

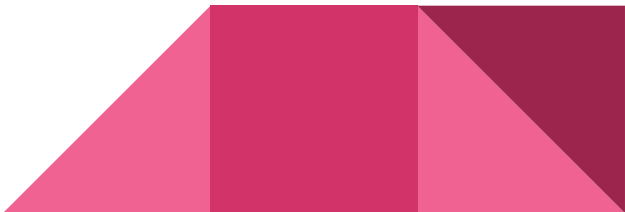
Object Relational Mapping

ORM

ORM - Mapeo Relacional de Objetos

Herramienta que permite **interactuar con bases de datos relacionales utilizando código orientado a objetos (POO)**.

Traduce operaciones en objetos de alto nivel a consultas SQL y viceversa. Ofrece **abstracción de la base de datos**.

- **Simplifica** la gestión de bases de datos.
 - **Elimina la necesidad de escribir SQL** manualmente.
 - Mejora la **mantenibilidad** del código.
 - Mejora la **escalabilidad** del código.
 - Proporciona **protección contra inyecciones SQL**.
- 

Ejemplos de ORMs

Python:

- SQLAlchemy
- Django ORM

Java:

- Hibernate

Ruby:

- ActiveRecord

PHP:

- **Eloquent** (Laravel)
- Doctrine (Symfony)

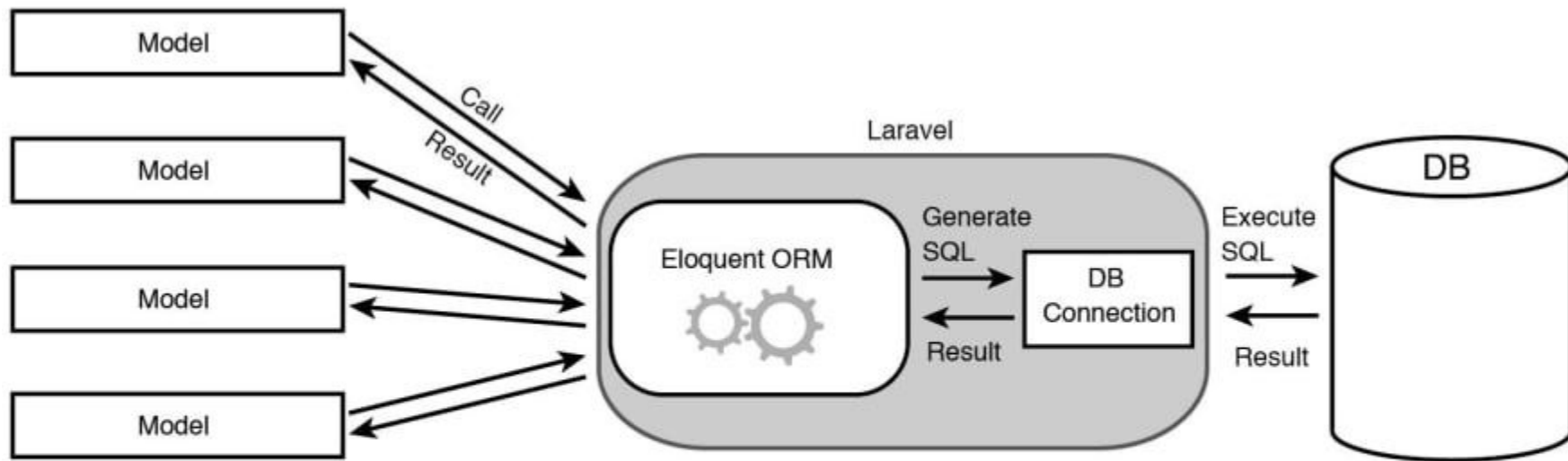


Funcionamiento básico

- El ORM define una **clase para representar una tabla** en la base de datos.
- Los **objetos** de la clase representan **filas** de la tabla.
- Los **métodos** de la clase permiten realizar operaciones **CRUD** (Create, Read, Update, Delete) sobre datos de la tabla.



Funcionamiento básico - Eloquent (ORM Laravel)




Funcionamiento básico - Eloquent (ORM Laravel)

Asumiendo que la conexión a la base de datos ya está configurada (en `config/database.php`).

Definir un Modelo: Crea una nueva clase de modelo que extiende la clase base `Illuminate\Database\Eloquent\Model`. Esta clase representa una tabla en la base de datos y define sus atributos, relaciones y otras configuraciones.

Definir el Nombre de la Tabla: Por defecto, Eloquent asume que el nombre de la tabla es la versión en plural y en snake_case del nombre de la clase del modelo. Si el nombre de la tabla es diferente, se puede especificar explícitamente en el modelo definiendo el atributo *protegida* `$table`.



Funcionamiento básico - Eloquent (ORM Laravel)

Definir Relaciones: Se utilizan los métodos de relaciones propios `hasOne`, `hasMany`, `belongsTo`, etc. para definir relaciones entre modelos. Esto permite realizar consultas fácilmente sobre datos relacionados.

Realizar Operaciones en la Base de Datos: Usar los métodos del constructor de consultas (query builder); `get`, `find`, `create`, `update`, `delete`, etc. para llevar a cabo operaciones en la base de datos. El constructor de consultas proporciona una interfaz fluida para montar sentencias SQL.



Ejemplos de uso - Continuara...

SE VERÁN JUNTO CON EL USO DE FRAMEWORK LARAVEL

