



**Universidad Nacional Autónoma de  
México**

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

**Diplomado**

**Desarrollo de Sistemas con Tecnología Java**

**Ejercicio Cinco**

**“Inyectar por constructores, propiedades y beans  
internos”**

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

## Tabla de contenido

1. Copiar un proyecto en IntelliJ IDEA .....	3
2. Borrar archivos .....	3
3. Interfaz Actividades .....	3
4. Clase PlanificacionProyectos.....	3
5. Clase GestionProyectos .....	4
6. Clase IntercambioConocimientos .....	4
7. Clase Empleado .....	5
8. Archivo de configuración bean-configuration.xml.....	6
9. Clase Inicio .....	7

## 1. Copiar un proyecto en IntelliJ IDEA

Para copiar un proyecto, hacemos lo siguiente:

1. Señalamos el proyecto spring-core-applicationcontext
2. Apretamos Ctrl + C
3. Luego Ctrl + V
4. New Name: spring-core-beans
5. File→Open, buscamos el nuevo proyecto
6. En el archivo pom.xml cambiar las siguientes líneas
  - `<artifactId>spring-core-applicationcontext</artifactId>` por lo siguiente
    - `<artifactId>spring-core-beans</artifactId>`
  - Cambiar la etiqueta `<name>` con
    - `<name>spring-core-beans</name>`
  - Cambiar la etiqueta `<description>` con
    - `<description>Ejemplo de Beans con Spring</description>`
7. Actualizamos maven

## 2. Borrar archivos

Borramos los siguientes archivos del nuevo proyecto.

- Persona.java
- Servicio.java
- beans.services.xml

## 3. Interfaz Actividades

Creamos la interfaz Actividades con las siguientes características.

- Nombre de la interfaz: Actividades
- Paquete: dgtic.core.modelo

```
package dgtic.core.modelo;
public interface Actividades {
    public void realiza();
}
```

## 4. Clase PlanificacionProyectos

Creamos la clase PlanificacionProyectos con las siguientes características.

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

```
package dgtic.core.modelo;
public class PlanificacionProyectos implements Actividades {
    @Override
    public void realiza() {
        System.out.println("Planifica Proyectos");
    }
}
```

```
}
```

## 5. Clase GestionProyectos

Creamos la clase GestionPoryectos con las siguientes características.

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

```
package dgtic.core.modelo;
public class GestionProyectos implements Actividades{
    @Override
    public void realiza() {
        System.out.println("Gestiona proyectos");
    }
}
```

## 6. Clase IntercambioConocimientos

Creamos la clase IntercambioConocimientos con las siguientes características.

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

```
package dgtic.core.modelo;
public class IntercambioConocimientos implements Actividades {
    @Override
    public void realiza() {
        System.out.println("Intercambia conocimientos");
    }
}
```

## 7. Clase Empleado

Creamos la clase Empleado con las siguientes características.

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

```
package dgtic.core.modelo;
public class Empleado {
    private String nombre;
    private Integer edad;
    private Actividades actividad;
    public Empleado() {
    }
    public Empleado(String nombre) {
        super();
        this.nombre = nombre;
    }

    public Empleado(Actividades actividad) {
        super();
        this.actividad = actividad;
    }

    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 + "]";
    }
}
```

## 8. Archivo de configuración bean-configuration.xml

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">
    <constructor-arg value="Rebeca" />
    <property name="edad" value="34" />
  </bean>
  <bean name="empleadoDos" class="dgtic.core.modelo.Empleado">
    <constructor-arg ref="gestionProyectos" />
  </bean>
  <bean name="empleadoTres" class="dgtic.core.modelo.Empleado">
    <property name="actividad" ref="intercambioConocimientos" />
  </bean>
  <bean name="empleadoCuatro" class="dgtic.core.modelo.Empleado">
    <property name="actividad">
      <bean class="dgtic.core.modelo.PlanificacionProyectos"></bean>
    </property>
  </bean>
  <bean name="empleadoCinco" class="dgtic.core.modelo.Empleado">
    <constructor-arg>
      <bean class="dgtic.core.modelo.PlanificacionProyectos"></bean>
    </constructor-arg>
    <property name="nombre" value="Maria"/>
    <property name="edad" value="45"></property>
  </bean>
  <bean id="gestionProyectos"
    class="dgtic.core.modelo.GestionProyectos" />
  <bean id="intercambioConocimientos"
    class="dgtic.core.modelo.IntercambioConocimientos" />
</beans>
```

## 9. Clase Inicio

Modificamos la clase Inicio como sigue:

```
package dgtic.core;
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=
            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();
        }
        System.out.println("-----");
        Empleado empDos=(Empleado)contexto.getBean("empleadoDos");
        System.out.println(empDos.toString());
        if(empDos.getActividad()!=null) {
            empDos.getActividad().realiza();
        }
        System.out.println("-----");
        Empleado empTres=(Empleado)contexto.getBean("empleadoTres");
        empTres.setNombre("Carlos");
        System.out.println(empTres.toString());
        if(empTres.getActividad()!=null) {
            empTres.getActividad().realiza();
        }
        System.out.println("-----");
        Empleado empCuatro=(Empleado)contexto.getBean("empleadoCuatro");
        empCuatro.setNombre("Tomas");
        System.out.println(empCuatro.toString());
        if(empCuatro.getActividad()!=null) {
            empCuatro.getActividad().realiza();
        }
        System.out.println("-----");
        Empleado empCinco=(Empleado)contexto.getBean("empleadoCinco");
        System.out.println(empCinco.toString());
        if(empCinco.getActividad()!=null) {
            empCinco.getActividad().realiza();
        }
        ((ClassPathXmlApplicationContext) contexto).close();
    }
}
```