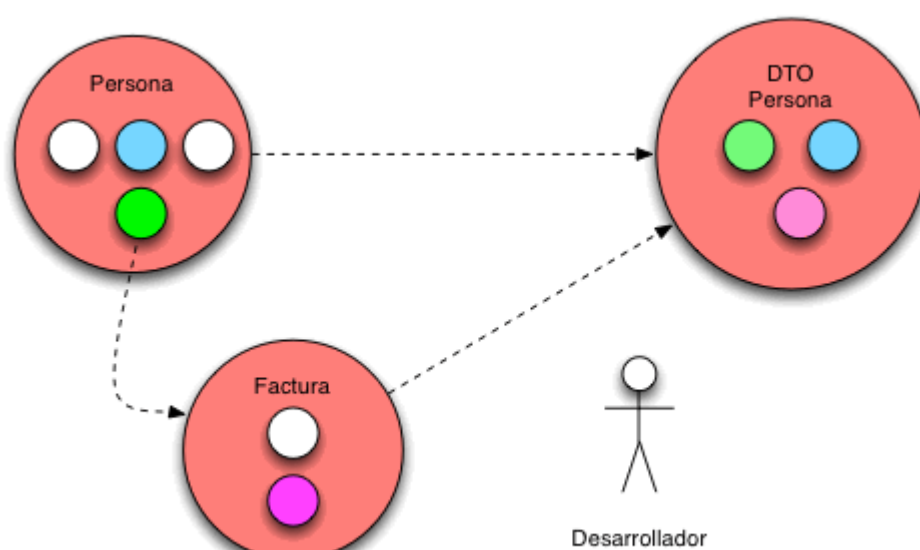
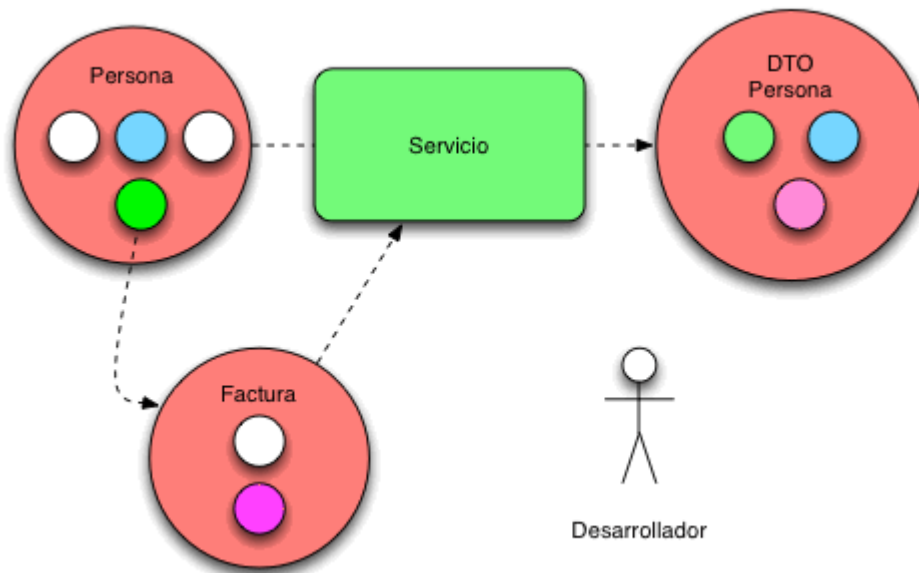


Cuando programamos en Java utilizamos JavaBeans de forma continua. Sin embargo a veces nos encontramos con algunos problemas a la hora de trabajar con ellos. Uno de los problemas más habituales es la conversión de una estructura de JavaBeans en otra estructura diferente como pueden ser los DTO. Es habitual encontrarnos con situaciones de este estilo cuando usamos frameworks de Persistencia tipo Hibernate.



Dozer

En muchas ocasiones recurrimos a una transformación manual normalmente a nivel de una clase de Servicio que terminará delegando en algún tipo de capa de presentación remota.



Sin embargo cuando la situación se repite de forma continuada puede acabar consumiéndonos mucho tiempo. Una solución a este problema es **Dozer** un framework especializado en realizar este tipo de tareas y bastante sencillo de utilizar. Imaginemonos que tenemos una estructura de clases Java como la siguiente:

```
package com.arquitecturajava;
```

```
public class PersonaDTO {
```

```
    private String nombre;
```

```
    private String edad;
```

```
    private double importe;
```

```
    public String getNombre() {
```

```
        return nombre;
```

```

    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getEdad() {
        return edad;
    }
    public void setEdad(String edad) {
        this.edad = edad;
    }
    public double getImporte() {
        return importe;
    }
    public void setImporte(double importe) {
        this.importe = importe;
    }
}

```

```

package com.arquitecturajava;

```

```

public class Persona {
    private String nombre;
    private String apellidos;
    private int edad;

    private Factura factura;

    public Factura getFactura() {
        return factura;
    }
    public void setFactura(Factura factura) {

```

```

    this.factura = factura;
}
public int getEdad() {
    return edad;
}
public void setEdad(int edad) {
    this.edad = edad;
}
public String getNombre() {
    return nombre;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getApellidos() {
    return apellidos;
}
public void setApellidos(String apellidos) {
    this.apellidos = apellidos;
}

}
package com.arquitecturajava;

public class Factura {
    private int id;
    private String concepto;
    private double importe;

    public int getId() {
        return id;
    }

```

```

}

public void setId(int id) {
    this.id = id;
}

public String getConcepto() {
    return concepto;
}

public void setConcepto(String concepto) {
    this.concepto = concepto;
}

public double getImporte() {
    return importe;
}

public void setImporte(double importe) {
    this.importe = importe;
}
}

```

Como podemos ver se trata de tres clases muy sencillas (Persona y Factura) que son JavaBeans y PersonaDTO que almacena algunos de los datos que tenemos en las Facturas. Vamos a usar Dozer para definir un fichero de Mapeo que se encargue de definir cual es el tipo de conversión.

Dozer.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<mappings xmlns="http://dozer.sourceforge.net"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://dozer.sourceforge.net
http://dozer.sourceforge.net/schema/beanmapping.xsd">

<mapping>
<class-a>com.arquitecturajava.Persona</class-a>
<class-b>com.arquitecturajava.PersonaDTO</class-b>
<field>
<a>nombre</a>
<b>nombre</b>
</field>
<field>
<a>edad</a>
<b>edad</b>
</field>
<field>
<a>factura.importe</a>
<b>importe</b>
</field>
</mapping>
</mappings>

```

El fichero de mapeo es sencillo y simplemente nos dice que el nombre y la edad son mapeados de Persona a PersonaDTO mientras que el importe de la Factura es mapeado a PersonaDTO como un campo adicional. Una vez construido el mapeo generamos un sencillo programa principal que realiza la transformación:

```

package com.arquitecturajava;

import java.util.ArrayList;
import java.util.List;

```

```
import org.dozer.DozerBeanMapper;

public class Principal {

    public static void main(String[] args) {

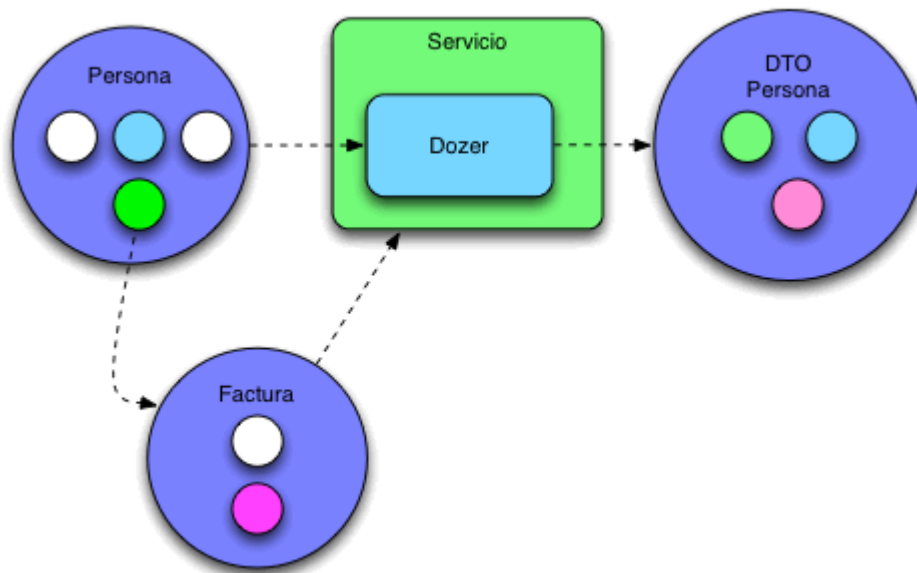
        Persona p= new Persona();
        p.setNombre("maria");
        p.setEdad(20);
        p.setApellidos("gonzalez");
        Factura f= new Factura();
        f.setId(1);
        f.setConcepto("nexus5");
        f.setImporte(400);
        p.setFactura(f);

        DozerBeanMapper mapper = new DozerBeanMapper();
        List lista = new ArrayList();
        lista.add("dozer.xml");
        mapper.setMappingFiles(lista);
        PersonaDT0 destino =
        mapper.map(p, PersonaDT0.class);
        System.out.println(destino.getNombre());
        System.out.println(destino.getEdad());
        System.out.println(destino.getImporte());

    }

}
```

De esta forma será Dozer el encargado de gestionarlo todo.



El programa nos mostrará la información del DTO por pantalla con la transformación de datos realizada correctamente.

```
Problems @ Javadoc Declaration Console Servers
<terminated> Principal (7) [Java Application] /usr/lib/jvm/java-7-openjdk-i386/bin/java (24/09/2014 16:22:15
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
maria
20
400.0
```