

ACADEMIA XIDERAL

# Inyección de dependencias y polimorfismo

GALERÍA DE ARTE

LISSET SÁNCHEZ

---

Reciclando un poco el tema del arte cree el proyecto “Galería de Arte con Polimorfismo e Inyección de Dependencias” lo pensé así, para que en un futuro talvez pueda unir este proyecto con el CRUD de pintores famosos, utilizando el modelo Obra como atributo dentro de mi modelo Pintor, relacionando ambos proyectos.

Este proyecto fue desarrollado para practicar y demostrar conceptos de polimorfismo e inyección de dependencias en Java utilizando Spring Boot. Consiste en una aplicación sencilla que administra obras de arte (pinturas y esculturas) en una galería, permitiendo crearlas y listarlas en memoria.

### Polimorfismo

Recordando un poco el tema del polimorfismo (del griego “muchas formas”) es la capacidad de una entidad (como una función, objeto o variable) de comportarse de diferentes maneras en distintos contextos, no sólo es la capacidad de un objeto de tomar distintas formas.

En este proyecto utilicé una clase padre abstracta llamada Obra que define un método abstracto descripcion(). Las clases Pintura y Escultura sobrescriben este método con su propia implementación, aparte de que se agregan atributos específicos en cada una: técnica o material. De esta manera, aunque trabajemos con una lista de tipo Obra, cada elemento responde de forma diferente según su tipo real.

Ejemplo:

```
Obra p1 = new Pintura("La noche estrellada", "Van Gogh", 1889, "Óleo");
```

```
Obra e1 = new Escultura("El Pensador", "Rodin", 1902, "Bronce");
```

```
System.out.println(p1.descripcion()); // Muestra info como Pintura
```

```
System.out.println(e1.descripcion()); // Muestra info como Escultura
```

### Inyección de Dependencias

Pasando ahora a la inyección de dependencias, este es un patrón de diseño que permite que los objetos no creen sus propias dependencias, sino que se las proporcione un contenedor (en el caso de mi proyecto, Spring). Al igual que con el polimorfismo, no hay que limitar el concepto, la inyección de dependencias es un principio dentro del diseño de software y existe desde antes de Spring. Spring simplemente la automatiza y facilita su uso. Acordamos que inyección de dependencias significa que una clase no crea sus dependencias, sino que recibe (o le “inyectan”) las instancias que necesita desde afuera. Esto favorece la desacoplación y facilita el mantenimiento y pruebas.

Existen 3 tipos: Inyección por constructor, Inyección por setter e Inyección por campo.

Dentro del proyecto:

- ObraRepository es la interfaz que define las operaciones sobre las obras.
- ObraRepositoryImpl es la implementación concreta usando un ArrayList en memoria.
- GaleriaServiceImpl recibe el repositorio inyectado automáticamente por Spring mediante el constructor.

Al ejecutar el proyecto, utilizando la clase PruebaGaleria que implementa CommandLineRunner, se crean dos obras (Pintura y Escultura) y se almacenan en la galería:

Pintura: La noche estrellada por Van Gogh (1889), técnica: Óleo

Escultura: El Pensador por Rodin (1902), material: Bronce