

U8-Tema8.1-Introducción a bases de datos

Fundamentos de Computación en la Nube (1ºDAM/1ºDAW)

Álvaro García Sánchez



Introducción a las bases de datos	2
Bases de Datos	2
Modelo de datos	2
Sistemas Gestores de Bases de Datos (DBMS)	3
Tipos de bases de datos y sistemas gestores de bases de datos	4
Bases de Datos Relacionales	4
Bases de Datos NoSQL	4
Servicios de bases de datos no administrados y administrados	6
Servicios de bases de datos no administrados	6
Servicios de Bases de Datos Administrados	7
Referencias	8

Introducción a las bases de datos

Las **bases de datos** y su tecnología desempeñan un papel crucial en casi todas las áreas de aplicación de la informática y por lo tanto en la sociedad actual (educación, medicina, negocios, ingeniería, investigación, etc.).

En los siguientes apartados se explica el concepto de bases de datos y que implica (objetivos y características) **en el ámbito de la informática y en los sistemas de información actuales**.

Bases de Datos

Una **Base de Datos (Database)** es un conjunto organizado de información estructurada o no estructurada que permite almacenar, gestionar y recuperar datos de manera eficiente.

Modelo de datos

Un **modelo de datos** es una representación abstracta y estructurada de cómo se organizan, almacenan y gestionan los datos dentro de un sistema de información o base de datos. Su objetivo es definir la estructura lógica de los datos, las relaciones entre ellos y las reglas que se aplican para garantizar su integridad y consistencia.

Los modelos de datos se utilizan **para conceptualizar y diseñar bases de datos**, sirviendo como un puente entre los requisitos del negocio y la implementación técnica. Los principales tipos de modelos de datos son:

Inicialmente, los modelos de datos usados en los sistemas informáticos se basaban en gestionar directamente **sistemas de ficheros**.

Posteriormente se definieron los modelos clásicos:

- **Modelo jerárquicos (finales de los 60):** almacenan la información en forma de árbol y unidireccional, de arriba hacia abajo. La información se almacena en nodos, y los nodos se relacionan mediante punteros, permitiendo que un nodo padre pueda tener muchos nodos hijos (relaciones de uno a muchos).
- **Modelos en red (principios de los 70):** a partir de los modelos jerárquicos, los datos se relacionan en forma de red o grafo, permitiendo la relaciones de muchos a muchos, de manera que un nodo hijo puede tener varios nodos padre, y viceversa.
- **Modelo Relacional (finales de los 70 hasta hoy en día):** utiliza tablas para representar datos y relaciones. Basado en teoría de conjuntos y lógica de predicados. **Es el modelo más difundido y utilizado.**

Tras los clásicos, hubo un movimiento por el uso de modelos avanzados que no terminaron de cuajar:

- **Modelos orientados a objetos (década de los 80):** soportan el paradigma de programación orientada a objetos, permitiendo la creación de tipos de datos compuestos y herencia entre los diferentes tipos.

Con la explosión de internet, y la necesidad de trabajar con grandes conjuntos de datos, aparecieron y se establecieron en el mercado los modelos:

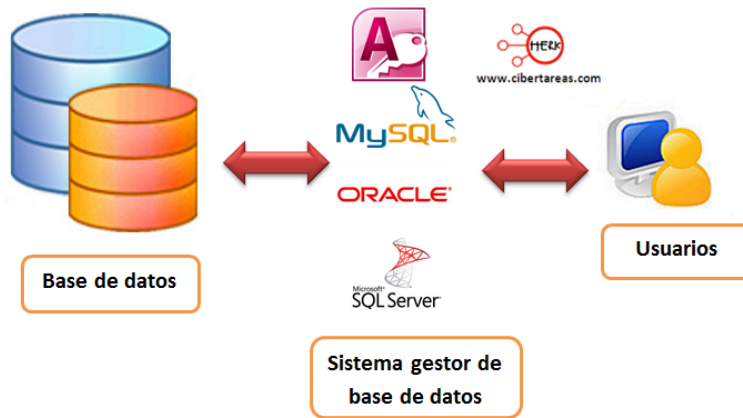
- **Modelos multidimensionales (década de los 90 y explosión en 2010 con el Big Data):** se utilizan para desarrollar aplicaciones de Data Warehouse de procesamiento analítico en línea (OLAP). Las tablas se estructuran como cubos de información, mediante una tabla de hechos y varias tablas de dimensiones (tiempo, lugar, etc..).
- **“Modelos” o soluciones NoSQL (década del 2000 hasta hoy en día):** como complemento a los modelos relacionales, ofrecen alternativas donde prima el rendimiento y escalabilidad de los datos. Formalmente, podemos decir que se trata de un conjunto de tecnologías que permiten el procesamiento rápido y eficiente de conjuntos de datos dando la mayor importancia al rendimiento, la fiabilidad y la agilidad.
- Destacan los modelos:
 - **Clave-Valor:** se almacenan datos a los cuales sólo se puede acceder mediante una clave.
 - **Documental:** utiliza estructuras de documentos JSON que contiene la información de varias entidades en un mismo documento
 - **Grafos:** Uso de nodos y vértices para navegar por los datos.
 - **Basado en columnas:** gira las tablas para almacenar los datos por columnas en vez de hacerlo por filas.

Los modelos de datos son esenciales para asegurar una estructura eficiente y coherente en cualquier sistema de gestión de datos.

Sistemas Gestores de Bases de Datos (DBMS)

Un **Sistema Gestor de Bases de Datos** (*DataBase Management System*, DBMS) es el software encargado de administrar y gestionar las bases de datos. El DBMS permite realizar operaciones como consultas, inserciones, actualizaciones y eliminaciones de datos, además de garantizar la seguridad, consistencia y disponibilidad de los datos.

Ejemplos: **Oracle, MySQL, MariaDB, SQL Server, PostgreSQL, OpenOffice/LibreOffice Base, Access, MongoDB, Cassandra, ...** . Son sistemas gestores de bases de datos, y no bases de datos en sí.



Fuente: Aitor Medrano. Especialización en Inteligencia Artificial y Big Data (IABD)

Tipos de bases de datos y sistemas gestores de bases de datos

Aunque no es una clasificación muy técnica es habitual hablar de.

Bases de Datos Relacionales

- **Base de Datos Relacional:** Colección de datos organizados en tablas con filas y columnas, según el **modelo relacional**.
- **DBMS Relacional:** Software que gestiona y controla las bases de datos relacionales mediante el uso de SQL (*Structured Query Language*).

Ejemplos de DBMS Relacionales: MySQL/MariaDB, PostgreSQL, Oracle Database, Microsoft SQL Server, ..

Bases de Datos NoSQL

- **Base de Datos NoSQL:** Han sido diseñadas para superar las limitaciones de las bases de datos relacionales para gestionar las demandas de los datos de estructura variable. Almacenan datos no estructurados o semiestructurados sin un esquema rígido. Se utilizan para manejar grandes volúmenes de datos distribuidos y dinámicos. Escalan horizontalmente.
- **DBMS NoSQL:** Software especializado para gestionar bases de datos NoSQL, ofreciendo flexibilidad y escalabilidad para datos heterogéneos.

Ejemplos de DBMS NoSQL:

- **Clave-Valor:** Los almacenes clave-valor son las bases de datos NoSQL más simples. Cada elemento de la base de datos se almacena con un nombre de atributo (o clave) junto a su valor, a modo de diccionario. Ejemplo: **Redis, Riak y AWS DynamoDB.**
- **Documentales:** Cada clave se asocia a una estructura compleja que se conoce como documento. Este puede contener diferentes pares clave-valor, o pares de clave-array o incluso documentos anidados, como en un documento JSON. Ejemplos: **MongoDB y CouchDB.**
- **Grafos:** Los almacenes de grafos se usan para almacenar información sobre redes, como pueden ser conexiones sociales. Ejemplos: **Neo4J, AWS Neptune y ArangoDB.**
- **Basados en columnas:** Los almacenes basados en columnas están optimizados para consultas sobre grandes conjuntos de datos, y almacenan los datos como columnas en vez de como filas. Ejemplos: **BigTable, Cassandra y HBase.**

key-value

Amazon
DynamoDB (Beta)

ORACLE[®]
BERKELEY DB 11g

redis

graph

Neo4j
the graph database

InfiniteGraph

sones

column

HBASE

riak

Cassandra

document

CouchDB
relax

mongoDB

terrastore

Fuente: Aitor Medrano. Especialización en Inteligencia Artificial y Big Data (IABD)

Servicios de bases de datos no administrados y administrados

Servicios de bases de datos no administrados

En un servicio **no administrado**, el administrador es responsable de instalar, configurar y mantener todo lo relacionado con el sistema gestor de bases de datos base de datos. Esto significa gestionar el servidor, el software y cualquier tarea operativa relacionada.

Algunas de las responsabilidades del administrador serían.

- Instalación y configuración del software de la base de datos.
- Configuración y mantenimiento del hardware o instancias de servidores. Mantenimiento del servidor y huella energética.
- Instalación y parches de software
- Gestión de backups y restauración ante fallos.
- Escalado manual del servidor y del almacenamiento.
- Configuración de alta disponibilidad y recuperación ante desastres.
- Monitoreo y ajuste del rendimiento.
- Aplicación de parches y actualizaciones de seguridad.
- ..

Ventajas:

- Mayor control y personalización completa del entorno.
- Ideal para configuraciones avanzadas y específicas.
- Puede tener costos más bajos en infraestructura pura.

Desventajas:

- Mayor complejidad y necesidad de conocimientos técnicos avanzados.
- Mayor carga operativa para el equipo de TI.
- Riesgo más alto de errores humanos y problemas de seguridad.

Ejemplo: Instalar MySQL o PostgreSQL en una instancia EC2 de AWS, una VM de Azure o un servidor físico y realizar toda la administración y el mantenimiento (parches , copias de seguridad de la base de datos, garantizar su alta disponibilidad, planificar la escalabilidad y la seguridad de los datos,).

Servicios de Bases de Datos Administrados

Los servicios **administrados** se encargan de la mayoría (o todas) las tareas operativas necesarias para que el sistema gestor de bases de datos funcione correctamente. El proveedor de la nube maneja la infraestructura y la operación diaria.

Algunas de las responsabilidades del proveedor serían.

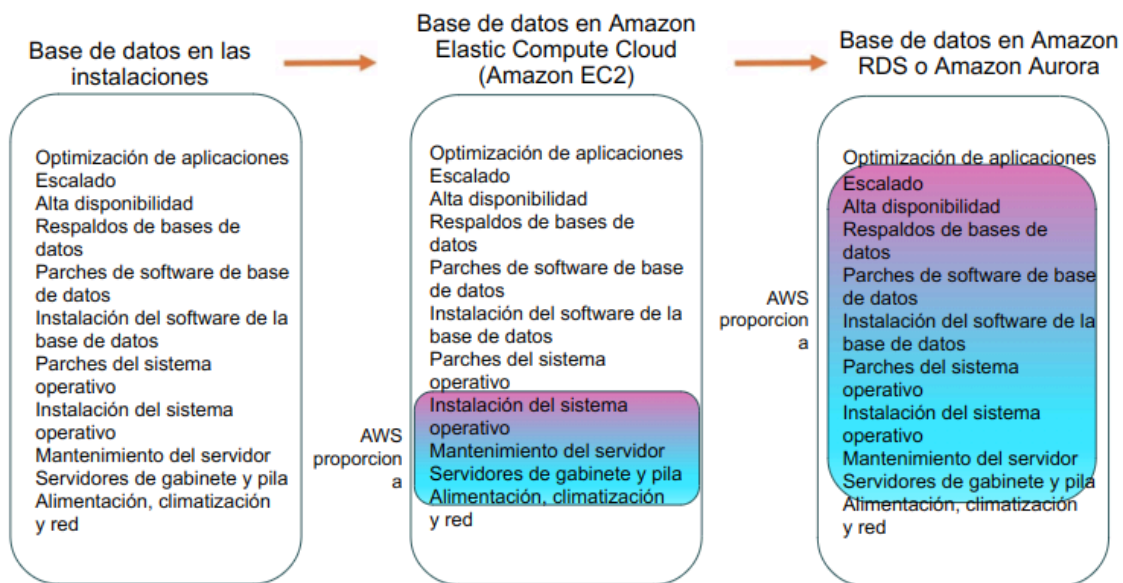
- Instalación, configuración y mantenimiento del software de base de datos.
- Escalado automático o sencillo del almacenamiento y cómputo.
- Gestión de backups automáticos y recuperación ante fallos.
- Alta disponibilidad y tolerancia a fallos integrada.
- Monitoreo continuo del rendimiento.
- Aplicación automática de parches y actualizaciones de seguridad.
- ...

Ventajas:

- Menor carga operativa y necesidad de administración manual.
- Alta disponibilidad y recuperación ante desastres integrada.
- Escalabilidad y elasticidad fáciles de manejar.
- Ideal para equipos con menos experiencia técnica en gestión de bases de datos.

Desventajas:

- Menor control sobre configuraciones avanzadas y personalización profunda.
- Costos más altos debido al servicio administrado.
- Puede haber restricciones de compatibilidad o limitaciones de personalización.



Fuente: Curso AWS Cloud Foundations

Ejemplos de servicios administrados: Amazon RDS, Azure SQL Database, Google Cloud Spanner y Firestore, Amazon DynamoDB , ...

Referencias

- Curso Academy Cloud Foundation de Amazon Web Services.
- Aitor Medrano. Especialización en Inteligencia Artificial y Big Data (IABD) <https://aitor-medrano.github.io/iabd/>. Licencia CC BY-NC-SA 4.0.
- OpenAI. (2025). Chat GPT.
- <https://aprendiendoaws.com/>