



**Universidad Nacional Autónoma de  
México**

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

**Diplomado**

**Desarrollo de Sistemas con Tecnología Java**

**Ejercicio Siete**

**“Autowire”**

**Mtro. ISC. Miguel Ángel Sánchez Hernández**

## Tabla de contenido

1.	Copiar un proyecto en IntelliJ IDEA .....	3
2.	Modificar la clase Empleado .....	3
3.	Archivo de configuración bean-configuration.xml (byName) .....	4
4.	Clase Inicio .....	4
5.	Archivo de configuración bean-configuration.xml (no hay propiedad) .....	5
6.	Archivo de configuración bean-configuration.xml (byType) .....	5
7.	Archivo de configuración bean-configuration.xml (más de una propiedad) .....	6
8.	Modificar la clase Empleado .....	6
9.	Archivo de configuración bean-configuration.xml (constructor) .....	6
10.	Archivo de configuración bean-configuration.xml (más de una propiedad) .....	7

## 1. Copiar un proyecto en IntelliJ IDEA

Para copiar un proyecto, hacemos lo siguiente:

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

## 2. Modificar la clase Empleado

Se cambia la clase Empleado como se indica abajo:

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

### 3. Archivo de configuración bean-configuration.xml (byName)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
        autowire="byName">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividad" class="dgtic.core.modelo.GestionProyectos"/>
  <bean id="actividadNueva" class="dgtic.core.modelo.GestionProyectos"/>
</beans>
```

### 4. Clase Inicio

Modificamos la clase Inicio como sigue:

```
package dgtic.core.inicio;
import org.springframework.beans.factory.UnsatisfiedDependencyException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import dgtic.core.modelo.Empleado;
public class Inicio {
    public static void main(String[] args) {
        ApplicationContext contexto = null;
        try {
            contexto = new ClassPathXmlApplicationContext(new String[] { "/src/main/java/dgtic/core/xml/bean-configuration.xml" });
            Empleado emp = (Empleado) contexto.getBean("empleado");
            System.out.println(emp.toString());
            if (emp.getActividad() != null) {
                emp.getActividad().realiza();
            } else {
                System.out.println(emp.getActividad());
            }
            System.out.println("-----");
        } catch (UnsatisfiedDependencyException ex) {
            System.out.println("No se pueden crear los beans");
        }
        if (contexto != null) {
            ((ClassPathXmlApplicationContext) contexto).close();
        }
    }
}
```

Correar la aplicación.

## 5. Archivo de configuración bean-configuration.xml (no hay propiedad)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
    autowire="byName">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividadCero" class="dgtic.core.modelo.GestionProyectos"/>
  <bean id="actividadNueva" class="dgtic.core.modelo.GestionProyectos"/>
</beans>
```

Correr la aplicación. La salida ahora para actividades es nula, dado que no encontró ningún bean que se llame igual a la propiedad.

## 6. Archivo de configuración bean-configuration.xml (byType)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
    autowire="byType">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividadCero" class="dgtic.core.modelo.GestionProyectos"/>
</beans>
```

Correr la aplicación. El bean es encontrado dado que se encontró un bean con un tipo que corresponde.

## 7. Archivo de configuración bean-configuration.xml (más de una propiedad)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
    autowire="byType">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividadCero" class="dgtic.core.modelo.GestionProyectos"/>
  <bean id="actividadNueva" class="dgtic.core.modelo.GestionProyectos"/>
</beans>
```

Correr la aplicación. Marca una excepción `UnsatisfiedDependencyException`, porque hay más de un bean del mismo tipo.

## 8. Modificar la clase Empleado

De la clase `Empleado`, quitar el constructor vacío.

## 9. Archivo de configuración bean-configuration.xml (constructor)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
    autowire="constructor">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividadCero" class="dgtic.core.modelo.GestionProyectos"/>
</beans>
```

Correr la aplicación.

## 10. Archivo de configuración bean-configuration.xml (más de una propiedad)

Modificamos el archivo bean-configuration.xml

```
<?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="empleado" class="dgtic.core.modelo.Empleado"
        autowire="constructor">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
  </bean>
  <bean id="actividadCero" class="dgtic.core.modelo.GestionProyectos"/>
  <bean id="actividadNueva" class="dgtic.core.modelo.PlanificacionProyectos"/>
</beans>
```

Correr la aplicación. Marca una excepción `UnsatisfiedDependencyException`, porque hay más de un bean con los parámetros de constructor.