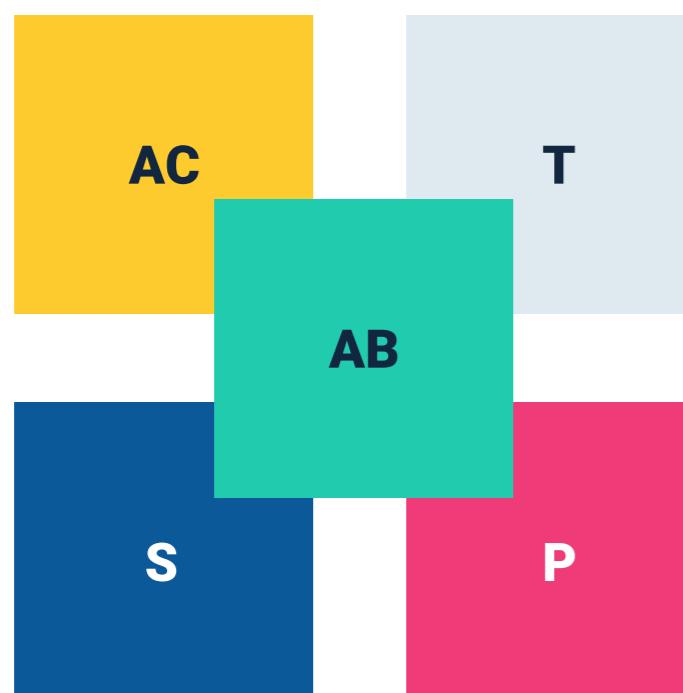


Diseño y modelo de datos.

Este componente formativo está diseñado para comprender los principios básicos de las bases de datos NoSQL, las bases relacionales, y las diferentes fuentes de datos y controles de acceso.

Iniciar >

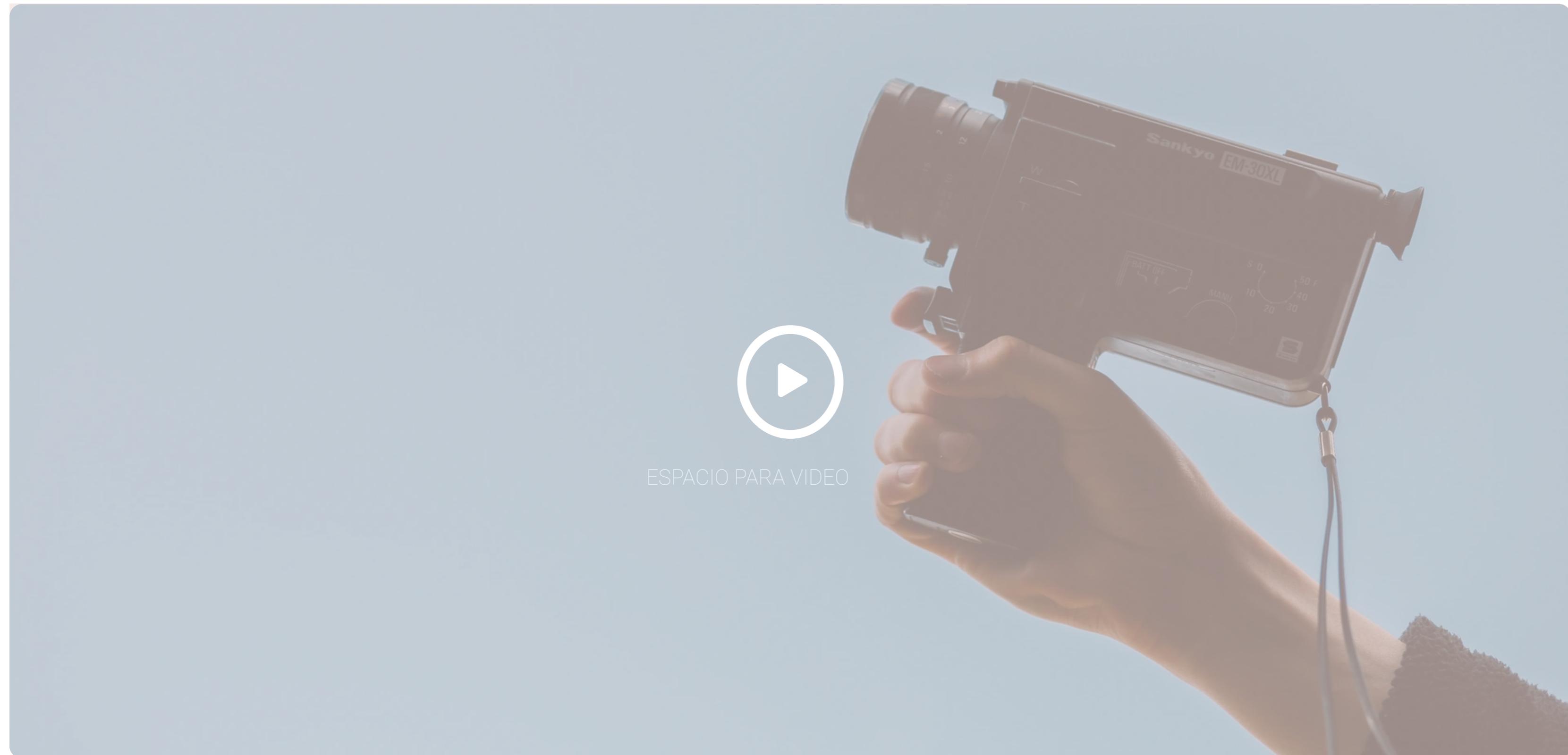


PRIMARIO #138AF8	ACENTO CONTENIDO #5ED1A9	CB
SECUNDARIO #1B3F5E	ACENTO BOTONES #FFD947	C
NEUTRAL 1 #EFEFEF	NEUTRAL 2 #F9F7EC	B
		A

i Introducción

Apreciado aprendiz, bienvenido a este componente formativo, donde encontrará diferentes temáticas orientadas al conocimiento de los sistemas de almacenamiento, desarrollando todas sus habilidades y capacidades en bases de datos.

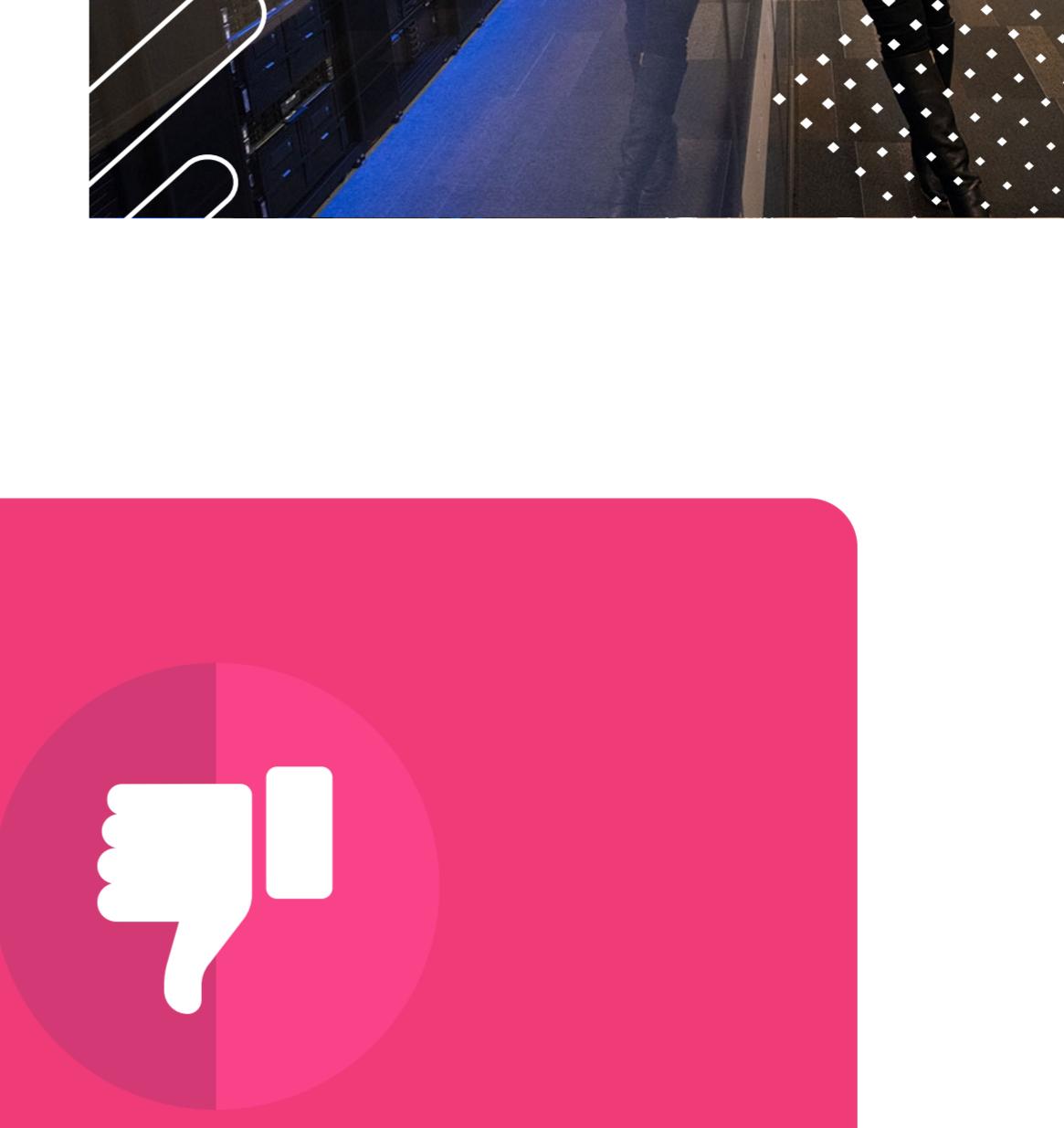
En el siguiente video conocerá, de forma general, la temática que se estudiará a lo largo del componente formativo.



1 Bases de Datos NoSQL

Las bases de datos NoSQL nacen como una alternativa de almacenamiento de información con una estructura distinta a la convencional, comúnmente conocida como 'base de datos relacional', y poseen un importante segmento del mercado, abarcando gran parte de los procesos de negocio que manejan las empresas hoy día.

Los procesos, por lo general, tienden a evolucionar de acuerdo a como se desarrollan los procesos de negocio dentro de las organizaciones, esto quiere decir que las bases de datos NoSQL, de alguna manera, se relacionan en gran medida con la evolución de los procesos de almacenamiento de información, lo cual implica, a su vez, que en muchos de los procesos que hoy se manejan, las bases de datos NoSQL se implementan no solo por compañías o empresas específicas, también por diferentes actores del mundo de la computación, incluyendo empresas, particulares o comunidad en general, todos esperando generar soluciones de almacenamiento a sus proyectos.



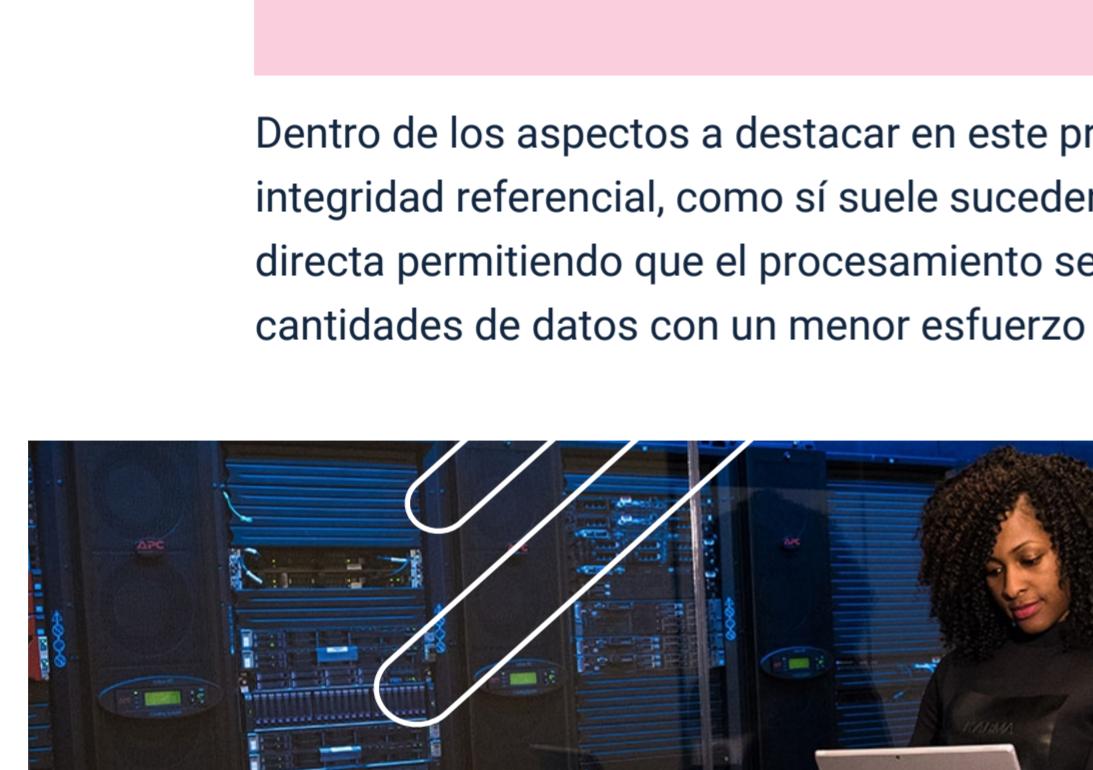
Las bases de datos NoSQL tienen muchas ventajas, aunque también algunas desventajas, por ejemplo:

Ventaja	Desventaja
<p>Su rendimiento, en comparación con las bases de datos tradicionales, es superior y utiliza menos recursos de cómputo, evita requerir una gran infraestructura de cómputo para llevar a cabo su instalación o el procesamiento de información.</p>	<p>Debido a que estas bases de datos son utilizadas para el manejo de grandes volúmenes de información, en algunos casos podría haber un margen de error al realizar el procesamiento de la información, diferenciándose de las bases de datos convencionales que cuentan con una mayor estabilidad y fiabilidad al momento de procesar información.</p>

Para ampliar la información, se le invita a ver el siguiente video.



Modelo de datos



El modelo de datos está determinado por la forma en la cual se organizan los datos y los campos de la base misma. Es importante tener en cuenta que las bases de datos NoSQL utilizan un nivel de archivo para organizar la información que contienen, es decir, se utilizan como si fuesen archivos planos que contienen una estructura sencilla pero organizada, en la que se ubica toda la información que debe contener y generar al momento de realizar el registro de los datos.

Al referirse a los tan populares archivos JSON o XML, los cuales contienen una estructura muy particular para la organización de la información implementada en las bases NoSQL, se evidencia la ventaja de almacenar información organizada y con poco espacio de almacenamiento. A continuación, se explica el almacenamiento de este tipo de modelo de datos.

Figura 1. Modelo de archivo JSON

Estructura del archivo JSON	Información de usuario	Información de contacto																					
<pre>{ "Codigo": 1, "Nombre": "Dulfran", "Apellido": "Montaño", "Ciudad O": "Barranquilla", "Profesion": "Ingeniero de sistemas", "Contacto": "3126954069" }</pre>	<table border="1"> <thead> <tr> <th>KEY</th> <th>NOMBRE</th> <th>APELLIDO</th> <th>PROFESIÓN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DULFRAN</td> <td>MONTAÑO</td> <td>INGENIERO DE SISTEMAS</td> </tr> <tr> <td>2</td> <td>JOSE</td> <td>MARTINEZ</td> <td>ABOGADO</td> </tr> </tbody> </table>	KEY	NOMBRE	APELLIDO	PROFESIÓN	1	DULFRAN	MONTAÑO	INGENIERO DE SISTEMAS	2	JOSE	MARTINEZ	ABOGADO	<table border="1"> <thead> <tr> <th>KEY_ID</th> <th>CIUDAD</th> <th>CONTACTO</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>BARRANQUILLA</td> <td>3126954069</td> </tr> <tr> <td>2</td> <td>MEDELLIN</td> <td>3004640398</td> </tr> </tbody> </table>	KEY_ID	CIUDAD	CONTACTO	1	BARRANQUILLA	3126954069	2	MEDELLIN	3004640398
KEY	NOMBRE	APELLIDO	PROFESIÓN																				
1	DULFRAN	MONTAÑO	INGENIERO DE SISTEMAS																				
2	JOSE	MARTINEZ	ABOGADO																				
KEY_ID	CIUDAD	CONTACTO																					
1	BARRANQUILLA	3126954069																					
2	MEDELLIN	3004640398																					



Como se observa en la imagen, el modelo de datos de una base de datos NoSQL se refiere a la equivalencia o relación que puedan estar compuestas por una o más relaciones de datos, estos archivos se encuentran estrechamente relacionados con la información complementaria de cada uno de ellos, es decir, la información se va acumulando hasta que forma lo que conocemos como base de datos NoSQL.

Dentro de los aspectos a destacar en este proceso, es que la información no se encuentra estrechamente relacionada o restringida por algún tipo de integridad referencial; como si suele suceder con las bases de datos convencionales, y al no existir dicha restricción, los datos fluyen de manera directa permitiendo que el procesamiento se haga al nivel de archivos planos, mejorando los tiempos de respuesta y, a su vez, procesando mayores cantidades de datos con un menor esfuerzo de cómputo.



Los modelos de datos son sumamente necesarios en los sistemas de información, permiten visualizar la manera en la que la información y los datos serán almacenados y procesados de acuerdo a las solicitudes del sistema de información. Es importante tener en cuenta que un sistema de información requiere de un sistema de almacenamiento eficiente y estable, no solo para almacenar la información, también, proporciona las bases necesarias que permite a los usuarios recibir los datos de manera íntegra, segura y en tiempos razonables; claro está, de acuerdo con la capacidad de respuesta de los sistemas implementados.

Para conocer el proceso de codificación de una base de datos en MongoDB, se invita a ver el video [Mongodb para Visual Studio Code Extensión](#), encontrado en el material complementario.

Gestores

Existen diversos gestores de bases de datos NoSQL, algunos de ellos destacan por su facilidad de uso y por la integración que ofrecen al usuario o a los administradores de bases de datos (DBA).



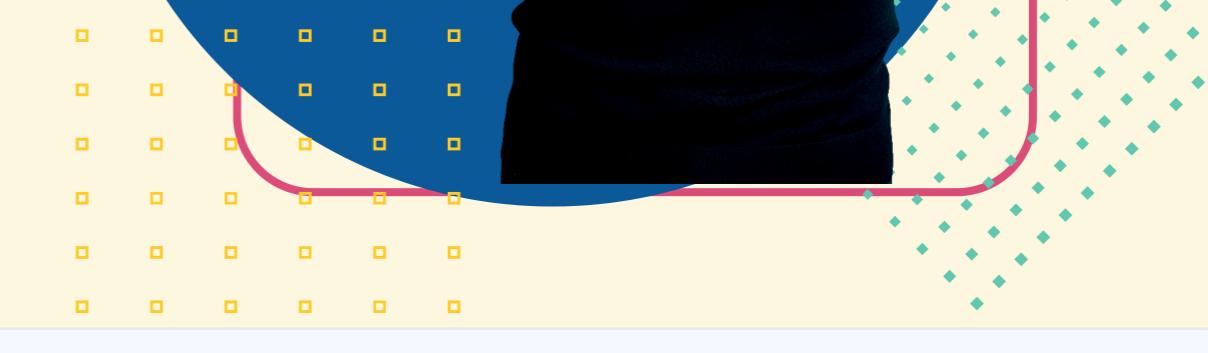
Algunos de estos gestores, son:

--	--

MongoDB.
Es un gestor de bases de datos que trabaja bajo la filosofía NoSQL, su proceso de manejo de la información es a nivel de archivos, MongoDB arma la secuencia de información que, posteriormente, puede ser relacionada mediante diferentes archivos planos. Adicional a esto, Mongo es una base de datos de código libre, es decir, puede ser modificada de acuerdo a las necesidades del usuario o de la organización que la implemente, su manejo interior se hace mediante archivos JSON que guardan las estructuras de la información para luego ser procesada y consultada.

Cassandra.
Es un gestor de bases de datos orientado a la filosofía NoSQL, permite trabajar con grandes volúmenes de información. Su código es abierto, permitiendo cualquier modificación de acuerdo a las necesidades tecnológicas de la organización o de quien requiera utilizar la herramienta con propósitos particulares o empresariales. Se encuentra escrito en lenguaje Java, posibilitando una amplia capacidad de adaptación en diferentes sistemas operativos, de manera distribuida, con balanceo de carga.

Los gestores de bases de datos permiten realizar las funciones de administración y mantenimiento de los datos de manera fácil y sencilla, esto ayuda a los administradores de estos sistemas a realizar labores de mantenimiento y gestión de manera rápida y eficiente. No obstante, en algunas ocasiones este proceso se suele llevar a cabo a través de consolas de administración, dilatando el proceso al ser una labor más compleja, especialmente con respecto a las copias de seguridad, ya que normalmente, en estos sistemas, es posible ejecutar las copias de manera automática, sin requerir una persona que se dedique a realizar este proceso a diario en consola, cometiendo así menos errores al momento de manipular la información. Otros procesos pueden ser actualización de datos, inserción de datos nuevos o eliminación de datos que ya no corresponden o no se necesitan en los procesos de negocio de la organización. Ahora bien, los gestores de bases de datos tienen muchas ventajas, una de ellas es la forma de conectarse a diferentes bases de datos, independiente del motor de las bases, es decir, se puede administrar varias bases de datos al tiempo con múltiples conexiones, significando una gran ventaja desde el punto de vista de licenciamiento y la eficiencia.



2 Bases de datos relacionales

Las bases de datos relacionales son un modelo de datos en el que los datos y la información, se encuentran relacionados por medio de llaves, estas permiten que la información pueda ser consultada mediante una serie de secuencia de comandos llamadas instrucciones SQL. Hay 4 comandos básicos que permiten controlar y manipular los procesos de la base de datos.

La siguiente imagen presenta los comandos DML para la gestión de la información de la base de datos.

Insert

Esta instrucción permite realizar el registro de nueva información dentro de la base de datos, es necesario aplicar las reglas de llaves, por ejemplo, si existe un registro que se repite, automáticamente se procederá a rechazar el ingreso del nuevo dato, indicando que ese dato ya se encuentra registrado en la base de datos para evitar problemas de ambigüedad de datos o duplicidad. `insert into tbl_clientes(PKid, Nombre, Contacto, Fecha_Nac, Dirección, Correo, FKcodigo_tbSexo) values ('123','Dulfran','3126954069','1983-04-20','Barranquilla','dulfraneling@gmail.com',1)`

Update

Esta instrucción permite realizar la actualización de un dato previamente registrado dentro de la base de datos, es importante destacar que para realizar la actualización, se debe cumplir con la regla de solo actualizar los campos que coincidan con el criterio de actualización, es decir, solo se deben actualizar los datos e información que coincidan con los datos relacionados.

`update tbl_clientes set Nombre = 'Pepito' where PKid = '123'`

El código realiza la actualización del nombre de la persona registrada, teniendo en cuenta, como restricción, la identificación de la persona, por lo cual solo se actualizará la información de este cliente en particular, no la de los demás clientes.

Delete

Esta instrucción permite eliminar un registro que se encuentre registrado previamente dentro de la base de datos, se utiliza cuando el registro no se requiera dentro de los procesos de negocio y su información debe desaparecer del sistema de almacenamiento.

`delete from tbl_clientes where PKid = '123'`

La instrucción procede a eliminar el registro del cliente, pero, como condición a cumplir, se elimina solo el cliente con la identificación registrada, sin afectar a los demás registros de la base de datos.

Select

Esta instrucción permite efectuar búsquedas de información de acuerdo con lo requerido por el usuario, estas consultas pueden realizarse de forma general o específica, es decir, se pueden traer un conjunto de datos asociados al registro o el de un usuario en particular.

`select * from tbl_clientes / select * from tbl_clientes where PKid = '123'`

Se presentan, así, dos posibilidades. La primera permite traer los datos de todos los clientes registrados, la segunda, ejecuta un filtro y solo traerá la información del cliente que coincide con la condición dada, que el documento de identificación sea 123.

Se ha abordado cada uno de los comandos que permiten manipular una base de datos relacional. A continuación se estudiará cuáles son los tipos de modelos que se manejan en este tipo de bases de datos.

Modelo conceptual

Se debe considerar que este modelo conceptual inicia con la recolección de requerimientos, a través de los cuales se tendrá como meta adquirir todas las entidades y conexiones de la información, dirigida por los procesos de negocio de la aplicación a desarrollar, considerando el proceso real a simular en el sistema a desarrollar, para así tomar los datos involucrados y se plasman en un diseño que pueda ser comprendido para el desarrollador de la aplicación. De esta forma, no queda información o datos por fuera de las diferentes entidades que conforman la base de datos a desarrollar.

Uno de los puntos importantes que tiene este tipo de modelo es que permite visualizar de manera sencilla cada entidad, sus conexiones y los datos que involucra.

La figura a continuación presenta un ejemplo de un modelo conceptual de una base de datos, teniendo en cuenta los siguientes pasos para su elaboración:

Modelo lógico

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

En este paso es primordial seleccionar el tipo de motor de base de datos que se utilizará para la implementación de la base de datos. En la actualidad, se pueden encontrar varios de ellos como Oracle, SQL Server, MySQL Server, tres de los motores de bases de datos más utilizados a nivel mundial en lo que se refiere a servidores de bases de datos relacionales.

Seguidamente, podrá observar un ejemplo de la creación física de una base de datos y cómo quedaría implementada en un servidor de bases de datos real.

Siguendo la imagen, la base de datos está implementada en un servidor de bases de datos real; para este caso se ha utilizado el motor de bases de datos SQL Server para la codificación, pero aplicaría igual para cualquiera de los motores de bases de datos mencionados anteriormente.

La imagen anterior presenta la codificación utilizada para crear la base de datos en el servidor, facultando la aplicación a almacenar la información de manera correcta dentro del sistema de almacenamiento. Una vez realizada la codificación de la base de datos y su creación, se deben generar las diferentes conexiones de la información, tal como lo presenta la siguiente imagen.

La imagen anterior presenta la codificación utilizada para crear la base de datos en el servidor, facultando la aplicación a almacenar la información de manera correcta dentro del sistema de almacenamiento. Una vez realizada la codificación de la base de datos y su creación, se deben generar las diferentes conexiones de la información, tal como lo presenta la siguiente imagen.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

El modelo lógico anterior propone identificar cada uno de los tipos de datos que tienen las entidades en la base de datos, facilitando que el programador, o la persona encargada de realizar la codificación de la base de datos en un sistema de gestión de base de datos, cuente con la información precisa para realizar el proceso, porque al conocer los diferentes tipos de datos que se conectan con las llaves y la restricción que cada una de ellas tiene, el desarrollador puede construir el sistema de almacenamiento con los tipos de datos y conexiones que este requiere.

</div

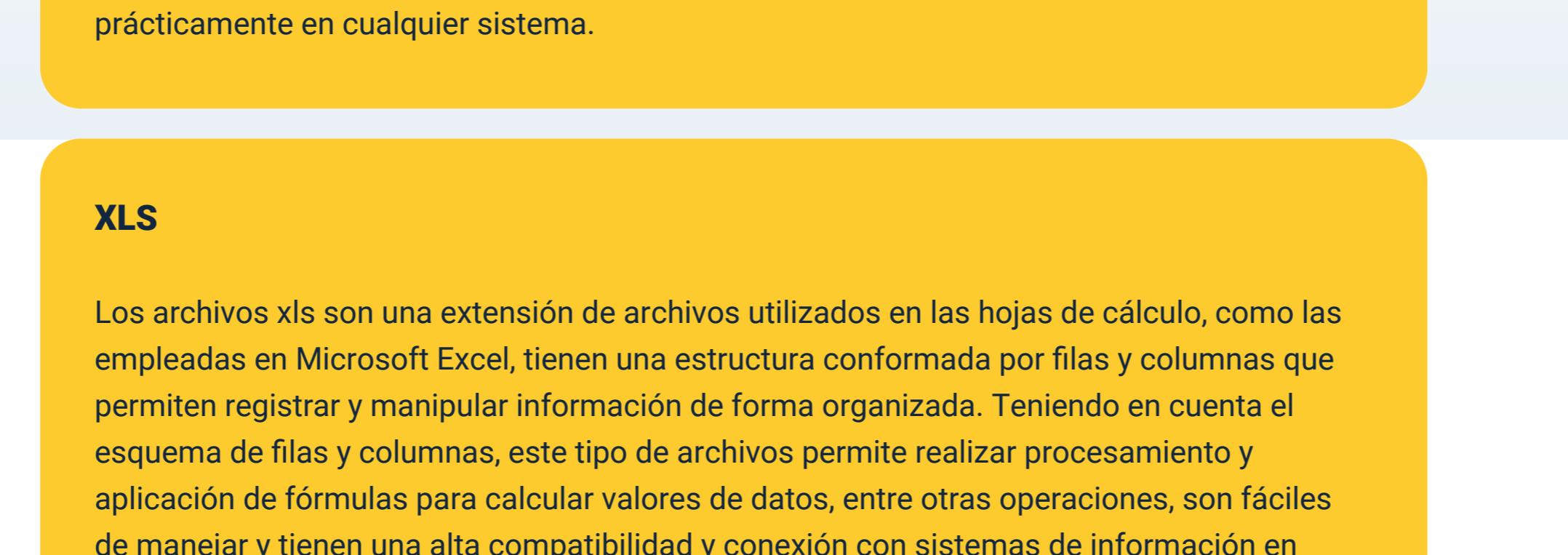
3 Fuentes de datos



Se denomina 'fuente de dato' a cualquier elemento que pueda suministrar datos e información, sin importar su formato o lugar de almacenamiento. Las fuentes de datos se originan a partir de las diferentes interacciones que los sistemas de información generan diariamente, por ejemplo, un archivo de Excel, empleado por una persona o compañía para disponer la información de la nómina de sus empleados, guarda, a su vez, información de nombres, contacto, correo, u otros datos diligenciados, y puede considerarse una fuente de información.

Existen diferentes fuentes de información como los archivos de intercambio de información, utilizados a través de la internet para almacenar información temporal, que son recopilados por sistemas de tratamiento de información como bases de datos, entre otros.

Conozca los diferentes tipos de fuentes de datos con sus respectivas características.



TXT

Los archivos con extensión TXT son archivos que se consideran planos debido a que no contienen ningún tipo de formato específico para el almacenamiento de la información. A pesar de esto, son utilizados para el almacenamiento de información porque ocupan muy poco espacio de almacenamiento en disco, además, otra de sus ventajas, es la compatibilidad que tienen con cualquier sistema operativo, es decir, con los editores de texto que tienen los principales sistemas operativos, por lo que pueden ser editados prácticamente en cualquier sistema.

XLS

Los archivos xls son una extensión de archivos utilizados en las hojas de cálculo, como las empleadas en Microsoft Excel, tienen una estructura conformada por filas y columnas que permiten registrar y manipular información de forma organizada. Teniendo en cuenta el esquema de filas y columnas, este tipo de archivos permite realizar procesamiento y aplicación de fórmulas para calcular valores de datos, entre otras operaciones, son fáciles de manejar y tienen una alta compatibilidad y conexión con sistemas de información en general.

JSON

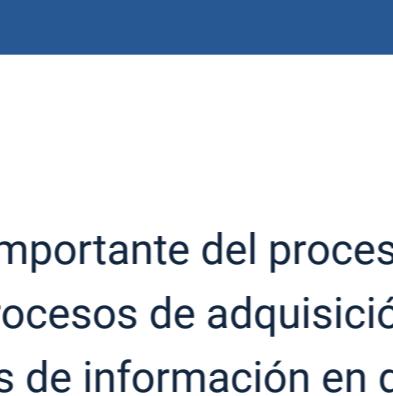
JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web, por ejemplo, para enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web o viceversa.

XML

Significa *Extensible Markup Language*. Es un lenguaje de marcado similar a HTML, pero es una especificación de W3C como lenguaje de marcado de propósito general. A diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que al trabajarla, se debe definir sus propias etiquetas. Su principal objetivo es compartir datos a través de diferentes sistemas como Internet.

Ahora, se presentan las principales características de cada uno de estos tipos de fuentes.

Características de los tipos de fuentes



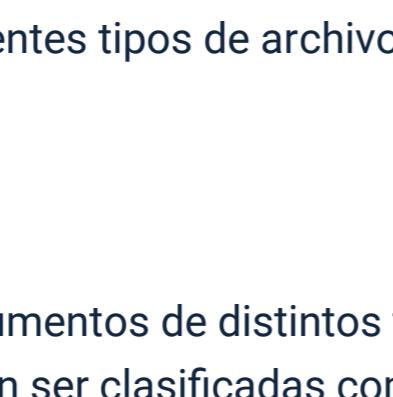
XML

- Es un lenguaje de etiquetado.
- Con él se crean estructuras propias de información.
- Es un estándar que permite intercambio de información entre varias plataformas.



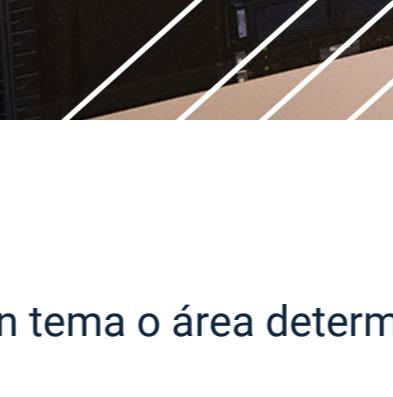
JSON

- Formato ligero de intercambio de datos.
- De fácil lectura y escritura para los usuarios.
- Fácilmente analizado y generados por parte de las máquinas.
- Formato de texto completamente independiente del lenguaje.
- Familiar para los programadores de lenguajes de la familia C, incluidos C, C++, C#, Java™, JavaScript, Perl, Python y mucho otros.



XLS

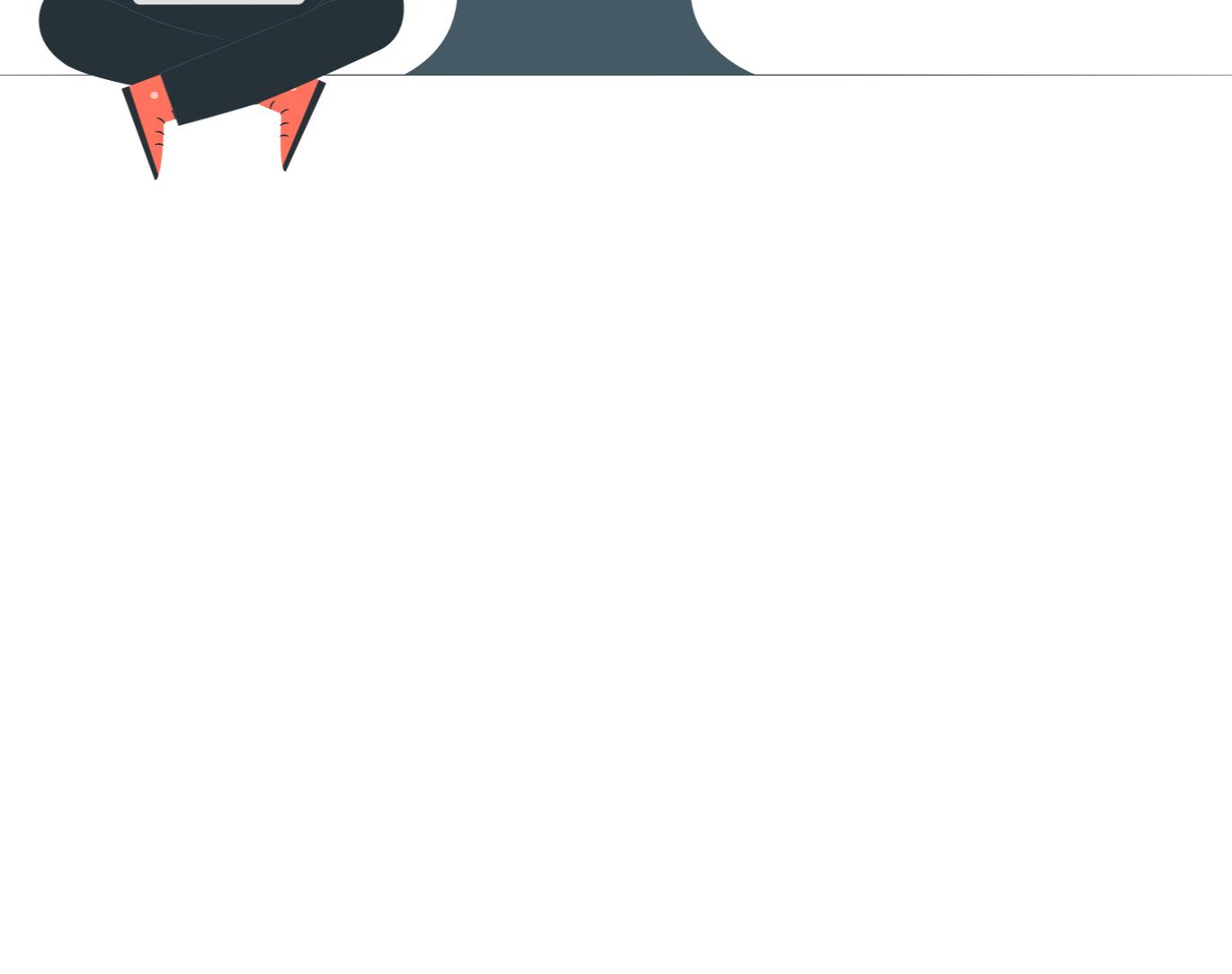
- Sustituido por un formato de archivo XML de Open Office (XML Spreadsheet o XMLSS).
- Empleado habitualmente por aplicaciones de terceros, como navegadores, para evitar la implementación del uso de formatos de archivo binario.
- Contiene información de hojas de cálculo e incluye gráficos, cálculos, tablas y macros.
- Esta extensión de archivo también se utiliza en el lenguaje de programación Visual Basic. Excel forma parte de Microsoft Office y está disponible para Windows y Mac OS.



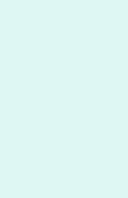
TXT

- Es universal debido a que cualquier procesador de texto puede leerlo.
- Puede utilizar Unicode para facilitar su lectura a usuarios de distintos idiomas.
- Exclusivamente en código ASCII pueden intercambiarse y leerse en Unix, Mac y Windows.

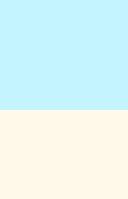
Las fuentes de datos son una parte importante del proceso de manejo de la información, proveen el insumo necesario para realizar los procesos de adquisición de la información. Como es conocido, las empresas producen muchos tipos de información en diferentes presentaciones, a eso se le puede llamar fuente de información. Ahora bien, es necesario tener en cuenta que la información tiene diferentes maneras de ser procesada, por ejemplo, si la fuente de información es un archivo .txt, un archivo plano, la información carecerá de algún tipo de tabulación, es decir, se deben utilizar mecanismos para garantizar su lectura de manera correcta y evitar tanto la pérdida de información como una lectura incorrecta de ella. En el caso de adquirir información de un archivo .xls, que está distribuido a través de filas y columnas, se debe establecer un método de lectura que garantice que se sustraiga la información de manera correcta, previniendo pérdidas o problemas en el proceso de recepción y, de igual forma, se aplicaría a los otros tipos de fuentes de información recopiladas en los diferentes tipos de archivos expuestos anteriormente.



Las fuentes de información son documentos de distintos tipos y orígenes que proveen datos y recursos sobre un tema o área determinada, según su nivel de información pueden ser clasificadas como:



Fuentes primarias



Fuentes secundarias



Fuentes terciarias



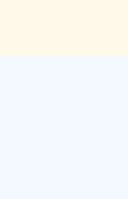
Fuentes primarias

Poseen información nueva que no ha sido modificada.



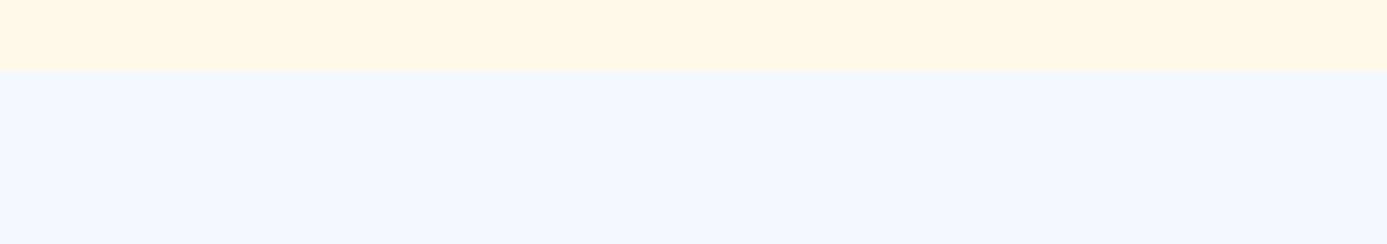
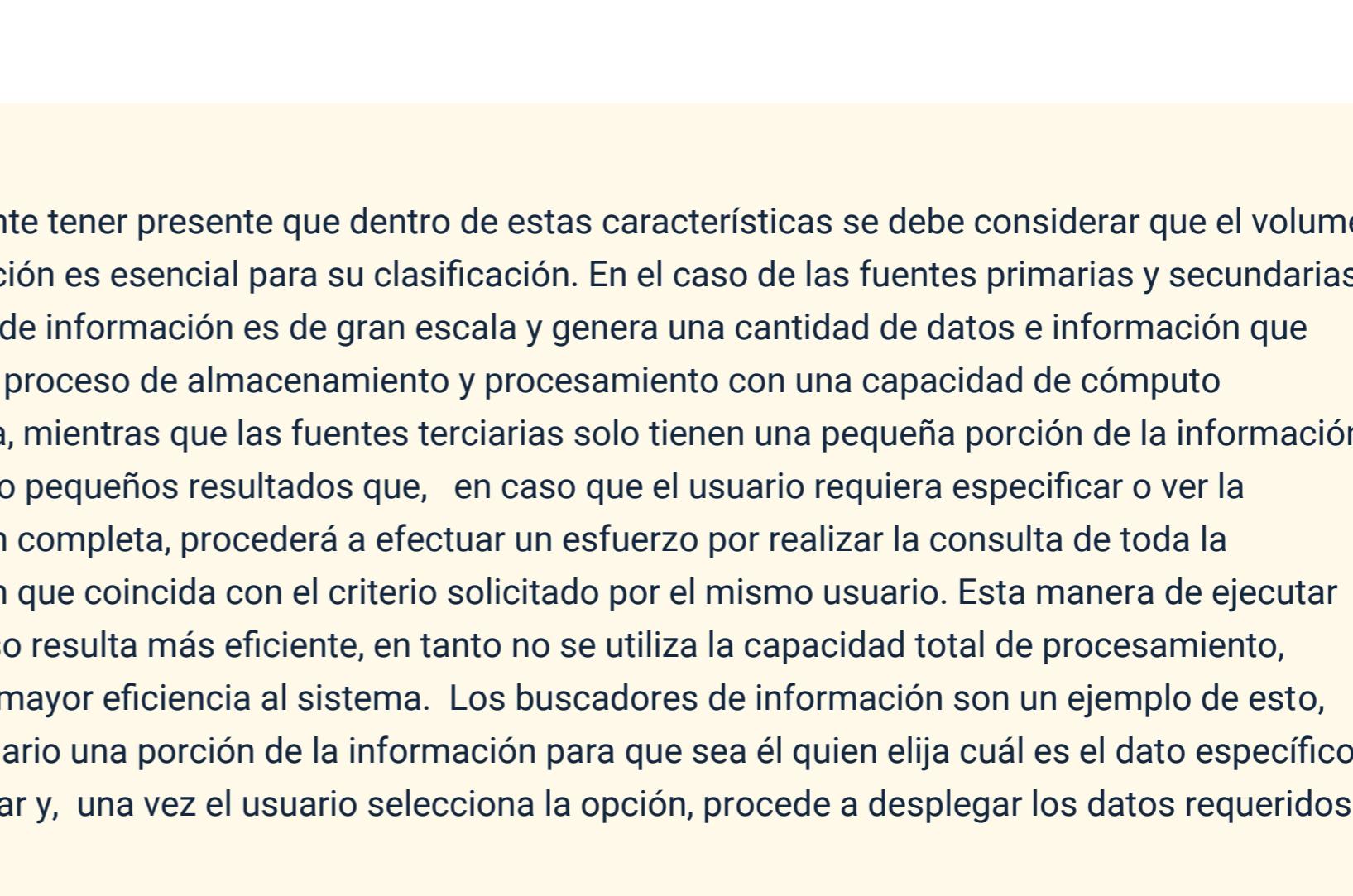
Fuentes secundarias

Contienen información que ha sufrido algún tipo de modificación.



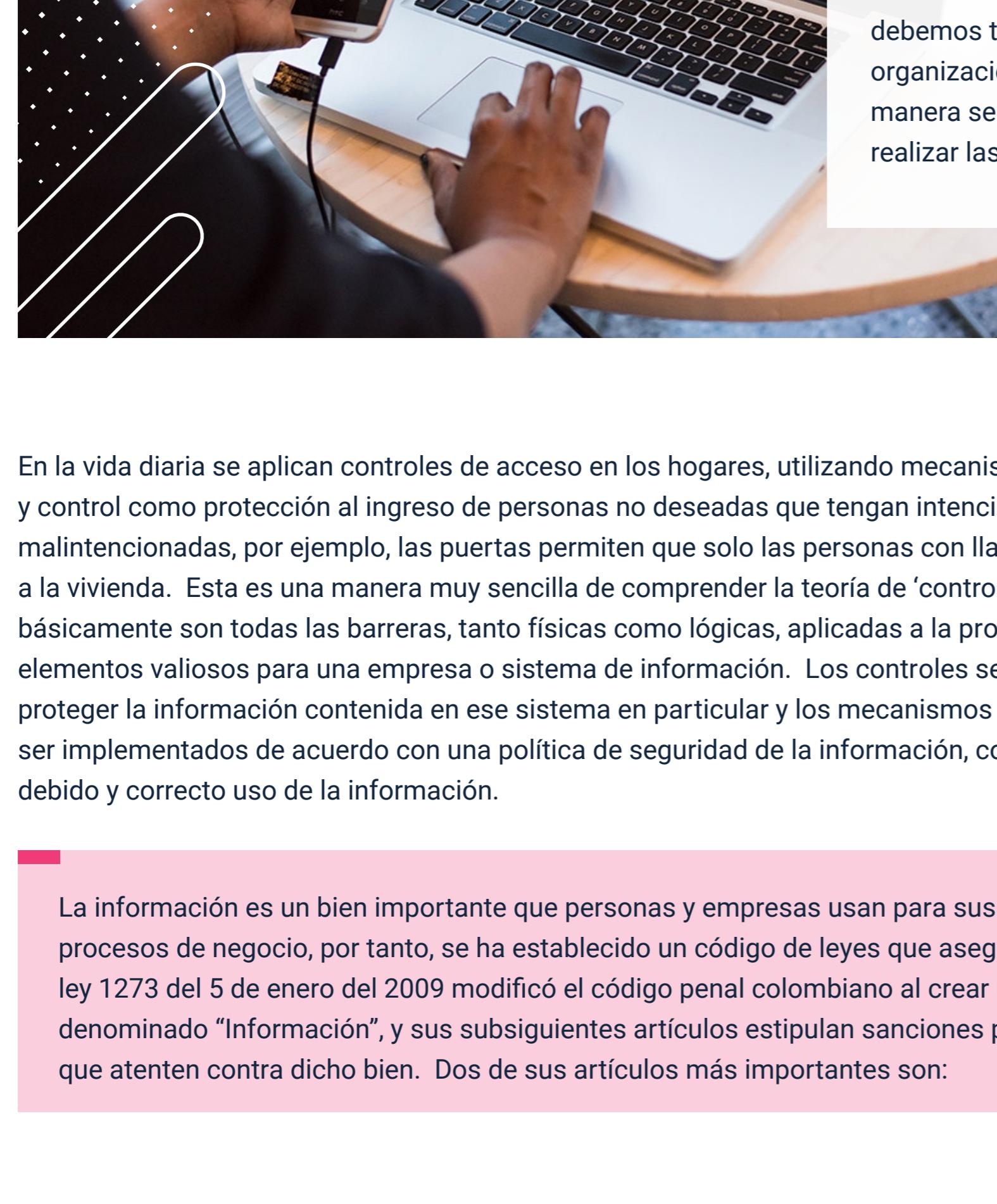
Fuentes terciarias

Comprenden solo una porción de información, la cual siempre debe ser consultada para obtener información de las fuentes anteriores.



Es importante tener presente que dentro de estas características se debe considerar que el volumen de información es esencial para su clasificación. En el caso de las fuentes primarias y secundarias, el volumen de información es de gran escala y genera una cantidad de datos e información que requiere un proceso de almacenamiento y procesamiento con una capacidad de cómputo significativa, mientras que las fuentes terciarias solo tienen una pequeña porción de la información, presentando pequeños resultados que, en caso que el usuario requiera especificar o ver la información completa, procederá a efectuar un esfuerzo por realizar la consulta de toda la información que coincida con el criterio solicitado por el mismo usuario. Esta manera de ejecutar este proceso resulta más eficiente, en tanto no se utiliza la capacidad total de procesamiento, generando mayor eficiencia al sistema. Los buscadores de información son un ejemplo de esto, traen al usuario una porción de la información para que sea él quien elija cuál es el dato específico por consultar y, una vez el usuario selecciona la opción, procede a desplegar los datos requeridos.

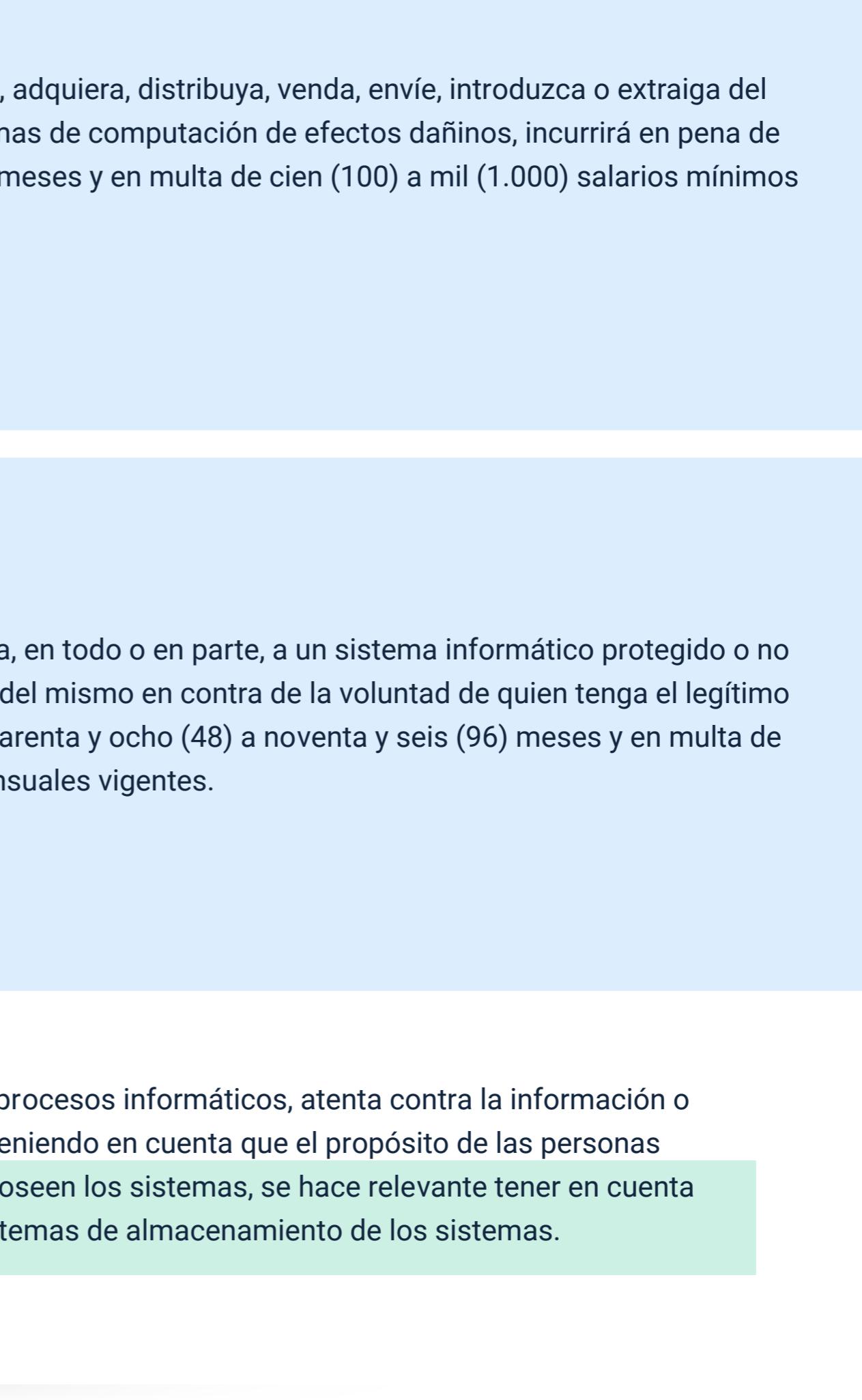
4 Control de acceso a datos



Los controles de acceso a datos son reglas y políticas que deben implementarse para realizar un proceso de administración adecuado de los datos e información almacenada en los sistemas de información, esto quiere decir que se debe no solo crear los mecanismos para el acceso de los datos, los cuales deben incluir las diferentes configuraciones y creación de usuarios, contraseñas, niveles de acceso y todo lo referente a las credenciales de acceso que se deben tener para acceder a la información almacenada, además, debemos tener en cuenta que la información es un activo importante para las organizaciones, y se debe velar porque esta información se encuentre almacenada de manera segura, y que las personas que tengan acceso a ella estén autorizados para realizar las operaciones requeridas para ser usada en la organización.

En la vida diaria se aplican controles de acceso en los hogares, utilizando mecanismos de protección y control como protección al ingreso de personas no deseadas que tengan intenciones malintencionadas, por ejemplo, las puertas permiten que solo las personas con llave puedan ingresar a la vivienda. Esta es una manera muy sencilla de comprender la teoría de 'controles de acceso', básicamente son todas las barreras, tanto físicas como lógicas, aplicadas a la protección de elementos valiosos para una empresa o sistema de información. Los controles se usan para proteger la información contenida en ese sistema en particular y los mecanismos de acceso deben ser implementados de acuerdo con una política de seguridad de la información, contemplando el debido y correcto uso de la información.

La información es un bien importante que personas y empresas usan para sus relaciones y procesos de negocio, por tanto, se ha establecido un código de leyes que aseguran tal bien. La ley 1273 del 5 de enero del 2009 modificó el código penal colombiano al crear un bien denominado "Información", y sus subsiguentes artículos estipulan sanciones para todo aquel que atenten contra dicho bien. Dos de sus artículos más importantes son:



Artículo 269A

Uso de software malicioso:

Quien sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envie, introduza o extraiga del territorio nacional software malicioso u otros programas de computación de efectos dañinos, incurrá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de cien (100) a mil (1.000) salarios mínimos legales mensuales vigentes.

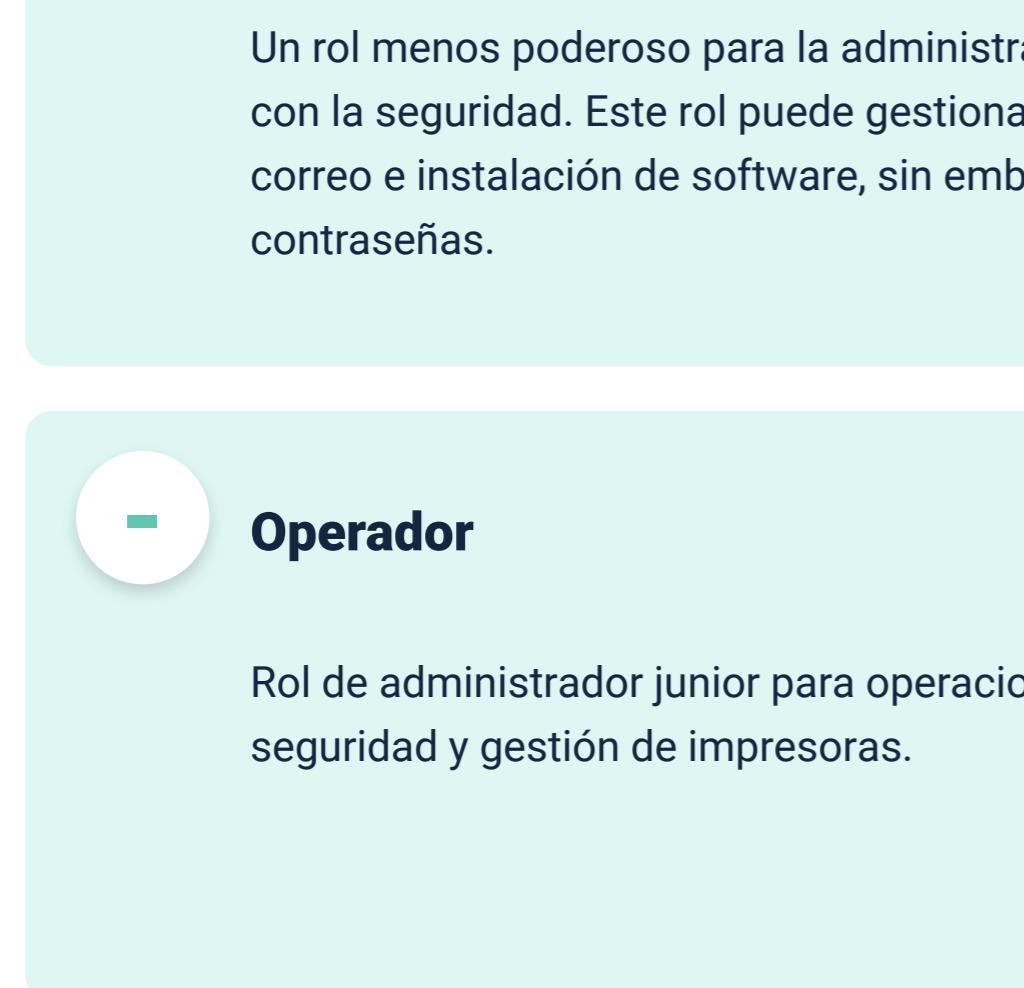
Artículo 269E

Acceso abusivo a un sistema informático

Quien, sin autorización o fuera de lo acordado, acceda, en todo o en parte, a un sistema informático protegido o no con una medida de seguridad, o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, incurrá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de cien (100) a mil (1.000) salarios mínimos legales mensuales vigentes.

La Ley 1273 explica las sanciones que se aplicarán si una persona, utilizando medidas o procesos informáticos, atenta contra la información o los datos almacenados en un sistema, de hacerlo, incurrirá en sanciones significativas. Teniendo en cuenta que el propósito de las personas dedicadas al crimen cibernetico es violar y atentar contra las medidas de seguridad que poseen los sistemas, se hace relevante tener en cuenta cada uno de estos procesos y estar atentos a todo lo ocurrido en una infraestructura y sistemas de almacenamiento de los sistemas.

Control de acceso basado en roles



El control de acceso basado en roles (RBAC), es una función de seguridad para controlar el acceso de usuarios a tareas que habitualmente están restringidas a un superusuario que, mediante la aplicación de atributos de seguridad a procesos y usuarios, el RBAC puede dividir sus capacidades entre varios administradores. Tal gestión de derechos de procesos se implementan mediante privilegios, a través de RBAC como alternativa al modelo de superusuario. (Oracle, 2010).

El usuario **root**, también conocido como superusuario, es omnipotente, como también los programas que se ejecutan como root, o los programas **setuid**. El usuario **root** puede leer y escribir en cualquier archivo, ejecutar todos los programas y enviar señales de terminación a cualquier proceso. Un programa usurpado puede realizar cualquier tarea en el sistema (Oracle, 2010), de hecho, cualquier persona que puede convertirse en superusuario puede modificar el cortafuegos de un sitio, modificar la pista de auditoría, leer registros confidenciales y apagar toda la red.

La flexibilidad en la configuración de los roles posibilita una variedad de políticas de seguridad, y aunque incluye pocos roles, es posible configurar fácilmente los tres roles recomendados, basados en perfiles de derechos con el mismo nombre (Oracle, 2010). Estos roles son:

- Administrador principal**
- root**
- Administrador del sistema**

- Operador**
- Administrador principal**
 - Un rol poderoso equivalente al usuario **root** o superusuario.

- root**
 - Al igual que el anterior es un rol potente con el impedimento de iniciar sesión. Un usuario debe iniciar primera sesión y luego asumir el rol **root** asignado.

- Administrador del sistema**
 - Un rol menos poderoso para la administración y no está relacionado con la seguridad. Este rol puede gestionar sistemas de archivos, correo e instalación de software, sin embargo, no puede definir contraseñas.

- Operador**
 - Rol de administrador junior para operaciones como copias de seguridad y gestión de impresoras.



Acceso basado en el computador principal

Este modelo de acceso es impreciso y poco seguro, consiste en tener una computadora que interactúa como intermediaria con el servidor de almacenamiento, si esta computadora tiene privilegios de modificación o privilegios de acceso como usuarias administradoras, en base de datos en ninguna restricción. Aunque es un sistema simple que no representa un gran proceso de implementación, es utilizado comúnmente en muchas organizaciones por su fácil implementación y mantenimiento. Igualmente, la conexión de acceso remoto, regularmente usada para acceder a los equipos desde cualquier lugar, siempre y cuando se proporcionen las credenciales necesarias para poder tener el acceso, podría generar un problema a futuro, en tanto no se pueden controlar de manera adecuada los niveles de acceso a la información.

Dentro de las características principales a resaltar de los controles de acceso, se encuentran las siguientes:

Protección de la información

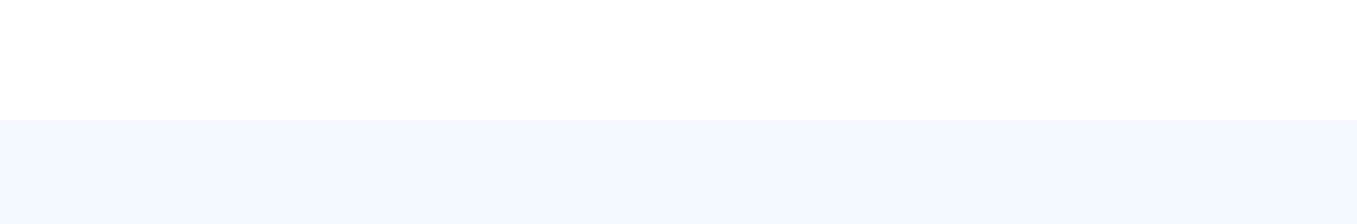
Integridad

Disponibilidad



Integridad

Es la capacidad que tiene un sistema de información para realizar las operaciones procedimentales de manera adecuada, es decir que los procesos que han sido programados dentro del sistema se realizan de manera adecuada y arrojan los resultados esperados. Como ejemplo se puede observar el funcionamiento de un cajero automático, no solo está programado para la entrega de dinero a los usuarios, también para realizar las transacciones de forma adecuada. Si un usuario tiene \$100.000 pesos en su cuenta y retira \$40.000 y no existe ningún proceso de descuento por transacción, lo ideal es que dentro de la cuenta resten \$60.000, porque el sistema realiza una operación matemática, la resta, y los valores resultantes deben ser precisos y acordes a lo solicitado por el usuario. En caso de arrojar un valor diferente, se podrá decir que el sistema no tiene integridad y debe ser revisado para que realice las operaciones de manera adecuada para los usuarios que utilicen este sistema de información.



Protección de la información

Esta característica es fundamental al momento de pensar en la implementación de un sistema de almacenamiento, porque la información, al ser uno de los activos más importantes de las empresas, sirve en la toma de decisiones o, en la mayoría de los casos, dependen de ella para realizar labores de adquisición de productos o servicios.

Disponibilidad

Esta es una característica de los sistemas de almacenamiento que establece los mecanismos y tiempos en los que el sistema de información deberá responder a los usuarios que requieren acceso a la información, esto implica la verificación de si esa persona tiene acceso o no a dicha información. El proceso de disponibilidad depende de una política de acceso a la información que determina los tiempos de negociación de entrega de la información y cómo el sistema estará operable para el proceso. Existen diferentes modelos de disponibilidad dos de ellos son:

- 24/7**: El sistema se encuentra disponible las 24 horas del día, los 7 días de la semana, a esto se le denomina también alta disponibilidad.

- 8/7**: El sistema está disponible 8 horas al día, los 7 días de la semana.

Estas son formas de controlar el acceso a la información y políticas de acceso a ella.

Confidencialidad

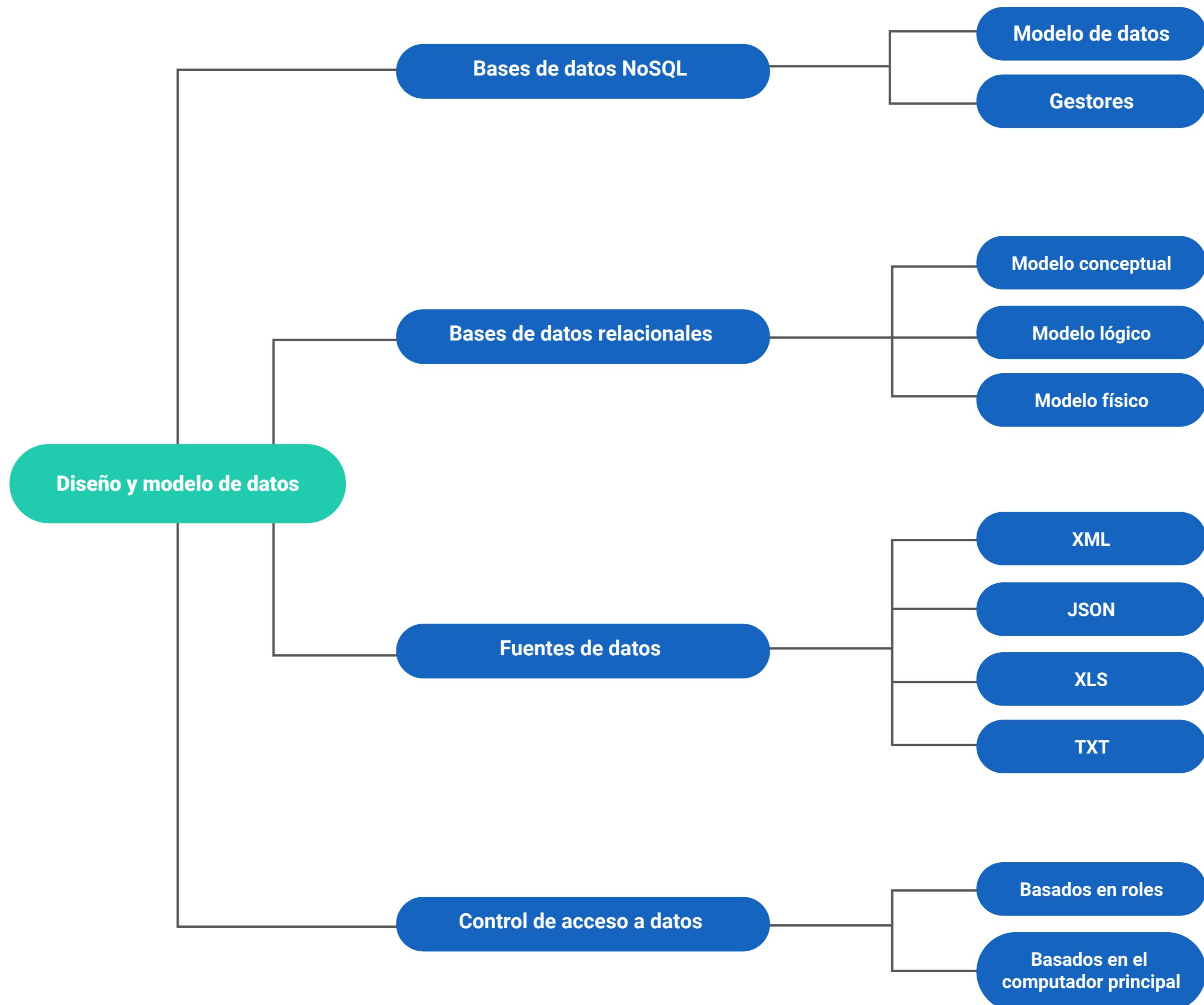
Esta característica se relaciona con las credenciales de acceso, la información requiere tener un dueño o responsable de su manipulación y uso. Ejemplo, las redes sociales o correo electrónico, ambos tienen mecanismos de acceso como usuarios y contraseñas o, en algunos casos, el acceso utilizan códigos enviados a dispositivos móviles con el fin de garantizar que quien solicita el acceso a la información sea la persona responsable o dueña de ella.

Recuerde explorar los demás recursos que se encuentran disponibles en este componente formativo; para ello, diríjase al menú principal, donde encontrará la síntesis, una actividad didáctica para reforzar los conceptos estudiados, material complementario, entre otros.

Desarrollo de aplicaciones web Full stack

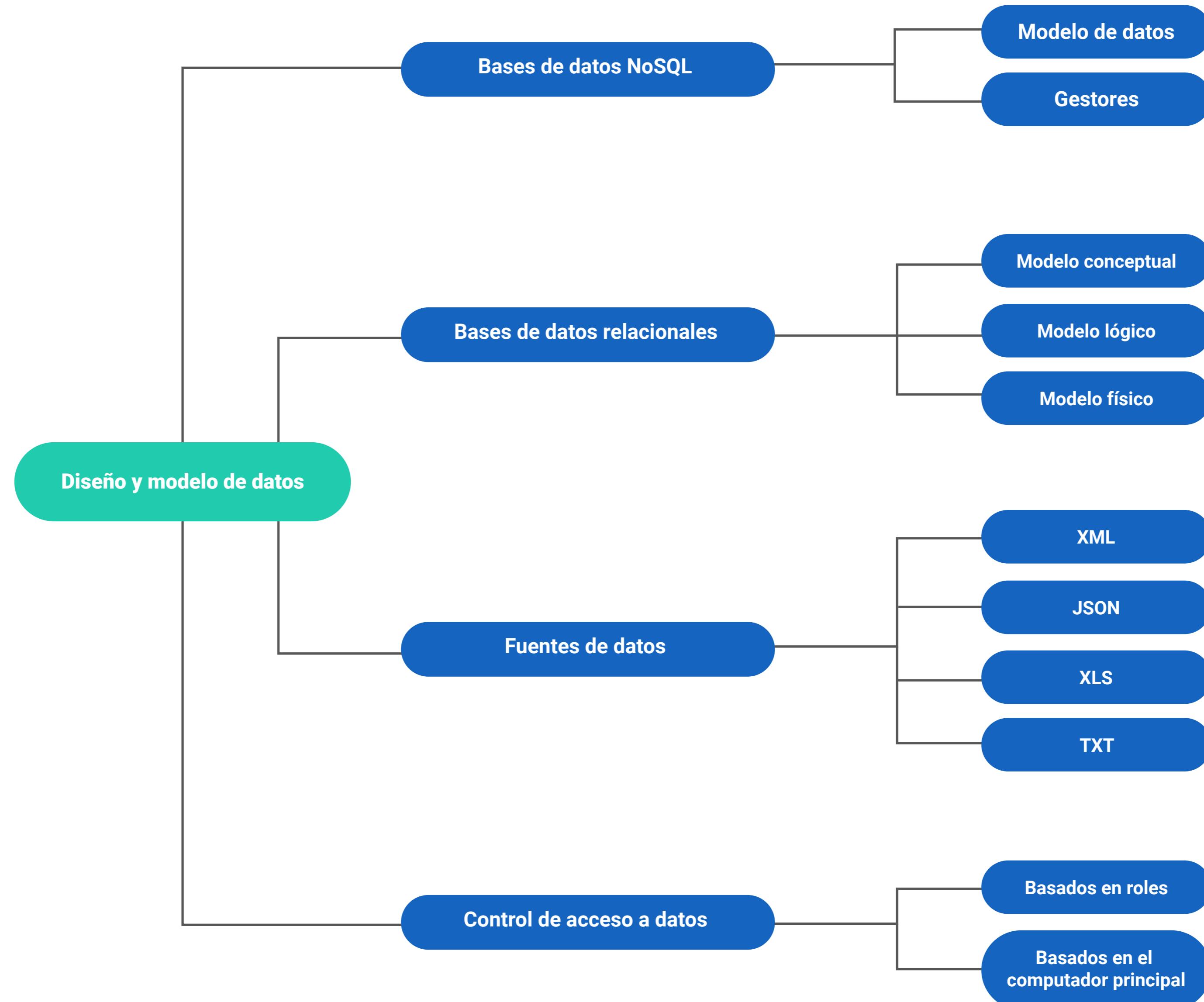
Síntesis: Diseño y modelo de datos.

La siguiente síntesis presenta un resumen de la temática estudiada en el componente formativo.



Desarrollo de aplicaciones web Full stack

Síntesis: Acuerdos y elaboración de propuesta técnico-económica.



Portada actividad

800 x 800



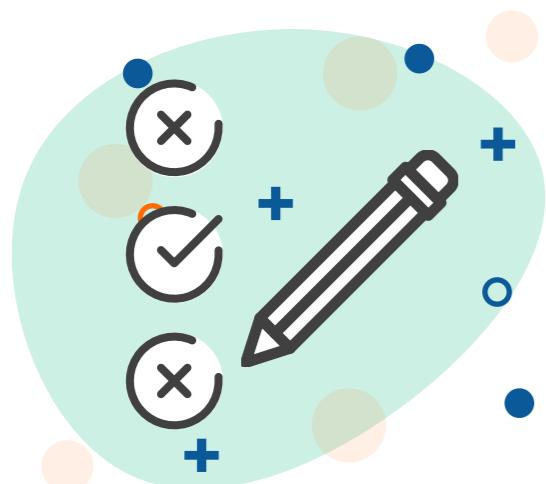
Imagen acompañamiento Actividad

350 x 480





Actividad didáctica



Selección múltiple

Evaluar la apropiación de las temáticas estudiadas en el componente formativo.

Selección múltiple

Realizar 