

# Hibernate

## ¿Qué es Hibernate?

**Hibernate** es una herramienta de mapeo objeto-relacional (ORM) para Java. Permite mapear clases Java a tablas de bases de datos, facilitando la persistencia de datos sin tener que escribir SQL directamente.

---

## Relación con JPA

Hibernate es una **implementación de la especificación JPA**, pero también ofrece **características adicionales propias**, más allá del estándar.

Puedes usar Hibernate:

- **Como proveedor de JPA** (recomendado con Spring)
  - **De forma nativa**, usando su propia API ( `Session` , `SessionFactory` , etc.)
- 

## Configuración básica (Hibernate nativo)

### 1. Dependencias Maven

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.6.15.Final</version>
  <!-- o la más reciente -->
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.36</version>
</dependency>
```

### 2. `hibernate.cfg.xml`

```
<hibernate-configuration>
  <session-factory>
    <property
      name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver
    </property>
```

```

    <property

name="hibernate.connection.url">jdbc:mysql://localhost:3306/mi_base
    </property>

    <property name="hibernate.connection.username">root</property>

    <property name="hibernate.connection.password">1234</property>

    <property name="hibernate.dialect">
        org.hibernate.dialect.MySQL5Dialect
    </property>

    <property name="hibernate.hbm2ddl.auto">update</property>

    <property name="show_sql">true</property>

    <mapping class="com.ejemplo.modelo.Usuario"/>
</session-factory>
</hibernate-configuration>

```

## Entidad ejemplo

```

@Entity
@Table(name = "usuarios")
public class Usuario {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(nullable = false)
    private String nombre;
    private String email;
    // Getters y Setters
}

```

## ⚡ Hibernate nativo vs JPA

Concepto	Hibernate nativo	JPA
Unidad de trabajo	Session	EntityManager

Concepto	Hibernate nativo	JPA
Configuración	<code>hibernate.cfg.xml</code>	<code>persistence.xml</code>
Creación	<code>SessionFactory</code>	<code>EntityManagerFactory</code>
API adicional	Sí (caching, filters, interceptors, etc.)	No

## Ciclo de vida de una entidad

1. **Transient:** No está en la base de datos ni en el contexto de Hibernate.
2. **Persistent:** Está siendo gestionada por Hibernate.
3. **Detached:** Fue persistente pero ya no lo es.
4. **Removed:** Marcada para ser eliminada.

## Operaciones básicas con Hibernate (nativo)

```
SessionFactory sf = new Configuration().configure().buildSessionFactory();
Session session = sf.openSession();
Transaction tx = session.beginTransaction();
// Crear Usuario u = new Usuario(); u.setNombre("Ana");
u.setEmail("ana@correo.com");
session.save(u);
// Leer Usuario encontrado = session.get(Usuario.class, 1L);
// Actualizar encontrado.setNombre("Ana María"); session.update(encontrado);
// Eliminar session.delete(encontrado); tx.commit(); session.close();
```

## Consultas en Hibernate

### 1. HQL (Hibernate Query Language)

```
Query<Usuario> q = session.createQuery("FROM Usuario WHERE email = :email",
Usuario.class);
q.setParameter("email", "ana@correo.com");
Usuario u = q.uniqueResult();
```

### 2. Criteria API (Hibernate)

```
CriteriaBuilder cb = session.getCriteriaBuilder();
CriteriaQuery<Usuario> cq = cb.createQuery(Usuario.class);
Root<Usuario> root = cq.from(Usuario.class);
```

```
cq.select(root).where(cb.like(root.get("nombre"), "A%"));  
List<Usuario> resultados = session.createQuery(cq).getResultList();
```

## Hibernate y Spring Boot

Spring Boot usa Hibernate como proveedor JPA por defecto.

```
spring:  
datasource:  
url: jdbc:mysql://localhost:3306/mi_base  
username: root  
password: 1234  
jpa:  
hibernate:  
ddl-auto: update  
show-sql: true  
database-platform: org.hibernate.dialect.MySQL5Dialect
```

## Opciones útiles de configuración

Propiedad	Descripción
hibernate.hbm2ddl.auto	create , update , validate , none
hibernate.show_sql	Muestra las consultas en consola
hibernate.format_sql	Formatea las consultas
hibernate.dialect	Dialecto para BD específica (MySQL, Oracle, etc.)
hibernate.cache.use_second_level_cache	Activa caché L2