



**Universidad Nacional Autónoma de  
México**

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

**Diplomado**

**Desarrollo de Sistemas con Tecnología Java**

**Ejercicio Seis**

**“Conectar Colecciones”**

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

## Tabla de contenido

1. Copiar un proyecto en IntelliJ IDEA .....	3
2. Modificar la clase Empleado .....	3
3. Clase EmpleadoM .....	4
4. Clase EmpleadoP .....	5
5. Archivo de configuración bean-configuration.xml .....	6
6. Clase Inicio .....	7

## 1. Copiar un proyecto en IntelliJ IDEA

Para copiar un proyecto, hacemos lo siguiente:

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

## 2. Modificar la clase Empleado

Se cambia la clase Empleado como se indica abajo:

```
package dgtic.core.modelo;
import java.util.Collection;
public class Empleado {
    private String nombre;
    private Integer edad;
    private Collection<Actividades> actividad;
    public Empleado() {
    }
    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 Collection<Actividades> getActividad() {
        return actividad;
    }
    public void setActividad(Collection<Actividades> actividad) {
        this.actividad = actividad;
    }
    @Override
    public String toString() {
        return "Empleado [nombre=" + nombre + ", edad=" + edad + "]";
    }
}
```

```
}
```

### 3. Clase EmpleadoM

Creemos la clase EmpleadoM con las siguientes características.

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

```
package dgtic.core.modelo;
import java.util.Map;
public class EmpleadoM {
    private String nombre;
    private Integer edad;
    private Map<String,Actividades> actividad;
    public EmpleadoM() {
    }
    public EmpleadoM(String nombre) {
        super();
        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 Map<String, Actividades> getActividad() {
        return actividad;
    }
    public void setActividad(Map<String, Actividades> actividad) {
        this.actividad = actividad;
    }
    @Override
    public String toString() {
        return "Empleado [nombre=" + nombre + ", edad=" + edad + "]";
    }
}
```

## 4. Clase EmpleadoP

Creamos la clase EmpleadoP con las siguientes características.

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

```
package dgtic.core.modelo;
package dgtic.core.modelo;
import java.util.Properties;
public class EmpleadoP {
    private String nombre;
    private Integer edad;
    private Properties actividad;
    public EmpleadoP() {
    }
    public EmpleadoP(String nombre) {
        super();
        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 Properties getActividad() {
        return actividad;
    }
    public void setActividad(Properties actividad) {
        this.actividad = actividad;
    }
    @Override
    public String toString() {
        return "Empleado [nombre=" + nombre + ", edad=" + edad + "]";
    }
}
```

## 5. 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">
    <property name="nombre" value="Tomas" />
    <property name="edad" value="34" />
    <property name="actividad">
      <list>
        <ref bean="gestionProyectos" />
        <ref bean="intercambioConocimientos" />
        <ref bean="planfacionProyectos" />
        <ref bean="gestionProyectos" />
      </list>
    </property>
  </bean>
  <bean id="empleadoDos" class="dgtic.core.modelo.Empleado">
    <property name="nombre" value="Alma" />
    <property name="edad" value="24" />
    <property name="actividad">
      <set>
        <ref bean="gestionProyectos" />
        <ref bean="intercambioConocimientos" />
        <ref bean="planfacionProyectos" />
        <ref bean="intercambioConocimientos" />
      </set>
    </property>
  </bean>
  <bean id="empleadoTres" class="dgtic.core.modelo.EmpleadoM">
    <property name="nombre" value="Roberto" />
    <property name="edad" value="45" />
    <property name="actividad">
      <map>
        <entry key="uno" value-ref="gestionProyectos" />
        <entry key="dos" value-ref="intercambioConocimientos" />
        <entry key="tres" value-ref="planfacionProyectos" />
      </map>
    </property>
  </bean>
  <bean id="empleadoCuatro" class="dgtic.core.modelo.EmpleadoP">
    <property name="nombre" value="Mario" />
    <property name="edad" value="19" />
    <property name="actividad">
      <props>
        <prop key="uno">"Gestion de Proyectos" </prop>
        <prop key="dos">"Planificación Proyectos" </prop>
      </props>
    </property>
  </bean>
  <bean id="gestionProyectos"
    class="dgtic.core.modelo.GestionProyectos" />
  <bean id="intercambioConocimientos"
    class="dgtic.core.modelo.IntercambioConocimientos" />
  <bean id="planfacionProyectos"
    class="dgtic.core.modelo.PlanfacionProyectos" />
</beans>
```

## 6. Clase Inicio

Modificamos la clase Inicio como sigue:

```
package dgtic.core;
import java.util.Iterator;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import dgtic.core.modelo.Actividades;
import dgtic.core.modelo.Empleado;
import dgtic.core.modelo.EmpleadoM;
import dgtic.core.modelo.EmpleadoP;
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());
        for (Actividades act : emp.getActividad()) {
            act.realiza();
        }
        System.out.println("-----");
        Empleado empDos = (Empleado) contexto.getBean("empleadoDos");
        System.out.println(empDos.toString());
        for (Actividades act : empDos.getActividad()) {
            act.realiza();
        }
        System.out.println("-----");
        EmpleadoM empTres = (EmpleadoM) contexto.getBean("empleadoTres");
        System.out.println(empTres.toString());
        for (String llave : empTres.getActividad().keySet()) {
            System.out.println("Llave: "+llave);
            empTres.getActividad().get(llave).realiza();
        }
        System.out.println("-----");
        EmpleadoP empCuatro = (EmpleadoP) contexto.getBean("empleadoCuatro");
        System.out.println(empCuatro.toString());
        for (Iterator<Object> iter=empCuatro.getActividad().keySet().iterator();
            iter.hasNext();) {
            String llave=(String) iter.next();
            System.out.println("Llave: "+llave);
            System.out.println(empCuatro.getActividad().get(llave));
        }
        ((ClassPathXmlApplicationContext) contexto).close();
    }
}
```