



**Universidad Nacional Autónoma de
México**

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

Diplomado

Desarrollo de Sistemas con Tecnología Java

Ejercicio Nueve

“Métodos de Fabrica Dos”

Mtro. ISC. Miguel Ángel Sánchez Hernández

Tabla de contenido

1. Copiar un proyecto en en IntelliJ IDEA.....	3
2. Limpiar contenido	3
3. Crear la clase Servicio	3
4. Crear la interfaz EmpleadoServicio	4
5. Crear la interfaz ReporteEmpleadoServicio	4
6. Crear la clase EmpleadoServicioImpl.....	4
7. Crear la clase ReporteEmpleadoServicioImpl	4
8. Crear la clase PredeterminadoServicio	5
9. Archivo de configuración bean-configuration.xml	5
10. Archivo de configuración bean-services.xml	6
11. Clase Inicio	7

1. Copiar un proyecto en en IntelliJ IDEA

Para copiar un proyecto, hacemos lo siguiente:

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

2. Limpiar contenido

Borrar el contenido de los paquetes:

- `dgctic.core.modelo`
- `dgctic.core.servicio`

3. Crear la clase Servicio

Creamos la clase Servicio con las siguientes características.

- Nombre de la clase: Servicio
- Paquete: `dgctic.core.servicio`

```
package dgctic.core.servicio;
import dgctic.core.modelo.Empleado;
public class Servicio {
    private static Servicio servicio=new Servicio();
    private Servicio() {
    }
    public static Servicio getInstance() {
        return servicio;
    }
}
```

4. Crear la interfaz EmpleadoServicio

Creemos la interfaz EmpleadoServicio con las siguientes características.

- Nombre de la interfaz: EmpleadoServicio
- Paquete: dgtic.core.servicio

```
package dgtic.core.servicio;  
public interface EmpleadoServicio {  
    public void servicioEmpleado();  
}
```

5. Crear la interfaz ReporteEmpleadoServicio

Creemos la interfaz ReporteEmpleadoServicio con las siguientes características.

- Nombre de la interfaz: ReporteEmpleadoServicio
- Paquete: dgtic.core.servicio

```
package dgtic.core.servicio;  
public interface ReporteEmpleadoServicio {  
    public void reporteDiario();  
}
```

6. Crear la clase EmpleadoServicioImpl

Creemos la clase EmpleadoServicioImpl con las siguientes características.

- Nombre de la clase: EmpleadoServicioImpl
- Paquete: dgtic.core.servicio

```
package dgtic.core.servicio;  
public class EmpleadoServicioImpl implements EmpleadoServicio{  
    @Override  
    public void servicioEmpleado() {  
        System.out.println("Servicio implementado");  
    }  
}
```

7. Crear la clase ReporteEmpleadoServicioImpl

Creemos la clase ReporteEmpleadoServicioImpl con las siguientes características.

- Nombre de la clase: ReporteEmpleadoServicioImpl
- Paquete: dgtic.core.servicio

```
package dgtic.core.servicio;  
public class ReporteEmpleadoServicioImpl implements ReporteEmpleadoServicio{  
    @Override  
    public void reporteDiario() {  
        System.out.println("Reporte Diario");  
    }  
}
```

8. Crear la clase PredeterminadoServicio

Creamos la clase PredeterminadoServicio con las siguientes características.

- Nombre de la clase: PredeterminadoServicio
- Paquete: dgctic.core.servicio

```
package dgctic.core.servicio;
public class PredeterminadoServicio {
    private static EmpleadoServicio empleadoServicio=new EmpleadoServicioImpl();
    private static ReporteEmpleadoServicio empleadoReporteServicio=new ReporteEmpleadoServicioImpl();
    private PredeterminadoServicio() {
    }
    public EmpleadoServicio getEmpleadoServicio() {
        return empleadoServicio;
    }
    public ReporteEmpleadoServicio getReporteEmpleadoServicio() {
        return empleadoReporteServicio;
    }
}
```

9. 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 -->

</beans>
```

10. Archivo de configuración bean-services.xml

Crear el archivo bean-services.xml, en el paquete dgtic.core.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="servicioLocal"
        class="dgtic.core.servicio.PredeterminadoServicio" />
  <bean id="empleadoServicio" factory-bean="servicioLocal"
        factory-method="getEmpleadoServicio" />
  <bean id="empleadoServicioReporte" factory-bean="servicioLocal"
        factory-method="getReporteEmpleadoServicio" />
</beans>
```

11. 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.servicio.EmpleadoServicio;
import dgtic.core.servicio.PredeterminadoServicio;
import dgtic.core.servicio.ReporteEmpleadoServicio;
import dgtic.core.servicio.Servicio;
public class Inicio {
    public static void main(String[] args) {
        ApplicationContext contexto = new ClassPathXmlApplicationContext(
            new String[] { "/src/main/java/dgtic/core/xml/bean-configuration.xml",
                          "/src/main/java/dgtic/core/xml/bean-services.xml" });

        //factory-bean
        EmpleadoServicio servUno=(EmpleadoServicio) contexto.getBean("empleadoServicio");
        servUno.servicioEmpleado();
        ReporteEmpleadoServicio servDos=(ReporteEmpleadoServicio) contexto.getBean("empleadoServicioReporte");
        servDos.reporteDiario();
        System.out.println("----");
        PredeterminadoServicio servTres=(PredeterminadoServicio) contexto.getBean("servicioLocal");
        servTres.getEmpleadoServicio().servicioEmpleado();
        servTres.getReporteEmpleadoServicio().reporteDiario();
        ((ClassPathXmlApplicationContext) contexto).close();
    }
}
```

Correar la aplicación.