



Universidad
de la Ciudad de
Aguascalientes

BASES DE DATOS PARA CIENCIA DE DATOS

Mentes que transforman el mundo

Bases de Datos Semi-estructuradas

Bases de Datos Semi-estructuradas

Las bases de datos semiestructuradas son sistemas de almacenamiento de datos que no requieren un esquema rígido como las bases de datos tradicionales. A diferencia de las bases de datos estructuradas, donde los datos se almacenan en tablas con filas y columnas predefinidas, las bases de datos semiestructuradas permiten que los datos tengan una estructura más flexible, con jerarquías y relaciones que no necesariamente siguen un formato fijo.

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

```
<University>
  <Student ID="1">
    <Name>John</Name>
    <Age>18</Age>
    <Degree>B.Sc.</Degree>
  </Student>
  <Student ID="2">
    <Name>David</Name>
    <Age>31</Age>
    <Degree>Ph.D. </Degree>
  </Student>
  ....
</University>
```

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Structured Data

vs

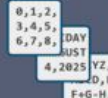
Unstructured Data

Can be displayed
in rows, columns and
relational databases



XY	1	2
A	A1	A2
B	B1	B2
C	C1	C2
D	D1	D2

Numbers, dates
and strings



0, 1, 2,
3, 4, 5,
6, 7, 8, DAY
AUGUST
4, 2025 YZ,
D, E
F+G-H,

Estimated 20% of
enterprise data (Gartner)

20%

Requires less storage

==
==
==

Easier to manage
and protect with
legacy solutions



Cannot be displayed
in rows, columns and
relational databases



Images, audio, video,
word processing files,
e-mails, spreadsheets



Estimated 80% of
enterprise data (Gartner)

80%

Requires more storage

==
==
==
==
==
==

More difficult to
manage and protect
with legacy solutions



Extensible Markup Language (XML),
permite definir etiquetas y atributos
para datos no estructurados y
pueden ser almacenados de forma
jerárquica.

```
<studentsList>
  <student id="1">
    <firstName>Greg</firstName>
    <lastName>Dean</lastName>
    <certificate>True</certificate>
    <scores>
      <module1>70</module1>
      <module12>80</module12>
      <module3>90</module3>
    </scores>
  </student>
  <student ind="2">
    <firstName>Wirt</firstName>
    <lastName>Wood</lastName>
    <certificate>True</certificate>
    <scores>
      <module1>80</module1>
      <module12>80.2</module12>
      <module3>80</module3>
    </scores>
  </student>
</studentsList>
```

JavaScript Object Notation (JSON) es una alternativa muy similar al XML pero optimizada para el intercambio en web

```
https://microsoftedge.github.io x +
https://microsoftedge.github.io/Demos/json-dummy-data/256KB-1
[
  {
    "name": "Adeel Solangi",
    "language": "Sindhi",
    "id": "V590F92YF627H FY0",
    "bio": "Donec lobortis eleifend condimentum. Cras dictum d
Maecenas quis nisi nunc. Nam tristique feugiat est vitae mollis. M
    "version": 6.1
  },
  {
    "name": "Afzal Ghaffar",
    "language": "Sindhi",
    "id": "ENTOCR13RSCLZ6KU",
    "bio": "Aliquam sollicitudin ante ligula, eget malesuada n
sem, scelerisque sit amet odio id, cursus tempor urna. Etiam congu
pharetra libero et velit gravida euismod.",
    "version": 1.88
  },
  {
    "name": "Aamir Solangi",
    "language": "Sindhi",
    "id": "IAKPO3R4761JDRVG",
    "bio": "Vestibulum pharetra libero et velit gravida euismo
porttitor sodales ac, lacinia non ex. Fusce eu ultrices elit, vel
    "version": 7.27
  },
  {
    "name": "Abla Dilmurat",
    "language": "Uyghur",
    "id": "5ZVOEPMJUI4MB4EN",
    "bio": "Donec lobortis eleifend condimentum. Morbi ac tell
    "version": 2.53
  },
  {
    "name": "Adil Eli",
```

Hyper-Text Markup Language (HTML) es utilizado para la web, provee una jeraquia para estructurar datos con etiquetas

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="pingback" href="https://www.datamation.com/xmlrpc.php" />
<meta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, ma
ink rel="icon" type="image/png" href="https://www.datamation.com/wp-content/uploads/2021/
<!-- This site is optimized with the Yoast SEO plugin v20.6 - https://yoast.com/wordpre
<meta name="description" content="IT & Tech Industry coverage focusing on Emerging
<link rel="canonical" href="https://www.datamation.com/" />
<meta property="og:locale" content="en_US" />
<meta property="og:type" content="website" />
<meta property="og:title" content="Technology News: Latest IT and Tech Industry News" /
<meta property="og:description" content="IT & Tech Industry coverage focusing on Em
<meta property="og:url" content="https://www.datamation.com/" />
<meta property="og:site_name" content="Datamation" />
<meta property="article:modified_time" content="2023-03-08T22:36:57+00:00" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:label1" content="Est. reading time" />
<meta
```



```
31.286950 15c8 Compile date 2022-10-14 09:07:51
31.287148 15c8 DB SUMMARY
31.287165 15c8 DB Session ID: 44GNK6D01WHC0FED75RW
31.288127 15c8 CURRENT file: CURRENT
31.288147 15c8 IDENTITY file: IDENTITY
31.288271 15c8 MANIFEST file: MANIFEST-000016 size: 531 Bytes
31.288290 15c8 SST files in C:\Users\cb\OneDrive\Pictures\Lightroom\Lightroom Catalog-v13.lrcat-data dir, Total
31.288302 15c8 Write Ahead Log file in C:\Users\cb\OneDrive\Pictures\Lightroom\Lightroom Catalog-v13.lrcat-data:
31.288550 15c8 Options.error_if_exists: 0
31.288556 15c8 Options.create_if_missing: 1
31.288560 15c8 Options.paranoid_checks: 1
31.288564 15c8 Options.flush_verify_memtable_count: 1
31.288568 15c8 Options.track_and_verify_wals_in_manifest: 0
31.288572 15c8 Options.verify_sst_unique_id_in_manifest: 0
31.288576 15c8 Options.env: 0000000022166060
31.288580 15c8 Options.fs: WinFS
31.288585 15c8 Options.info_log: 000000002220A2B0
31.288589 15c8 Options.max_file_opening_threads: 16
31.288592 15c8 Options.statistics: 0000000000000000
31.288596 15c8 Options.use_fsync: 0
31.288600 15c8 Options.max_log_file_size: 0
31.288604 15c8 Options.max_manifest_file_size: 536870912
31.288608 15c8 Options.log_file_time_to_roll: 0
31.288612 15c8 Options.keep_log_file_num: 1000
31.288616 15c8 Options.recycle_log_file_num: 0
31.288620 15c8 Options.allow_fallocate: 1
31.288623 15c8 Options.allow_mmap_reads: 0
31.288627 15c8 Options.allow_mmap_writes: 0
31.288631 15c8 Options.use_direct_reads: 0
31.288635 15c8 Options.use_direct_io_for_flush_and_compaction: 0
31.288639 15c8 Options.create_missing_column_families: 0
31.288642 15c8 Options
31.288646 15c8 Options
31.288650 15c8 Options
31.288654 15c8 Options.WAL_ttl_seconds: 0
31.288658 15c8 Options.WAL_size_limit_MB: 0
31.288661 15c8 Options.max_write_batch_group_size_bytes: 1048576
31.288665 15c8 Options.manifest_preallocation_size: 4194304
```

Log Files

ST*850*1001 ☐
BEG*00*SA*4768*65*050410 ☐
N1*123 MAIN STREET ☐
N4*FAIRVIEW*CA*94618 ☐
PO1*1*100*EA*27.65**VN*331896-42☐
CTT*1*100 ☐
SE*8*1001 ☐

Legend:

ST*Transaction Set ID*Transaction Set Control Number
BEG*Transaction Set Purpose*Purchase Order Date
Number*Release Number*Purchase Order Date
N1*Name Type*Name
N3*Address
N4*City*State*Zip Code
P01*Line Number*Quantity Ordered*Unit of Measure*Product
Basis*Product ID
Qualifier*Product ID
CTT*Number of Line Items*HashTotal
SE*Number of Included Segments*Transaction Set Control

Electronic Data Interchange (EDI) se utiliza ampliamente en diversas industrias, incluidas el comercio, la manufactura, la atención médica, la logística, las finanzas y entre otros, principalmente se utiliza para la digitalización de documentos.

Características

1. Los datos no se rigen por un modelo de datos, pero tienen estructura
2. No es posible almacenar en forma de filas y columnas
3. Contienen metadatos que detallan la estructura en los propios datos
4. Las entidades pueden o no tener los mismo atributos o propiedades

Ventajas

1. Flexibilidad

A diferencia de las bases de datos relacionales, donde se requiere un esquema fijo y predefinido, las bases de datos semiestructuradas permiten almacenar datos sin necesidad de un esquema rígido. Esto es particularmente útil cuando los datos pueden cambiar con el tiempo o cuando se integran datos de múltiples fuentes con diferentes estructuras.

- Facilita la adaptación a cambios en los datos sin necesidad de reestructurar toda la base de datos.

2. Uso de datos jerárquicos o complejos

Los datos que tienen una estructura jerárquica o relaciones complejas (como documentos XML o JSON) se manejan de manera eficiente en bases de datos semiestructuradas.

- Permite representar relaciones complejas y anidadas de manera natural y sencilla.

3. Integración de datos heterogéneos y No Normalizados

Al no requerir un esquema uniforme, es más fácil integrar y manejar datos provenientes de diferentes fuentes, cada una con su propia estructura.

- Mejora la capacidad de fusionar datos de diversas fuentes sin necesidad de normalización o conversión previa.

- Facilita la captura y almacenamiento de datos tal como se encuentran, sin necesidad de un preprocesamiento exhaustivo.

4. Eficiencia en el almacenamiento de datos incompletos (porosidad)

En bases de datos semiestructuradas, no es necesario que todos los registros contengan todas las propiedades, lo que permite almacenar datos incompletos sin generar datos nulos o redundantes.

- Optimiza el almacenamiento y evita desperdicio de espacio en casos donde no toda la información está disponible.

5. Facilidad para manejar cambios en los datos

La estructura flexible permite agregar, eliminar o modificar propiedades de los datos sin necesidad de alterar un esquema fijo.

- Permite a las organizaciones adaptarse rápidamente a nuevos requerimientos sin incurrir en grandes costos de reestructuración.

- Reduce el tiempo y los recursos necesarios para mantener la base de datos cuando hay cambios en los requisitos de los datos.

6. Compatibilidad con Formatos como XML y JSON

Estos formatos son ampliamente utilizados en aplicaciones web y móviles, lo que facilita la integración con tecnologías modernas.

- Mejora la interoperabilidad con aplicaciones y servicios web, y facilita la manipulación de datos en estos formatos.

7. Consulta dinámicas según las necesidades

Aunque puede ser más complicado en comparación con bases de datos relacionales, las consultas en bases de datos semiestructuradas permiten recuperar datos sin necesidad de conocer la estructura completa de los mismos.

- Permite realizar consultas dinámicas y flexibles, adaptándose a las necesidades de la aplicación en tiempo real.

8. Escalabilidad

Muchas bases de datos semiestructuradas, como las bases de datos NoSQL, están diseñadas para escalar horizontalmente, permitiendo manejar grandes volúmenes de datos y solicitudes simultáneas.

- Ofrece la capacidad de crecer junto con las necesidades de la aplicación, manejando grandes volúmenes de datos y tráfico con mayor eficiencia.

Desventajas

1. Integridad de los datos

Es posible que dadas las características de los datos que se almacenen sea complejo aplicar medidas de integridad de datos, así como validación de la información.

- Posibles inconsistencias en los datos que generar errores al momento de explotar la información

2. Rendimiento inferior en ciertos escenarios

Las bases de datos semiestructuradas pueden ser menos eficientes en operaciones que involucren grandes uniones, agregaciones complejas o consultas que requieren explorar múltiples niveles de jerarquía.

- El rendimiento puede degradarse significativamente en comparación con bases de datos relacionales optimizadas para estos tipos de consultas.

3. Menor número de herramientas

- Aunque las herramientas para manejar bases de datos semiestructuradas han avanzado, no siempre son tan maduras o completas como las herramientas disponibles para bases de datos relacionales.

4. Dificultad de análisis

- Se puede complicar el tema del análisis de información en ciertos escenarios dada la naturaleza de la información que puede ser complejo etiquetar o indexar..

Introducción a XML

XML

XML (Extensible Markup Language) permite describir y organizar la información de maneras que son fácilmente comprensibles tanto para los seres humanos como para los sistemas. A continuación, puede compartir esa información y su descripción con otros a través de Internet, una extranet, una red o de otras formas.

Es posible utilizar XML para crear su propio lenguaje de marcación que incluya un conjunto de reglas y etiquetas que describan información que se adapte a sus necesidades, por ejemplo, nombre, título, dirección y código postal. Este lenguaje de marcación se define en una definición de tipo de documento (DTD) o un archivo de esquema XML que funciona como la forma estándar de describir la información. El uso de XML para compartir información estandarizada significa que ya no está obligado a escribir programas para centrarse en software propietario o convertir y traducir diferentes formatos de datos.

Aunque tanto XML como HTML utilizan etiquetas para describir el contenido, también son muy diferentes:

1. HTML describe cómo dar formato a la información para su visualización y está destinado a la interacción de equipo a humano.
2. XML describe lo que la información es y está pensada para la interacción de sistema a sistema.

XML utiliza lenguaje humano, no informático. XML es legible y comprensible, incluso por principiantes, y no es más difícil de codificar que HTML.

XML es completamente compatible con lenguajes de programación y 100% portable, cualquier aplicación que pueda procesar XML puede utilizar su información, independientemente de la plataforma.

XML es ampliable, permite crear sus propias etiquetas, o utilizar etiquetas creadas por otros, que utilicen el lenguaje natural de su dominio, que tengan los atributos que necesita y que tengan sentido para usted y sus usuarios.

XML

Intercambio de Datos: XML es comúnmente utilizado para intercambiar datos entre sistemas diferentes, ya que es independiente de la plataforma y el lenguaje de programación. Por ejemplo, servicios web (Web Services) suelen usar XML para enviar y recibir datos.

Almacenamiento de Datos: XML puede ser utilizado para almacenar datos estructurados de forma similar a una base de datos, pero en un formato de texto. Esto es útil para configuraciones, datos de aplicaciones y otros tipos de información que necesitan ser fácilmente transportados o leídos por diferentes sistemas.

Estandarización de Documentos: XML es la base para muchos formatos estándar de documentos, como XHTML (una versión de HTML), SVG (para gráficos vectoriales), y muchos otros. Esto permite que los documentos sean compatibles con diferentes herramientas y plataformas.

Configuración de Aplicaciones: Muchas aplicaciones utilizan archivos XML para configurar diferentes parámetros y ajustes. Por ejemplo, archivos de configuración de software como los utilizados en frameworks de desarrollo, servidores web, entre otros.

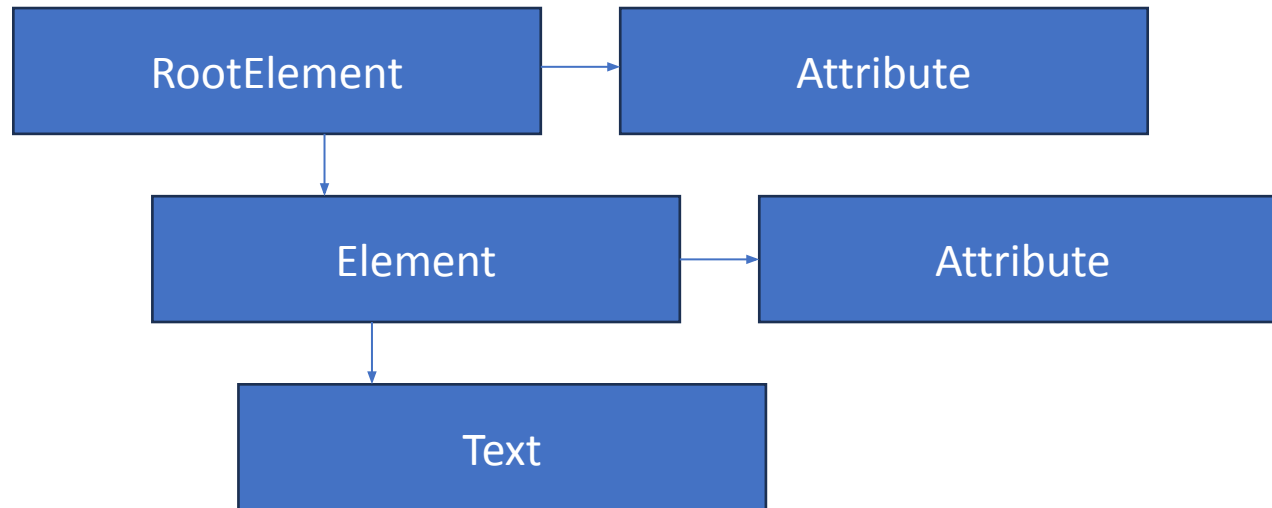
Documentación Técnica: XML es utilizado para la creación de documentación técnica, debido a su capacidad para estructurar información de manera clara y organizada. Ejemplos incluyen DITA (Darwin Information Typing Architecture) para la creación de documentación técnica modular.

Desarrollo de Interfaces de Usuario (UI): En algunos entornos de desarrollo, XML es usado para definir interfaces de usuario, como en Android con los archivos de layout, o en XAML para aplicaciones en .NET.

Bases de Datos XML: Algunas bases de datos están diseñadas específicamente para almacenar y gestionar datos en formato XML, permitiendo consultas y transformaciones de los datos de manera eficiente.

Integración de Aplicaciones: XML facilita la integración entre diferentes aplicaciones y servicios, al proporcionar un formato común para representar datos, lo que es particularmente útil en entornos de empresas donde múltiples sistemas deben interactuar entre sí.

Estructura XML



Estructura XML

Element

Un elemento de XML es todo lo que incluya una etiqueta de inicio `<algo>` y una etiqueta de fin `</algo>`

Puede contener text, atributos, otros elementos o una combinación de ambos

Attribute

Información adicional de los elementos que debe ser entrecomillada `<algo atributo="1"></algo>`

```
<library>
  <book>
    <title>XML Basics</title>
    <author>John Doe</author>
    <price>29.99</price>
  </book>
  <book>
    <title>Advanced XML</title>
    <author>Jane Smith</author>
    <price>39.99</price>
  </book>
  <book>
    <title>XQuery in Action</title>
    <author>Mary Johnson</author>
    <price>49.99</price>
  </book>
</library>
```


XPath

XPath (XML Path Language) es un lenguaje utilizado para navegar y seleccionar nodos en un documento XML. Es una herramienta poderosa y flexible que permite extraer, manipular y analizar datos en XML.

1. Selección de Nodos:

Rutas Absolutas y Relativas: XPath permite seleccionar nodos específicos usando rutas absolutas (comenzando desde la raíz) o relativas (comenzando desde el nodo actual).

Ejemplo de una ruta absoluta: `/catalog/book/title` (selecciona el nodo `<title>` dentro de `<book>` en el nivel raíz `<catalog>`).

Ejemplo de una ruta relativa: `book/title` (selecciona todos los nodos `<title>` que son hijos de cualquier nodo `<book>`).

2. Predicados:

Los predicados en XPath se usan para filtrar nodos seleccionados en función de ciertas condiciones.

Ejemplo: `//book[price>30]` selecciona todos los nodos `<book>` cuyo hijo `<price>` tiene un valor mayor a 30.

Ejemplo: `//book[@category='fiction']` selecciona todos los nodos `<book>` que tienen un atributo `category` igual a "fiction".

XPath

XPath (XML Path Language) es un lenguaje utilizado para navegar y seleccionar nodos en un documento XML. Es una herramienta poderosa y flexible que permite extraer, manipular y analizar datos en XML.

3. Ejes en XPath:

Los ejes en XPath definen la relación entre los nodos en el documento XML.

child: Selecciona los hijos directos del nodo actual.

parent: Selecciona el nodo padre del nodo actual.

ancestor: Selecciona todos los ancestros del nodo actual.

descendant: Selecciona todos los descendientes (hijos, nietos, etc.) del nodo actual.

following-sibling: Selecciona todos los nodos hermanos siguientes al nodo actual.

preceding-sibling: Selecciona todos los nodos hermanos anteriores al nodo actual.

Ejemplo: `/library/book/child::title` Seleccionar todos los <title> elementos hijos de los <book> en la biblioteca.

Ejemplo: `/library/book/title[text()='XML Basics']/parent::book` Seleccionar el nodo <book> que es el padre del <title> cuyo valor es "XML Basics".

XQuery

XQuery (XML Query Language) es un lenguaje diseñado para consultar, transformar y manipular datos almacenados en formato XML. Similar en propósito a SQL en bases de datos relacionales, XQuery permite realizar consultas complejas y obtener resultados específicos de documentos XML, lo que lo convierte en una herramienta esencial que utiliza XML como formato principal de datos.

Características Principales de XQuery

Lenguaje Declarativo: XQuery permite especificar qué datos se desean extraer sin tener que detallar cómo extraerlos. Esto lo hace similar a otros lenguajes de consulta como SQL.

Extensión de XPath: XQuery extiende el lenguaje XPath, lo que significa que todas las capacidades de XPath están disponibles en XQuery, junto con funciones adicionales para manipular datos.

Transformación de Datos: XQuery no solo extrae datos, sino que también puede transformar XML en otros formatos, como HTML, JSON, o incluso otro XML.

Manipulación de Secuencias: XQuery trata los resultados como secuencias de elementos, lo que permite realizar operaciones sobre conjuntos de datos, como ordenar, filtrar, y agrupar.

Soporte para Funciones y Variables: XQuery permite definir funciones y variables, lo que facilita la creación de consultas más complejas y reutilizables.

XQuery

Estructura Básica de XQuery

XQuery utiliza una estructura flexible y poderosa que suele incluir:

FLWOR Expressions: Estas son el núcleo de muchas consultas en XQuery y se componen de cinco cláusulas principales:

For: Itera sobre una secuencia de nodos.

Let: Define variables para almacenar valores.

Where: Filtra los resultados.

Order by: Ordena los resultados.

Return: Especifica el valor o los valores que se deben devolver.

Funciones Integradas: XQuery incluye una variedad de funciones integradas para manipular cadenas, realizar cálculos matemáticos, manejar fechas, etc.

XPath: La sintaxis de XPath se utiliza dentro de XQuery para seleccionar nodos específicos en un documento XML.

XQuery

1. Seleccionar Todos los Títulos de Libros

```
for $book in /library/book return $book/title
```

2. Filtrar Libros por Precio

```
for $book in /library/book where $book/price > 30 return $book/title
```

3. Ordenar Libros por Precio

```
for $book in /library/book order by $book/price return $book/title
```

4. Crear un Resumen en Texto Plano

```
for $book in /library/book return concat($book/title, ' by ', $book/author)
```

5. Transformar XML a HTML

```
for $book in /library/book return <div> <h2>{ $book/title }</h2> <p>Author: {  
$book/author }</p> <p>Price: ${ $book/price }</p> </div>
```


Motores de XML

BaseX

BaseX es un motor de base de datos XML de código abierto que se especializa en la gestión y consulta de grandes volúmenes de datos XML. Es conocido por su alta eficiencia en el manejo de documentos XML y ofrece soporte completo para las tecnologías relacionadas, como XPath, XQuery y XSLT

eXist-db

eXist-db es una base de datos XML nativa de código abierto que ofrece almacenamiento nativo y consulta de documentos XML. Es muy utilizada en proyectos que necesitan un manejo sofisticado de XML, incluyendo aplicaciones web que se basan en tecnologías XML.

Motores de XML

MarkLogic

MarkLogic es un motor de base de datos NoSQL comercial que ofrece un fuerte soporte para datos XML. Es conocido por su capacidad para manejar grandes volúmenes de datos y consultas complejas en XML, con características avanzadas para la seguridad, escalabilidad y replicación.

Oracle XML DB

Oracle XML DB es una característica de la base de datos Oracle que ofrece un repositorio nativo para almacenar y gestionar datos XML dentro de una base de datos relacional Oracle. Combina la robustez de Oracle con capacidades avanzadas de gestión de XML.

Caso de uso



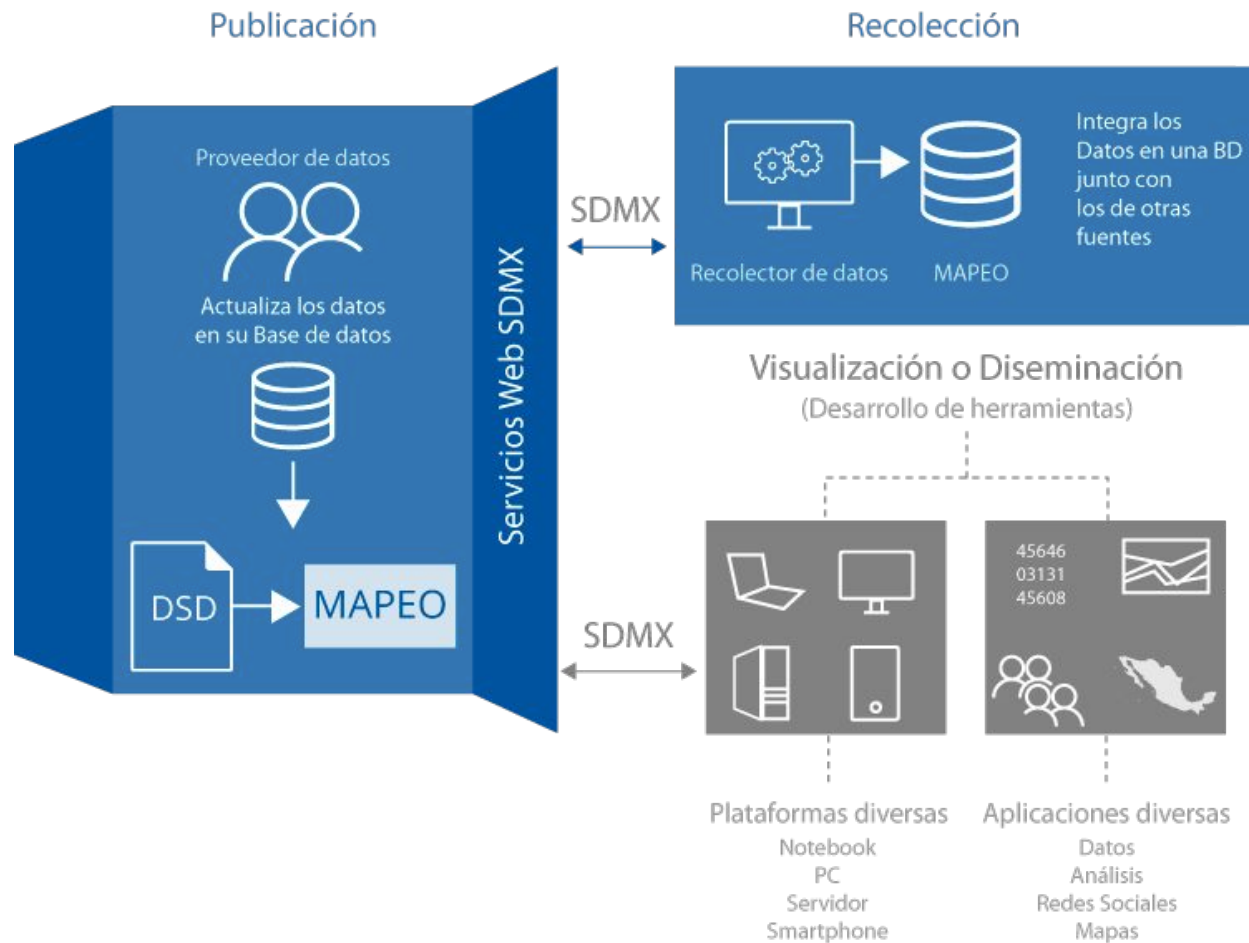
ISO International Standard (ISO 17369).

<https://sdmx.org/>

<https://sdmx.snieg.mx/home>

<https://registry.sdmx.org/>

Caso de uso



Cómputo en la nube

Cómputo en la nube en el tiempo

1961	1970	1990	1999	2002
<p>El concepto básico de Cómputo en la Nube (Cloud Computing) se le atribuye a John McCarthy, también responsable de introducir el término "Inteligencia Artificial". En 1961 fue el primero en sugerir que la tecnología de tiempo compartido (Time-Sharing) de las computadoras podría conducir a un futuro donde el poder del cómputo e incluso las aplicaciones podrían venderse como servicio similar a la electricidad o el agua, sentando las bases conceptuales para lo que hoy conocemos como cómputo en la nube</p>	<p>IBM introduce el sistema operativo VM, que permite ejecutar múltiples máquinas virtuales en un solo equipo físico. Este concepto de virtualización es fundamental para el desarrollo del cómputo en la nube.</p>	<p>El auge de internet permite la proliferación de los Proveedores de Servicios de Aplicaciones (ASPs), que ofrecen software hospedado en sus propios servidores, accesible de manera remota por los usuarios.</p>	<p>En 1999, Salesforce.com introdujo el concepto de entrega de aplicaciones empresariales a través de una sencilla página web.</p>	<p>En 2002, Amazon lanza Amazon Web Services. En 2006 llegó Google Docs, que realmente trajo el Cloud Computing a la conciencia del público. En ese año también se vio la introducción de Elastic Compute Cloud de Amazon (EC2) como un servicio web comercial que permitió a las empresas pequeñas y particulares alquilar equipos en los que pudieran ejecutar sus propias aplicaciones informáticas.</p>

Cómputo en la nube en el tiempo

2008	2009	2010	2011	2012
Google lanza Google App Engine, una plataforma como servicio (PaaS) que facilita a los desarrolladores la creación y despliegue de aplicaciones en la infraestructura de Google	En 2009, Microsoft lanza Windows Azure . Y en 2010 surgieron servicios en distintas capas de servicio: cliente, aplicación, plataforma, infraestructura y servidor.	Se lanza OpenStack , una plataforma de computación en la nube de código abierto que se convierte en un actor clave en el mercado de nubes privadas	En 2011, Apple lanzó su servicio iCloud , un sistema de almacenamiento en la Nube para documentos, música, vídeos, fotografías, aplicaciones y calendarios.	Google lanza Google Drive , reforzando el papel del almacenamiento en la nube en la vida cotidiana

Cómputo en la nube en el tiempo

2014	2015	2020	2022	Actualidad
AWS introduce Lambda , un servicio de computación basado en eventos que inicia la tendencia del "serverless computing", donde el código se ejecuta en respuesta a eventos sin necesidad de gestionar la infraestructura	Microsoft Azure supera a AWS en número de clientes, reflejando el crecimiento y la competencia en la industria de la nube.	La pandemia de COVID-19 acelera la adopción de servicios en la nube a medida que las empresas de todo el mundo migran al trabajo remoto, requiriendo soluciones escalables, accesibles y seguras.	Las tecnologías nativas de la nube, como los contenedores y Kubernetes , se vuelven estándar, permitiendo un despliegue de aplicaciones más flexible y escalable	Comienzan a tomar forma innovaciones como el cómputo en el borde (edge computing), servicios en la nube impulsados por inteligencia artificial , y el cómputo cuántico , con los proveedores de la nube expandiendo constantemente sus ofertas y capacidades

Datacenter

Un **centro de datos** es una infraestructura física que alberga servidores y otros componentes de TI críticos, incluyendo sistemas de almacenamiento, equipos de redes, y dispositivos de seguridad, con el fin de gestionar y almacenar datos, ejecutar aplicaciones, y ofrecer servicios a usuarios finales o a otras aplicaciones

Infraestructura Física:

Servidores: Máquinas físicas que ejecutan aplicaciones, almacenan datos y procesan información.

Almacenamiento: Sistemas de almacenamiento de datos (como NAS, SAN) que conservan grandes volúmenes de información.

Equipos de Red: Routers, switches, firewalls y otros dispositivos que gestionan el tráfico de datos dentro y fuera del datacenter.

Sistemas de Energía: Suministro eléctrico redundante, generadores de emergencia, y sistemas de alimentación ininterrumpida (UPS) para garantizar la continuidad operativa

Datacenter

Infraestructura Física:

Servidores: Máquinas físicas que ejecutan aplicaciones, almacenan datos y procesan información.

Almacenamiento: Sistemas de almacenamiento de datos (como NAS, SAN) que conservan grandes volúmenes de información.

Equipos de Red: Routers, switches, firewalls y otros dispositivos que gestionan el tráfico de datos dentro y fuera del datacenter.

Sistemas de Energía: Suministro eléctrico redundante, generadores de emergencia, y sistemas de alimentación ininterrumpida (UPS) para garantizar la continuidad operativa

Redundancia y Alta Disponibilidad:

Redundancia de Componentes: Los datacenters están diseñados con redundancia en todos los niveles (energía, almacenamiento, redes) para minimizar el riesgo de fallos.

Alta Disponibilidad: Asegura que los sistemas y servicios estén disponibles de manera continua, minimizando el tiempo de inactividad.

Seguridad Física y Lógica:

Seguridad Física: Controles de acceso, vigilancia 24/7, y medidas de protección contra incendios y desastres naturales.

Seguridad Lógica: Firewalls, sistemas de detección de intrusos, cifrado de datos y otras medidas para proteger la integridad y confidencialidad de la información.

Datacenter

Control de Clima y Ambiente:

Sistemas de Refrigeración: Mantienen la temperatura y la humedad adecuadas para evitar el sobrecalentamiento de los equipos.

Monitoreo Ambiental: Sensores y sistemas que monitorean las condiciones ambientales dentro del datacenter.

Escalabilidad:

Escalabilidad : Capacidad para agregar más recursos (servidores, almacenamiento, ancho de banda) a medida que aumentan las necesidades de la organización.

Virtualización: Uso de tecnologías de virtualización para maximizar el uso de los recursos físicos y mejorar la flexibilidad.

Automatización y Gestión:

Monitoreo en Tiempo Real: Sistemas para supervisar el rendimiento de los servidores, redes y almacenamiento en tiempo real.

Automatización de Procesos: Herramientas que permiten automatizar tareas repetitivas, como el aprovisionamiento de servidores y la gestión de redes.

Datacenter

Conectividad y Latencia:

Conectividad de Alta Velocidad: Conexiones de red de alta velocidad para asegurar la transmisión rápida de datos entre los sistemas internos y externos.

Latencia Baja: Optimización de la infraestructura para reducir el tiempo de respuesta en la comunicación de datos.

Resiliencia y Recuperación ante Desastres:

Planes de Continuidad del Negocio: Estrategias para garantizar la operación continua en caso de fallos o desastres.

Recuperación ante Desastres: Infraestructura y procesos para restaurar rápidamente las operaciones después de un evento adverso.

Eficiencia Energética

Cumplimiento Normativo

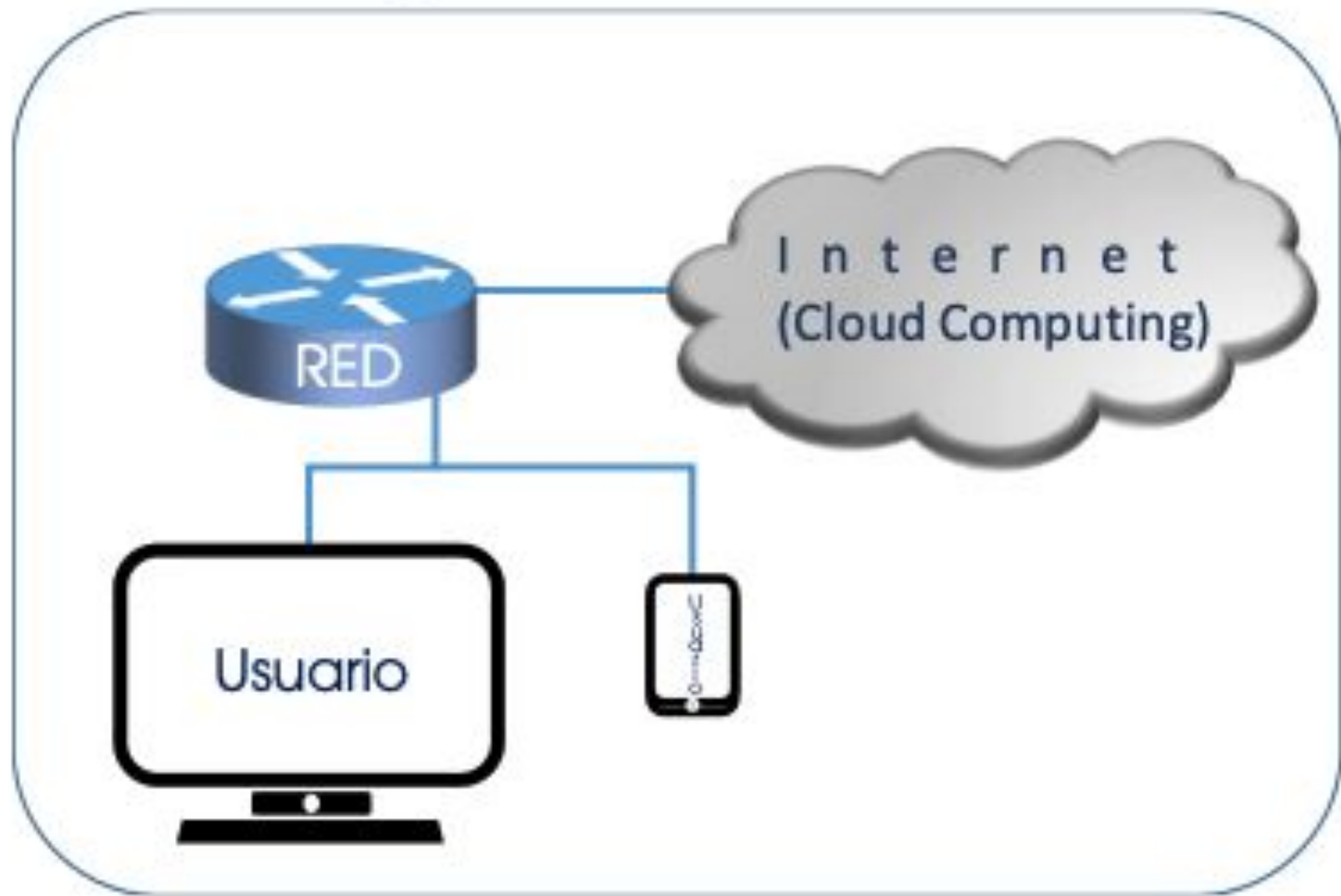
Definición

El **cómputo en la nube**, también conocido como cloud computing, es un paradigma de la informática que permite acceder a recursos y servicios a través de internet, sin necesidad de gestionarlos o mantenerlos localmente en dispositivos físicos propios. Esta tecnología ha revolucionado la forma en que las empresas y los usuarios acceden a software, almacenamiento y capacidades de procesamiento

La **arquitectura de nube** se refiere a cómo varios componentes de la tecnología de nube, como el hardware, los recursos virtuales, las capacidades del software y los sistemas de red virtual, interactúan y se conectan para crear entornos de computación en la nube. Actúa como un plano que define la mejor manera de combinar de manera estratégica los recursos a fin de crear un entorno de nube para una necesidad empresarial específica.

El **cloud computing** es un modelo para permitir el acceso ubicuo, conveniente y bajo demanda de la red a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar y lanzar rápidamente con un mínimo esfuerzo de gestión o Interacción del proveedor de servicios.

El **cómputo en la nube** es un paradigma para establecer el acceso de red a una fuente de recursos físicos o virtuales de una forma escalable y elástica con auto aprovisionamiento y administración bajo demanda



Características

Autoservicio: Los usuarios pueden acceder a recursos computacionales, como almacenamiento, capacidad de procesamiento y redes, según lo necesiten, sin intervención humana directa del proveedor de servicios.

Acceso ubicuo en la red: Los servicios en la nube están disponibles a través de internet, lo que permite el acceso desde cualquier lugar, en cualquier momento y desde cualquier dispositivo que tenga conexión a la red, como computadoras, teléfonos inteligentes o tabletas.

Agrupación de Recursos (Multitenancy): Los recursos computacionales se agrupan y comparten entre múltiples usuarios de manera eficiente. Estos recursos se asignan dinámicamente según la demanda, lo que permite a los usuarios aprovechar al máximo la capacidad disponible.

Escalamiento: La nube permite escalar los recursos de manera automática o manual, aumentando o disminuyendo según las necesidades del momento. Esto asegura que los recursos estén disponibles para cubrir la demanda, sin tener que pagar por más de lo necesario.

Características

Mantenimiento automatizado: El mantenimiento de la infraestructura, como actualizaciones de software, parches de seguridad y copias de seguridad, es gestionado automáticamente por el proveedor de servicios, lo que minimiza la carga administrativa para los usuarios.

Alta Disponibilidad y Resiliencia: Los servicios en la nube están diseñados para ser altamente disponibles, con redundancia y recuperación ante desastres integradas. Esto asegura que las aplicaciones y los datos sean accesibles incluso en caso de fallos en el hardware o en el centro de datos.

Seguridad: Los proveedores de servicios en la nube implementan avanzadas medidas de seguridad para proteger los datos y las aplicaciones, incluyendo cifrado, autenticación multifactor y firewalls. Sin embargo, la seguridad es una responsabilidad compartida entre el proveedor y el cliente.

Acceso ubicuo en la red: Las soluciones en la nube suelen ser compatibles con una amplia gama de tecnologías y permiten la integración de diferentes sistemas y servicios. Esto facilita la interoperabilidad y la adaptación a los requisitos específicos de los usuarios.

Beneficios

Rentabilidad: En lugar de invertir costos iniciales para los servidores, puedes optar por usar la infraestructura de un proveedor de servicios en la nube. El aprovisionamiento dinámico te permite optimizar aún más las inversiones, ya que pagas solo por los recursos de procesamiento que usas.

Tiempo más rápido de salida al mercado: Ya no es necesario esperar para adquirir, preparar y configurar la infraestructura de procesamiento. Las arquitecturas de nube te permiten ponerte en marcha con rapidez a fin de dedicar más tiempo al desarrollo y la entrega de productos nuevos.

Escalabilidad: Las arquitecturas de nube te brindan más flexibilidad para aumentar o disminuir la escala de los recursos de procesamiento según los requisitos de tu infraestructura. Puedes escalar con facilidad para satisfacer una demanda más alta, ya sea de crecimiento o aumentos repentinos de temporada en el tráfico.

Beneficios

Acceso Global y Movilidad: La nube permite el acceso a aplicaciones y datos desde cualquier lugar con conexión a internet.

Innovación: Las arquitecturas de nube te permiten aprovechar las tecnologías más recientes para el almacenamiento, la seguridad, la IA y las estadísticas, como el aprendizaje automático.

Continuidad del Negocio y Recuperación ante Desastres: La nube ofrece soluciones de backup y recuperación ante desastres que aseguran la continuidad del negocio en caso de fallos o desastres.

Transformación digital acelerada: Las arquitecturas nativas de la nube como Kubernetes te permiten aprovechar los servicios en la nube y los entornos automatizados al máximo para acelerar la modernización y generar transformación digital.



Riesgos

Seguridad de los Datos: La naturaleza del cloud computing implica que los datos se almacenan en servidores controlados por terceros, lo que puede presentar riesgos de seguridad.

Privacidad de los Datos: Los datos almacenados en la nube pueden ser accedidos o monitoreados por terceros, incluidos los proveedores de servicios en la nube o entidades gubernamentales.

Control de los Datos: Las organizaciones pueden perder cierto control sobre la gestión de sus datos y aplicaciones cuando los mueven a la nube.

Cumplimiento Normativo y Legal: Diferentes jurisdicciones tienen diferentes regulaciones sobre cómo deben ser manejados y protegidos los datos

Dependencia del Proveedor: Una vez que una empresa se compromete con un proveedor de servicios en la nube, puede ser difícil y costoso cambiar a otro proveedor.

Costos Ocultos: Aunque la nube suele ser más económica que mantener infraestructura propia, pueden surgir costos inesperados.

Riesgos

Mitigación de riesgos

- Evaluar cuidadosamente a los proveedores de servicios en la nube.
- Implementar medidas de seguridad robustas, como cifrado de datos y autenticación multifactor.
- Asegurarse de cumplir con las normativas y leyes aplicables.
- Establecer acuerdos claros de nivel de servicio (SLA) con los proveedores.
- Desarrollar planes de contingencia y recuperación ante desastres que incluyan alternativas en caso de fallo del proveedor.

Modelos de despliegue

Nube privada: La infraestructura de la nube es de uso exclusivo de una sola organización conformada por diversos consumidores. La nube puede pertenecer, ser operada y administrada por la organización, un tercero o una combinación de ambos, y puede existir dentro o fuera de las instalaciones.

Beneficios

- **Seguridad y Cumplimiento:** Mayor control sobre los datos y las políticas de seguridad, facilitando el cumplimiento de normativas.
- **Personalización y Control:** Configuración y gestión personalizada según las necesidades específicas de la organización.
- **Rendimiento y Confiabilidad:** Recursos dedicados que ofrecen un rendimiento más consistente y predecible.
- **Privacidad:** Aislamiento total de los datos y las aplicaciones de la organización, sin compartir recursos con terceros.

Desventajas

- **Costo:** Generalmente, una nube privada es más costosa de implementar y mantener que una nube pública, debido a la necesidad de adquirir y gestionar la infraestructura.
- **Escalabilidad Limitada:** La escalabilidad puede estar limitada por la capacidad de la infraestructura existente.
- **Gestión Compleja:** Requiere un equipo de TI capacitado para gestionar y mantener la infraestructura, especialmente en una nube privada interna.

Modelos de despliegue

Nube comunitaria: La infraestructura en la nube se proporciona para uso exclusivo de una comunidad específica de consumidores o de organizaciones que tienen requerimientos o propósitos comunes, la nube puede pertenecer, ser operada y administrada por una o más de las organizaciones de la comunidad, un tercero o alguna combinación de ellas, y puede existir dentro o fuera de las instalaciones.

Beneficios

- **Costo:** Compartir la infraestructura y los recursos entre varias organizaciones reduce los costos en comparación con una nube privada individual.
- **Seguridad y Cumplimiento Específico:** Las nubes comunitarias pueden diseñarse para cumplir con regulaciones y estándares específicos del sector, proporcionando un entorno más seguro y conforme a las normativas.
- **Colaboración Facilitada:** Facilita la colaboración entre organizaciones que tienen objetivos o necesidades comunes, como en sectores como la salud, la educación, o la administración pública.
- **Control Compartido:** Ofrece un nivel de control y personalización más alto que una nube pública, sin los costos y la complejidad de una nube privada total.

Modelos de despliegue

Nube comunitaria: La infraestructura en la nube se proporciona para uso exclusivo de una comunidad específica de consumidores o de organizaciones que tienen requerimientos o propósitos comunes, la nube puede pertenecer, ser operada y administrada por una o más de las organizaciones de la comunidad, un tercero o alguna combinación de ellas, y puede existir dentro o fuera de las instalaciones.

Desventajas

- **Compromisos de Gestión:** Dado que la gobernanza es compartida, puede haber desacuerdos entre las organizaciones participantes sobre la gestión y las políticas.
- **Escalabilidad Limitada:** La escalabilidad puede estar limitada por la capacidad de la infraestructura compartida y las necesidades diversas de las organizaciones.
- **Riesgos de Seguridad Compartida:** Aunque la nube es segura, el hecho de compartir la infraestructura con otras organizaciones podría aumentar los riesgos si no se gestionan adecuadamente.

Modelos de despliegue

Nube pública: La infraestructura de la nube es de uso abierto al público en general.

La nube puede pertenecer, ser operada y administrada por una organización comercial, académica o gubernamental, o una combinación de ellas. Existe en las instalaciones del proveedor de servicios de la nube.

Beneficios

- **Costos:** Al compartir la infraestructura entre múltiples clientes, los costos se reducen significativamente, permitiendo a las empresas pagar solo por lo que usan.
- **Escalabilidad y Flexibilidad:** Las organizaciones pueden escalar rápidamente sus recursos de acuerdo con la demanda, lo que es ideal para cargas de trabajo variables.
- **Acceso a Tecnologías Avanzadas:** Las empresas pueden acceder a las últimas tecnologías sin necesidad de invertir en su desarrollo o mantenimiento.
- **Fiabilidad y Recuperación ante Desastres:** Los proveedores garantizan altos niveles de disponibilidad y planes de recuperación ante desastres, lo que protege contra interrupciones del servicio.
- **Rapidez en la Implementación:** Las empresas pueden desplegar nuevas aplicaciones y servicios rápidamente, lo que acelera el tiempo de comercialización.

Modelos de despliegue

Nube pública: La infraestructura de la nube es de uso abierto al público en general.

La nube puede pertenecer, ser operada y administrada por una organización comercial, académica o gubernamental, o una combinación de ellas. Existe en las instalaciones del proveedor de servicios de la nube.

Desventajas

- **Menor Control y Personalización:** Las organizaciones tienen menos control sobre la infraestructura subyacente y las políticas de seguridad en comparación con una nube privada.
- **Riesgos de Seguridad:** Aunque los proveedores implementan medidas de seguridad avanzadas, los datos están almacenados en una infraestructura compartida, lo que puede ser una preocupación para algunas organizaciones.
- **Dependencia del Proveedor:** Las organizaciones dependen del proveedor para la disponibilidad y rendimiento del servicio, lo que puede generar riesgos en caso de interrupciones del servicio.
- **Costos:** Para cargas de trabajo constantes y predecibles, los costos a largo plazo pueden ser más altos que la inversión en infraestructura local o en una nube privada.

Modelos de despliegue

Nube híbrida: La infraestructura de la nube es una combinación de dos o más infraestructuras de la nube (privada, comunitaria o pública) que se mantienen como entidades únicas, pero que están unidas por tecnología estandarizada o patentada que permite la portabilidad de datos y aplicaciones

Beneficios

- **Agilidad Empresarial:** Las empresas pueden adaptarse rápidamente a las demandas cambiantes del mercado y a las necesidades de TI, utilizando el entorno más adecuado para cada tarea.
- **Optimización de Recursos:** Permite a las organizaciones usar la nube pública para tareas temporales o variables, mientras que las operaciones críticas y constantes permanecen en la nube privada.
- **Mayor Seguridad:** Ofrece un enfoque equilibrado donde las partes más sensibles de las operaciones se mantienen seguras y controladas, mientras que otras se benefician de la escalabilidad de la nube pública.
- **Mejora del Rendimiento:** Las aplicaciones y datos pueden distribuirse de manera óptima, utilizando la nube privada para aplicaciones de alto rendimiento y la nube pública para tareas menos críticas o que requieren alta escalabilidad.

Modelos de despliegue

Nube híbrida: La infraestructura de la nube es una combinación de dos o más infraestructuras de la nube (privada, comunitaria o pública) que se mantienen como entidades únicas, pero que están unidas por tecnología estandarizada o patentada que permite la portabilidad de datos y aplicaciones

Desventajas

- **Complejidad en la Gestión:** Gestionar una infraestructura híbrida puede ser más complejo, ya que requiere integrar y coordinar ambos entornos de manera eficiente.
- **Costo:** Inicialmente, la implementación de una nube híbrida puede ser costosa debido a la necesidad de integrar dos infraestructuras diferentes.
- **Desafíos de Seguridad:** Aunque la nube híbrida ofrece mayores controles, el hecho de mover datos entre nubes públicas y privadas puede presentar riesgos de seguridad si no se gestionan adecuadamente.
- **Dependencia de la Conectividad:** La comunicación constante entre la nube privada y la pública requiere una conectividad sólida y confiable, lo que puede ser un desafío en algunas situaciones.

Bases de datos para la toma de decisiones

Bodegas de datos (Datawarehouse)

Datawarehouse: Un almacén de datos es un sistema utilizado para la recopilación, almacenamiento y análisis de grandes volúmenes de datos provenientes de diversas fuentes. Su diseño permite la integración y organización eficiente de datos históricos, facilitando así la toma de decisiones informadas en las organizaciones. A diferencia de las bases de datos tradicionales, un almacén de datos está optimizado para consultas complejas y análisis multidimensionales, lo que lo convierte en una herramienta clave en el ámbito de la inteligencia empresarial.

Ventajas

Mejora la toma de decisiones

Permite análisis más profundos

Integra datos de diversas fuentes

Facilita la identificación de tendencias

Optimiza el rendimiento organizacional

Apoya la planificación estratégica

Desventajas

Costos de implementación elevados

Requiere mantenimiento constante

Puede ser complejo de gestionar

Dependencia de tecnología

Riesgo de obsolescencia

Necesidad de capacitación especializada

Beneficios datawarehouse

Acceso a Información Integrada

La implementación de un almacén de datos permite a las organizaciones acceder a información consolidada y actualizada de diversas fuentes, lo que facilita la identificación de patrones y tendencias, mejorando así la calidad y rapidez en la toma de decisiones estratégicas.

Mejora en la Toma de Decisiones

Análisis de datos históricos

El análisis de datos históricos en un almacén de datos permite a las organizaciones identificar tendencias a largo plazo, evaluar el rendimiento pasado y predecir comportamientos futuros, lo que es crucial para la planificación estratégica y la toma de decisiones informadas.

Soporte para informes y visualización de datos

Un almacén de datos proporciona una base sólida para la generación de informes detallados y visualizaciones efectivas, permitiendo a los usuarios extraer resúmenes significativos a partir de grandes volúmenes de datos, lo que mejora la comprensión y comunicación de la información dentro de la organización.

Facilita la explotación de los datos

Componentes clave

Fuentes de datos: Las fuentes de datos son esenciales, ya que alimentan el almacén con información proveniente de sistemas operativos, bases de datos externas y aplicaciones, garantizando una integración completa y precisa, como APIs y archivos planos.

La diversidad de fuentes enriquece el almacén de datos y permite un análisis más completo.

ETL (Extracción, Transformación y Carga): Este proceso es fundamental para la preparación de datos, donde se extraen, transforman y cargan los datos en el almacén, asegurando que sean consistentes y útiles para el análisis.

Modelo de datos: Un modelo de datos bien diseñado es crucial para estructurar la información en el almacén, facilitando el acceso y análisis eficiente mediante esquemas como estrella o copo de nieve.

Calidad de datos: La calidad de los datos es crucial en el proceso de extracción. Datos inexactos o incompletos pueden llevar a decisiones erróneas, por lo que se deben implementar controles y validaciones durante el proceso ETL.

Componentes clave

Estrategias de almacenamiento: La organización de datos en un almacén se basa en estrategias como la normalización y desnormalización, que optimizan el acceso y la consulta.

Estas técnicas permiten estructurar los datos de manera eficiente, facilitando su recuperación y análisis en tiempo real.

Clasificación de los datos: La clasificación de datos es esencial para una gestión efectiva. Se utilizan categorías y etiquetas que permiten agrupar información similar, mejorando la organización y facilitando el acceso a los usuarios, lo que resulta en un análisis más ágil y preciso.

Diseño: Cuando una organización se propone diseñar un almacén de datos, debe comenzar por definir sus requisitos comerciales específicos, acordar el alcance y preparar un diseño conceptual.

Cualquier diseño de data warehouse debe incluir los siguientes conceptos:

- *Contenido específico de datos.*
- *Relaciones dentro de los grupos de datos y entre ellos.*
- *El entorno de sistemas que dará soporte al data warehouse.*
- *Los tipos de transformaciones de datos necesarios.*
- *La frecuencia de actualización de los datos.*

Mercados de datos (Data Mart).

Data Mart: Es una estructura de almacenamiento de datos que se centra en un área específica de negocio, permitiendo a los usuarios acceder y analizar información relevante de manera más eficiente. A diferencia de un datawarehouse, que abarca toda la organización, un datamart está diseñado para satisfacer las necesidades de un departamento o función particular, facilitando el análisis y la toma de decisiones informadas.

Ventajas: Enfoque específico, permite análisis más rápido, menor costo de implementación, fácil de usar, integración sencilla, mejor rendimiento en consultas.

Desventajas: Alcance limitado, datos no centralizados, mantenimiento adicional requerido, posible redundancia de datos, menos escalabilidad, puede generar silos de información.

Beneficios datamart

Acceso a datos específicos

Los datamarts permiten a las organizaciones acceder a datos específicos y relevantes de manera rápida, lo que mejora la capacidad de los analistas y gerentes para tomar decisiones informadas basadas en información actualizada y precisa.

Facilita en la Toma de Decisiones

Optimizados

Al estar diseñados para áreas específicas, los datamarts reducen la complejidad de las consultas y mejoran el rendimiento analítico, permitiendo a los usuarios realizar análisis más profundos sin afectar el rendimiento general del sistema de datos.

Soporte para informes y visualización de datos

Un almacén de datos proporciona una base sólida para la generación de informes detallados y visualizaciones efectivas, permitiendo a los usuarios extraer resúmenes significativos a partir de grandes volúmenes de datos, lo que mejora la comprensión y comunicación de la información dentro de la organización.

Facilita la explotación de los datos

Procesamiento y análisis en línea (OLAP)

OLAP: Significa Procesamiento Analítico en Línea, es una tecnología que permite a los usuarios realizar análisis complejos de grandes volúmenes de datos multidimensionales. Facilita la toma de decisiones al proporcionar herramientas para explorar y analizar datos desde diferentes perspectivas, permitiendo a las organizaciones identificar tendencias, patrones y relaciones significativas en sus datos.

Ventajas: Análisis multidimensional eficiente, permite explorar datos desde múltiples ángulos, Mejora la toma de decisiones, proporciona información relevante y oportuna, Acelera el procesamiento de consultas, optimiza el tiempo de respuesta, Facilita la identificación de tendencias, ayuda a descubrir patrones ocultos en los datos.

Desventajas: Costos de implementación elevados, requiere inversión en infraestructura tecnológica, Complejidad en la configuración inicial, puede ser difícil de integrar con sistemas existentes, Necesita capacitación especializada, demanda habilidades técnicas avanzadas, Limitaciones en el manejo de datos no estructurados, puede no ser adecuado para todos los tipos de análisis.

Diferencias entre OLAP y OLTP

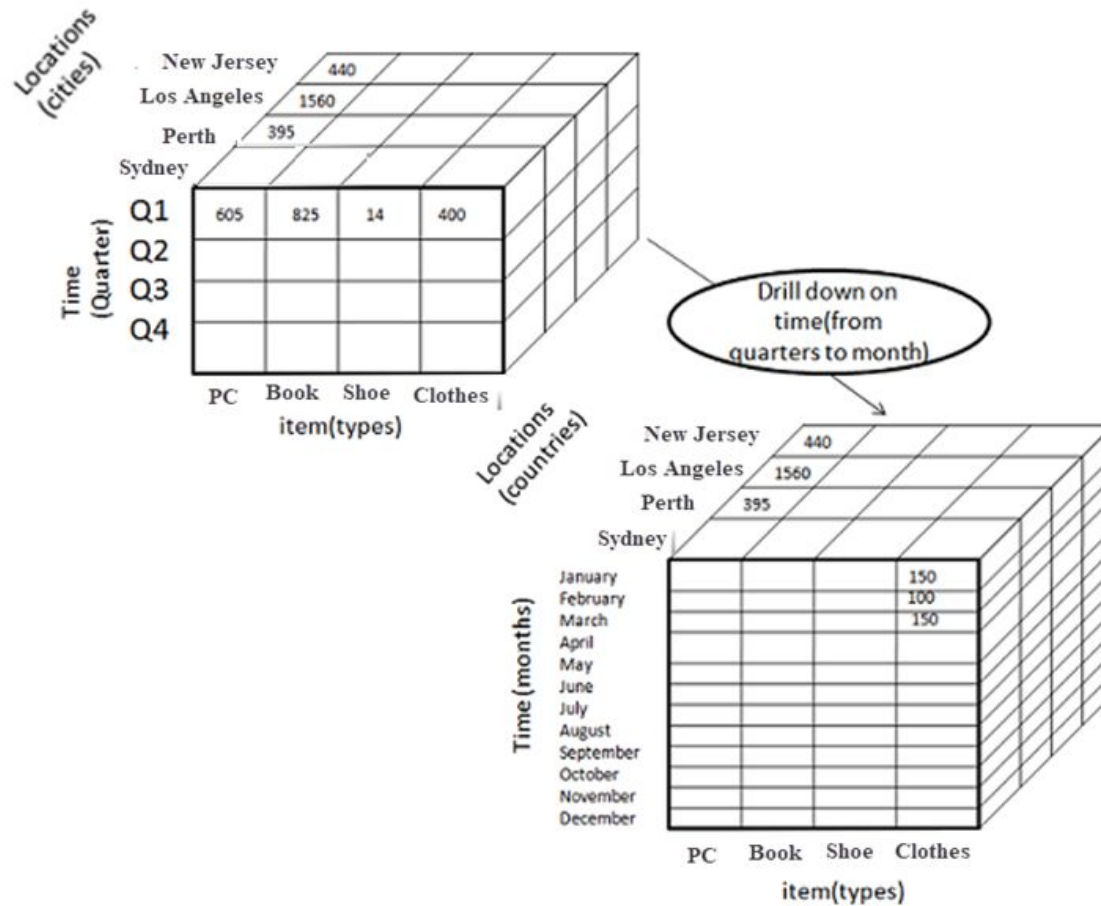
Uso: OLAP (Procesamiento Analítico en Línea) se utiliza para análisis de datos complejos y toma de decisiones estratégicas, mientras que OLTP (Procesamiento de Transacciones en Línea) se centra en la gestión de transacciones diarias y operaciones comerciales.

Estructura de datos: OLAP organiza datos en estructuras multidimensionales, facilitando el análisis, mientras que OLTP utiliza bases de datos relacionales optimizadas para transacciones rápidas y eficientes.

Rendimiento y consultas: OLAP permite consultas complejas y análisis históricos, lo que puede llevar más tiempo, mientras que OLTP está diseñado para respuestas rápidas a consultas simples y transacciones en tiempo real.

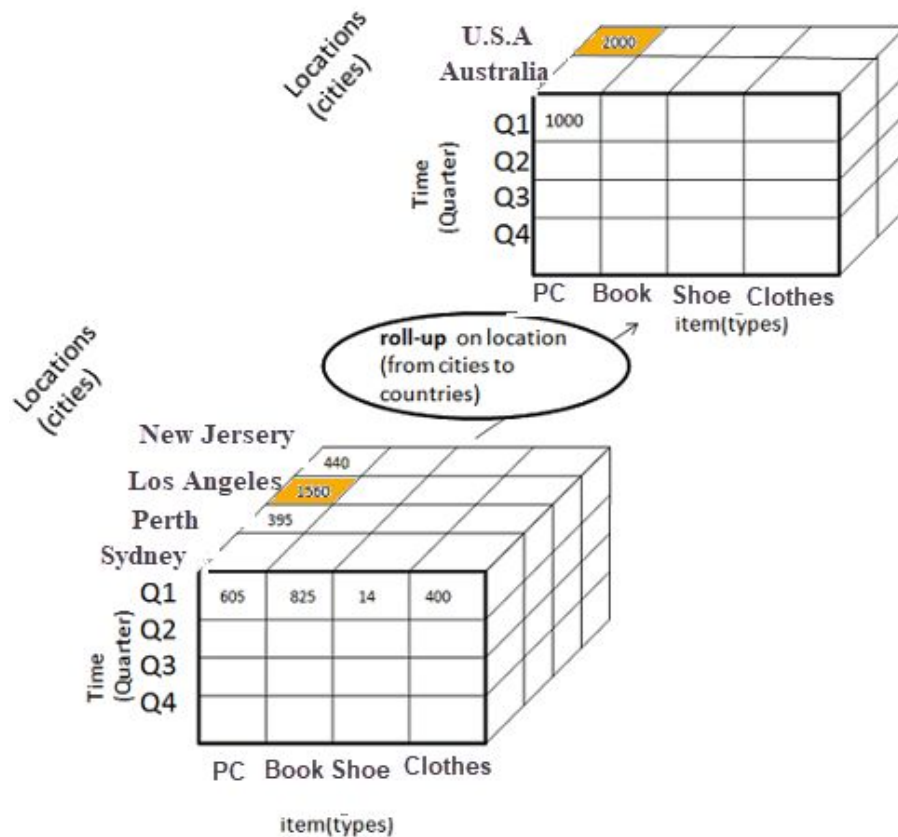
Operaciones OLAP

Drill down: Permite moverse de un alto nivel a un nivel más detallado



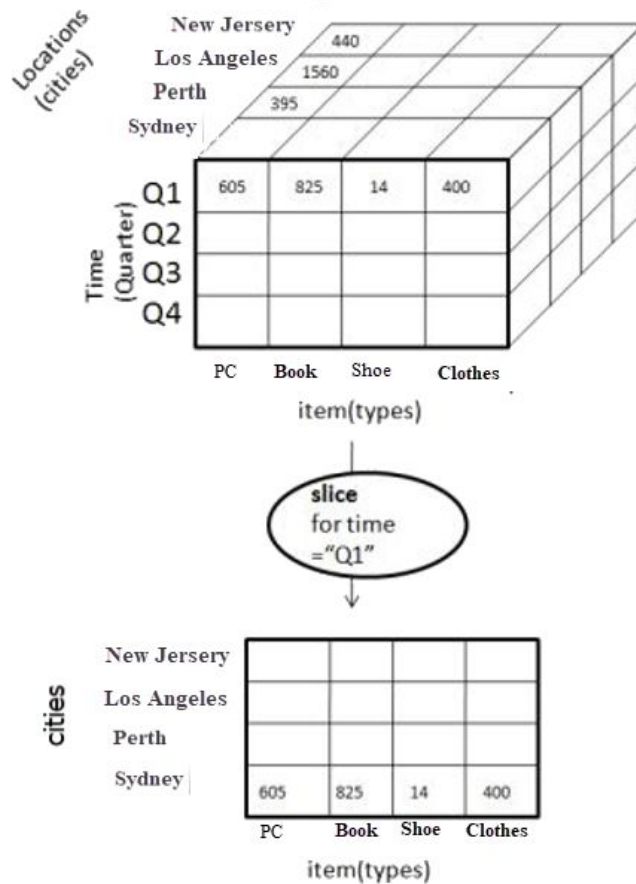
Operaciones OLAP

Roll up: Permite realizar agregaciones de la información



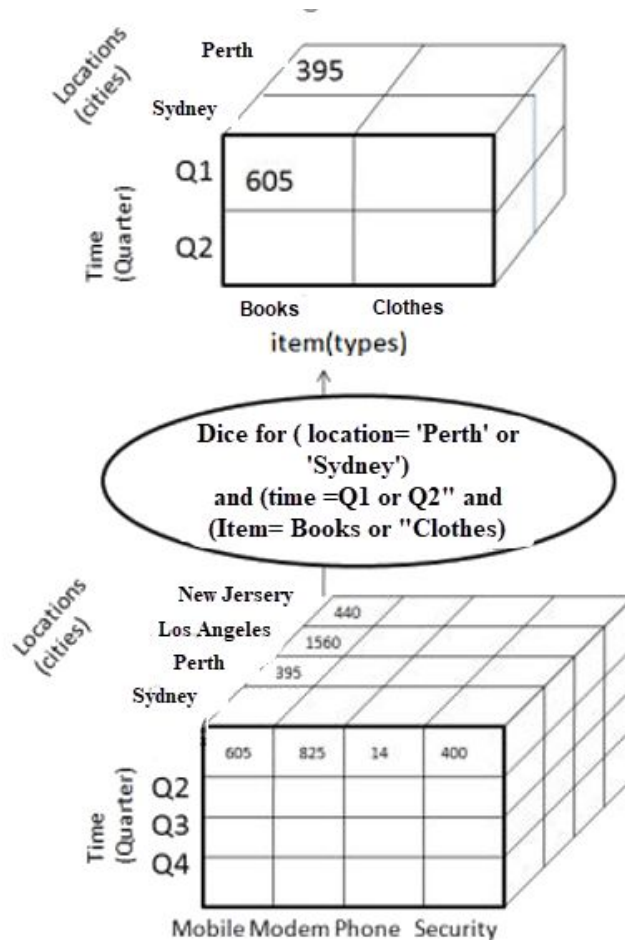
Operaciones OLAP

Slice: Divide una dimension en otra vista



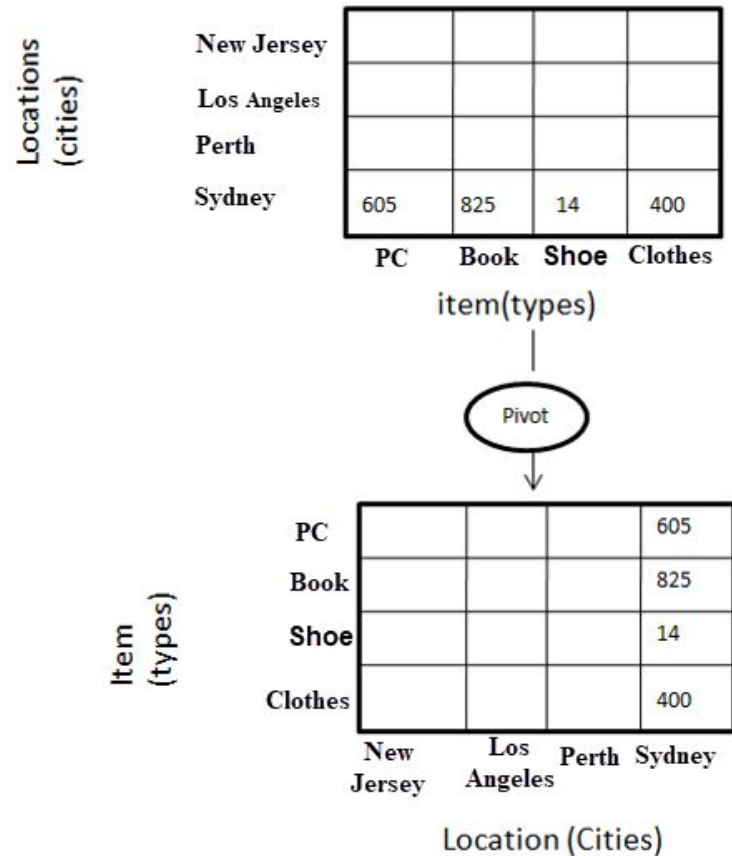
Operaciones OLAP

Dice: Divide en más de una dimension en otra vista, lo que produce un cubo de menor dimensión.



Operaciones OLAP

Pivot: Permite rotar el cubo entre sus dimensiones.



Ejemplos de consultas OLAP

Consulta de ventas por región: Esta consulta permite analizar las ventas totales en diferentes regiones geográficas, facilitando la identificación de áreas con mejor rendimiento y aquellas que requieren atención adicional para mejorar las ventas.

Análisis de tendencias temporales: A través de esta consulta, se pueden observar las variaciones en las ventas a lo largo del tiempo, permitiendo a los analistas detectar patrones estacionales y prever futuras demandas en función de datos históricos.

Comparación de productos: Esta consulta permite evaluar el rendimiento de diferentes productos dentro de una misma categoría, ayudando a las empresas a identificar cuáles son los más rentables y cuáles necesitan estrategias de marketing o mejoras.

Minería de datos (Data mining).

Data mining: La minería de datos, también conocida como data mining, es un proceso de análisis de grandes volúmenes de datos con el objetivo de descubrir patrones, tendencias y relaciones ocultas que no son evidentes a simple vista. Este proceso se lleva a cabo utilizando métodos de inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos

Valor en la toma de decisiones: La minería de datos es crucial en el análisis de datos porque permite identificar tendencias y patrones ocultos que pueden influir en la toma de decisiones estratégicas, optimizando recursos y mejorando la eficiencia operativa en diversas industrias.

Ventajas: Descubrimiento de información no esperada.

Análisis de grandes volúmenes de datos.

Resultados fáciles de interpretar.

Mejora en la atención al cliente y fidelización.

Aumento de ventas y nuevas oportunidades de negocio.

Reducción de costos

Desventajas: Requiere una elección cuidadosa de algoritmos.

Puede ser complejo y costoso implementar.

Riesgos de privacidad y seguridad de datos

Proceso de la minería de datos

El proceso de minería de datos generalmente incluye los siguientes pasos:

Definición del problema: Identificar los objetivos y el alcance del proyecto, trabajando conjuntamente con las partes interesadas de la empresa para determinar la información necesaria.

Recopilación de datos: Reunir los datos relevantes, que pueden provenir de diversas fuentes internas y externas.

Preparación de datos: Limpieza y organización de los datos para eliminar duplicados, registros incompletos o formatos antiguos.

Análisis de datos: Utilización de algoritmos y técnicas de análisis para descubrir patrones y relaciones ocultas.

Interpretación y Visualización de resultados: Presentar los hallazgos de manera que sean comprensibles y útiles para la toma de decisiones.

Aplicaciones

Sector Financiero: La minería de datos se utiliza para detectar fraudes, analizar riesgos crediticios y personalizar ofertas, mejorando la seguridad y la satisfacción del cliente en instituciones financieras.

Salud y Medicina: En el ámbito de la salud, permite el análisis de datos clínicos para predecir brotes de enfermedades, optimizar tratamientos y mejorar la gestión hospitalaria mediante la identificación de patrones en los datos de pacientes.

Marketing y Ventas: Las empresas aplican minería de datos para segmentar mercados, predecir comportamientos de compra y personalizar campañas publicitarias, lo que resulta en un aumento significativo en la efectividad de sus estrategias comerciales.

Métodos y Técnicas Utilizadas

Clasificación de datos: Los algoritmos de clasificación, como árboles de decisión y máquinas de soporte vectorial, permiten categorizar datos en clases predefinidas, facilitando la toma de decisiones basadas en patrones aprendidos a partir de datos históricos.

Agrupamiento efectivo: Los métodos de agrupamiento, como k-means y jerárquico, agrupan datos similares sin etiquetas previas, ayudando a identificar segmentos naturales dentro de los datos y revelando estructuras ocultas.

Regresión predictiva: Los algoritmos de regresión, como la regresión lineal y polinómica, se utilizan para modelar relaciones entre variables y hacer predicciones sobre resultados futuros, siendo esenciales en análisis de tendencias y pronósticos.

Big data

Dimensiones del big data

Big Data se refiere a la capacidad de gestionar y analizar grandes volúmenes de datos provenientes de diversas fuentes, permitiendo la extracción de información valiosa a través de técnicas avanzadas como el aprendizaje automático y la inteligencia artificial, lo que facilita la toma de decisiones informadas y la innovación en múltiples sectores.

Volumen

El volumen de datos generados en la actualidad es inmenso, lo que requiere soluciones de almacenamiento y procesamiento que puedan manejar eficientemente esta gran cantidad de información.

Velocidad

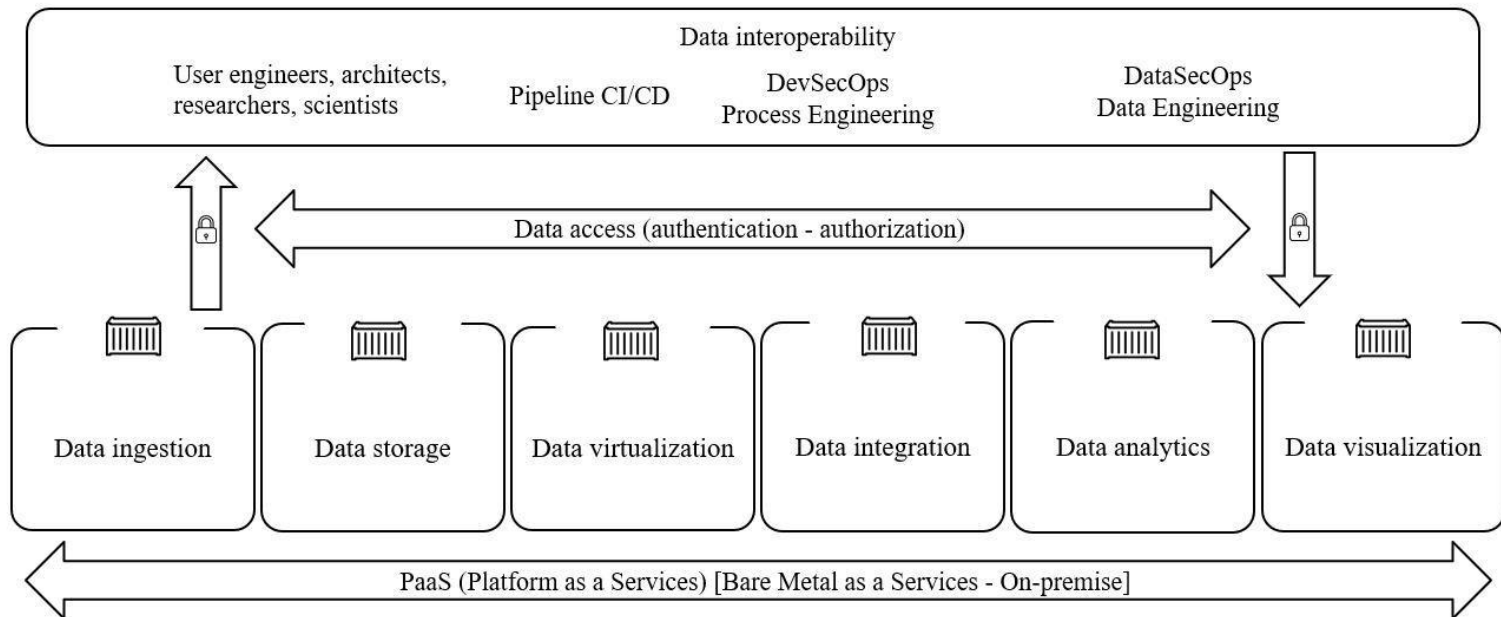
La velocidad de generación de datos es crucial, ya que las organizaciones deben procesar información en tiempo real para responder rápidamente a cambios y oportunidades del mercado.

Variedad

La variedad de datos, que incluye formatos estructurados y no estructurados, exige herramientas analíticas avanzadas para integrar y extraer insights significativos de múltiples fuentes.

Tecnologías de Big Data

Las tecnologías de Big Data abarcan un conjunto amplio de herramientas, plataformas y frameworks diseñados para recopilar, procesar, almacenar y analizar grandes volúmenes de datos (estructurados, no estructurados y semiestructurados).



Tecnologías de Big Data

1. Almacenamiento de Datos

1.1. Almacenamiento distribuido

- **Hadoop Distributed File System (HDFS):** Sistema de archivos distribuido diseñado para ejecutarse en hardware común y manejar datos masivos.
- **Amazon S3:** Solución de almacenamiento en la nube ampliamente usada en Big Data.
- **Google Cloud Storage:** Servicio de almacenamiento altamente escalable.
- **MinIO:** Almacenamiento de objetos compatible con S3 para entornos locales.

1.2. Bases de datos NoSQL

- **MongoDB:** Base de datos NoSQL orientada a documentos.
- **Cassandra:** Base de datos distribuida ideal para alta disponibilidad y escalabilidad.
- **HBase:** Base de datos basada en Hadoop, diseñada para Big Data.
- **Redis:** Base de datos en memoria para cargas rápidas y en tiempo real.

Tecnologías de Big Data

2. Procesamiento de Datos

2.1. Procesamiento Batch

- **Apache Hadoop:** Ecosistema para procesamiento distribuido por lotes.
- **Apache Spark:** Framework de procesamiento en memoria para cargas batch y en tiempo real.
- **Apache Flink:** Plataforma para procesamiento de datos a gran escala.

2.2. Procesamiento en tiempo real

- **Apache Kafka:** Plataforma para transmisión de datos y sistemas de mensajería.
- **Apache Storm:** Framework para procesamiento distribuido en tiempo real.
- **Apache Flink:** Procesamiento tanto batch como en streaming.

3. Herramientas de Consulta y Analítica

- **Apache Hive:** Motor de consulta SQL sobre Hadoop.
- **Presto:** Motor de consulta distribuido para datos a gran escala.
- **Apache Drill:** Herramienta para ejecutar consultas SQL en datos heterogéneos.
- **Druid:** Plataforma para análisis de datos OLAP en tiempo real.

Tecnologías de Big Data

4. Ingeniería de Datos

- **Apache Airflow**: Orquestador de flujos de trabajo para procesamiento de datos.
- **Gitlab**: Herramienta de apoyo para pipelines de procesamiento de datos.
- **NiFi**: Plataforma de integración de datos basada en flujo de Apache.

5. Machine Learning e IA en Big Data

5.1. Plataformas de ML

- **TensorFlow**: Biblioteca para machine learning y deep learning.
- **PyTorch**: Framework enfocado en investigación y producción de ML.
- **Apache Mahout**: Herramienta para construir algoritmos de machine learning a gran escala.
- **MLflow**: es una plataforma de código abierto diseñada para administrar el ciclo de vida completo de un modelo de aprendizaje automático (Machine Learning).

5.2. Entornos distribuidos

- **MLlib**: Librería de Spark para machine learning.
- **H2O.ai**: Plataforma para IA y aprendizaje automático distribuido.

Tecnologías de Big Data

6. Visualización de Datos

- **Tableau:** Herramienta avanzada de análisis y visualización.
- **Power BI:** Plataforma de análisis e informes de Microsoft.
- **Grafana:** Panel de control para monitoreo y visualización de datos en tiempo real.
- **Apache Superset:** Herramienta de visualización y exploración de datos.

7. Seguridad y Gobernanza

- **Apache Ranger:** Control de acceso y seguridad en Big Data.
- **Apache Atlas:** Gestión de metadatos y gobernanza de datos.
- **AWS Lake Formation:** Herramienta para crear y gestionar lagos de datos.

Importancia del Big Data

Optimización empresarial

El Big Data permite a las empresas identificar patrones y tendencias, lo que se traduce en una mejora en la toma de decisiones estratégicas y un aumento en la eficiencia operativa, impulsando así la competitividad en el mercado.

Avances en salud

La integración de Big Data en el sector salud facilita la personalización de tratamientos y la mejora en la gestión de enfermedades, permitiendo a los profesionales de la salud ofrecer cuidados más precisos y basados en datos concretos.

Políticas públicas efectivas

Los gobiernos utilizan el análisis de Big Data para entender mejor las necesidades de la población, lo que les permite diseñar políticas más efectivas y responder de manera ágil a situaciones críticas, mejorando la calidad de vida de los ciudadanos.

Casos de uso

Análisis en tiempo real de transacciones financieras para detección de fraude

Descripción del problema

Un banco necesita detectar transacciones fraudulentas en tiempo real para proteger a sus clientes y prevenir pérdidas. El sistema debe procesar datos de múltiples fuentes, aplicar modelos de machine learning, y generar alertas instantáneas en caso de anomalías.

Ingestión de datos

Los datos se recogen de múltiples fuentes:

- Fuentes de entrada:
 - Terminales de punto de venta (POS).
 - Aplicaciones móviles del banco.
 - Registros de actividades de cuentas.

Tecnología:

- Apache Kafka: Se utiliza para capturar y transmitir datos en tiempo real desde las fuentes. Cada tipo de evento (transacciones, geolocalización, historial de usuario) se gestiona en tópicos separados.

Casos de uso

Almacenamiento de datos

- Datos en tiempo real (eventos recientes):
 - Apache HBase: Para almacenar eventos recientes que requieren consultas rápidas para validar patrones.
- Datos históricos:
 - Amazon S3 o Hadoop HDFS: Almacenamiento a largo plazo para análisis históricos y entrenamiento de modelos de machine learning.

Procesamiento de datos

- Streaming en tiempo real:
 - Apache Flink o Apache Spark Streaming: Procesa los eventos en tiempo real y realiza tareas como:
 - Unión de datos de múltiples fuentes.
 - Aplicación de reglas de negocio.
 - Extracción de características para modelos de predicción.
- Procesamiento batch:
 - Apache Spark: Se usa para análisis más complejos y entrenar modelos de detección de fraude usando datos históricos.

Casos de uso

Modelos de Machine Learning

- Entrenamiento del modelo:
 - Se entrena un modelo de clasificación (como árboles de decisión o redes neuronales) usando TensorFlow o Scikit-learn.
 - Datos de entrenamiento incluyen características como monto, ubicación, frecuencia de transacciones, y comportamiento histórico del usuario.
- Despliegue del modelo:
 - MLlib (de Apache Spark) : Para aplicar el modelo en tiempo real y marcar transacciones sospechosas.

Visualización y Alertas

- Generación de alertas:
 - Apache Kafka: Las transacciones sospechosas se publican en un tópico específico para ser consumidas por sistemas de alerta.
- Monitoreo y visualización:
 - Grafana o Tableau: Panel para visualizar métricas en tiempo real, como:
 - Transacciones procesadas por segundo.
 - Tasas de fraude detectadas.
 - Distribución geográfica de actividades sospechosas.

Casos de uso

Seguridad y Gobernanza

- Control de acceso:
 - Apache Ranger: Para gestionar permisos en bases de datos y flujos de trabajo.
- Metadatos y trazabilidad:
 - Apache Atlas: Para rastrear cómo se procesan los datos y qué transformaciones se aplican.

Casos de uso

Monitoreo y Análisis de Logs de Servidores para Detección de Anomalías

Descripción del problema

Una empresa de tecnología necesita analizar en tiempo real los logs generados por sus servidores y aplicaciones para identificar anomalías como errores frecuentes, accesos sospechosos o cambios en el rendimiento. El objetivo es mejorar la detección proactiva de problemas y garantizar la alta disponibilidad de los servicios.

Ingestión de datos

Los logs se recopilan desde diversas fuentes:

- Servidores de aplicaciones.
- Servidores web (Apache/Nginx).
- Bases de datos.
- Firewalls o sistemas de seguridad.

Tecnologías:

- Filebeat: Agente ligero para capturar logs y enviarlos al sistema central.
- Logstash: Procesa y transforma los logs para agregar etiquetas, filtrar datos irrelevantes y estructurar la información.

Casos de uso

Transmisión y almacenamiento de logs

- Apache Kafka: Actúa como una capa de mensajería para transmitir los logs en tiempo real.
- Elasticsearch: Para almacenar los logs procesados y permitir búsquedas rápidas y análisis avanzado.
- Amazon S3: Se utiliza para almacenar logs históricos y realizar análisis batch.

Procesamiento y análisis

- En tiempo real:
 - Apache Spark Streaming: Consume datos de Kafka para detectar patrones o anomalías en los logs. Ejemplo: Incrementos repentinos en errores HTTP 500 o múltiples intentos fallidos de inicio de sesión.
 - Librerías de Machine Learning: Como MLlib o Python para aplicar modelos que predicen fallos en sistemas o identifican patrones de ataque.
- Análisis por lotes:
 - Apache Hadoop: Procesa logs históricos almacenados en S3 para extraer tendencias a largo plazo, como picos de tráfico o uso anómalo de recursos.

Casos de uso

Visualización y alertas

- Visualización:
 - Kibana: Herramienta para visualizar datos almacenados en Elasticsearch. Permite crear paneles que muestren métricas clave como:
 - Frecuencia de errores en diferentes aplicaciones.
 - Distribución geográfica de accesos.
 - Uso de recursos del sistema.
- Alertas:
 - Grafana: Genera alertas basadas en datos de métricas de logs.

Seguridad y gobernanza

- Cifrado de logs:
 - Se utiliza TLS para asegurar la transmisión de datos desde Filebeat hasta Elasticsearch.
- Gestión de acceso:
 - Elasticsearch Security o Apache Ranger: Para gestionar permisos de acceso a los logs.



Universidad
de la Ciudad de
Aguascalientes

Mentes que transforman el mundo

ucags.edu.mx

📞 449 181 2621

📍 Jesús F Contreras #123, Aguascalientes, Mexico, 20070