



JPA vs Hibernate

[1. JPA \(Java Persistence API\) 🌟](#)

[2. Hibernate ⚙️](#)

[3. Principales Diferencias 🤔](#)

[4. Similitudes ✨](#)

[5. Diagrama Explicativo 📊](#)

1. JPA (Java Persistence API) 🌟

- **Definición:** JPA es una **especificación** de Java que ofrece los estándares para trabajar con **mapeo objeto-relacional (ORM)**.
- **Lanzamiento:** Introducido el 11 de mayo de 2006.
- **Propósito:** Facilita el **mapeo** entre objetos Java y bases de datos relacionales.
- **Funcionalidad:** Proporciona los **estándares** y **reglas** para la persistencia, pero necesita una implementación concreta (como Hibernate).
- **Paquete:** `jakarta.persistence` o `javax.persistence` para las clases e interfaces principales.
- **Consultas:** Admite consultas usando **JPQL** (Java Persistence Query Language).

2. Hibernate ⚙️

- **Definición:** Framework ORM **implementación** de JPA, desarrollado para simplificar la interacción con bases de datos.
- **Lanzamiento:** Lanzado el 23 de mayo de 2007.
- **Propósito:** Proporciona **funcionalidad concreta** para la persistencia de objetos Java en bases de datos relacionales.
- **Paquete:** `org.hibernate` para las clases principales.
- **Consultas:** Utiliza **HQL** (Hibernate Query Language) para las consultas.
- **Interacción:** Proporciona clases como `SessionFactory` , `Session` para realizar acciones CRUD.

3. Principales Diferencias 🤔

JPA	Hibernate
Está descrito en el paquete <code>jakarta.persistence</code> (+3.0) <code>javax.persistence</code> (2.3 o inferior).	Está descrito en el paquete <code>org.hibernate</code> .
Describe el manejo de datos relacionales en aplicaciones Java.	Hibernate es una herramienta de Mapeo Objeto-Relacional (ORM) que se utiliza para guardar objetos Java en un sistema de base de datos relacional.
No es una implementación, es solo una especificación de Java.	Hibernate es una implementación de JPA. Por lo tanto, sigue el estándar común proporcionado por JPA.
Es una API estándar que permite realizar operaciones en la base de datos.	Se utiliza para mapear tipos de datos Java con tipos de datos SQL y tablas de base de datos.
Utiliza Java Persistence Query Language (JPQL) como lenguaje de consulta orientado a objetos.	Utiliza Hibernate Query Language (HQL) como lenguaje de consulta orientado a objetos.
Utiliza la interfaz <code>EntityManagerFactory</code> para interactuar con la fábrica del administrador de entidades para la unidad de persistencia.	Utiliza la interfaz <code>SessionFactory</code> para crear instancias de sesión.
Utiliza la interfaz <code>EntityManager</code> para realizar acciones de crear, leer y eliminar para instancias de	Utiliza la interfaz <code>Session</code> para realizar acciones de crear, leer

clases de entidad mapeadas.	y eliminar para instancias de clases de entidad mapeadas.
Actúa como una interfaz de tiempo de ejecución entre una aplicación Java y Hibernate.	Actúa como una interfaz de tiempo de ejecución entre una aplicación Java y Hibernate.

4. Similaridades ✨

- Ambas herramientas están orientadas al **mapeo objeto-relacional**.
- Ambas utilizan **JPQL** o **HQL** para crear consultas orientadas a objetos.
- Ambas permiten la persistencia usando **EntityManager** y **Session**.

5. Diagrama Explicativo 📊

JPA --- Implementación —→ Hibernate
Especificación Framework ORM