

## ¿Qué es un ORM?

Un **ORM** (Object-Relational Mapping o **Mapeo Objeto-Relacional**) es una herramienta que permite interactuar con bases de datos relacionales usando el paradigma de programación orientada a objetos.

En lugar de escribir consultas SQL manualmente para acceder o manipular datos en una base de datos, un ORM permite trabajar con **objetos del lenguaje de programación** (como las clases en C#), y el ORM se encarga de traducir esas operaciones en comandos SQL que la base de datos entiende.

## ¿Cómo funciona?

A grandes rasgos, podríamos resumir su funcionamiento con 3 funcionalidades como las siguientes:

# - **Mapeo de clases a tablas:** Cada clase en tu código representa una tabla en la base de datos.

# - **Mapeo de atributos a columnas:** Cada atributo de una clase representa una columna de la tabla.

# - **Instancias como filas:** Cada instancia de una clase representa una fila o registro en la tabla.

## Ventajas y desventajas de usar un ORM.

Ventajas:

# - Permite mayor rapidez al desarrollar aplicaciones.

# - El código de la aplicación resulta más limpio y mantenible.

# - Ofrece independencia del sistema de gestión de bases de datos que utilice. (Por ejemplo, puedes cambiar de MySQL a PostgreSQL con pocos cambios).

# - Ayuda a evitar errores comunes de SQL.

# - Nos permite crear automáticamente una base de datos partiendo de las clases de nuestro modelo de dominio, o a la viceversa, nos permite crear las clases de nuestro modelo de dominio partiendo de una base de datos existente.

# - Mediante el uso de migraciones nos permite mantener actualizado nuestro modelos de datos con la base de datos.

Desventajas:

# - Puede ocultar lo que realmente sucede en la base de datos (menor control en las consultas).

# - En escenarios, puede ser menos eficiente que escribir SQL optimizado a mano.

# - Conlleva una curva de aprendizaje para usarlo correctamente como cualquier herramienta.

### **Algunos ejemplos de ORMs populares según el lenguaje de programación:**

# - **En Python:** SQLAlchemy, Django ORM.

# - **Java:** Hibernate.

# - **PHP:** Eloquent (Laravel).

# - **JavaScript/Node.js:** Sequelize, Prisma.

# - **C# (.NET):** Entity Framework Core.