguaje para la consulta y modificación de los datos.
- El comando SELECT permite consultar información de tablas.
- Los comandos INSERT, UPDATE y DELETE permiten la inserción, edición y eliminación de registros respectivamente.

Ideas clave

1.1. Introducción y objetivos

Este tema realiza una **introducción a la asignatura** y presenta las **definiciones necesarias** sobre las que se basarán los temas posteriores. Para estudiar este tema se deben **leer las Ideas clave** y, si se desea información adicional sobre un concepto específico, se debería consultar las **lecturas sugeridas en la bibliografía** (sección A fondo).

La lectura de los casos de estudio también será de ayuda, ya que proporcionan información de las decisiones tomadas para la captura de datos en un entorno determinado. Este capítulo servirá tanto de introducción como de repaso de los conceptos básicos sobre bases de datos en general y bases de datos relacionales en específico.

En tiempos pasados, la obtención de datos solía ser un proceso costoso y a menudo requería del trabajo humano para la digitalización de documentos físicos. El continuo avance de la tecnología permite que los métodos disponibles para la captura de datos sean cada vez **más diversos**. Esto, en adición a la disminución del **coste por unidad de almacenamiento**, ha resultado en la posibilidad de almacenar grandes cantidades de datos provenientes de distintas fuentes.

A raíz de esta diversidad, es necesario contar con tres elementos que conforman la base para capturar datos de forma eficiente y adecuada al uso que se le brindará a dichos datos:

- El primer elemento es la definición y distinción de los términos *dato, información* y *conocimiento.*

- El segundo, las dimensiones o criterios de calidad que se analizan en un conjunto de datos.
- El último elemento es la experiencia de analizar casos verdaderos en los que ha sido necesario capturar datos y almacenarlos.

El aprendizaje de formatos de ficheros de texto como **CSV** y **JSON** será de utilidad en los próximos temas, ya que son formatos utilizados por el sistema de almacenamiento.

Después de capturar información y poder utilizarla de forma eficiente, será necesario almacenarla de **forma permanente**. Así, la información obtenida no residirá únicamente en la memoria volátil del ordenador, sino que estará disponible para futuras ocasiones.

El método más básico para almacenar datos es mediante el uso del **sistema de ficheros** del sistema operativo. Los ficheros pueden tener un **formato plano**, donde toda la información es legible para una persona; o un **formato binario**, donde la información puede escribirse y leerse de forma directa por una aplicación, pero no puede ser analizada directamente de forma manual.

Este tema proporciona un **repaso** de los mecanismos comúnmente utilizados para el **almacenamiento de información**. Se hablará de dos aproximaciones: la utilización de ficheros planos, que son comúnmente utilizados para el almacenamiento y compartición de datos, y las bases de datos, que dan un paso más allá y proporcionan una **consistencia en la información** y el hecho de poder **consultar y modificar de manera eficiente** un conjunto de datos en específico. Es por esto por lo que se revisará el concepto de base de datos y, en particular, las **bases de datos relacionales**.

Además de ser una de las herramientas más comunes en la actualidad, se puede hacer una analogía entre estas herramientas y el sistema que se utilizarán en otros temas de la asignatura.

1.2. Origen y calidad de los datos

Datos, información y conocimiento

En situaciones informales es común usar indiscriminadamente los términos ***dato, información*** y ***conocimiento***. En ámbitos profesionales y académicos, es conveniente distinguir estos conceptos para evitar malinterpretaciones durante las distintas fases de la analítica de datos.

Existen varias aproximaciones para la distinción de estos términos. En el contexto de esta asignatura se utilizará la definición descrita por Davenport y Prusak (2000).

Un dato puede definirse como un hecho concreto y discreto acerca de un evento. La característica de ser discreto significa que, **semánticamente, es la unidad mínima que puede comunicarse o almacenarse**. Por sí solos, los datos no brindan detalles significantes del entorno del que fueron obtenidos. Ejemplos de datos pueden ser:

- 2010.
- 443.
- DE.

La **información** puede distinguirse simplemente como un **mensaje** formado por la **composición de varios datos**. Esto significa que, a diferencia del dato, la información sí posee un significado para un receptor u observador. Por ejemplo, utilizando los datos anteriores se podría obtener la siguiente información:

- El año de establecimiento de la empresa ACME fue el 2010.
- La altura del edificio Empire State es de 443 metros.
- DE es el código ISO que identifica al idioma alemán.

Los datos deben ser transformados para añadirles valor y convertirlos en información. Estas transformaciones incluyen métodos como:

- **Contextualización**: conocer el propósito del dato obtenido.
- **Categorización**: conocer la unidad de medida y los componentes del dato.
- **Cálculo**: realizar una operación matemática sobre el dato.
- **Corrección**: eliminar errores del dato.
- **Agregación**: resumir o minimizar un dato de forma más concisa.

El conocimiento implica una combinación de experiencias, información contextual y relevancia sobre cierta información. Así como la información se genera a partir de datos, el conocimiento surge de la agregación de información. Ejemplos de métodos que generan esta transformación son:

- **Comparación**: relación entre información obtenida en distintas experiencias.
- **Repercusión**: implicación de la información en decisiones y acciones.
- **Conexión**: relación entre distintos tipos de información.
- **Conversación**: opinión de otras personas sobre la información.

La jerarquía del conocimiento suele representarse gráficamente por una pirámide, siendo los datos la base y el conocimiento la cima.

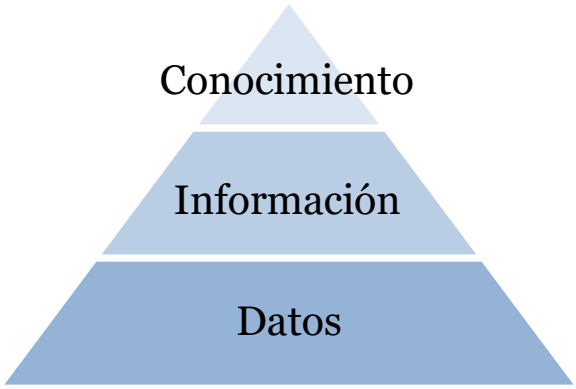


Figura 1. Jerarquía del conocimiento.

Distinguir estos conceptos básicos proporciona un nivel de abstracción útil para la separación de características en el proceso de análisis. El hecho de que un dato sea inválido o erróneo **debe distinguirse fácilmente** de que la **información que se obtiene** de dicho conjunto de datos **sea adecuada o no** al problema que se intenta resolver.

Por ejemplo, es importante conocer si la malinterpretación de un análisis de datos se debe a un error en la fuente de datos, a un problema al combinar los datos en el proceso de análisis o a una confusión por parte del usuario final, debido a experiencias en otros contextos.

Evaluación de la calidad

Las métricas o dimensiones utilizadas para describir la calidad de un conjunto de datos pueden agruparse con base en los actores que interactúan con los datos. Los **diseñadores y administradores** de almacenes de datos tratan con métricas que afectan el diseño o esquema de los datos obtenidos y no con los datos directamente. Entre estas métricas se podría mencionar la *completitud* de los datos obtenidos y el minimalismo del conjunto de datos, el cual se interpreta como la eliminación de redundancias en el diseño del almacén de datos.

Los **desarrolladores de *software*** tratan con métricas específicas de la producción de productos informáticos. Si bien en esta categoría entran métricas que no están relacionadas con los datos directamente, es importante tener en cuenta que estas métricas afectan al proceso de almacenaje, acceso y manipulación de los datos.

El último actor que interactúa con los datos es el **usuario final**, es decir, quien utilizará los datos presentados para crear una conclusión y tomar una decisión acorde. Las métricas relevantes para este actor pueden ser el nivel de disponibilidad de los datos y el nivel de interpretación requerido para entender los datos presentados.

A partir de estas perspectivas, Jarke et al. (1998) presentan un conjunto de dimensiones para evaluar las propiedades de un conjunto de datos:

- La **completitud** o cobertura describe el porcentaje de datos disponibles respecto a la población total que representa dichos datos. Por ejemplo, un conjunto de datos con información de 90 de 100 tratamientos médicos realizados en un hospital presenta una cobertura de 90 %. Esto también se aplica a subconjuntos de los datos obtenidos, por ejemplo, el 5 % de los tratamientos médicos no incluyen la fecha de finalización del tratamiento, lo cual disminuiría la cobertura debido a esta característica.

- La **credibilidad** representa la fiabilidad que se le brinda al organismo que proporciona el conjunto de datos. Esta métrica puede reflejarse en el conjunto de datos por aquellas características que presenten un valor por defecto. Por ejemplo, si se habla de postres, es fácil ver que, si incluye valores por defecto como tarta, helado, fruta, etc., el dato es fiable, pero nunca lo será si se habla de pincho moruno.

- La **precisión** indica el porcentaje de datos correctos respecto al total disponible. También se representa por medio de un porcentaje.

- La **consistencia** describe el nivel con el que los datos son coherentes entre ellos. Un ejemplo de la aplicación de esta métrica se da en datos geográficos: se puede concluir que si una misma entidad tiene asociada una ciudad y un país que no tienen relación, existe un problema de consistencia.

- La **interpretabilidad** define el grado en el que los datos pueden ser entendidos correctamente por una persona. Entre los atributos que definen la interpretabilidad de un conjunto de datos se puede mencionar la documentación de elementos importantes y si el formato en el que se proporcionan los datos es entendible.

A continuación, se proporciona un conjunto de datos de fuente desconocida representadas en la siguiente tabla:

Datos de ejemplo					
Marca	**Modelo**	**Precio**	**Motor**	**Consumo**	**Fecha**
Audi	La Ferrari	15 000 €	963 CV diésel	alto	2014
Lamborghini	Murciélago	300 000 €	580 CV 7500 rpm		2011
BMW	M6	158 000€	560 CV 4935 rpm	13,6/7,6/9,9	2012
Porsche	911 turbo	127 000 €	420 CV 7500 rpm	11 litros	2017

Tabla 1. Datos de ejemplo.

Si analizamos estos datos con respecto a los indicadores de calidad antes mencionados:

- **Completitud**: 75 % respecto a los datos de consumo, o el mismo porcentaje con respecto a las revoluciones por minuto del motor del coche.
- **Credibilidad**: al tratarse de una fuente desconocida, la credibilidad es baja debido a que no se puede comprobar la fuente. Además, por defecto, se sabe que Ferrari solo hace motores de gasolina y que, por defecto, no existen motores diésel de ese caballaje.
- **Precisión:** es claro ver que los datos no son precisos, ya que el precio de un súper deportivo no puede ser de 15 000€ y las revoluciones de algunos coches son demasiado redondas. Además, un consumo «alto» no es una cifra de consumo.
- **Consistencia**: la consistencia de los datos es baja porque a poco que se entienda algo de coches, el modelo de Ferrari nunca puede pertenecer a la marca Audi, sino a Ferrari.
- **Interpretabilidad**: podría mejorarse, puesto que hay muchos datos que no incluyen unidades o que ofrecen un conjunto de números que si no se está familiarizado con el entorno no se conocerá su significado, como ocurre con los 3 valores de consumo.

Fuentes de información

Los métodos para la captura de información se pueden clasificar con base en las características del elemento que genera el conjunto de datos. Las cinco categorías más utilizadas en la actualidad son:

- La **captura manual** de datos es la categoría más tradicional y también una de las más frecuentes en el contexto de investigación en ámbitos sociales y naturales. Entre los métodos que encajan en esta categoría se encuentra el uso de encuestas y las mediciones a través de **observaciones**.

  Si bien esta es la única categoría que no tiene una dependencia directa de las tecnologías de información, para su utilización es necesario que la información se digitalice, ya sea en el momento de la captura o en un procesado posterior.

- El **procesado de documentos estructurados** consiste en la obtención directa de datos disponibles en documentos, cuyo fin inicial no es ser consultados como fuente de datos. Uno de los métodos más comunes en esta categoría es el procesado de páginas HTML en un sitio web, conocido como ***web scraping***.

  Otro ejemplo es el análisis de ***logs*** o ficheros que contienen un listado secuencial de los eventos ocurridos dentro de un sistema y son creados con el objetivo de tener una bitácora y no para ser accedido por otras aplicaciones.

- La **salida de aplicaciones** es una de las categorías más triviales. Los métodos de este tipo involucran el acceso a almacenes de datos tradicionales, tales como **bases de datos relacionales**, ficheros con valores separados por comas (**CSV**), etc.

- Los **datos obtenidos a través de sensores**. En este conjunto se pueden mencionar ejemplos como sensores meteorológicos, sensores de ambiente (ruido o luz), sensores corporales (ritmo cardíaco o conductividad de la piel) y sensores de dispositivos móviles (acelerómetro o giroscopio).

Actualmente existe un gran interés en el uso de sensores para la captura de datos en el contexto personal; este movimiento es conocido como ***quantified self***.

- El **acceso a datos públicos**, ya sea mediante la descarga de conjuntos de datos o a través de interfaces de programación de aplicaciones (**API**, por sus siglas en inglés). Por un lado, muchas de las entidades públicas (como gobiernos centrales y locales) publican catálogos de datos para que sean analizados y se desarrollen nuevas aplicaciones a partir de ellos.

  La publicación de estos datos de carácter público suele realizarse en un portal dedicado, por ejemplo http://datos.gob.es en España, http://data.gov.uk en Reino Unido y http://data.gov en Estados Unidos. Por otro lado, algunas empresas brindan acceso a datos de forma pública, siendo uno de los fines el formar parte de una plataforma de desarrollo, como, por ejemplo, los servicios de redes sociales (Facebook o Twitter) que suelen brindar un API tanto para obtener información sobre un usuario en particular, como para alterar dicha información.

1.3. Organización de los datos

Ficheros planos

Los **ficheros planos** suelen ser un mecanismo utilizado para el intercambio de información entre sistemas. Una de sus ventajas es que es posible ver y editar el contenido del fichero con una herramienta de edición de texto.

Estos ficheros suelen ser mucho más verbosos que los ficheros en formato binario, lo cual implica que su tamaño en el sistema de ficheros será mayor, así como las operaciones necesarias para procesar el contenido desde un programa de *software*. Entre los formatos de fichero plano más comunes se pueden mencionar CSV, JSON y XML. A continuación, se describen los detalles básicos de cada uno de ellos.

El formato **CSV** (*Comma Separated Values* o valores separados por coma), se documenta en la RFC 4180 y presenta las siguientes características:

- Cada registro se delimita por un **cambio de línea** (combinación de dos caracteres: CR y LF).
- Como su nombre indica, los valores de cada registro se separan mediante el **uso de comas**. Es requerido que el número de valores sea constante para todos los registros disponibles en el fichero.
- Los valores pueden estar encapsulados con **comillas dobles**. Esto es obligatorio en aquellos casos donde el valor incluye un cambio de línea, una coma o comillas dobles.
- Si un valor contiene comillas dobles, estas deben escaparse precediéndolas con otro carácter de comillas dobles. Por ejemplo: «Encargado de ""Business Model""»
- Opcionalmente, puede incluir una primera línea con los **nombres de los campos** que se incluyen en el fichero.

A continuación, se muestra un ejemplo de datos en formato CSV. Se puede apreciar que el fichero tiene 4 registros, con 3 valores cada uno. Este ejemplo incluye los títulos de cada valor en la primera fila.

```
Nombre, Edad, Cargo
Juan, 45, Director
Antonio, 35, “Gestor de proyectos”
Pablo, 34, “Analista”
Pedro, 32, “Administrador de bases de datos”
```

El siguiente formato, **JSON** (*JavaScript Object Notation* o notación de objetos en JavaScript) se basa en el lenguaje de programación JavaScript y organiza su notación en dos estructuras:

- Un objeto o registro, definido como un conjunto de pares nombre/valor.
- Un *array* o lista ordenada de valores.

Al ser estructuras de datos muy genéricas, suelen estar soportadas por la mayoría de los lenguajes de programación modernos. La definición de las estructuras sigue las siguientes condiciones:

- Un objeto se delimita por llaves ({ }) y los pares nombre/valor se separan por medio de comas, y entre el nombre y el valor se coloca el carácter dos puntos (*:*).
- Un *array* se delimita por corchetes ([ ]) y los valores se separan por comas.
- Cada valor, tanto dentro de un objeto como de un *array,* puede ser una cadena de texto delimitada por comillas dobles, un número, un valor booleano *(true/false),* el valor nulo (*null*), un objeto o un *array.*

La representación de los primeros dos registros del ejemplo en CSV sería la siguiente en formato JSON:

```
[{
    "Nombre": "Juan",
    "Edad": 45,
    "Cargo": "Director"
}, {
    "Nombre": "Antonio",
    "Edad": 35,
    "Cargo": "Gestor de proyectos"
}]
```

Puede observarse que el formato JSON es más verboso, lo cual permite que los objetos en un documento contengan información heterogénea, al contrario que en un documento CSV.

Finalmente, el formato **XML** (*eXtended Markup Language* o lenguaje de marcas extensible) es el más verboso de los tres que se representan en este tema. Como su nombre lo indica, utiliza marcas o **etiquetas** como parte de la estructura y formato de los datos que contiene.

Entre las condiciones que debe cumplir un documento que siga el formato XML se encuentran las siguientes:

- El documento inicia con la línea: `<?xml version=”1.0”>`.
- Un documento XML tiene solamente un elemento raíz.
- Un elemento en XML se abre mediante una etiqueta delimitada por los signos menor que y mayor que (`<etiqueta>`) y se cierra con la misma etiqueta, pero incluyendo una barra (/) inmediatamente después del menor que (`</etiqueta>`).
- Los elementos pueden tener atributos. Estos se incluyen como pares nombre/valor, separados por el carácter de igualdad (=) dentro de la etiqueta del elemento. El valor debe delimitarse por comillas simples o dobles.
- El contenido de un elemento puede ser texto, uno o varios elementos, o la combinación de ambos.

Siguiendo con el mismo ejemplo que en los formatos anteriores, la información de los dos primeros registros podría representarse con el siguiente contenido:

```
<empleados>
  <empleado>
    <Nombre>Juan</Nombre>
    <Edad>45</Edad>
    <Cargo>Director</Cargo>
  </empleado>
  <empleado>
    <Nombre>Antonio</Nombre>
    <Edad>35</Edad>
    <Cargo>Gestor de proyectos</Cargo>
  </empleado>
<empleados>
```

Aunque las condiciones para crear los ficheros son claras, estas no siempre garantizan que el fichero XML quede bien formado. Un fichero XML estará bien formado cuando cumpla todas las características de formato, las cuales podrán analizarse mediante alguna herramienta de análisis sintáctico, que incluya la norma correspondiente.

Validación de ficheros XML

Recuerda que todo fichero XML cumple con una jerarquía de etiquetas que garantiza su estructura. Para ello, las etiquetas estarán contenidas adecuadamente unas dentro de otras, con su cierre correspondiente. Recuerda también que el fichero XML debe incluir un único elemento raíz que contendrá las otras etiquetas que existan. El valor de los atributos, por su parte, deben estar entre comillas simples o dobles; y en lo que respecta a las etiquetas, estas son sensibles a mayúsculas y minúsculas.

Cuando un XML cumple con estas consideraciones, se dice que este está «bien formado». Para que se pueda asegurar que el documento XML es «válido», será necesario validarlo mediante algún otro documento que garantice dicha validez.

Validación de ficheros XML mediante DTD

Uno de esos documentos se llama **DTD** (*Document Type Definition* o definición de tipo de documento). Este documento recoge las reglas que debe cumplir el contenido del XML (estructura, elementos y atributos que sí puede incluir el fichero), de ahí que sea útil para garantizar la validez de **todo fichero XML** que haga referencia a él.

Veamos un ejemplo práctico con el siguiente fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mail SYSTEM "ruler.dtd">
<mail>
<to>Pepe</to>
<from>Boss</from>
<subject>Empty Stock</subject>
<body>Remember to buy supplies!</body>
</mail>
```

Observa cómo en la sentencia `DOCTYPE` se hace referencia al fichero DTD llamado **ruler.dtd**. Este fichero es el encargado de definir la estructura correcta para el documento llamado «mail» y para las etiquetas que este contiene.

Ejemplo del fichero **ruler.dtd**:

```
<!DOCTYPE mail [
<!ELEMENT mail (to,from,subject,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

La forma de interpretar el DTD es el siguiente:

- `!DOCTYPE mail`: establece que el elemento raíz es `mail`.
- `!ELEMENT mail`: indica los elementos que deben estar contenidos dentro del raíz, esto son `to`, `from`, `subject` y `body`.
- `!ELEMENT to`: indica el tipo del elemento `to`, en este caso es `#PCDATA`.
- `!ELEMENT from`: establece que el elemento `from` es de tipo `#PCDATA`.
- `!ELEMENT subject`: establece que el elemento `subject` es de tipo `#PCDATA`.
- `!ELEMENT body`: establece que el elemento `body` es de tipo `#PCDATA`.

Para la DTD, `#PCDATA` significa de tipo texto.

Aunque los DTD son documentos que se usan cada vez menos, es importante que sepas que estos existen y cuál es su papel de cara a formar y validar ficheros XML.

Validación de ficheros XML mediante esquemas (SCHEMA)

Otra alternativa que existe para validar la estructura y el contenido de los ficheros XML es mediante esquemas. Un esquema, al igual que un DTD, indica la estructura

que debe llevar un documento XML. Un fichero esquema lleva la extensión XSD y está definido como un fichero XML, a diferencia de la DTD que utiliza su propio metalenguaje.

Los XSD ofrecen muchas más opciones para declarar la estructura de los ficheros XML. De dichas opciones se puede destacar la extensa lista de tipos de datos predefinidos para elementos y atributos, los cuales a su vez pueden ampliarse o restringirse para crear nuevos tipos.

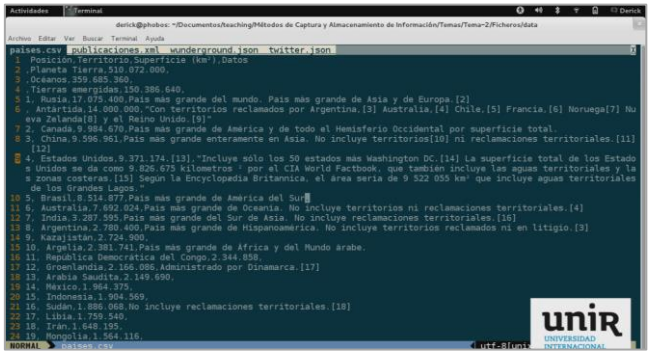
Asimismo, permiten indicar con gran precisión la cardinalidad de un elemento (las veces que dicho elemento puede aparecer en un documento XML). Otra ventaja que tienen es que, gracias a los espacios de nombres, soportan la mezcla de distintos conjuntos de etiquetas (vocabularios).

Siguiendo con el ejemplo anterior, un SCHEMA ideal para nuestro fichero XML de ejemplo podría ser el siguiente:

Ejemplo de fichero **ruler.xsd**:

```
<xs:element name="mail">
<xs:complexType>
  <xs:sequence>
    <xs:element name="to" type="xs:string"/>
    <xs:element name="from" type="xs:string"/>
    <xs:element name="subject" type="xs:string"/>
    <xs:element name="body" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

Aprovecha este punto para repasar con la clase «Revisión de formatos CSV, XML y JSON». En ella se presenta un *screencast* con la revisión en detalle de ficheros con los formatos vistos en el tema.



Vídeo 1. Revisión de formatos CSV, XML y JSON.

Accede al vídeo a través del aula virtual

Bases de datos

Una **base de datos** es un conjunto de datos persistente utilizado por un sistema de *software*. Siguiendo con las definiciones, y tal y como se menciona en la bibliografía, un sistema de base de datos es un sistema computarizado para el **almacenamiento de registros**. Se pueden mencionar cuatro componentes de un sistema de esta categoría:

- **Datos**. Los datos en un sistema de base de datos pueden definirse como **integrados**, en aquellos casos en que todos los datos se mantienen unificados y comúnmente serán accedidos por una sola persona, así como **compartidos**, para aquellos casos en los que se desea mantener los conjuntos de datos separados y otorgar privilegios de acceso distintos a varias personas.

- ***Hardware***. Como en otros métodos de almacenamiento, los componentes de *hardware* que intervienen en un sistema de base de datos son los **volúmenes de almacenamiento**, así como los **procesadores y memoria principal**.

- ***Software.*** La capa de *software* entre el usuario y la base de datos física se conoce como DBMS (*Database Management System* o sistema gestor de la base de datos).

- **Usuarios**. Existen tres clases de usuarios en un sistema de bases de datos:
  - **Programadores**: encargados de crear aplicaciones que permitan la interacción con la base de datos.
  - **Usuarios finales**: utilizan las distintas aplicaciones y herramientas para interactuar con la base de datos.
  - **Administrador de base de datos**: también conocido como DBA por sus siglas en inglés, se encarga de gestionar la estructura, disponibilidad y eficiencia del sistema de base de datos.

En el contexto de bases de datos se utiliza el término *entidad* para describir a cualquier objeto que puede almacenarse en el sistema. Por ejemplo, en una base de datos utilizada por un almacén se puede tener una entidad *producto* para describir los productos disponibles en el almacén y el término *bodega* para describir las bodegas con las que cuenta.

Además, se utilizan los términos *vínculo* o *relación* para representar las relaciones entre las entidades. En el ejemplo del almacén, puede haber una relación bodega-producto para indicar que un producto se almacena en una bodega específica.

Los datos almacenados en una base de datos se pueden categorizar de forma jerárquica. La unidad más pequeña es el campo, el cual suele tener un tipo (número, fecha, etc.) y la base de datos tendrá muchas ocurrencias.

Al conjunto de datos que tienen relación entre sí se le denomina **registro**. Por ejemplo, un registro de tipo producto puede tener campos como «nombre», «precio» y «descripción». Finalmente, al conjunto de registros del mismo tipo se le denomina archivo almacenado.

Bases de datos relacionales y SQL

En un sistema de base de datos **relacional**, los archivos de datos son representados por **tablas**. Cada **columna** de la tabla representa un campo del archivo, mientras que cada **fila** representa un registro de datos. Además, cuando un usuario realiza una operación sobre una tabla, el resultado de dicha operación también será una tabla.

Para ejemplificar estos sistemas se ha utilizado el ejemplo de los productos en un almacén. En este caso, la información de los productos puede almacenarse en una tabla productos cuyo contenido podría ser el siguiente:

Identificador	Nombre	Precio	Bodega
1	Mesa	150,00	1
2	Silla	50,00	1

Para interactuar con una base de datos de manera programática, se utiliza el estándar **SQL**. Los comandos para la manipulación de datos se pueden resumir en cuatro:

- **SELECT**: utilizado para obtener un conjunto de datos a partir de una o varias tablas en un DBMS relacional.
- **INSERT**: comando para agregar un conjunto de registros dentro de una tabla.
- **UPDATE**: permite la modificación de un conjunto de campos sobre un conjunto de registros en una tabla específica.
- **DELETE**: como su nombre indica, permite eliminar un conjunto de registros de una tabla.

En general, las funciones básicas que se pueden llevar a cabo sobre una base de datos se conocen con el acrónimo **CRUD**, que significa ***C**reate, **R**ead, **U**pdate and **D**elete.*

Después de leer gran parte del tema, el reto siguiente es que repases los aspectos más relevantes de los métodos de captura de información con la clase «Resumen de los métodos de captura de información». También será útil recordar las principales características de las bases de datos relacionales y NoSQL.

Vídeo 2. Resumen de los métodos de captura de información.

Accede al vídeo a través del aula virtual

1.4. Casos de estudio

Análisis de *logs* de un servidor web

En este caso de uso se cuenta con un sitio web en el que un equipo de trabajo puede descargar varios documentos. Los documentos se descargan directamente, los usuarios se autentican en el servidor y no existe ninguna aplicación web intermediaria que pueda mantener un registro de los documentos descargados por cada persona.

Se plantea el problema de generar un informe sobre a qué documentos se accede con mayor frecuencia y qué usuarios han tenido mayor actividad con estos documentos. Al no contar con ningún almacenamiento de datos con esta

información, se plantea la solución de capturar los datos necesarios mediante el **procesamiento de *logs*** en el servidor web.

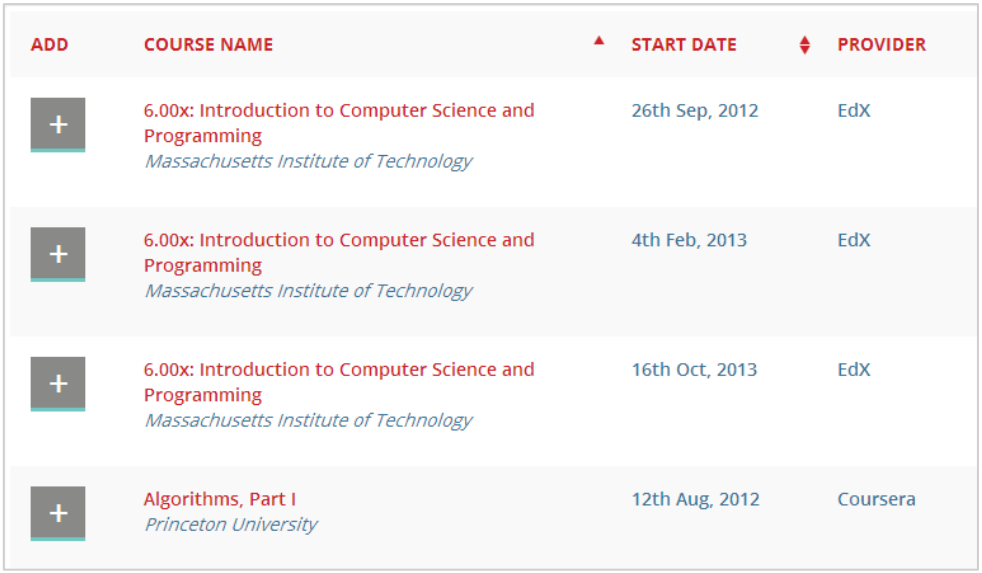
Un ejemplo del contenido de los *logs* en un servidor web, en este caso el Servidor HTTP Apache, puede ser el siguiente:

```
192.168.1.101 - juan [21/Nov/2013:12:08:03 +0100] "GET /resource/152 HTTP/1.1" 200 1368023 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"

192.168.1.101 - juan [21/Nov/2013:12:10:22 +0100] "GET /resource/601 HTTP/1.1" 200 92349 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"

192.168.1.101 - juan [21/Nov/2013:16:36:19 +0100] "GET /resource/34 HTTP/1.1" 200 5928327 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"
```

Puede observarse que el fichero almacena varios datos sobre la conexión HTTP realizada hacia el servidor: dirección IP, usuario de autenticación, fecha y hora, recurso descargado y el agente o navegador utilizado para la conexión. Todos estos datos pueden capturarse mediante el procesado del fichero y ser almacenados en una base de datos relacional para realizar el análisis correspondiente y generar el informe solicitado.

Web scraping de listado de cursos *online*

El requisito presentado en este caso es capturar la información relacionada a un conjunto de cursos masivos en línea (MOOCs) disponible en el sitio web Class Central. La información de estos cursos se presenta en una serie de tablas.



ADD	COURSE NAME	START DATE	PROVIDER
+	6.00x: Introduction to Computer Science and Programming *Massachusetts Institute of Technology*	26th Sep, 2012	EdX
+	6.00x: Introduction to Computer Science and Programming *Massachusetts Institute of Technology*	4th Feb, 2013	EdX
+	6.00x: Introduction to Computer Science and Programming *Massachusetts Institute of Technology*	16th Oct, 2013	EdX
+	Algorithms, Part I *Princeton University*	12th Aug, 2012	Coursera

Figura 2. Listado de cursos MOOCs WebScraping.

Al no contar con un API que brinde acceso a los datos, la solución ha sido utilizar la herramienta de *web scraping* **Scraper**. Esta herramienta es una extensión del navegador Google Chrome y permite capturar los datos presentados de forma estructurada en formato HTML. Después de capturados, los datos se pueden exportar a una hoja de cálculo del servicio Google Docs.

Acceso a transacciones bancarias mediante API

La tarea solicitada en este caso es crear un conjunto de visualizaciones sobre los patrones de compras de un conjunto de personas. Para esto, se dispone de acceso a información agregada sobre las compras realizadas con tarjeta de débito o crédito en una ciudad.

Con el fin de evitar dar información personal de los clientes del banco, los datos publicados se agrupan por código postal, tipo de establecimiento (alimentación, supermercados, ocio, etc.), día de la semana y hora del día. Los datos disponibles son:

- Número total de compras.
- Suma total de las transacciones realizadas dentro de un código postal y por tipo de establecimiento.

- Diez códigos postales asociados a la tarjeta utilizada y que presentan la mayor frecuencia de aparición.

Los datos se publican a través de un API con servicios web, por lo que el método utilizado se cae en la categoría **acceso a datos públicos**. El primer paso necesario para capturar estos datos ha sido crear una cuenta en el portal de acceso que contiene el catálogo de datos. A continuación, se ha desarrollado una serie de *scripts* para la captura de datos, siguiendo la documentación proporcionada por el mismo portal.

Productos en formato CSV

En este caso, se cuenta con información de un inventario de productos. Los datos capturados de cada producto son el **identificador** (de tipo numérico), el **nombre** (de tipo cadena de texto) y la **cantidad** (de tipo numérico).

Se plantea el problema de definir el formato de un fichero CSV que almacene los datos del inventario. Una de las condiciones presentadas es incluir el nombre de los campos para que no se genere ninguna confusión al procesar el fichero.

Un ejemplo del formato y el contenido del fichero CSV, siguiendo los requisitos planteados, es el siguiente:

```
identificador, nombre, cantidad
1, Plato, 150
2, Sartén, 100
3, Jarra, 200
4, Vaso, 150
```

Como puede observarse, ninguno de los valores incluidos en el campo «nombre» incluye caracteres especiales tales como las comillas dobles, el cambio de línea o la coma. Por esta razón estos valores no están delimitados por comillas dobles.

Información geolocalizada en formato JSON

En esta situación se cuenta con información de la actividad geolocalizada de una persona. Esta característica de geolocalización implica que entre los datos disponibles se encuentra la ubicación del usuario en términos de longitud y latitud, los cuales se representan por un número decimal. Además de la ubicación, se ha capturado un identificador de usuario y el momento de la captura, en formato de fecha y hora.

El requisito en este caso es especificar el formato de un fichero JSON que incluya esta información. Una posible solución es la siguiente:

```
[{
    “usuario”: 1,
    “ubicacion”: {
        “longitud”: 40,
        “latitud”: -3
    },
    “timestamp”: “2012-12-01 13:00”
}, {
    “usuario”: 2,
    “ubicacion”: {
        “longitud”: 40.1,
        “latitud”: -3.2
    },
    “timestamp”: “2012-12-03 10:30”
}]
```

Puede observarse que la ubicación del usuario se ha definido como un objeto perteneciente al de la actividad del usuario. Esta estructura toma ventaja de la recurrencia brindada por el formato JSON.

Información de clientes en una base de datos relacional

Este último caso contempla el almacenamiento de información sobre clientes en una base de datos relacional. La información para almacenar consiste en un identificador de cliente, el primer nombre, los apellidos y la fecha en la que se suscribieron al servicio.

Al tratarse de una base de datos relacional, la información capturada se representará en la forma de tabla. Así, un ejemplo de esta tabla se puede visualizar en la siguiente estructura.

Ejemplo de datos en tabla			
Identificador	**Nombre**	**Apellidos**	**Fecha de registro**
1	David	Gómez Pérez	2011-01-20
2	Francisco	León García	2011-02-01

Tabla 2. Información de clientes, datos de ejemplo.

En la lección «Captura de información a través de Web Scraping» se presenta un *screencast* de la captura de datos desde una página web utilizando la herramienta Scraper.

Vídeo 3. Captura de información a través de Web Scraping.

Accede al vídeo a través del aula virtual

En este último vídeo podrás hacer un repaso general de lo visto en este tema, especialmente lo que concierne al origen, la calidad y la organización de los datos.

Vídeo 4. Repaso general del primer tema de la asignatura.

Accede al vídeo a través del aula virtual

1.5. Referencias bibliográficas

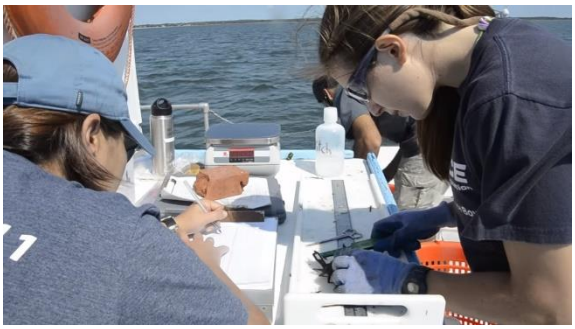
Davenport, T., y Prusak, L. (2000). *Working knowledge: how organizations manage what they know.* Harvard Business Review Press.

Jarke, M., Jeusfeld, M. A., Quix, C., y Vassiliadis, P. (1998). Architecture and quality of data warehouses: an extended repository approach. *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, 1413, 243-260.

A fondo

How do scientists collect data?

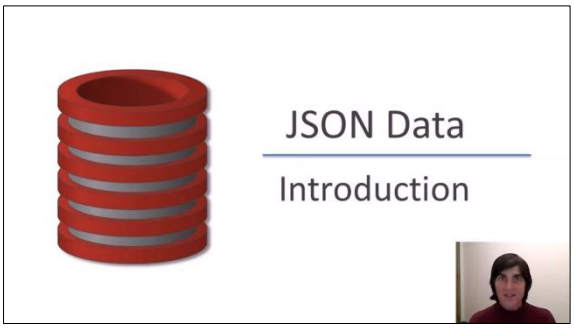
Genoino, C. (5 de diciembre de 2012). *How do scientists collect data?* [Archivo de vídeo]. http://www.youtube.com/watch?v=mPmiW0x3s3k



Este breve vídeo describe los datos obtenidos por un grupo de científicos especializados en biología marina. Es interesante observar la cantidad de datos contextuales que serán utilizados en el análisis.

Introduction to JSON data

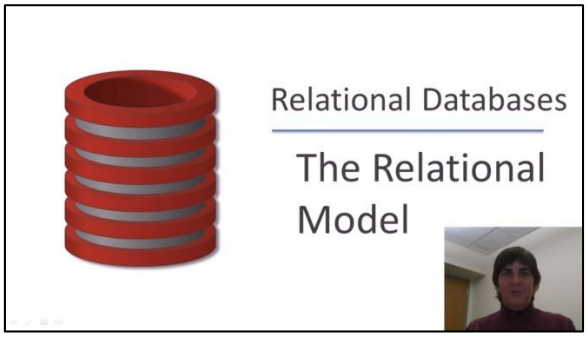
Stanford Dbclass. (26 de diciembre de 2013). *04 01 json intro part1* [Archivo de vídeo]. https://www.youtube.com/watch?v=lutlvfe8YHU



Este vídeo explica detalladamente los aspectos básicos del formato JSON. El vídeo forma parte del curso abierto masivo «Introducción a Base de Datos», ofrecido por la Universidad de Stanford.

The relational model

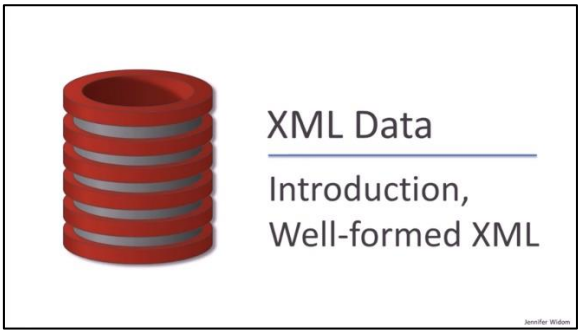
Stanford Dbclass. (4 de enero de 2013). *02-01-relational-model.mp4* [Archivo de vídeo]. https://www.youtube.com/watch?v=spQ7IFksP9g



Otro vídeo del curso «Introducción a Base de Datos» de la profesora Jennifer Widom. En este caso se presenta una explicación muy detallada del modelo de base de datos relacional.

Well-formed XML

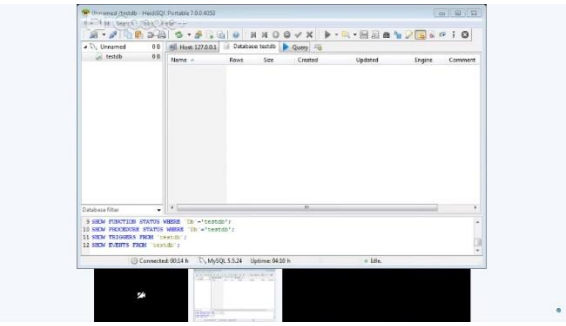
Stanford Dbclass. (4 de enero de 2013). *03-01-well-formed-xml.mp4* [Archivo de vídeo]. https://www.youtube.com/watch?v=x8kMELlNaYg



Un último vídeo del curso «Introducción a Bases de Datos» con relación al contenido de este tema. Este vídeo brinda una introducción muy ilustrativa al estándar XML, además de analizar las características de un documento con sintaxis correcta.

Introducción a SQL

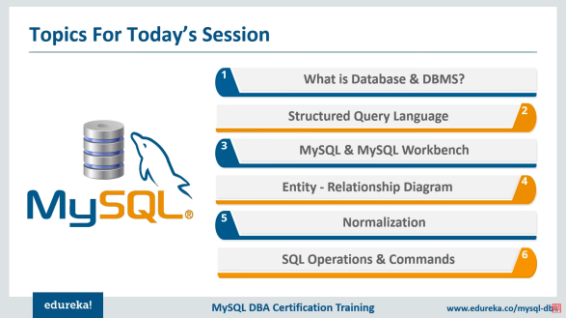
Exitae Informática. (25 de abril de 2013). *Introducción a SQL* [Archivo de vídeo]. http://www.youtube.com/watch?v=fDNgSTF1dVs



Este vídeo presenta de forma muy guiada el uso de SQL a través de una consola. Se muestran ejemplos tanto de comandos para la creación y modificación de tablas, así como consultas, actualización y eliminación de datos.

Introducción a MySQL

edureka! (23 de octubre de 2018). *MySQL Tutorial For Beginners | Introduction to MySQL | Learn MySQL | MySQL Training | Edureka* [Archivo de vídeo]. https://www.youtube.com/watch?v=WmGgxTpGs_8



Este vídeo hace un recorrido por las principales características de uno de los motores de bases de datos SQL gratuitos, muy usado tanto en la industria como en la investigación.

Aunque no es una base de datos que veas en la asignatura, es recomendable que hagas el ejercicio de instalarla e intentar usarla con alguna aplicación para que asimiles su uso de cara a futuros retos profesionales.

7 command-line tools for data science

Janssens, J. (19 de septiembre de 2013). 7 Command-Line Tools for Data Science. *Data Science Workshops* [Página web]. http://jeroenjanssens.com/2013/09/19/seven-command-line-tools-for-data-science.html

Post en el blog de Jeroen Janssens con una recopilación de herramientas para el tratamiento de ficheros JSON y CSV desde la línea de comando de Unix. Aun si Unix no es la plataforma que utilizas regularmente, es interesante analizar el uso de estos ficheros en un entorno real de analítica de datos (nunca pasan de moda).

SQL for Web Nerds

Greenspun, P. (s.f.). SQL for Web Nerdsweb [Página web]. http://philip.greenspun.com/sql/

Completo tutorial sobre bases de datos relacionales y SQL. El autor, Philip Greenspun, profesor del Instituto Tecnológico de Massachusetts, describe de forma muy detallada las razones por las que los desarrolladores de aplicaciones web tienen la necesidad de utilizar bases de datos relacionales.

Dataportals.org

Open Knowledge Foundation. (s.f.). Data Portals [Página web]. http://dataportals.org/

Listado de catálogos de datos abiertos. Los catálogos están organizados por nivel (local, estatal, nacional, etc.) y por grupos.

Research-Quality Data Sets

Kaggle. (2019). Página web. https://www.kaggle.com/

Listado de conjuntos de datos abiertos y con la calidad requerida como para utilizarlos con fines de investigación.

JSON.org

JSON. (s.f.). Introducción a JSON [Página web]. http://json.org/json-es.html

Página oficial de la especificación del formato JSON.

Bibliografía

Chakrabarti, S. (2003). *Mining the web: discovering knowledge from hypertext data* (pp. 17-43). Morgan Kaufmann.

Debenham, J. (1998). *Knowledge engineering. Unifying knowledge base and database design* (pp. 15-22). Springer.

Matallah, H., Belalem, G., & Bouamrane, K. (2021). Comparative study between the MySQL relational database and the MongoDB NoSQL database. International *Journal of Software Science and Computational Intelligence (IJSSCI),* 13(3), 38-63.

Meier, A., & Kaufmann, M. (2019). *SQL & NoSQL databases.* Springer Fachmedien Wiesbaden.

de Oliveira, V. F., Pessoa, M. A. D. O., Junqueira, F., & Miyagi, P. E. (2021). SQL and NoSQL Databases in the Context of Industry 4.0. *Machines,* 10(1), 20.

Redman, T. C. (1996). *Data quality for the information age* (pp. 245-267). Artech House.

Shafranovich, Y. (2005). Common format and MIME type for Comma-Separated Values (CSV) files. *Internet Engineering Task Force IETF RFC 4180.*

Sumalatha, A., Vookanti, R., & Vannala, S. (2021). Study on Applications of SQL and Not only SQL Databases used for Big Data Analytics. *International Journal For Research & Development In Technology,* 15, 127-130.

Wanumen, L. F. (2018). *Bases de datos en SQL server.* Ecoe Ediciones. Disponible en la Biblioteca Virtual de UNIR.

Test

1. ¿Cuál es la unidad semántica mínima que puede almacenarse o comunicarse?

A. Dato.

B. Información.

C. Conocimiento.

D. Las respuestas A y B son correctas.

El dato es la mínima expresión semántica que puede almacenarse.

2. ¿Qué métodos pueden utilizarse para la transformación de información a conocimiento?

A. Contextualización, agregación y cálculo.

B. Repercusión, conexión y conversación.

C. Categorización, corrección y agregación.

D. Análisis, investigación y discusión.

Son los métodos que, entre otros, permiten la transformación adecuada de la información en conocimiento.

3. ¿Qué métrica de calidad describe la proporción en la que un conjunto de datos contiene a la población que representa?

A. Precisión.

B. Consistencia.

C. Completitud.

D. Interpretabilidad.

Solo puede ser la completitud la métrica que indique lo representativo que son los datos con respecto a una población.

4. ¿Cuál de los siguientes es un ejemplo de método de captura manual?

A. *Web scraping.*

B. Encuestas.

C. Acceso a bases de datos relacionales.

D. Lectura de termómetro digital.

Excepto las encuestas, los otros métodos pueden ser automatizados; la encuesta, por su parte, no.

5. ¿En qué categoría de captura de datos entra la lectura de información del acelerómetro y giroscopio de un teléfono móvil?

A. Captura manual.

B. Procesamiento de documentos.

C. Acceso a datos públicos.

D. Sensores.

El acelerómetro y el giroscopio son los sensores que vienen integrados en el teléfono para capturar el movimiento del teléfono y su posición o inclinación, por ende, los sensores determinan la captura de este tipo de datos.

6. ¿Qué elemento es utilizado para delimitar valores en un fichero CSV?

A. Coma.

B. CRLF.

C. Comillas dobles.

D. Espacio.

Dentro de los limitares posibles de CSV, la coma es uno de ellos.

7. ¿Sobre qué estructuras se basa el formato JSON?

A. Objetos y diccionarios.

B. Tablas *hash.*

C. Objetos y *arrays.*

D. Listas enlazadas.

Los objetos y los *arrays* son los elementos que conforman la estructura de JSON.

8. ¿Cuál de las siguientes condiciones sobre XML es verdadera?

A. Un documento solo puede tener un elemento raíz.

B. El contenido de un elemento debe ser otro elemento.

C. Todo elemento debe tener un atributo llamado «id».

D. Los atributos deben de ser de tipo numérico.

Es la única afirmación que es cierta, un documento XML solo puede tener un único elemento raíz.

9. ¿Qué nombre recibe un conjunto de datos persistente utilizado por un sistema de *software*?

A. Archivo.

B. Base de datos.

C. Registro.

D. Las respuestas A y B son correctas.

La persistencia se puede dar o bien en una base de datos o bien en un archivo.

10. ¿Cuál es la instrucción de SQL para consultar información?

A. SELECT.

B. INSERT.

C. UPDATE.

D. DELETE.

La instrucción SELECT permite consultar información en cualquier base de datos.