

Persistencia de datos



Persistencia de datos

La persistencia de datos es la capacidad de conservar datos en un sistema de almacenamiento no volátil para que puedan recuperarse en el futuro.

Ventajas

Evita la pérdida de información en caso de interrupciones o fallas del sistema

Conserva los datos históricos y las preferencias de los usuarios

Garantiza que los datos estén disponibles, consistentes e intactos

Desventajas

Puede permitir que información sensible se exponga si el medio de almacenamiento se pierde o se da a terceros

Los restos de datos pueden existir en varios lugares del sistema operativo.

Importancia

Para las empresas que operan con dispositivos usados, es importante prevenir la persistencia de datos. Al vender un dispositivo de segunda mano, se debe proporcionar al cliente la prueba de que se borraron los datos con éxito

ORM (Object Relational Mapping)



ORM

Sus siglas significan Object Relational Mapping (mapeo relacional de objetos) y su función es abstraer la base de datos, de modo que tú como programador puedas hacer consultas sin conocer SQL, y en su lugar, seguir usando el lenguaje de programación que ya conoces.

Lo que hace un ORM es mapear las bases de datos a objetos (por eso su nombre) y estos objetos tendrán métodos para interactuar con ellos y hacer el CRUD sin comunicarte directamente a la base de datos.

Principales ORM

Gorm para Go

SQLAlchemy para Python.

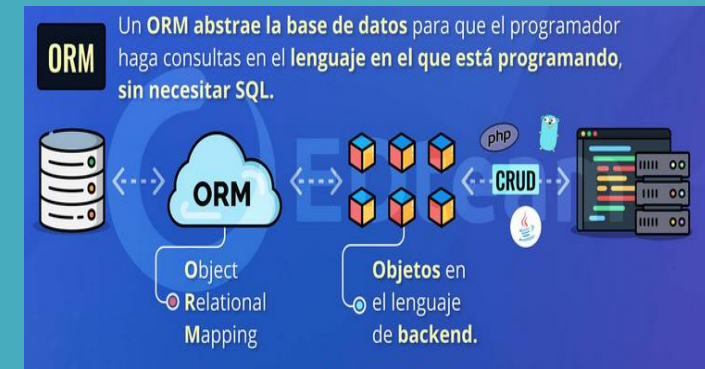
Hibernate usa Java.

TypeORM para JavaScript

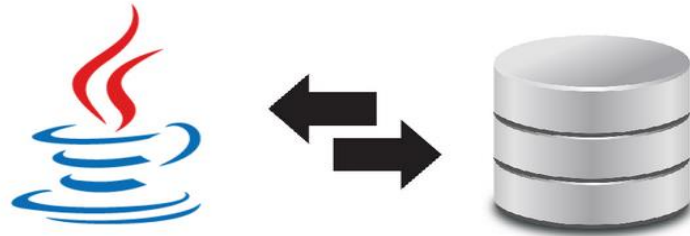
Entity framework para C#

Eloquent para PHP

Si bien un ORM te evita escribir SQL, no significa que debas tachar de tu lista aprender SQL. Conocerlo va a aumentar tu nivel y te permitirá hacer frente a proyectos mas complejos. Sin lugar a dudas, los ORM son una excelente alternativa que nos da flexibilidad y rapidez en el desarrollo aunque también tienen sus limites.



JPA (Java Persistence API)



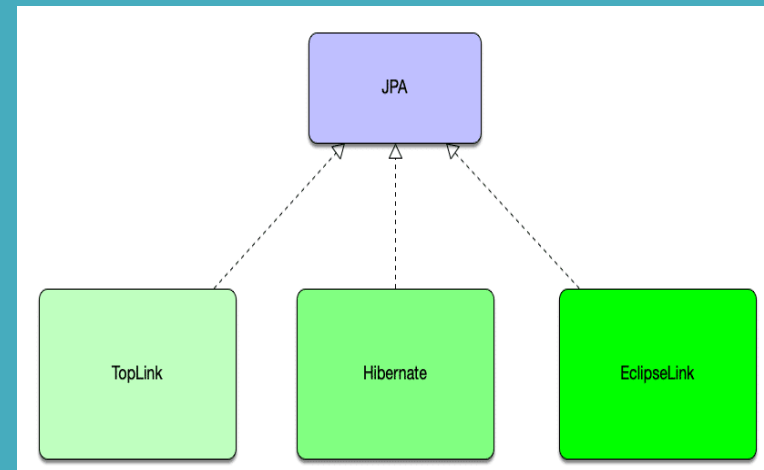
JAVA PERSISTENCE API

JPA

JPA o Java Persistence API es una especificación concretamente la JSR 338 . Una especificación no es más que un documento en el que se define como se debe gestionar una funcionalidad X. En este caso una capa de persistencia con objetos Java . Por ejemplo que anotaciones han de usarse, como han de persistirse los objetos como han de buscarse , cuál es su ciclo de vida etc .

Al tratarse de un documento evidentemente no implementa nada . Para poder trabajar con ella necesitaremos tener un Framework que implemente la especificación, uno de los más conocidos es Hibernate

Existen más implementaciones de JPA que al fin y al cabo es únicamente una especificación: EclipseLink, DataNucleus, TopLink, ObjectDB.



Hibernate



Hibernate

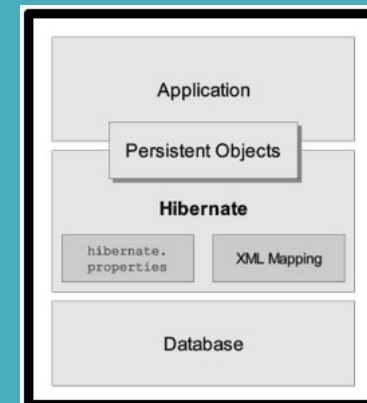
Hibernate es un framework de mapeo objeto-relacional (ORM) para Java que permite la comunicación entre una aplicación y una base de datos SQL.

¿Qué hace Hibernate?

- Convierte los datos entre los tipos de Java y los de SQL
- Genera sentencias SQL
- Permite mapear objetos de dominio de aplicación a tablas de bases de datos relacionales y viceversa
- Agiliza la relación entre la aplicación y la base de datos
- Optimiza el flujo de trabajo
- Evita el código repetitivo

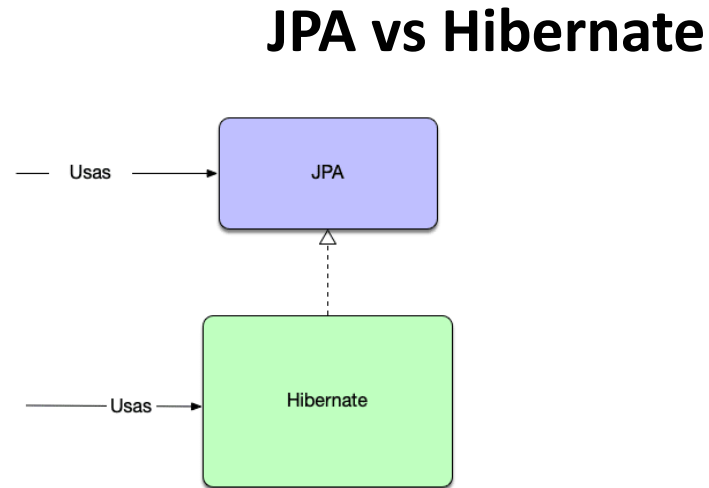
Cómo funciona Hibernate?

En la mayoría de programas actuales existen dos modelos de datos que coexisten: el usado en la memoria de la computadora (orientación a objetos), y el usado en las bases de datos (modelo relacional). Hibernate busca solucionar el problema de la diferencia entre ambos modelos.



JPA vs Hibernate

Normalmente la pregunta de mucha gente es que usar si JPA o Hibernate , es decir si usar directamente la especificación o usar directamente el framework.



¿Usar Hibernate?

La gente prefiere usar el framework (Hibernate) de forma directa porque el framework tiene capacidades adicionales que JPA como standard y especificación no soporta.

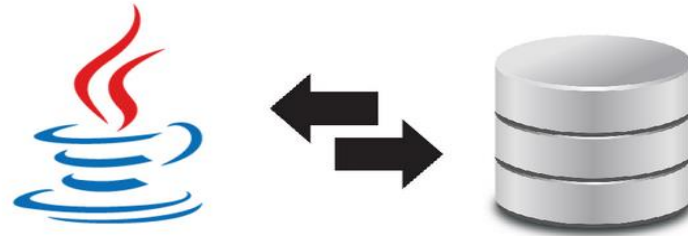
En la mayoría de las situaciones el usar directamente el framework aunque aporte alguna cosa adicional esos aportes no son críticos y las pocas ventajas que aporta no son suficientes comparadas con las ventajas de usar directamente la especificación.

¿Usar la especificación?

Vamos a ver un poco sus ventajas:

- La especificación esta muy trabajada y a nivel cultural muy extendida entre todos los desarrolladores . El uso concreto de un framework no lo esta tanto.
- La especificación nos permitirá cambiar de implementación de forma transparente si tenemos la necesidad .

Anotaciones JPA



JAVA PERSISTENCE API

Anotaciones JPA

Define an Entity Class

`@Entity`

`@Table`

Basic Column Mappings

`@Column`

`@Id`

`@GeneratedValue`

`@Enumerated`

`@Temporal`

`@Lob`

Association Mappings

`@ManyToMany`

`@ManyToOne` and `@OneToMany`

`@ManyToOne`

`@OneToMany`

`@OneToOne`

Gracias

CESDE | comfama

cesde.edu.co

