

Requisitos para usar Hibernate con Spring Boot

- ✓ Instalar SGBD:
 - MariaDB. Nosotros usaremos este.
 - MySQL Workbench
 - PHPMyAdmin
- ✓ Librerías Hibernate (Mínimo Java 8):

La dependencia de **Spring Data JPA** en Spring Boot, automáticamente incluye Hibernate como el proveedor JPA predeterminado. Viene con todas las bibliotecas necesarias para trabajar con JPA e Hibernate, facilitando la configuración y el uso de estas tecnologías en tu aplicación Spring Boot.

La dependencia se puede agregar tanto manualmente en el archivo pom.xml de Maven (copiamos las dependencias de la última versión estable y la incluimos en el fichero pom.xml), como con el asistente o inicializador de Spring Boot.

- ✓ Dependencias a driver JDBC: Usaremos **mariaDB**. Se puede agregar manualmente en el pom (Maven) o utilizando el asistente o inicializador de Spring Boot.
- ✓ Conexión BBDD: en application.properties configuramos la cadena de conexión.

```
# Configuración de la base de datos
spring.datasource.url=jdbc:mariadb://localhost:3306/hibernate
spring.datasource.username=user
spring.datasource.password=password
```

```
server.port = 8083 // Sólo en caso que queramos modificar el puerto por defecto 8080
```

```
#Permite ver las consultas SQL en la consola
spring.jpa.show-sql=true
```

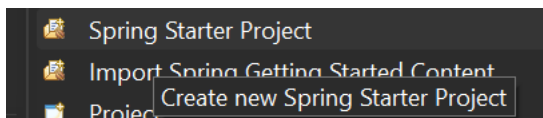
```
#Habilita el formato de las consultas, haciendo que sean más legibles al agregar saltos de
línea y #sangrías
spring.jpa.properties.hibernate.format_sql=true
```

```
# Elimina las tablas y las crea de nuevo en cada inicio. (SOLO EN DESARROLLO)
spring.jpa.hibernate.ddl-auto=create
```

Valores de `spring.jpa.hibernate.ddl-auto`:

Valor	Crea tablas	Elimina tablas	Actualiza tablas	Valida esquema	Uso recomendado
create	✓	✓	✗	✗	Desarrollo o prototipos.
create-drop	✓	✓ (al final)	✗	✗	Pruebas unitarias.
update	✓ (si faltan)	✗	✓ (incremental)	✗	Desarrollo continuo.
validate	✗	✗	✗	✓	Producción (solo validación).
none	✗	✗	✗	✗	Producción (manual).

Ejemplo: Crear un proyecto de prueba en eclipse STS con Spring Starter Project:



Service URL	https://start.spring.io		
Name	Ejemplo		
<input type="checkbox"/> Use default location			
Location	C:\CEU\Spring\SPRING\Tema 4. Ejercicios JPA\Ejemplo		Browse
Type:	Maven	Packaging:	Jar
Java Version:	17	Language:	Java
Group	com.ej01		
Artifact	ej01		
Version	0.0.1-SNAPSHOT		
Description	Ejercicio 01		
Package	com.ej01		

Añadimos Spring Data JPA y el conector de MariaDB para que se añadan las dependencias de Hibernate y el Driver.

New Spring Starter Project Dependencies

Spring Boot Version: ▾

Frequently Used:

<input checked="" type="checkbox"/> MariaDB Driver	<input checked="" type="checkbox"/> Spring Boot DevTools	<input checked="" type="checkbox"/> Spring Data JPA
<input checked="" type="checkbox"/> Spring Web	<input type="checkbox"/> Thymeleaf	

Available:

x

▼ Developer Tools

☒ Lombok

Selected:

- X Spring Boot DevTools
- X Lombok
- X Spring Data JPA
- X MariaDB Driver
- X Spring Web

Actualizamos dependencias, botón derecho sobre el proyecto : Maven / Update Project
Para compilar por consola: **mvn clean package -U**

Configurar la base de datos

- 1) En MySQL Workbench o MariaDB creamos una BD de prueba. Por ejemplo: hibernate

```
CREATE DATABASE hibernate;
```

```
show databases;  
use hibernate;
```

- 2) Crear usuario

```
CREATE USER 'mi_usuario'@'localhost' IDENTIFIED BY 'mi_contraseña';  
GRANT ALL PRIVILEGES ON nombre_BD.* TO 'mi_usuario'@'localhost';  
FLUSH PRIVILEGES;
```

Estructura en src/main/java:

- **paquete controller**
 - clase controller: Ej ClienteController.java
- **paquete dao:**
 - interfaz dao: Ej ClienteDAO.java
- **paquete modelo:** Clases que se mapean: Ej Cliente.java
- **paquete servicio:**
 - interfaz servicio: Ej ClienteService.java
 - clase servicio: Ej ClienteServiceImpl.java

Probar aplicación

Usaremos la interfaz CommandLineRunner para poder ejecutar código después de que la aplicación se haya inicializado correctamente y antes de que el servidor comience a aceptar solicitudes HTTP.

Es necesario el método **run** que se invoca automáticamente cuando la aplicación arranca.

```
@SpringBootApplication
public class Ejercicio06Application implements CommandLineRunner {

    public static void main(String[] args) {
        SpringApplication.run(Ejercicio06Application.class, args);
    }

    @Autowired
    ClienteService clienteService;

    @Override
    public void run(String... args) throws Exception {
```