

¿ Qué es la Inyección de Dependencia?

Es suministrar a un objeto una referencia de otro(s) objeto(s) que necesite según la relación

Resuelve el problema de reutilización y modularidad entre componentes

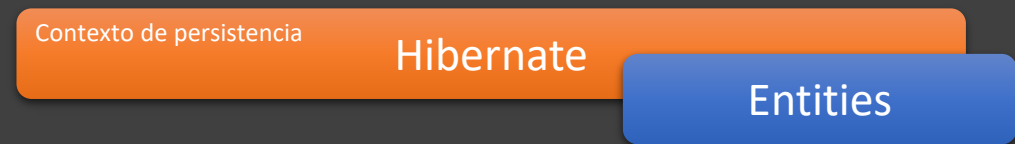
Cómo Funciona

El Contenedor se encarga de gestionar las instancias y dependencias de componentes mediante la relación e inyección de objetos

En contra-oposición de la creación explícita (operador new) de objetos

Esto permite un bajo acoplamiento entre los objetos

Tiene que plasmarse mediante la anotación @Autowired



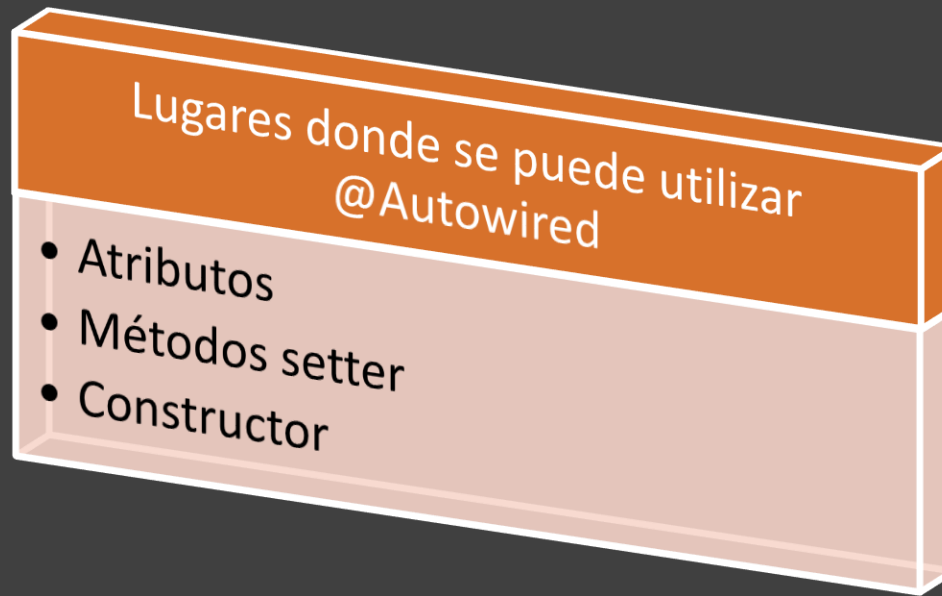
Anotación @Autowired

Especifica que se inyectará un objeto de spring (bean o componente) en un atributo de otro objeto

Es decir para inyectar un bean de spring en el componente actual

Por defecto, la inyección falla si no encuentra candidatos disponibles

Anotación @Autowired



@Autowired en atributo

```
@Component
public class UsuarioComponent {

    @Autowired
    private IUserarioService servicio;

    // ...
}
```

@Autowired en método Setter

```
@Component
public class UsuarioComponent {

    private IUserarioService servicio;

    @Autowired
    public void setUserarioService(IUserarioService servicio) {
        this.servicio = servicio;
    }

    // ...
}
```

@Autowired en constructor

```
@Component
public class UsuarioComponent {

    private IUserarioService servicio;

    @Autowired
    public UsuarioComponent(IUserarioService servicio) {
        this.servicio = servicio;
    }

    // ...
}
```

Anotación @Component y sus derivados

@Component es un estereotipo genérico para cualquier componente manejado por Spring

@Repository, @Service, y @Controller son especializaciones de @Component para usos más específicos:

- @Repository para persistencia, componentes que acceden a los datos (DAOs)
- @Service para servicios de lógica de negocio
- @Controller para controladores MVC

Componentes o beans

El término “bean” se utiliza para referirse a cualquier componente manejado por Spring

Los beans deben ser anotado con la anotación `@Component` o sus derivados

Debe tener un constructor vacío por defecto, sin argumentos

Cualquier beans anotado con `@Component` (o derivados) bajo el package base serán instanciados y manejados por el contenedor DI de Spring (Auto-scanning)

Anotaciones

@Configuration y @Bean

Una clase anotada con @Configuration indica que la clase puede ser utilizada por el contenedor Spring como una fuente de definiciones beans

```
@Configuration
public class AppConfig {

    @Bean
    public IUserarioService registrarUsuarioService() {
        return new UsuarioServiceImpl();
    }
}
```

@Configuration y @Bean

La anotación @Bean juega el mismo papel que anotar la clase con @Component

Lo definido anteriormente en la clase AppConfig, es equivalente a:

```
@Component
public class UsuarioService implements IUsuarioService {

}
```
