

# Cómo funciona la conexión a base de datos en Spring Boot

## ¿Qué es una base de datos?

Una base de datos es un sistema para guardar información de forma permanente y organizada. Sirve para que los datos no se pierdan cuando cerramos la aplicación. Por ejemplo, en un sistema de artículos, guardamos cada artículo en una tabla para poder recuperarlo, modificarlo o eliminarlo más adelante.

## ¿Qué herramientas usamos?

- Spring Boot: El framework principal de la aplicación.
- MySQL (con XAMPP): El sistema donde se guardan los datos.
- JPA (Java Persistence API): La forma estándar de Java para trabajar con bases de datos.
- Hibernate: La implementación interna de JPA.
- Spring Data JPA: Permite acceder a datos con métodos listos para usar.

## ¿Dónde se configura la conexión?

En el archivo application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/articulos_db  
spring.datasource.username=root  
spring.datasource.password=  
spring.jpa.hibernate.ddl-auto=update
```

Esto le dice a Spring cómo conectarse a MySQL, qué usuario usar, y qué hacer si las tablas no existen (crearlas automáticamente).

## ¿Cómo se define qué guardar?

Se usa una clase modelo, por ejemplo Articulo.java:

```
@Entity: esta clase representa una tabla.  
@Table(name = "articulo"): el nombre de la tabla.  
@Id y @GeneratedValue: indican que el campo 'id' es clave primaria y autoincremental.
```

Spring va a convertir objetos Articulo en filas de la tabla.

# Cómo funciona la conexión a base de datos en Spring Boot

## ¿Cómo se accede a los datos?

Spring usa una interfaz que extiende JpaRepository:

```
public interface ArticuloRepository extends JpaRepository<Articulo, Long> {}
```

Con esto ya podes hacer:

- findAll() - Traer todos
- findById(id) - Buscar uno
- save(objeto) - Guardar o actualizar
- deleteById(id) - Eliminar

## ¿Cómo se usa desde el código?

Desde el servicio, simplemente se llama al repositorio:

```
articuloRepository.save(articulo);
```

Spring se encarga de ejecutar la consulta SQL correcta por detras sin que el programador la escriba.

## Resumen

- Configuras la conexion en application.properties.
- Creas una clase con @Entity.
- Usas JpaRepository.
- Spring se conecta a la base, crea la tabla si no existe, y ejecuta SQL por vos.
- Solo usas metodos simples como save, findAll, etc.
- No necesitas escribir SQL para lo basico!