



DESARROLLO DE SOFTWARE I

Eric Gustavo Coronel Castillo

gcoronelc.blogspot.com

ecoronel@uch.edu.pe



TEMA: Spring Core con Anotaciones



En esta sesión
comprenderás como
funciona Spring Framework
con anotaciones.

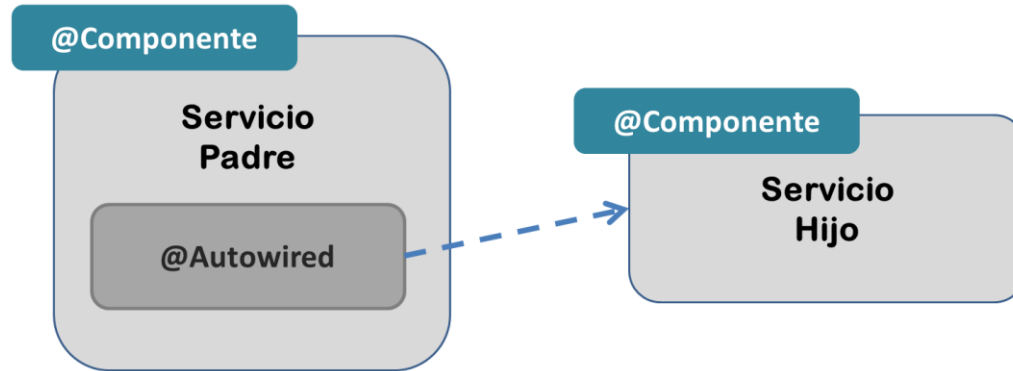




Contexto

Usar anotaciones dentro de las clases permite que la configuración y la implementación estén en un único sitio.

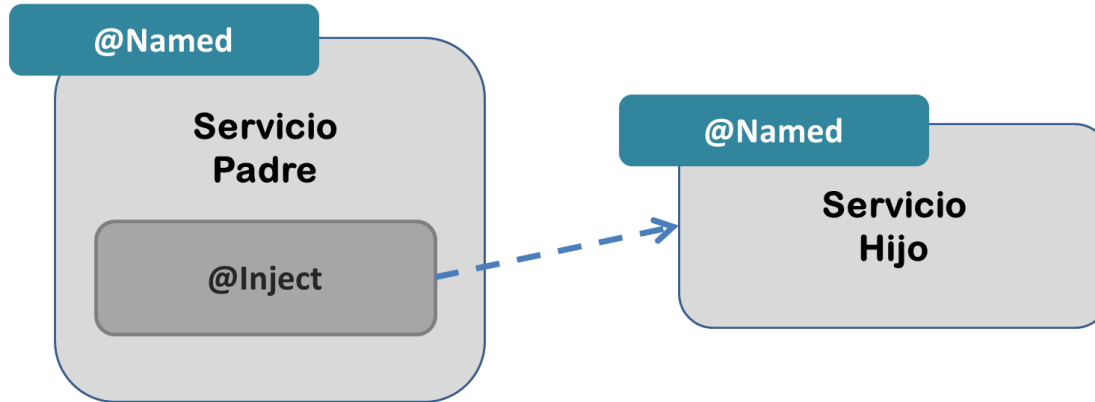
Spring soporta que toda la configuración esté en un fichero XML, que se haga todo en forma de anotaciones, o que se mezclen ambos formatos.





Anotaciones Estándares de Java EE

La competencia entre los estándares de Java EE y el Spring Framework es cada vez más dura ya que las similitudes entre ambos son muchas. Elegir uno u otro depende de muchas cosas.





Configuración

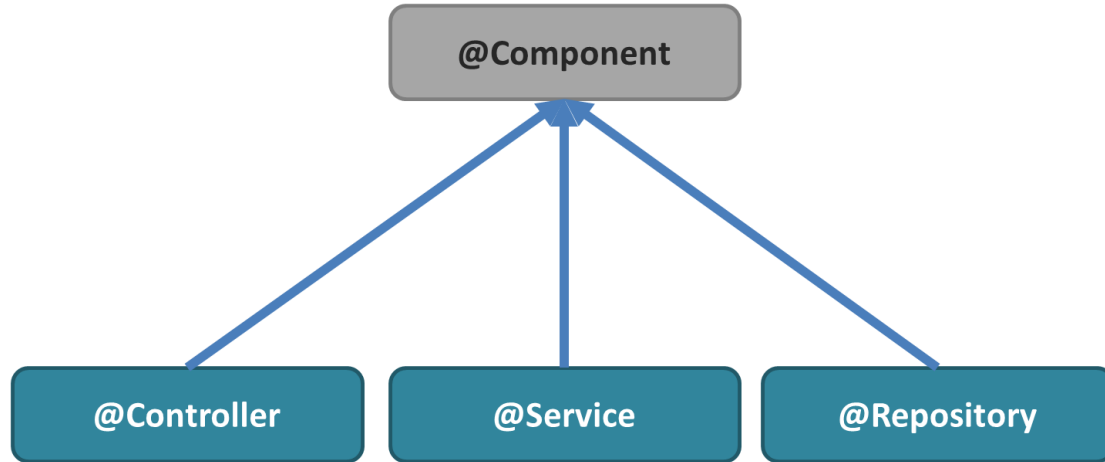
```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.1.xsd">

  <context:annotation-config />
  <context:component-scan base-package="pe.egcc.app.beans"/>

</beans>
```



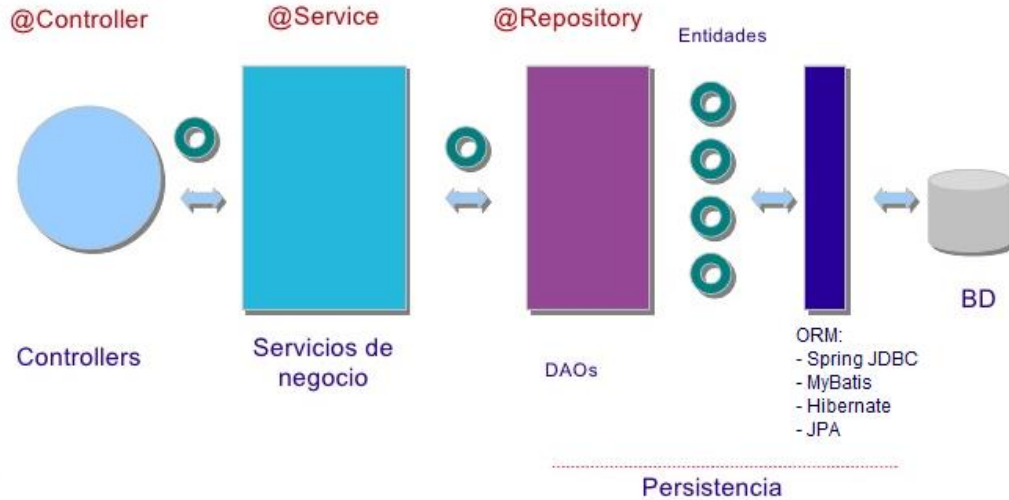
Anotaciones de Beans





Arquitectura

Arquitectura con Spring





@Autowired

```
// Sobre el método setter
@Autowired
public void setMotor(Motor motor) {
    this.motor = motor;
}

// Sobre una propiedad
@Autowired
private Motor motor;
```



@Autowired

```
// Sobre un constructor
@Autowired
public Coche(Motor motor) {
    this.motor = motor;
}

// Sobre un método
@Autowired
public void montaje(Motor motor, Volante volante) {
    this.motor = motor;
    this.volante = volante;
}
```

Configuración

Para que puedas utilizar estas anotaciones es necesario añadir la librería respectiva.

Es necesario que añadas la dependencia en el fichero pom.xml:

```
<dependency>
  <groupId>javax.inject</groupId>
  <artifactId>javax.inject</artifactId>
  <version>1</version>
</dependency>
```

@Named

Esta anotación es similar a @Component.

```
import javax.inject.Named;

@Named
public class Motor {
    . . .
    . . .
}
```

@Inject

Esta anotación tiene el mismo comportamiento que la anterior @Autowired, aunque carece del parámetro required.

```
import javax.inject.Named;
import javax.inject.Inject;

@Named
public class Coche {

    @Inject
    private Motor motor;

}
```