



**Universidad Nacional Autónoma de
México**

**Dirección General de Cómputo de
Tecnologías de Información y
Comunicación**

Diplomado

Desarrollo de Sistemas con Tecnología Java

Ejercicio Doce

**“Configuración basada en Java
(@Configuration,@Bean)”**

Mtro. ISC. Miguel Ángel Sánchez Hernández

Tabla de contenido

1. Copiar un proyecto en IntelliJ IDEA	3
2. Borrado de Archivos	3
3. Clase Estudiante.....	3
4. Clase ConfiguracionServicio	4
5. Modificar la clase Inicio (versión 1)	4
6. Modificar la clase Inicio (versión 2)	5
7. Modificar la clase ConfiguracionServicio.....	6
8. Crear el archivo bean-configuration.xml.....	6
9. Clase Materia.....	7
10. Modificar la clase ConfiguracionServicio.....	7
11. Modificar la clase Inicio (versión 3)	8
12. Modificar la clase Inicio (versión 4)	9
13. Modificar la clase ConfiguracionServicio.....	10
14. Modificar la clase Inicio (versión 5)	10
15. Crear la clase ConfiguracionServicioDos	11
16. Crear el archivo bean-servicios.xml	11
17. Configurar ConfiguracionServicio.....	12
18. Modificar la clase Inicio (versión 6)	12

1. Copiar un proyecto en IntelliJ IDEA

Para copiar un proyecto, hacemos lo siguiente:

1. Señalamos el proyecto spring-core-initdestroy
2. Apretamos Ctrl + C
3. Luego Ctrl + V
4. New Name: spring-core-javabean
5. En el archivo pom.xml cambiar las siguientes líneas
 - `<artifactId>spring-core-initdestroy</artifactId>` por lo siguiente
 - `<artifactId>spring-core-javabean</artifactId>`
 - Cambiar las etiquetas `<name>` con
 - `<name> spring-core-javabean </name>`
 - Cambiar la etiqueta `<description>` con
 - `<description> Ejemplo de configuración en Java con Spring </description>`
6. Actualizar maven

2. Borrado de Archivos

Borrar los archivos de las carpetas (**no las carpetas**):

- dgtic.core.modelo
- dgtic.core.servicio
- dgtic.core.xml

3. Clase Estudiante

Crear la clase Estudiante con los siguientes datos:

- Nombre de la clase: Estudiante
- Paquete: dgtic.core.modelo

```
package dgtic.core.modelo;
public class Estudiante {
    private String nombre;
    private int edad;
    public Estudiante() {
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public int getEdad() {
        return edad;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }
    @Override
    public String toString() {
        return "Estudiante [nombre=" + nombre + ", edad=" + edad + "]";
    }
}
```

4. Clase ConfiguracionServicio

Crear la clase ConfiguracionServicio con los siguientes datos:

- Nombre de la clase: ConfiguracionServicio
- Paquete: dgtic.core.config

```
package dgtic.core.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import dgtic.core.modelo.Estudiante;
@Configuration
public class ConfiguracionServicio {
    @Bean(name = "estudiante")
    public Estudiante servicioEstudiante() {
        Estudiante est=new Estudiante();
        est.setNombre("Pedro");
        est.setEdad(13);
        return est;
    }
}
```

5. Modificar la clase Inicio (versión 1)

Modificar la clase Inicio como sigue:

```
package dgtic.core;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import dgtic.core.modelo.Estudiante;
import dgtic.core.config.ConfiguracionServicio;

public class Inicio {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                (ConfiguracionServicio.class);
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        System.out.println(estudiante.toString());
        contexto.close();
    }
}
```

Correr la aplicación. Comprobaremos que tengo beans de alcance singleton.

6. Modificar la clase Inicio (versión 2)

Modificar la clase Inicio como sigue:

```
package dgtic.core;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import dgtic.core.modelo.Estudiante;
import dgtic.core.config.ConfiguracionServicio;

public class Inicio {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                (ConfiguracionServicio.class);
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteDos = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        //cambiar propiedad a estudianteDos
        estudianteDos.setNombre("Martha");
        System.out.println("-----");
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        contexto.close();
    }
}
```

Correr la aplicación. Como podemos notar tenemos beans de alcance singleton, para cambiar a prototype, haremos uso de la notación @Scope

7. Modificar la clase ConfiguracionServicio

Modificamos la clase ConfiguracionServicio como sigue:

```
package dgtic.core.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Scope;
import dgtic.core.modelo.Estudiante;
@Configuration
public class ConfiguracionServicio {
    @Bean(name = "estudiante")
    @Scope("prototype")
    public Estudiante servicioEstudiante() {
        Estudiante est=new Estudiante();
        est.setNombre("Pedro");
        est.setEdad(13);
        return est;
    }
}
```

Correr la aplicación. Como podemos notar ahora los beans ya no son singleton. En la siguiente parte mezclaremos las configuraciones basadas en Java y XML.

8. Crear el archivo bean-configuration.xml

Crear un archivo XML con las siguientes características:

- Nombre del archivo xml: bean-configuration.xml
- Dirección:resources

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       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.xsd">

    <!-- bean definitions here -->
    <bean id="materia" class="dgtic.core.modelo.Materia">
        <property name="nombre" value="Cálculo"/>
        <property name="creditos" value="10"/>
    </bean>
</beans>
```

9. Clase Materia

Crear la clase Materia con los siguientes datos:

- Nombre de la clase: Materia
- Paquete: dgtic.core.modelo

```
package dgtic.core.modelo;
public class Materia {
    private String nombre;
    private Integer credits;
    public Materia() {
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public Integer getCredits() {
        return credits;
    }
    public void setCredits(Integer credits) {
        this.credits = credits;
    }
    @Override
    public String toString() {
        return "Materia [nombre=" + nombre + ", credits=" + credits + "]";
    }
}
```

10. Modificar la clase ConfiguracionServicio

Modificamos la clase ConfiguracionServicio como sigue:

```
package dgtic.core.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Scope;
import dgtic.core.modelo.Estudiante;
@Configuration
@ImportResource(locations = {"classpath:bean-configuration.xml"})
public class ConfiguracionServicio {
    @Bean(name = "estudiante")
    @Scope("prototype")
    public Estudiante servicioEstudiante() {
        Estudiante est=new Estudiante();
        est.setNombre("Pedro");
        est.setEdad(13);
        return est;
    }
}
```

11. Modificar la clase Inicio (versión 3)

Modificamos la clase Inicio como sigue:

```
package dgtic.core;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import dgtic.core.modelo.Estudiante;
import dgtic.core.modelo.Materia;
import dgtic.core.config.ConfiguracionServicio;

public class Inicio {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                (ConfiguracionServicio.class);
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteDos = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        //cambiar propiedad a estudianteDos
        estudianteDos.setNombre("Martha");
        System.out.println("-----");
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        System.out.println("-----");
        Materia materia=(Materia)contexto.getBean("materia");
        System.out.println(materia.toString());
        contexto.close();
    }
}
```

Correr la aplicación. Ya tenemos las dos formas de configuración XML y Java conviviendo en nuestra aplicación. Por último, le indicaremos a Spring que descubra los beans que se crean en el contexto de aplicación.

12. Modificar la clase Inicio (versión 4)

Modificamos la clase Inicio como sigue:

```
package dgtic.core;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import dgtic.core.modelo.Estudiante;
import dgtic.core.modelo.Materia;
import dgtic.core.servicio.ConfiguracionServicio;

public class Inicio {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                ("dgtic.core");
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteDos = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        //cambiar propiedad a estudianteDos
        estudianteDos.setNombre("Martha");
        System.out.println("-----");
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        System.out.println("-----");
        Materia materia=(Materia)contexto.getBean("materia");
        System.out.println(materia.toString());
        contexto.close();
    }
}
```

Correr la aplicación. Spring crea todos los beans que se encuentran en el contexto de la aplicación, que en nuestro caso es el paquete base “**dgtic.core**”. Todavía Spring ofrece escaneos más especializados con la anotación `@ComponentScan`. El código siguiente lo muestra.

13. Modificar la clase ConfiguracionServicio

Modificamos la clase ConfiguracionServicio como sigue:

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.ImportResource;
import org.springframework.context.annotation.Scope;

import dgtic.core.modelo.Estudiante;

@Configuration
@ImportResource(locations={"classpath:/src/main/java/dgtic/core/xml/*.xml"})
@ComponentScan(basePackages = "dgtic.core")
public class ConfiguracionServicio {
    @Bean(name = "estudiante")
    @Scope("prototype")
    public Estudiante servicioEstudiante() {
        Estudiante est=new Estudiante();
        est.setNombre("Pedro");
        est.setEdad(13);
        return est;
    }
}
```

14. Modificar la clase Inicio (versión 5)

Modificamos la clase Inicio como sigue:

```
package dgtic.core.inicio;
import dgtic.core.config.ConfiguracionServicio;
import dgtic.core.modelo.Estudiante;
import dgtic.core.modelo.Materia;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class InicioCinco {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                (ConfiguracionServicio.class);
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteDos = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        //cambiar propiedad a estudianteDos
        estudianteDos.setNombre("Martha");
        System.out.println("-----");
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        System.out.println("-----");
        Materia materia=(Materia)contexto.getBean("materia");
        System.out.println(materia.toString());
        contexto.close();
    }
}
```

Correr la aplicación

15. Crear la clase ConfiguracionServicioDos

Crear la clase ConfiguracionServicioDos con los siguientes datos:

- Nombre de la clase: ConfiguracionServicioDos
- Paquete: dgtic.core.config

```
import dgtic.core.modelo.Estudiante;
import dgtic.core.modelo.Materia;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Scope;

@Configuration
public class ConfiguracionServicioDos {
    @Bean(name = "estudianteTres")
    @Scope("prototype")
    public Estudiante servicioEstudiante(){
        Estudiante estudiante=new Estudiante();
        estudiante.setNombre("Miguel");
        estudiante.setEdad(46);
        return estudiante;
    }
}
```

16. Crear el archivo bean-servicios.xml

Crear un archivo XML con las siguientes características:

- Nombre del archivo xml: bean-servicios.xml
- Dirección: resources

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       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.xsd">

    <!-- bean definitions here -->
    <bean id="materiaDos" class="dgtic.core.modelo.Materia">
        <property name="nombre" value="Programación" />
        <property name="creditos" value="10" />
    </bean>
</beans>
```

17. Configurar ConfiguracionServicio

Configurar la clase ConfiguracionServicio.java

```
@Configuration
@ImportResource(locations = {"classpath:bean-configuration.xml",
    "classpath:bean-services.xml"})
@Import({ConfiguracionServicioDos.class})
public class ConfiguracionServicio {
    @Bean(name = "estudiante")
    @Scope("prototype")
    public Estudiante servicioEstudiante(){
        Estudiante estudiante=new Estudiante();
        estudiante.setNombre("Manuel");
        estudiante.setEdad(46);
        return estudiante;
    }
}
```

18. Modificar la clase Inicio (versión 6)

Modificamos la clase Inicio como sigue:

```
package dgtic.core;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import dgtic.core.modelo.Estudiante;
import dgtic.core.modelo.Materia;
import dgtic.core.config.ConfiguracionServicio;
public class Inicio {
    public static void main(String[] args) {
        ConfigurableApplicationContext contexto =
            new AnnotationConfigApplicationContext
                (ConfiguracionServicio.class);
        Estudiante estudiante = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteDos = (Estudiante) contexto.getBean("estudiante", Estudiante.class);
        Estudiante estudianteTres = (Estudiante) contexto.getBean("estudianteTres", Estudiante.class);
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        System.out.println(estudianteTres.toString());
        //cambiar propiedad a estudianteDos
        estudianteDos.setNombre("Martha");
        System.out.println("-----");
        System.out.println(estudiante.toString());
        System.out.println(estudianteDos.toString());
        System.out.println("-----");
        Materia materia=(Materia)contexto.getBean("materia");
        Materia materiaDos=(Materia)contexto.getBean("materiaDos");
        System.out.println(materia.toString());
        System.out.println(materiaDos.toString());
        contexto.close();
    }
}
```

Correr la aplicación