



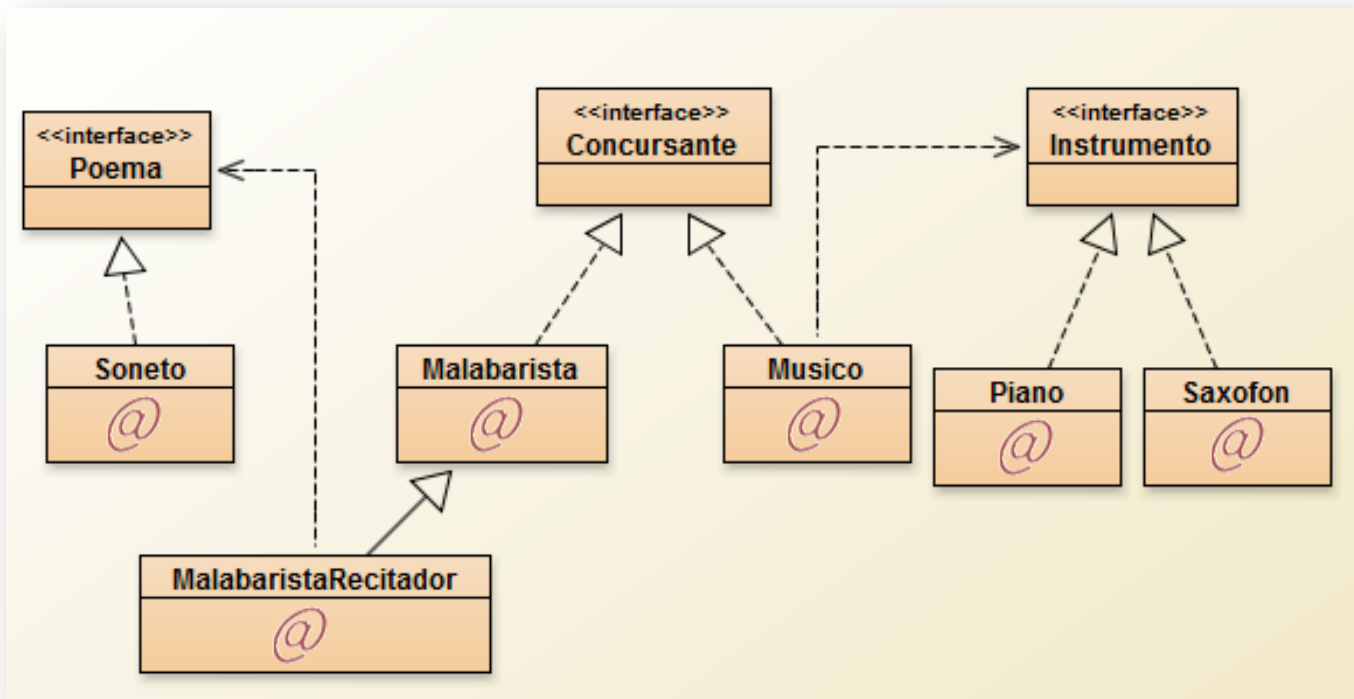
## Ejercicio

Proyecto Concurso de Talentos v4

## Objetivo del Ejercicio

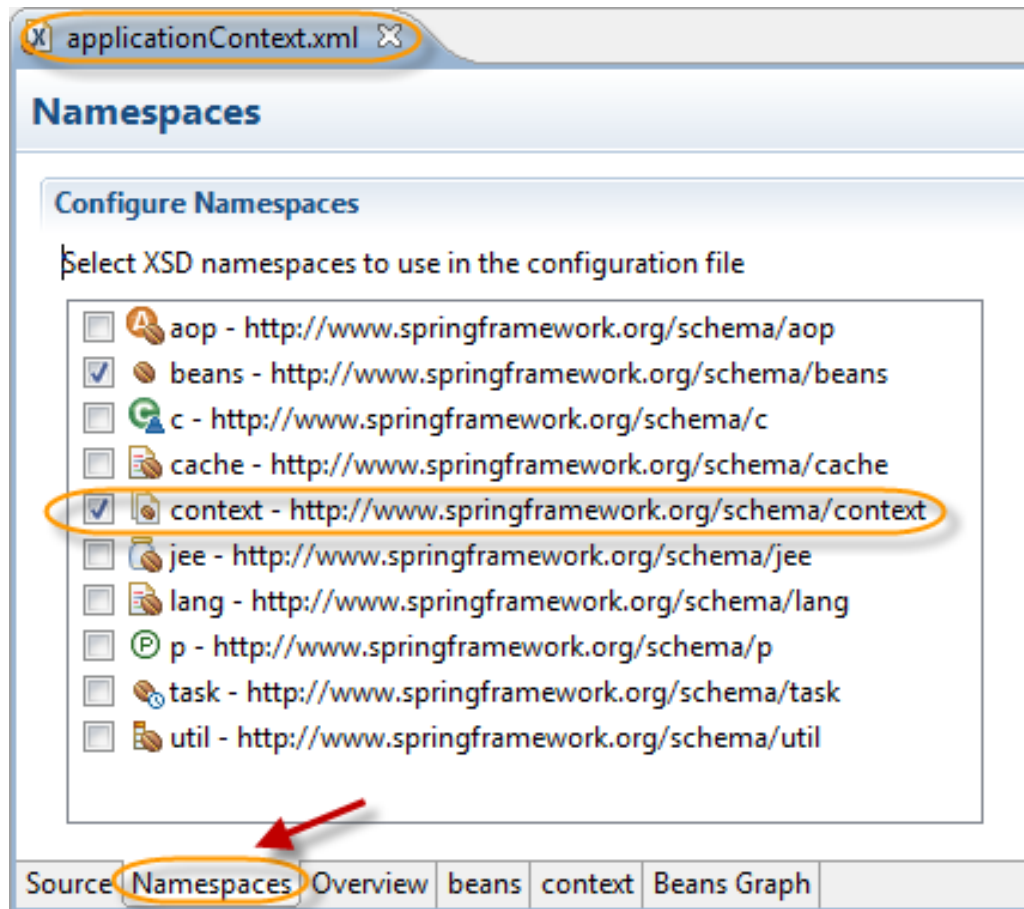
- El objetivo del ejercicio es modificar el proyecto de Concurso de Talentos para poner en práctica el concepto del Uso de Anotaciones con Spring.

Al finalizar deberemos tener el Proyecto Concurso Talentos con las siguientes clases, pero cada clase utilizando anotaciones.



## Paso 1. Modificación Archivo applicationContext.xml

Modificaremos el archivo para agregar el concepto de autodiscovering:



## Paso 1. Modificación Archivo applicationContext.xml (cont)

Deberemos poner entre comentarios cada uno de los beans (excepto el bean de “*JasonSax*”) definidos en el archivo de applicationContext, y agregamos la definición de autodiscovering:

```
<context:component-scan base-package="concursantes" />
```

El resultado es similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context"
       >

    <context:component-scan base-package="concursantes" />

    <!-- Código equivalente en Java:
    Malabarista concursante1 = new Malabarista(); -->
    <!--
    <bean id="solei" class="concursantes.Malabarista" >
        <constructor-arg value="10" />
    </bean>
    -->
    <!-- Código Solei Recitador -->
    <!--
    -->
    <!-- cambio de instrumento -->

    <bean id="jasonSax" class="concursantes.Musico">
        <property name="cancion" value="Equinox"/>
        <property name="instrumento">
            <bean class="concursantes.Saxofon"/>
        </property>
    </bean>

</beans>
```

Es el único que queda configurado de la forma clásica

## Paso 2. Refactorizar la Clase Malabarista

Agregar la anotación `@Component` a la clase `Malabarista.java`, el código queda como sigue:

```
package concursantes;

import org.springframework.stereotype.Component;

@Component("solei")
public class Malabarista implements Concursante {

    private int pelotas = 10;

    public Malabarista() {
    }

    public Malabarista(int pelotas) {
        this.pelotas = pelotas;
    }

    public void ejecutar() throws EjecucionException {
        System.out.println("Malabareo " + pelotas + " pelotas");
    }

    public int getPelotas() {
        return pelotas;
    }

    public void setPelotas(int pelotas) {
        this.pelotas = pelotas;
    }
}
```

## Paso 3. Refactorizar la Clase MalabaristaRecitador

Agregar la anotación `@Component` a la clase `MalabaristaRecitador.java`, el código queda como sigue:

```
package concursantes;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component("soleiRecitador")
public class MalabaristaRecitador extends Malabarista {

    private Poema poema;

    public MalabaristaRecitador(Poema poema) {
        super();
        this.poema = poema;
    }

    //Inyeccion automática por constructor, inyectamos valores primitivos y referencias
    @Autowired
    public MalabaristaRecitador(@Value("15") int pelotas, Poema poema) {
        super(pelotas);
        this.poema = poema;
    }

    public void ejecutar() throws EjecucionException {
        super.ejecutar();
        System.out.println("Mientras recita...");
        poema.recitar();
        System.out.println("Termina recitación...");
    }
}
```



## Paso 4. Refactorizar la Clase Musico

Agregar la anotación @Component a la clase Musico.java:

```
package concursantes;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component("jasonPiano")
public class Musico implements Concursante {

    //Inyección de valor primitivo (podemos eliminar el método set)
    @Value("Noche de Paz")
    private String cancion;
    //Inyección automática por propiedad (podemos eliminar el método set)
    @Autowired
    @Qualifier("piano")
    private Instrumento instrumento;

    public Musico() {
    }

    public void ejecutar() throws EjecucionException {
        System.out.println("Tocando " + cancion + ": ");
        instrumento.tocar();
    }

    public String getCancion() {
        return cancion;
    }

    public void setCancion(String cancion) {
        this.cancion = cancion;
    }

    public Instrumento getInstrumento() {
        return instrumento;
    }

    public void setInstrumento(Instrumento instrumento) {
        this.instrumento = instrumento;
    }
}
```



## Paso 5. Refactorizar la Clase Piano

Agregar la anotación `@Component` a la clase `Piano.java`, el código queda como sigue:

```
package concursantes;

import org.springframework.stereotype.Component;

@Component
public class Piano implements Instrumento {

    public void tocar() {
        System.out.println("Clin clin clin clin...");
    }
}
```





## Paso 6. Refactorizar la Clase Saxofón

Agregar la anotación `@Component` a la clase `Saxofon.java`, el código queda como sigue:

```
package concursantes;

import org.springframework.stereotype.Component;

@Component
public class Saxofon implements Instrumento {

    public void tocar() {
        System.out.println("Tuu tuu tuu tuu...");
    }
}
```

## Paso 7. Refactorizar la Clase Soneto

Agregar la anotación `@Component` a la clase `Soneto.java`, el código queda como sigue:

```
package concursantes;

import org.springframework.stereotype.Component;

@Component("poema")
public class Soneto implements Poema {

    public void recitar() {
        // Implementación del soneto
        String sonetoPasionSorJuana =
            "Este, que ves, engaño colorido, \n"
            + "que del arte ostentando los primores,\n"
            + "con falsos silogismos de colores\n"
            + "es cauteloso engaño del sentido; \n"
            + "éste, en quien la lisonja ha pretendido\n"
            + "excusar de los años los horrores, \n"
            + "y venciendo del tiempo los rigores \n"
            + "triunfar de la vejez y del olvido, \n"
            + "es un vano artificio del cuidado, \n"
            + "es una flor al viento delicada, \n"
            + "es un resguardo inútil para el hado, \n"
            + "es una necia diligencia errada, \n"
            + "es un afán caduco y, bien mirado, \n"
            + "es cadáver, es polvo, es sombra, es nada. \n";

        System.out.println("\nSoneto:" + sonetoPasionSorJuana);
    }
}
```

## Paso 8: Modificar la clase TestConcursoTalentos

- Remplazar el código de la clase TestConcursoTalentos.java por el siguiente:

```
package test;

import static org.junit.Assert.assertEquals;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import concursantes.Concursante;
import concursantes.Malabarista;
import concursantes.Musico;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations =
{"classpath:applicationContext.xml"})
public class TestConcursoTalentos {

    private static Log logger =
LogFactory.getLog("TestConcursoTalentos");
    @Autowired
    @Qualifier("solei")
    private Concursante malabarista1;
    @Autowired
    @Qualifier("soleiRecitador")
    private Concursante malabarista2;
    @Autowired
    @Qualifier("jasonPiano")
    private Concursante musico1;
    @Autowired
    @Qualifier("jasonSax")
    private Concursante musico2;
```

```
@Test
public void testMalabarista() {
    logger.info("Inicio de ejecutar Malabarista");

    int noPelotas = 10;
    malabarista1.ejecutar();
    assertEquals(noPelotas, ((Malabarista) malabarista1).getPelotas());

    logger.info("Fin de Ejecutar Malabarista");

    logger.info("Inicio de ejecutar MalabaristaRecitador");

    noPelotas = 15;
    malabarista2.ejecutar();
    assertEquals(noPelotas, ((Malabarista) malabarista2).getPelotas());

    logger.info("Fin de Ejecutar MalabaristaRecitador");

    logger.info("Inicio de ejecutar Pianista");

    String cancion = "Noche de Paz";
    musico1.ejecutar();
    assertEquals(cancion, ((Musico) musico1).getCancion());

    logger.info("Fin de Ejecutar Pianista");

    logger.info("Inicio de ejecutar Saxofonista");

    cancion = "Equinox";
    musico2.ejecutar();
    assertEquals(cancion, ((Musico) musico2).getCancion());

    logger.info("Fin de Ejecutar Saxofonista");
}
```

## Paso 9: Ejecutar el Test

- Ejecutamos el test como JUnit y obtenemos el siguiente resultado:

Spring - spring-concurso-talentos-01/src/test/java/test/TestConcursoTalentos.java - SpringSource Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

Finished after 0.536 seconds

Runs: 1/1 Errors: 0 Failures: 0

test.TestConcursoTalentos [Runner: JUnit 4] (0.520 s)

```
package test;

import static org.junit.Assert.assertEquals;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
```

Console

<terminated> TestConcursoTalentos [JUnit] C:\Program Files\Java\jdk1.7.0\_01\bin\javaw.exe (22/03/2012)

Malabareo 10 pelotas

547 [main] INFO TestConcursoTalentos - Fin de Ejecutar Malabarista

547 [main] INFO TestConcursoTalentos - Inicio de ejecutar MalabaristaRecita...

Malabareo 15 pelotas

Mientras recita...

547 [main] INFO TestConcursoTalentos - Inicio de ejecutar Saxofonista

Tocando Equinox:

Tuu tuu tuu tuu...

547 [main] INFO TestConcursoTalentos - Fin de Ejecutar Saxofonista

550 [Thread-0] INFO org.springframework.context.support.GenericApplicationContext

550 [Thread-0] INFO org.springframework.beans.factory.support.DefaultListable