



Sistemas de Gestión de Bases de Datos: Arquitectura e Implementación

Una guía completa para comprender la arquitectura, los lenguajes y las herramientas para una gestión efectiva de los SGBD

Temario del curso

1 Fundamentos de DBMS

Definición, rol e importancia en los sistemas de información

3 Lenguajes de base de datos

Sintaxis y aplicaciones de DDL y DML

2 Arquitectura RDBMS

Modelo de tres niveles y componentes funcionales principales

4 Herramientas gráficas

Implementación práctica utilizando phpMyAdmin



Objetivo del módulo

Comprender el funcionamiento de los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS), sus componentes clave y cómo interactuar con ellos a través de lenguajes de definición y manipulación de datos, utilizando tanto interfaces gráficas como SQL.

¿Qué es un SGBD?

Definición

Un Sistema de Gestión de Bases de Datos (SGBD) es un software que permite a los usuarios crear, mantener y manipular bases de datos de manera eficiente, segura y controlada.

Sirve como interfaz entre la base de datos y sus usuarios finales o programas de aplicación, asegurando que los datos se organicen de manera consistente y permanezcan fácilmente accesibles.



El papel de los SGBD en los sistemas de información



Centralización de datos

Proporciona un repositorio central para todos los datos de la organización, eliminando los problemas de redundancia e inconsistencia que aquejan a los sistemas basados en archivos.



Acceso concurrente

Facilita el acceso seguro y simultáneo de múltiples usuarios, manteniendo la integridad de los datos a través de la gestión de transacciones.



Independencia de datos

Separa la lógica de la aplicación de los detalles de almacenamiento de datos, permitiendo que cualquiera de los dos cambie sin afectar al otro.

Arquitectura RDBMS: Modelo de tres capas ANSI/SPARC



Esta arquitectura proporciona independencia de datos, permitiendo cambios en un nivel sin afectar a los demás.

Tres capas explicadas

Capa externa (vista del usuario)

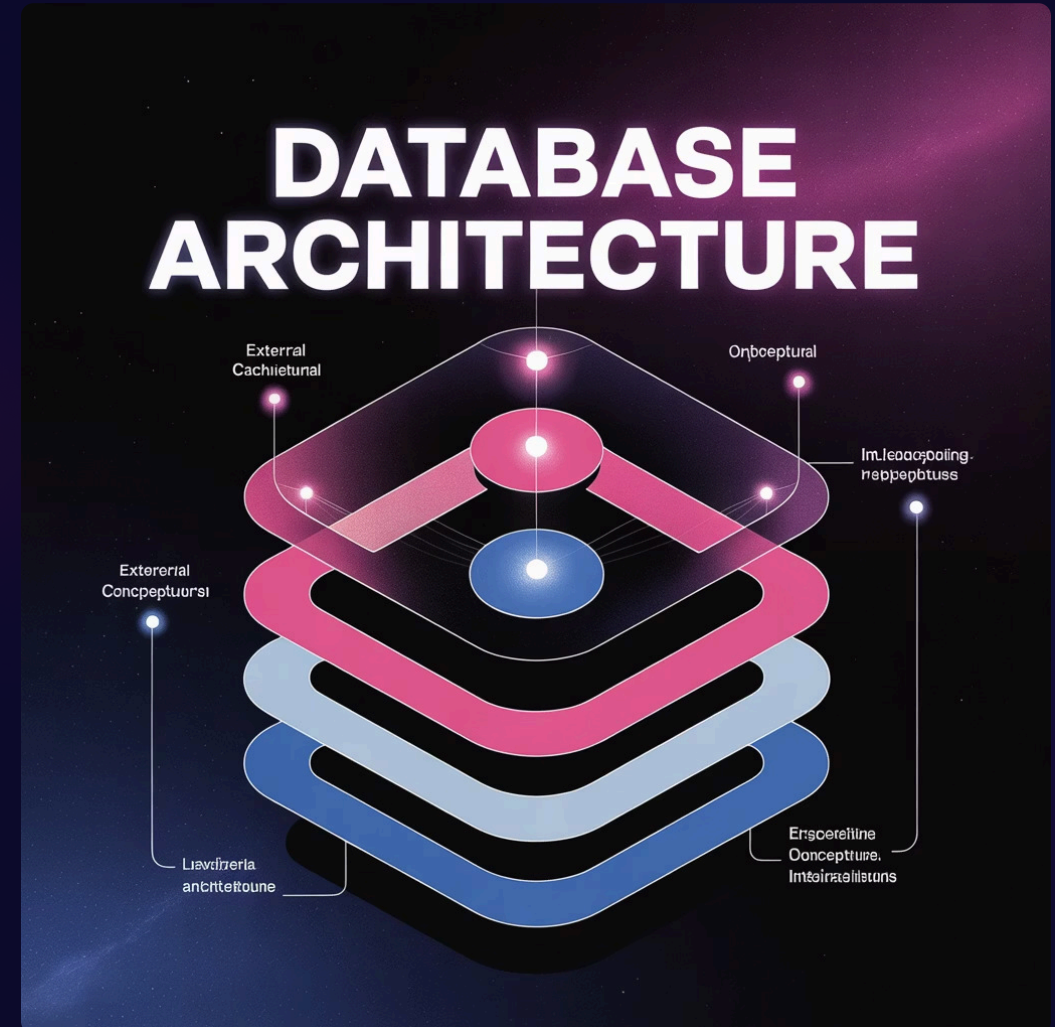
Representa cómo los diferentes usuarios o aplicaciones perciben la base de datos. Pueden existir múltiples vistas, cada una adaptada a las necesidades específicas de los usuarios.

Capa conceptual (estructura lógica)

Define la estructura lógica de toda la base de datos independientemente del almacenamiento físico. Incluye entidades, relaciones, restricciones y reglas de seguridad.

Capa interna (almacenamiento físico)

Describe cómo se almacenan los datos físicamente utilizando organizaciones de archivos, estructuras de indexación y rutas de acceso.



Componentes funcionales de un SGBD



Administrador de almacenamiento

Maneja el almacenamiento físico de datos, la organización de archivos y la administración del búfer



Procesador de consultas

Interpreta y optimiza las consultas SQL para una ejecución eficiente



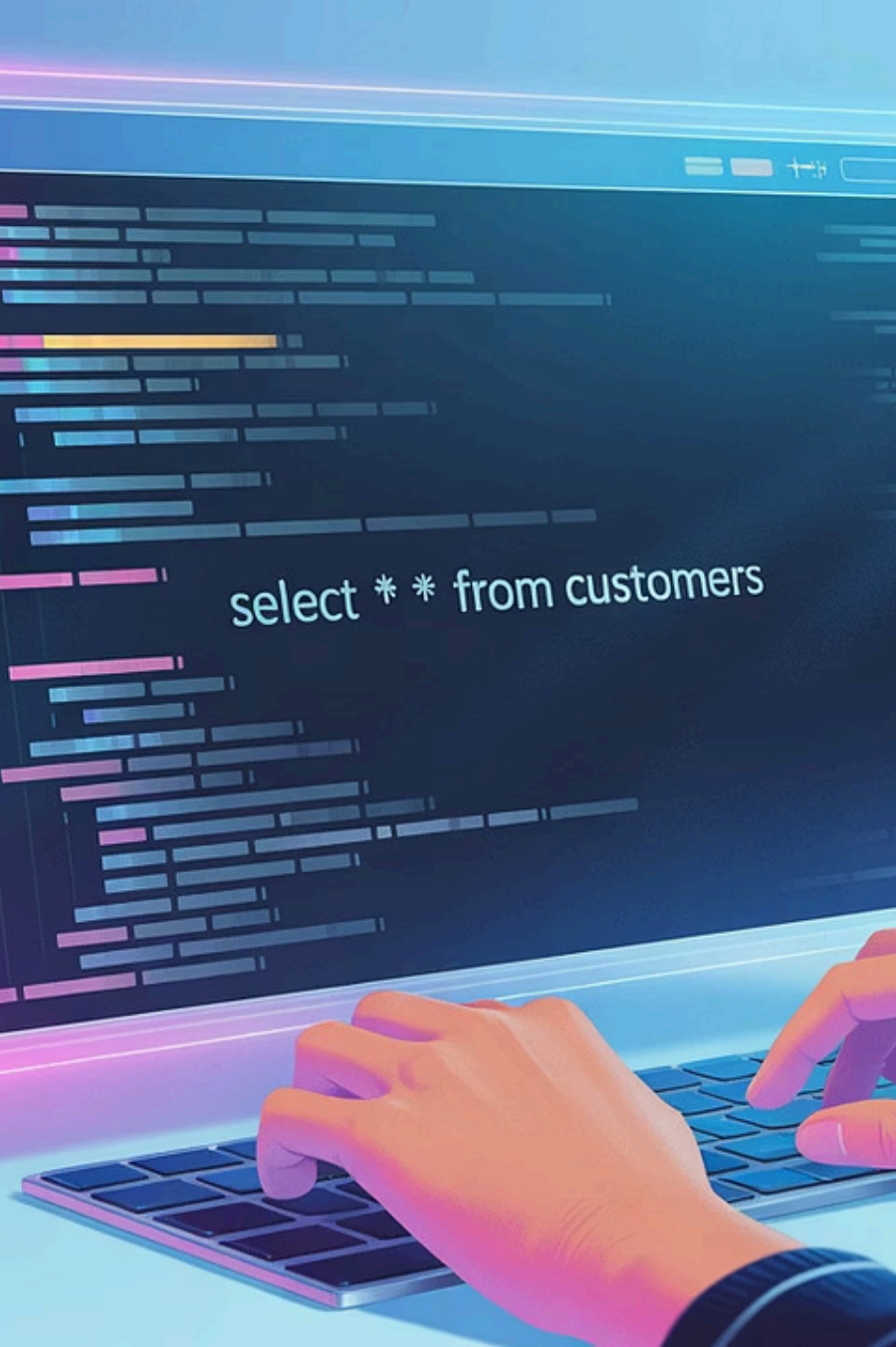
Controlador de concurrencia

Administra las transacciones simultáneas para evitar conflictos



Módulo de seguridad y respaldo

Aplica controles de acceso y mantiene la seguridad de los datos



Lenguajes utilizados en DBMS

Los sistemas DBMS utilizan lenguajes especializados para definir estructuras y manipular datos. Estos lenguajes proporcionan una interfaz estandarizada entre los usuarios y el sistema de base de datos.

DDL - Lenguaje de Definición de Datos

Propósito

Se utiliza para definir y modificar la estructura de la base de datos, incluyendo:

- Crear tablas, vistas y esquemas
- Definir restricciones y relaciones
- Crear índices para mejorar el rendimiento
- Modificar estructuras existentes

```
CREATE TABLE estudiantes (  
  id_estudiante INT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  apellido VARCHAR(50) NOT NULL,  
  fecha_matricula DATE,  
  carrera VARCHAR(100)  
);
```

```
ALTER TABLE estudiantes  
ADD COLUMN promedio DECIMAL(3,2);
```

DML - Lenguaje de manipulación de datos

Propósito

Se utiliza para trabajar con los datos reales dentro de las estructuras de la base de datos:

- Recuperar datos (SELECT)
- Insertar nuevos registros (INSERT)
- Modificar datos existentes (UPDATE)
- Eliminar registros (DELETE)

-- Recuperar datos

```
SELECT * FROM estudiantes
```

```
WHERE carrera = 'Ciencias de la Computación';
```

-- Insertar nuevo registro

```
INSERT INTO estudiantes VALUES
```

```
(1001, 'María', 'García',
```

```
'2023-09-01', 'Ciencia de Datos');
```

-- Actualizar registro existente

```
UPDATE estudiantes
```

```
SET carrera = 'Ingeniería Informática'
```

```
WHERE id_estudiante = 1001;
```

Operaciones DDL comunes



Creación de objetos

CREATE TABLE, CREATE VIEW,
CREATE INDEX

Establecer las estructuras fundamentales que contendrán sus datos



Modificación de objetos

ALTER TABLE, ALTER VIEW

Adaptar las estructuras existentes a los requisitos cambiantes sin pérdida de datos



Eliminación de objetos

DROP TABLE, DROP VIEW,
DROP INDEX

Eliminar estructuras que ya no se necesitan en la base de datos

Operaciones DML comunes



Consulta de datos

```
SELECT columna FROM tabla  
WHERE condición
```

Recuperar información específica en función de criterios de filtrado precisos



Modificación de datos

```
INSERT INTO, UPDATE, DELETE  
FROM
```

Agregar, cambiar o eliminar registros específicos manteniendo la integridad



Control de transacciones

```
COMMIT, ROLLBACK,  
SAVEPOINT
```

Garantizar la consistencia de los datos agrupando las operaciones en unidades atómicas



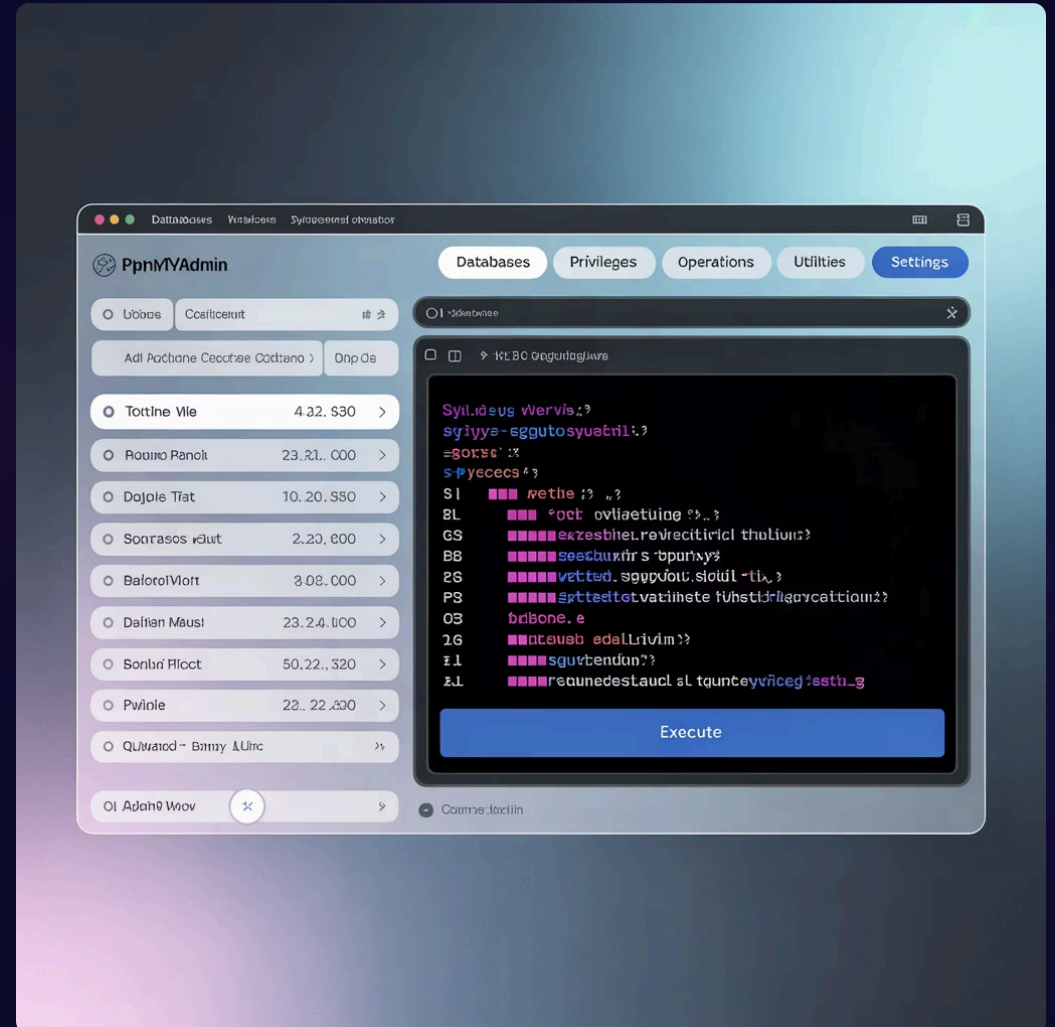
Herramientas gráficas para la gestión de bases de datos

Si bien el dominio de SQL es esencial, las herramientas gráficas proporcionan interfaces intuitivas que simplifican las tareas de gestión de bases de datos y aceleran el aprendizaje.

phpMyAdmin: Un ejemplo práctico

Características clave

- Interfaz web para MySQL/MariaDB
- No se requiere SQL para operaciones básicas
- Creación y modificación visual del esquema
- Constructor y ejecutor de consultas integrados
- Funcionalidad de importación/exportación
- Gestión de permisos de usuario



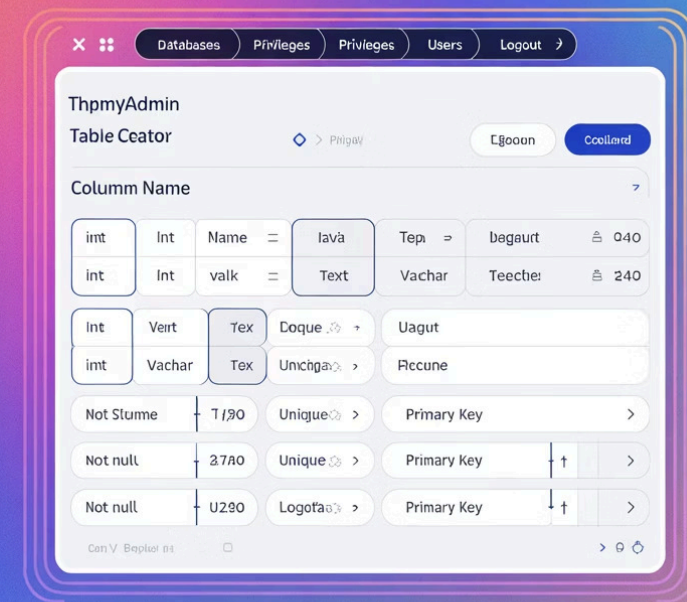
Trabajar con tablas en phpMyAdmin

Creación de tablas

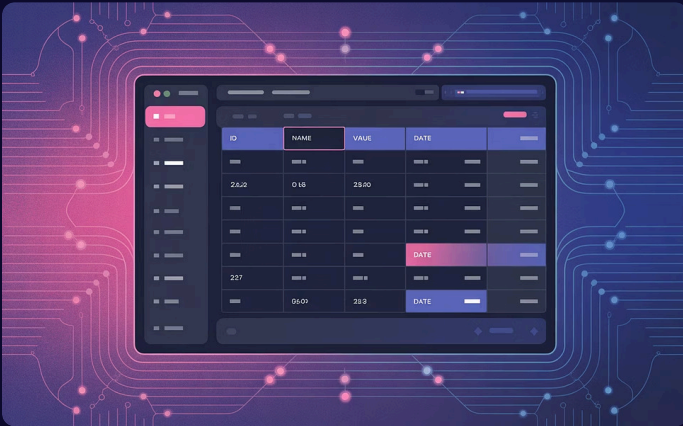
Utilice la interfaz visual para definir:

- Nombres de columnas y tipos de datos
- Claves primarias y foráneas
- Restricciones y valores predeterminados
- Índices para un mejor rendimiento

La herramienta genera SQL automáticamente, ayudando a los usuarios a aprender la sintaxis.



Administración de datos en phpMyAdmin



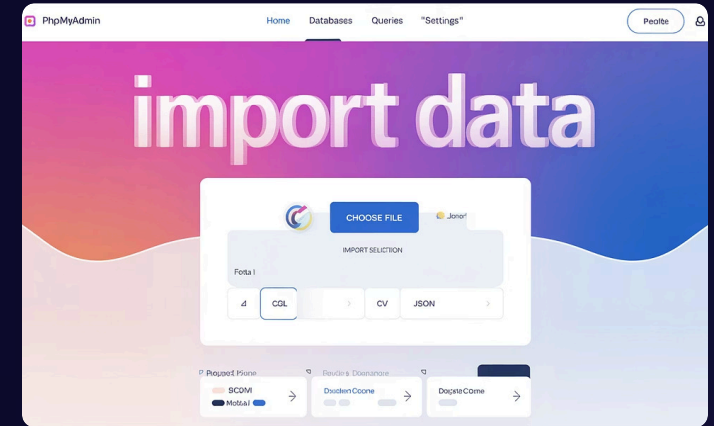
Examinar datos

Ver y filtrar el contenido de la tabla con opciones de clasificación y navegación para conjuntos de datos grandes



Editor SQL

Escribir y ejecutar consultas personalizadas con resaltado de sintaxis e historial de ejecución



Importar/Exportar

Transferir datos entre sistemas utilizando varios formatos como SQL, CSV y XML

Ventajas de usar ambos enfoques

Herramientas gráficas

- Curva de aprendizaje más baja para principiantes
- Más rápido para tareas simples y rutinarias
- Representación visual de los objetos de la base de datos
- Reducción de errores de sintaxis

SQL directo

- Mayor control y flexibilidad
- Más eficiente para operaciones complejas
- Mejor comprensión de lo que está sucediendo
- Automatizable mediante scripts
- Habilidades transferibles a través de sistemas

Los administradores de bases de datos profesionales suelen usar ambos enfoques dependiendo de la tarea que tengan entre manos.

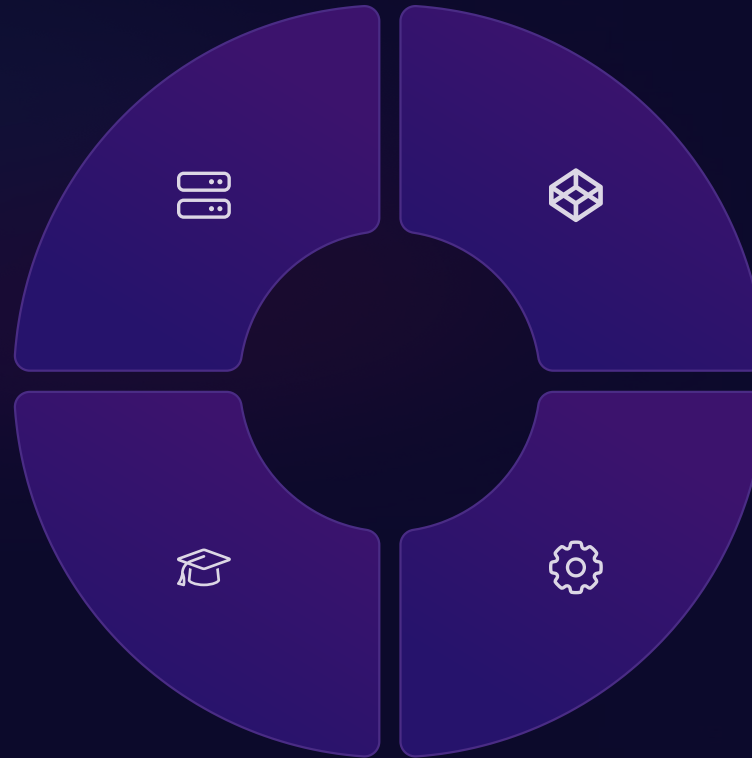
Puntos clave

Arquitectura de DBMS

El modelo de tres capas proporciona abstracción e independencia entre el almacenamiento físico y la representación lógica

Habilidades profesionales

Comprender los modelos conceptuales es esencial independientemente de la interfaz utilizada



Lenguajes SQL

DDL y DML sirven propósitos distintos en la definición de estructuras y la manipulación de datos

Opciones de implementación

Tanto las herramientas gráficas como el SQL directo tienen fortalezas complementarias en la gestión de bases de datos

Pregunta de discusión

¿Por qué es importante entender cómo se definen y manipulan las bases de datos a nivel conceptual, más allá de simplemente usar herramientas gráficas?

Considera estos aspectos en tu respuesta:

- Capacidades de solución de problemas
- Optimización del rendimiento
- Versatilidad entre plataformas
- Desarrollo profesional a largo plazo

