



Introducción a Bases de Datos: Bienvenida y Objetivos

Presentamos conceptos clave y la estructura del curso. Repasamos la importancia de las bases de datos en la industria informática y su aplicación en redes sociales, bancos y comercio electrónico.

Los datos son unidades básicas sin significado propio, mientras que la información es el resultado de procesar y contextualizar esos datos para darles sentido.

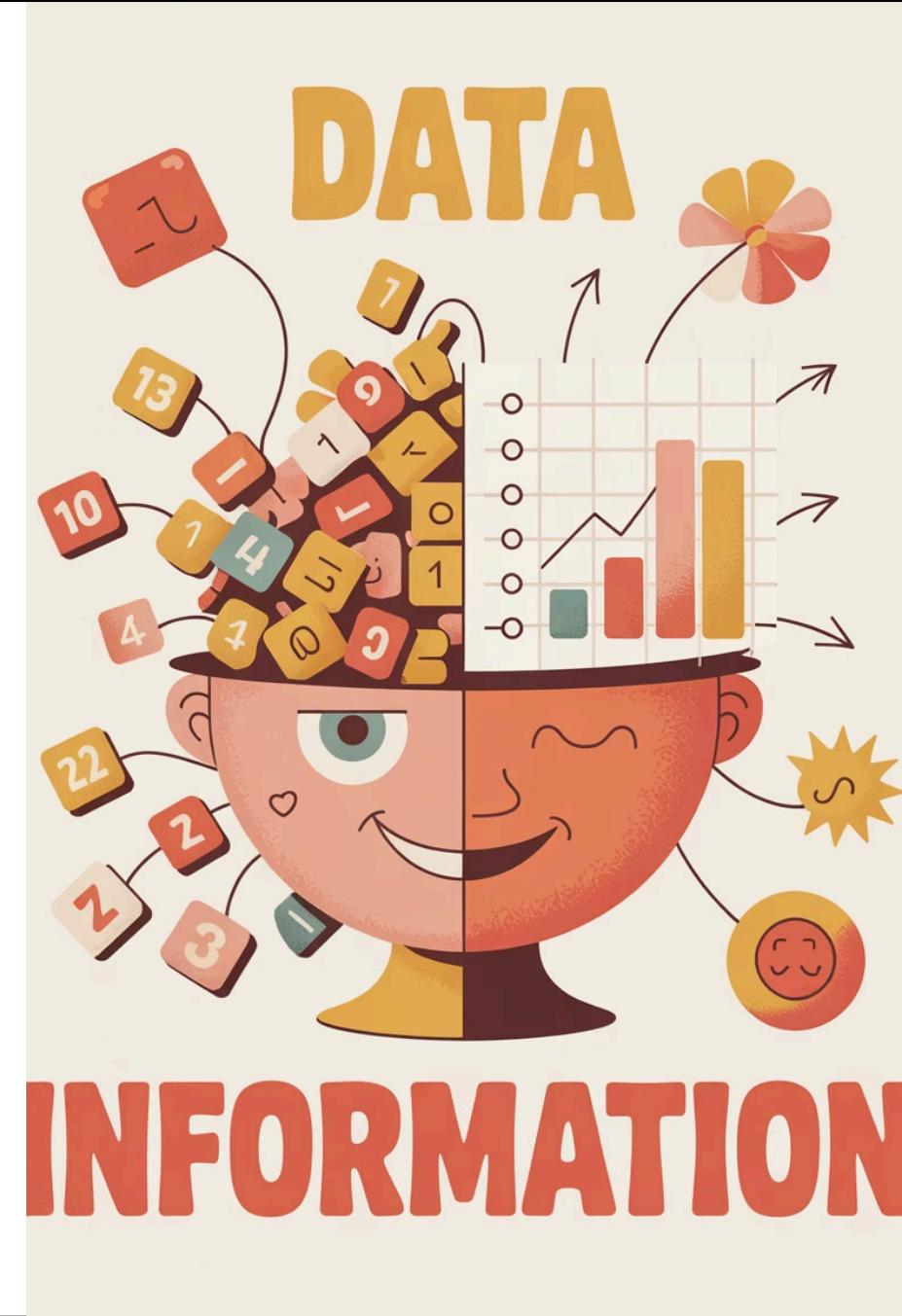
Dato e Información

Dato

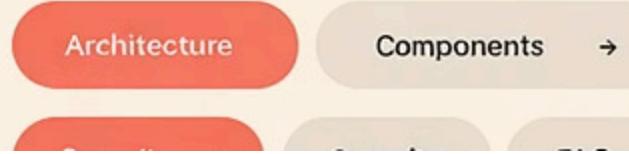
Unidad mínima de información, como un número, palabra o símbolo. Los datos por sí solos no tienen significado, pero son la materia prima para crear información.

Información

Datos procesados y organizados con significado y relevancia para quien los utiliza. La información resulta del análisis, interpretación y contextualización de los datos.



Systems Overview



Sistemas de Información y Bases de Datos

Un sistema de información integra elementos para gestionar datos y facilitar información; una base de datos organiza y administra datos estructurados electrónicamente.

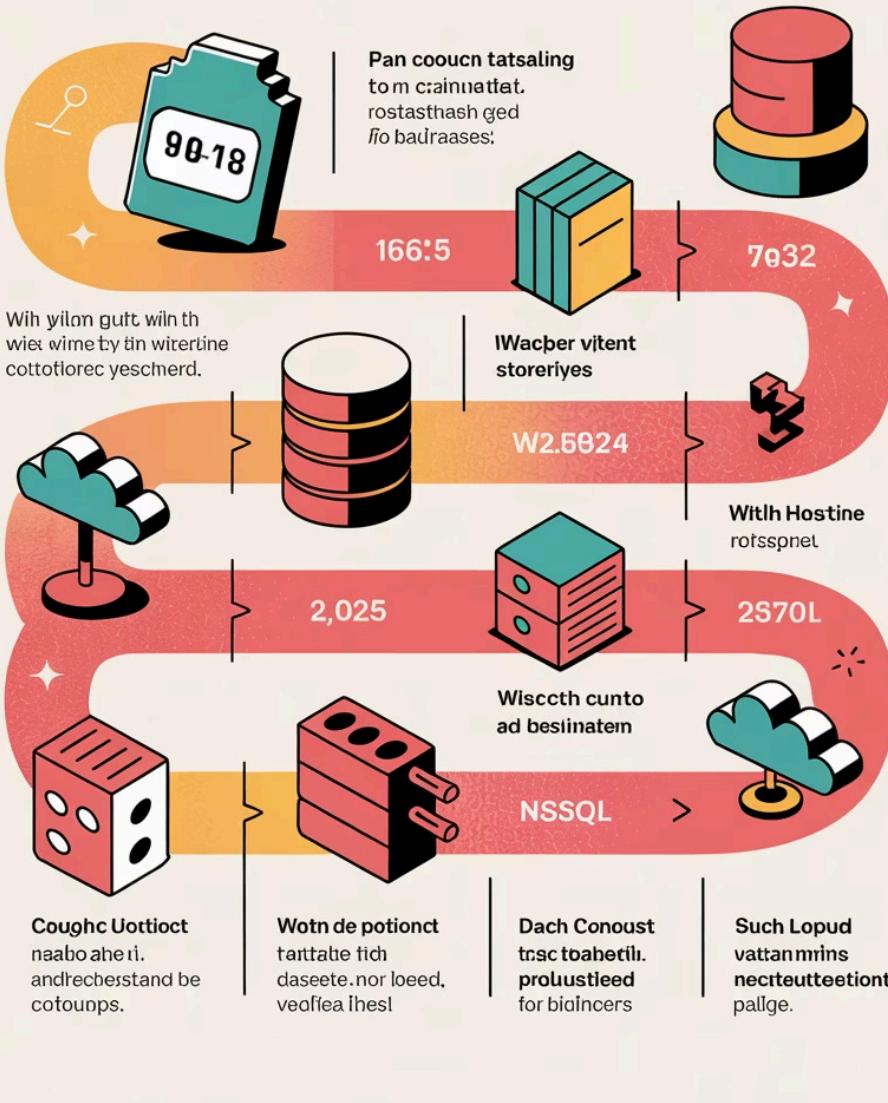
Sistema de Información

Conjunto de elementos (hardware, software, datos, procedimientos y personas) organizados para procesar, almacenar y distribuir información de manera eficiente y efectiva en una organización.

Base de Datos

Colección organizada de datos estructurados, almacenados y gestionados electrónicamente. Las bases de datos permiten el almacenamiento, la actualización y la recuperación eficiente de la información.

Database Evolution: From Relational to NOSQL



Resumen: Una visión rápida de la evolución de las bases de datos, desde modelos antiguos hasta tecnologías modernas como NoSQL.

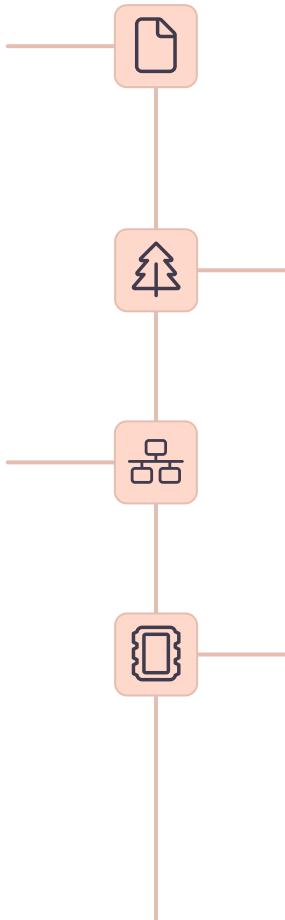
Historia y Evolución de las Bases de Datos

- Años 1960**
Archivos planos, modelos jerárquicos y de red.
- 1970**
Modelo relacional por Edgard. F. Codd.
- Posterior**
Bases orientadas a objetos y NoSQL desde 2000.

Evolución Histórica de las Bases de Datos

Archivos Planos (1960-1965)

Almacenamiento en ficheros de texto simples.
Difícil mantenimiento y alta redundancia de datos.



Modelo Jerárquico (1966-1970)

Estructura de árbol con relaciones padre-hijo.
Implementado en IMS de IBM.

Modelo de Red (1969-1973)

Relaciones complejas entre registros.
Estandarizado por CODASYL para mejorar
flexibilidad.

Hacia el Modelo Relacional

Fin de los 60s: Codd establece bases teóricas que
revolucionarán el campo.

Modelo Relacional: Surgimiento y Características

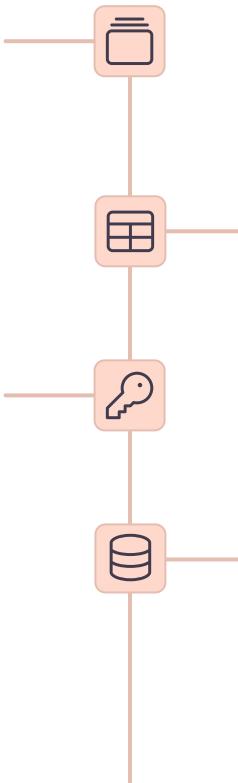
El modelo relacional revolucionó la gestión de datos en los años 70.

1970: Publicación Revolucionaria

Edgar F. Codd publica "A Relational Model of Data for Large Shared Data Banks" en IBM.

Claves y Relaciones

Introduce claves primarias y foráneas para establecer relaciones entre tablas.



Estructura en Tablas

Los datos se organizan en tablas con filas y columnas. Cada tabla representa una entidad.

Primeras Implementaciones

System R (IBM) y Oracle V2 surgen como primeras bases de datos relacionales comerciales.

El modelo relacional superó limitaciones de los modelos jerárquicos y de red anteriores.

Modelo Relacional: Surgimiento y Características

El modelo relacional, propuesto en 1970, organiza datos en tablas y utiliza SQL para su manejo, destacando por su simplicidad, integridad y eficiencia.

Contexto

- Propuesto por Edgard F. Codd en 1970 como una alternativa a los modelos jerárquicos y de red.
- Organiza los datos en tablas bidimensionales compuestas por filas (tuplas) y columnas (atributos).
- Permite independencia lógica al separar la estructura conceptual de la física, facilitando cambios sin afectar las aplicaciones.
- Introducción del lenguaje SQL como estándar para manipulación y consulta de datos.

Características

- Simplicidad en el diseño y facilidad de comprensión para usuarios y desarrolladores.
- Reducción significativa de la redundancia de datos a través de normalización.
- Integridad referencial mediante restricciones que aseguran consistencia entre tablas relacionadas.
- Capacidad para manejar grandes volúmenes de datos con eficiencia y flexibilidad.

Evolución Post-Relacional: Bases de Datos Modernas

La evolución de las bases de datos continuó tras el dominio del modelo relacional, adaptándose a nuevos desafíos tecnológicos.

Bases Orientadas a Objetos (1990s)

Surge para apoyar lenguajes de programación orientados a objetos. Almacena directamente objetos complejos.



Cassandra (2008)

Desarrollada por Facebook. Diseñada para escalabilidad masiva en múltiples servidores.



MongoDB (2009)

Base de datos documental NoSQL. Almacena datos en documentos BSON similares a JSON.

Neo4j (2007)

Base de datos de grafos. Optimizada para relaciones complejas entre entidades.

Era Cloud (2010s)

Bases de datos distribuidas en la nube. Ofrecen alta disponibilidad y escalabilidad automática.

Diferencia entre bases de datos SQL y NoSQL

Bases de datos SQL

Utilizan el modelo relacional con tablas estructuradas y esquemas predefinidos.

- Esquema rígido y predefinido
- Garantizan **integridad** referencial
- Transacciones **ACID**
- Escalamiento vertical
- Ejemplos: MySQL, PostgreSQL, Oracle

Bases de datos NoSQL

Diseñadas para modelos de datos flexibles sin esquemas estrictos.

- Esquemas dinámicos y flexibles
- Escalamiento horizontal
- Optimizadas para grandes **volúmenes**
- Consistencia eventual
- Ejemplos: MongoDB, Cassandra, Neo4j

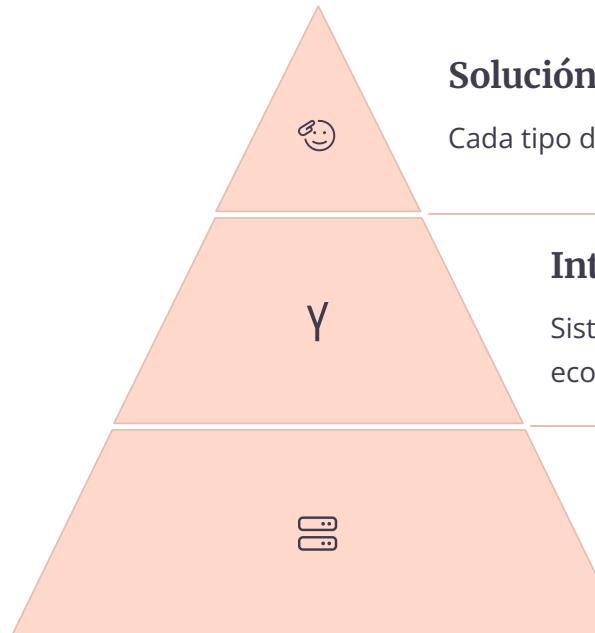
Casos de uso

La elección depende de los requisitos específicos del proyecto.

- SQL: aplicaciones financieras, ERP
- NoSQL: redes sociales, IoT, big data
- Tendencia actual: bases de datos políglota

Bases de Datos Políglotas

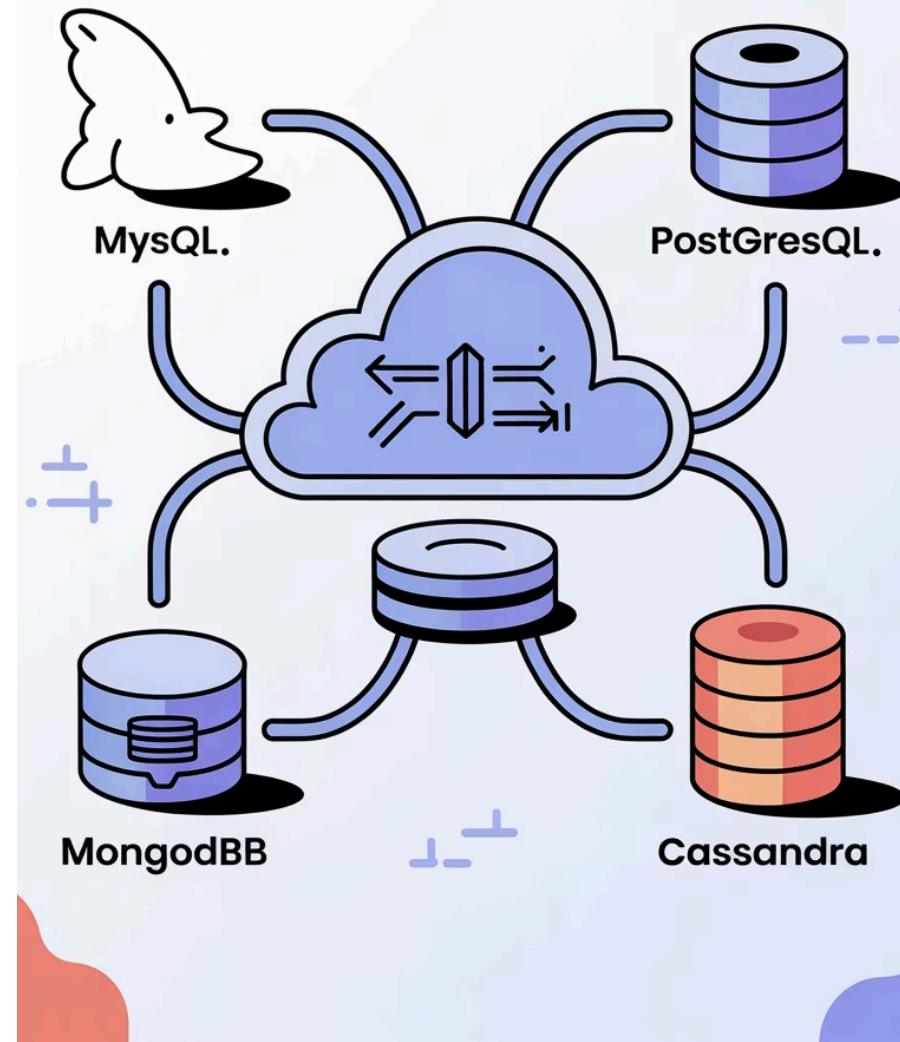
El enfoque pragmático que combina múltiples tecnologías de bases de datos para resolver problemas específicos.



Las arquitecturas políglotas permiten utilizar la herramienta más adecuada para cada necesidad específica dentro de una misma aplicación.

El reto principal reside en la correcta integración y sincronización entre los diferentes sistemas de almacenamiento.

Polyglot Database Architecture





Connections

Schemas

Schemas

Queries

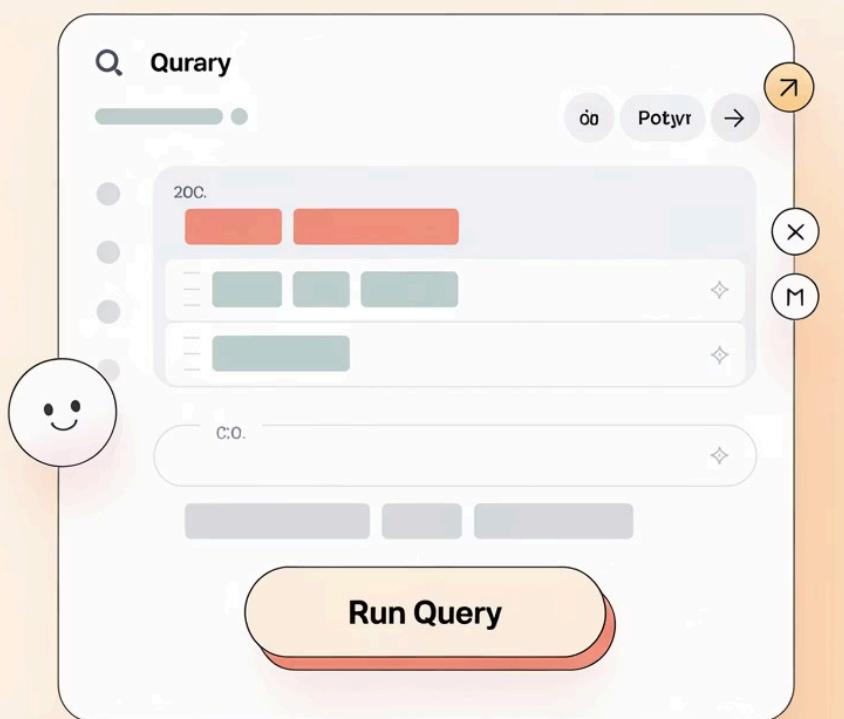


Help

MySQL

Datcabace Workbench

Datawe Mondy



Presentamos dos herramientas esenciales para la administración y modelado de bases de datos MySQL.

Entorno de Trabajo: Instalación de Herramientas



MySQL Workbench

Herramienta gráfica para modelado y consultas SQL.



XAMPP + phpMyAdmin

Paquete con Apache, MySQL y gestión web de bases de datos.

Entorno de Trabajo: Instalación de Herramientas

Workbench es una herramienta visual para administrar bases de datos MySQL que facilita la creación de objetos, ejecución de consultas y monitoreo. Su instalación es sencilla siguiendo unos pocos pasos.

Workbench permite a los desarrolladores y administradores realizar tareas como:

- Crear y editar tablas, vistas, procedimientos y otros objetos de la base de datos
- Ejecutar consultas SQL y ver los resultados
- Importar y exportar datos en diferentes formatos
- Monitorear el rendimiento y la actividad de la base de datos

Para instalar Workbench, sigue estos pasos:

1. Descarga el instalador de [**la página oficial de MySQL**](#)
2. Ejecuta el instalador y sigue las instrucciones en pantalla
3. Una vez instalado, abre Workbench y conéctate a tu servidor de base de datos

Entorno de Trabajo: Instalación de XAMPP y phpMyAdmin

XAMPP y phpMyAdmin, cómo instalar XAMPP, y cómo acceder a phpMyAdmin para gestionar bases de datos MySQL localmente.

¿Qué es XAMPP y phpMyAdmin?

XAMPP es un paquete que incluye Apache, MySQL, y PHP para desarrollo local.

phpMyAdmin es una herramienta web para gestionar bases de datos MySQL de forma gráfica.

Pasos para instalar XAMPP

1. Descarga XAMPP desde su sitio oficial.
2. Ejecuta el instalador y sigue las instrucciones.
3. Inicia el panel de control y activa Apache y MySQL.

Acceder a phpMyAdmin

Abre un navegador y escribe **localhost/phpmyadmin**.

Gestiona bases de datos, crea tablas y ejecuta consultas SQL fácilmente.



Studyspark:

Gracias!

Continuamos aprendiendo ...