

JPA (Java Persistence API)

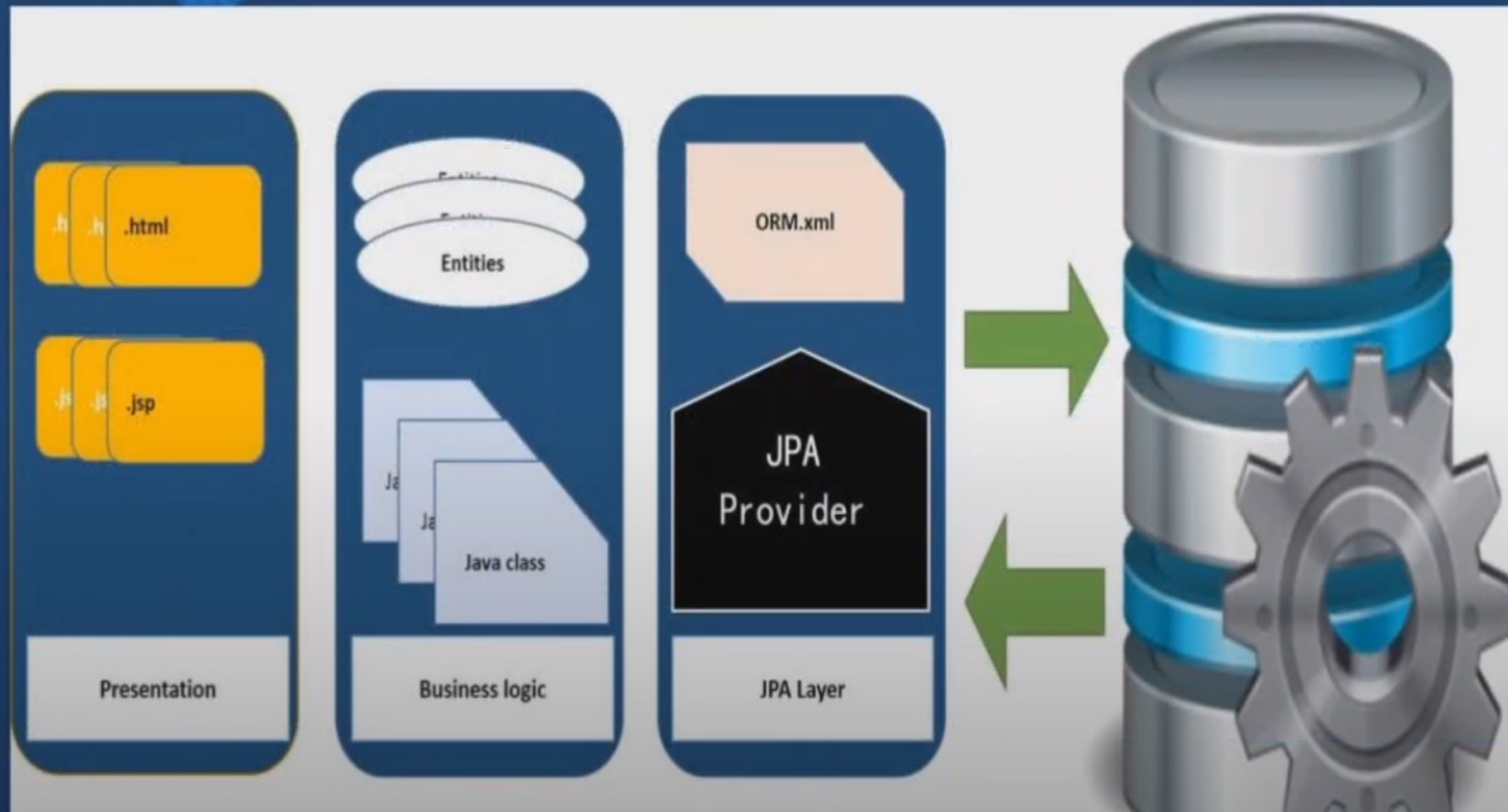
¿Qué es JPA?

- Es una API (Application Programming Interface) desarrollada principalmente para la plataforma Java EE.
- Es la propuesta estándar que ofrece Java para implementar un Framework Object Relational Mapping (ORM), que permite interactuar con la base de datos por medio de objetos.
- JPA es el encargado de convertir los objetos Java en instrucciones para el Manejador de Base de Datos (MDB).



JPA
Java Persistence API

¿Cómo funciona JPA?



Proveedores de JPA

- Como JPA es una API open source, existen distintos proveedores como Oracle, Redhat, Eclipse, etc que proporcionan nuevos productos mediante la adición de la persistencia.
- Algunos de estos productos incluyen:
 - **Hibernate**
 - **Eclipselink**
 - **Toplink**
 - **ObjectDB**
 - **Open JPA**
 - **Entre otros**



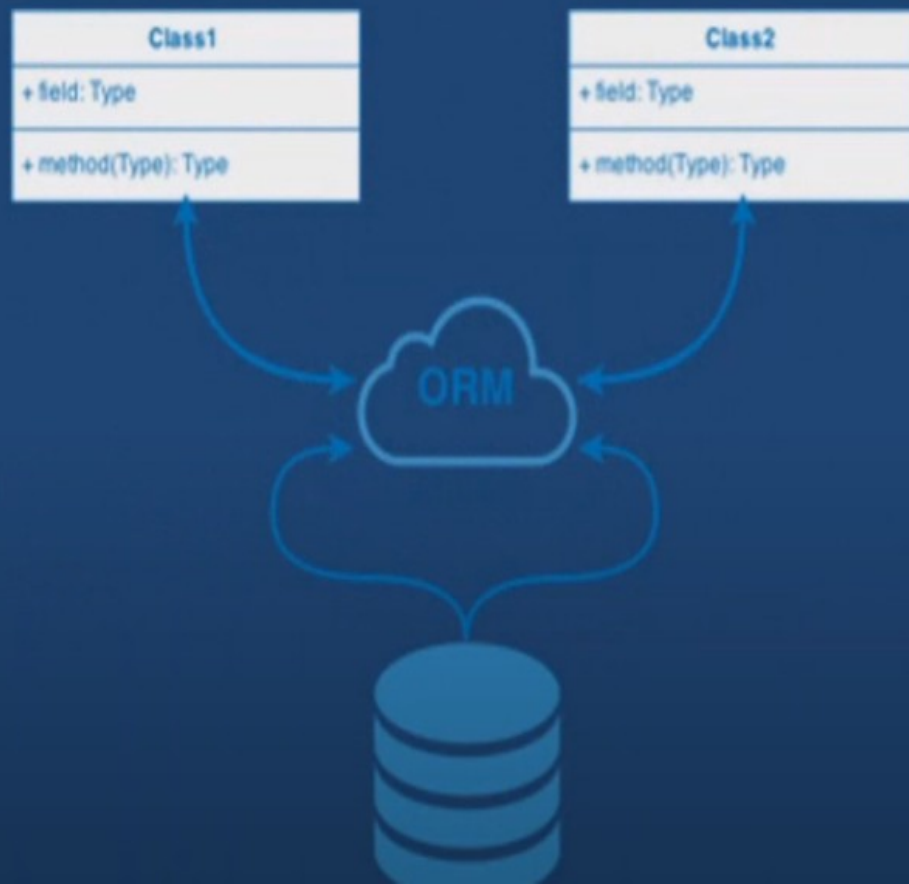
JPA
Java Persistence API

JPA + ORM

(Object-Relational Mapping)

Mapeo ORM (Object Relational Mapping)

El mapeo Objeto-Relacional, ORM en inglés, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia



Anotaciones más usadas

- **@Entity**: Especifica la creación de una entidad. Se coloca al inicio de la definición de una clase.
- **@Id**: Primary key de la entidad
 - **@GeneratedValue(strategy = GenerationType.SEQUENCE)**: Establece que la ID se va a generar de forma automática y secuencial.
- **@Basic**: Para hacer referencia atributos comunes.
- **@Temporal**: Se usa normalmente en fechas.
 - Si se quiere tener en cuenta la hora se utiliza: **@Temporal(TemporalType.TIMESTAMP)**
 - Si solo se tiene en cuenta la fecha (DATE): **@Temporal(TemporalType.DATE)**
- **@OneToMany**: Indicar una relación unidireccional de 1 a n.

Salir del modo de pantalla completa (f)

Métodos JPA

- Dentro de las clases controladoras de JPA, existen métodos creados por defecto para realizar operaciones ABML (Altas, bajas, modificaciones y lectura) en las bases de datos.

ALTA DE REGISTROS

- Método Create
 - create (ObjetoACrear)

BAJA DE REGISTROS

- Método Destroy
 - destroy (idRegistroADestruir)

ENCONTRAR UN REGISTRO

- Método Find
 - find (idRegistroABuscar)

TRAER TODOS LOS REGISTROS

- Método Find Entities
 - EJ: List<Alumno> findAlumnoEntities()

MODIFICAR UN REGISTRO

- Método Edit
 - edit (ObjetoAEditar)