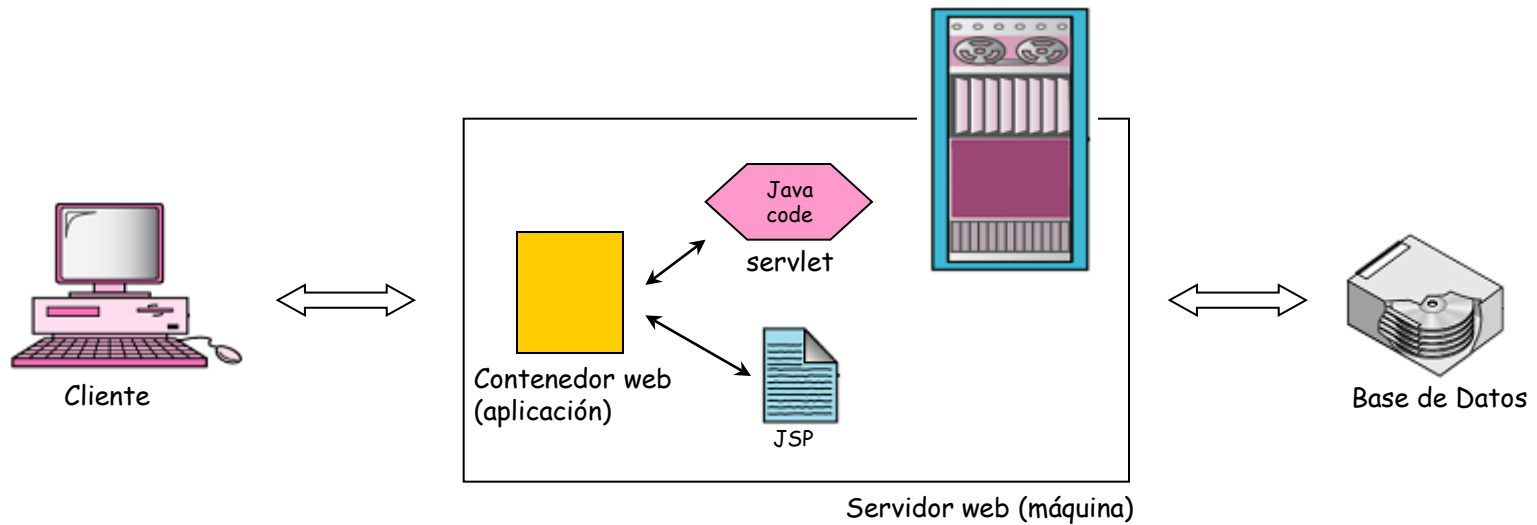


Desarrollo de Aplicaciones Web

JSP. JavaBeans.

Java Server Pages (JSP)



JSP. JavaBeans.

Limitaciones de los *scriptlets*:

- i) A medida que la página JSP se complica, los *scriptlets* insertados se hacen largos y complicados.
- ii) La funcionalidad del scriplet está ligada a la JSP para la que ha sido desarrollada.

Solucion.- mover datos y funcionalidad presentes en los *scriptlets* a componentes (*JavaBeans*), lo cual permite:

- Reutilización del código
- Separación de papeles (diseñador, desarrollador).

JSP. JavaBeans.

JavaBeans.- son componentes que exponen parte de sus datos, llamados propiedades. Los datos son almacenados por los JavaBeans y se pueden recuperar posteriormente.

JavaBeans (definición formal).- clase que mantiene algunos datos (propiedades), sigue unas determinadas convenciones de codificación y recupera/asigna valores a las propiedades mediante métodos getxxx/setxxx.

Ej.-

```
<jsp:setProperty name="miCoche" property="fabricante" value="Renault"/>
```

JSP. JavaBeans.

Construcción de un Java Bean:

```
package pbi;

public class CarBean {

    public CarBean() {
    }

    private String fabricante= "Fiat";

    public String getFabricante() {
        return fabricante;
    }

    public void setFabricante(String fabricante) {
        this.fabricante= fabricante;
    }

}
```

JSP. JavaBeans.

Uso de un JavaBean mediante *scripts*:

```
<html>
  <head>
    <title> Uso de JavaBeans </title>
  </head>
  <body>
    <% pbi.CarBean miCoche= new pbi.CarBean(); %>

    Mi coche es un <%= miCoche.getFabricante() %> <br/>

    <% miCoche.setFabricante("Renault"); %>

    Ahora mi coche es un <%= miCoche.getFabricante() %>
  </body>
</html>
```

JSP. JavaBeans.

Etiquetas JavaBean.- mecanismo de llamada a JavaBeans, mediante etiquetas análogas a las del HTML. Existen tres etiquetas:

- `<jsp:useBean>` .- localiza y hace una llamada a un JavaBean.

Ej.- `<jsp:useBean id="miCoche" class="pbi.CarBean" />`

- `<jsp:setProperty>` .- establece el valor de una propiedad de un componente.

Ej.- `<jsp:setProperty name="miCoche" property="fabricante" value="Renault" />`

- `<jsp:getProperty>` .- obtiene el valor de una propiedad almacenada en un componente.

Ej.- `<jsp:getProperty name="miCoche" property="fabricante" />`

JSP. JavaBeans.

Uso de un JavaBean mediante etiquetas:

```
<html>
  <head>
    <title> Uso de JavaBeans </title>
  </head>
  <body>
    <jsp:useBean id="miCoche" class="pbi.CarBean" scope="request"/>

    Mi coche es un <jsp:getProperty name="miCoche" property="fabricante" /><br/>

    <jsp:setProperty name="miCoche" property="fabricante" value="Renault" />

    Ahora mi coche es un <jsp:getProperty name="miCoche" property="fabricante" />
  </body>
</html>
```


JSP. JavaBeans.

<jsp:useBean>, también permite la creación de un *bean* en caso de no existir.

```
<jsp:useBean id="miCoche" class="pbi.CarBean" scope="request"/>
```

```
pbi.CarBean miCoche= null;
```

```
Synchronized(request){
```

```
    miCoche= (pbi.CarBean)_jspx_page_context.getAttribute("miCoche",  
                                                            PageContext.REQUEST_SCOPE);
```

```
    if (miCoche == null){  
        miCoche= new pbi.CarBean();  
        _jspx_page_context.setAttribute("miCoche", miCoche,  
                                         PageContext.REQUEST_SCOPE);  
    }
```

JSP. JavaBeans.

`<jsp:useBean>`, con un cuerpo. *Setting* condicional.

```
<jsp:useBean id="miCoche" class="pbi.CarBean" scope="page"/>
```

```
<jsp:setProperty name="miCoche" property="fabricante" value="Audi"/>
```

```
</jsp:useBean >
```

JSP. JavaBeans.

Referencias polimórficas

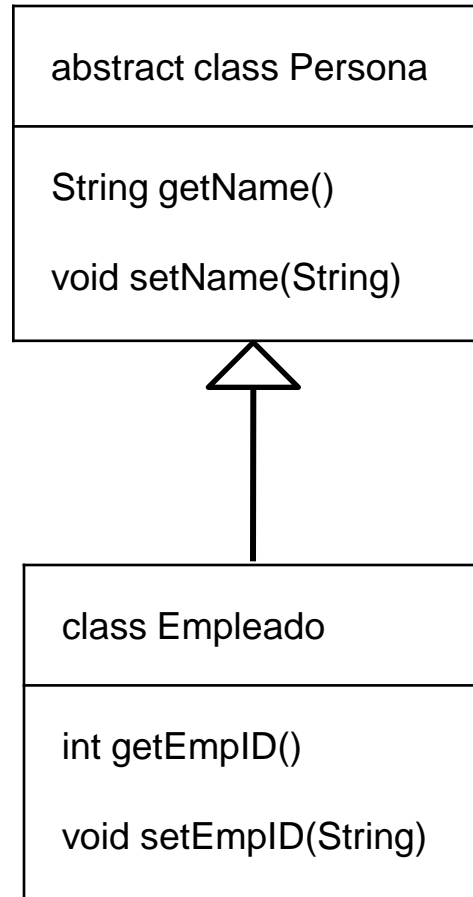
class Persona
String getName() void setName(String)

```
Persona persona= null;  
if (persona == null)  
    persona= new Persona();
```

```
<jsp:useBean id="persona" class="Persona" scope="page"/>
```

JSP. JavaBeans.

Referencias polimórficas



JSP. JavaBeans.

Referencias polimórficas

abstract class Persona
String getName() void setName(String)

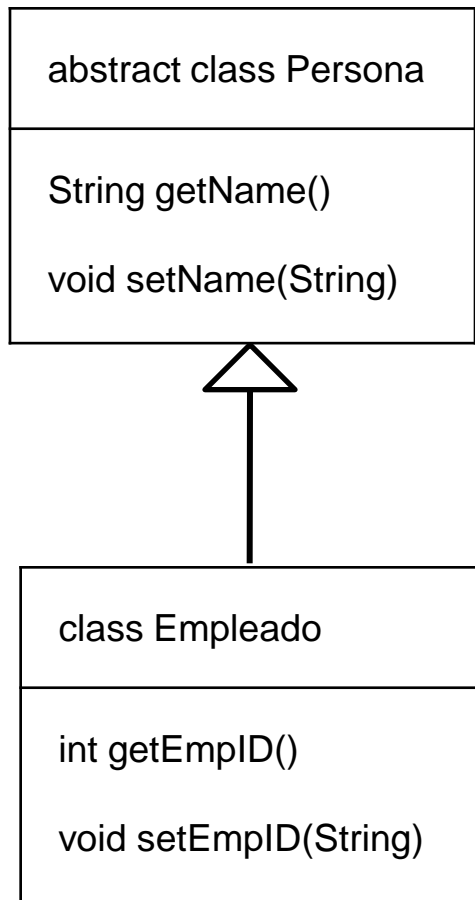
```
Persona persona= null;  
if (persona == null)  
    persona= new Persona();
```

```
<jsp:useBean id="persona" class="Persona" scope="page"/>
```

ERROR → *InstantalizationException*

JSP. JavaBeans.

Referencias polimórficas

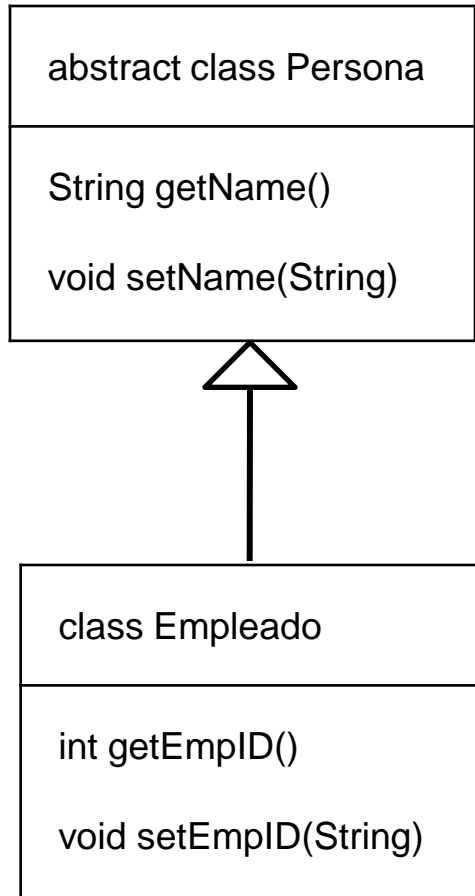


```
Persona persona= null;
    if (persona == null)
        persona= new Empleado();
```

```
<jsp:useBean id="persona" type="Persona"
    class=Empleado scope="page"/>
```

JSP. JavaBeans.

Formularios:



```
public abstract class Persona {
    private String name;

    public void setName (String name) {
        this.name=name;
    }

    public String getName() {
        return name;
    }
}
```

```
public class Empleado extends Persona {
    private int emplID;

    public void setEmpID (int emplID) {
        this.emplID=emplID;
    }

    public int getEmpID() {
        return emplID;
    }
}
```

JSP. JavaBeans.

Formularios:

```
<html>
  <body>
    <form action="miBean.jsp">
      nombre: <input type="text" name="userName">
      ID#: <input type="text" name="userID">
      <input type="submit">
    </form>
  </body>
</html>
```

```
<jsp:useBean id="persona" type="Persona" class=Empleado/>
<% persona.setName(request.getParameter("userName")); %>
```


JSP. JavaBeans.

Formularios:

```
<jsp:useBean id="persona" type="Persona" class=Empleado/>  
    <jsp: setProperty name="persona" property="name"  
        value="<%=request.getParameter("userName"); %>" />  
</ jsp:useBean>
```

```
<jsp:useBean id="persona" type="Persona" class=Empleado/>  
    <jsp: setProperty name="persona" property="name"  
        param="userName" />  
</ jsp:useBean>
```

JSP. JavaBeans.

Formularios:

```
<html>
  <body>
    <form action="miBean.jsp">
      nombre: <input type="text" name="name">
      ID#: <input type="text" name="userID">
      <input type="submit">
    </form>
  </body>
</html>
```

```
<jsp:useBean id="persona" type="Persona" class=Empleado/>
  <jsp: setProperty name="persona" property="name" />
</ jsp:useBean>
```

JSP. JavaBeans.

Formularios:

```
<html>
  <body>
    <form action="miBean.jsp">
      nombre: <input type="text" name="name">
      ID#: <input type="text" name="empID">
      <input type="submit">
    </form>
  </body>
</html>
```

```
<jsp:useBean id="persona" type="Persona" class=Empleado/>
  <jsp: setProperty name="persona" property="*" />
</ jsp:useBean>
```

JSP. JavaBeans.

Objetos:

class Persona
String getName() void setName(String) public Dog getDog() public void setDog(Dog)

class Dog
public String getName() public void setName(String)

<jsp:useBean id="persona" type="Persona" class=Empleado/>

El nombre del perro es: <jsp: setProperty name="persona" property="dog" />

ERROR: me da el código del objeto (*Dog@876592*).

JSP. JavaBeans.

Objetos:

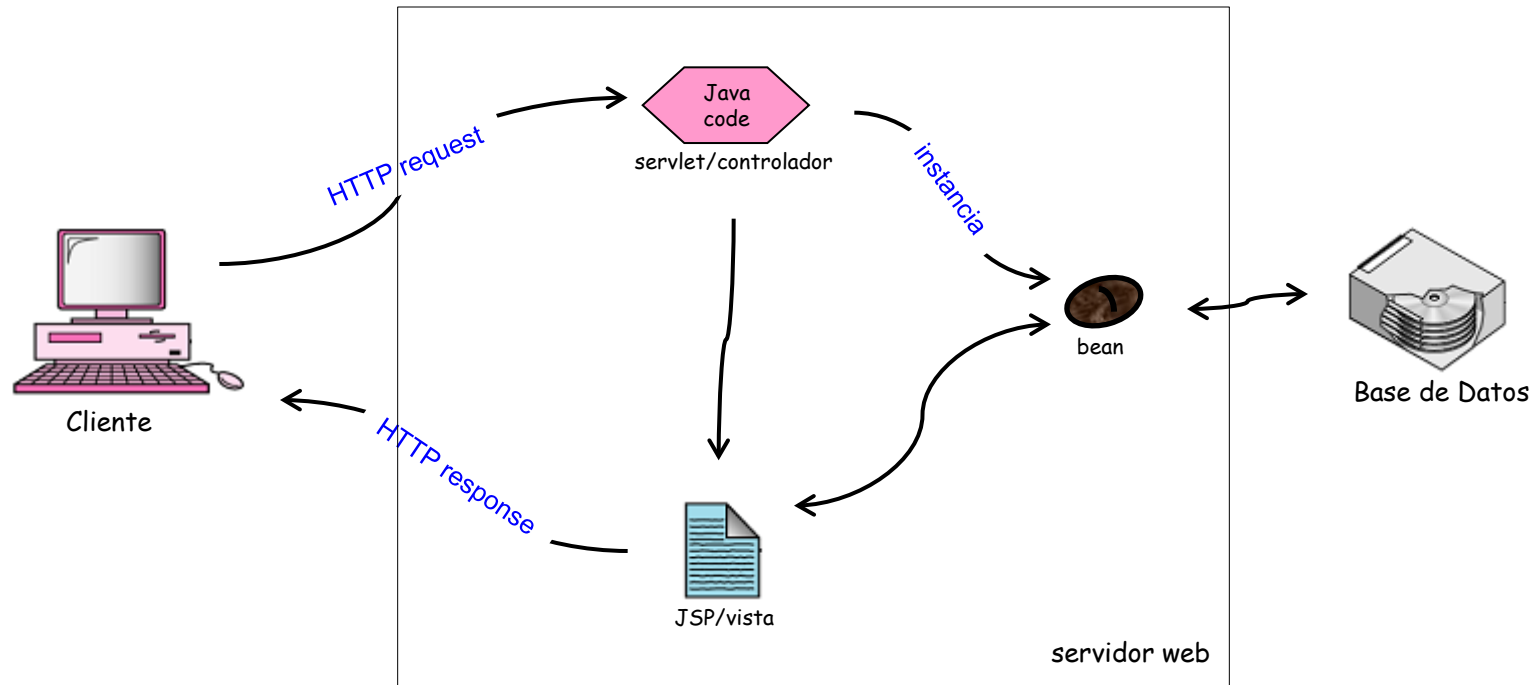
class Persona
String getName() void setName(String) public Dog getDog() public void setDog(Dog)

class Dog
public String getName() public void setName(String)

```
<%= ((Persona) request.getAttribute("persona")).getDog().getName() %>
```

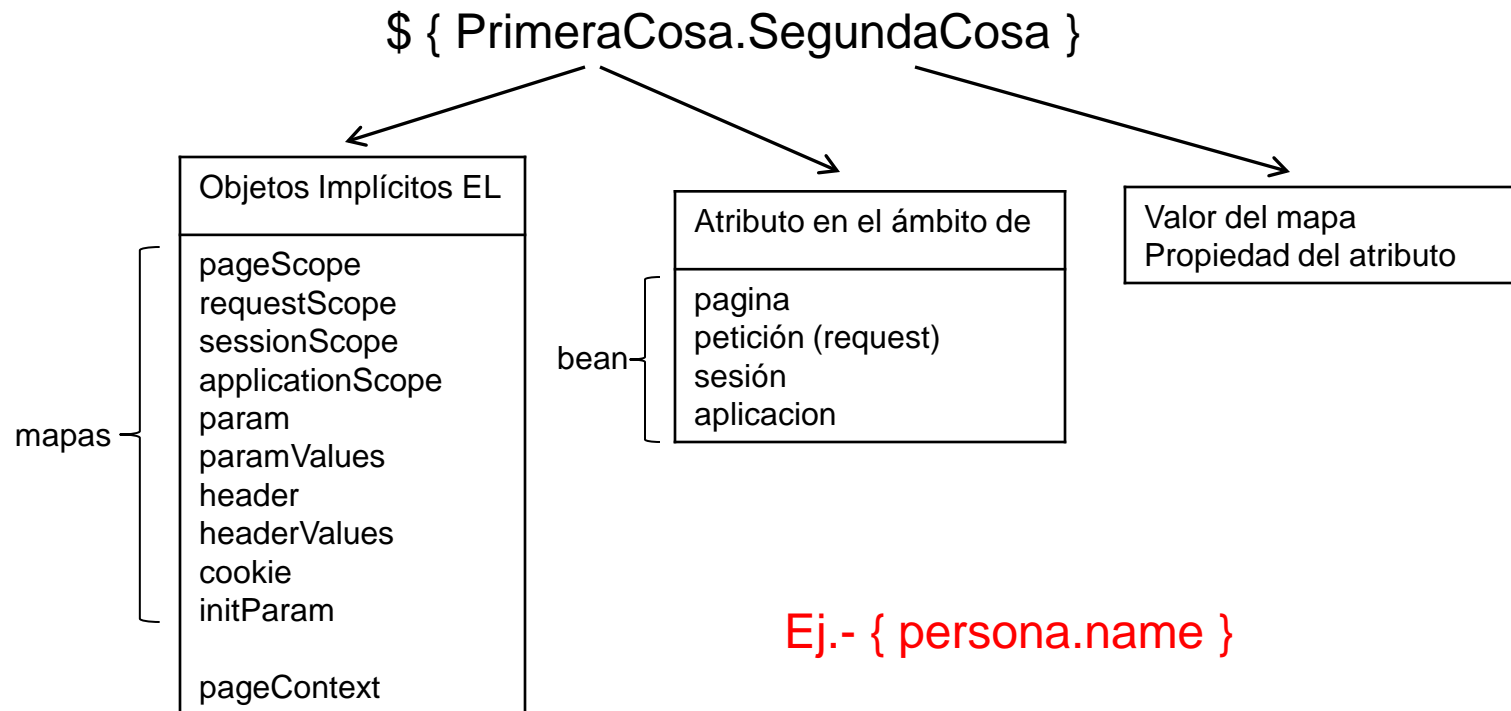
Demasiado engorroso. Solución EL.

Java Server Pages (JSP)



JSP. Expression Lenguaje (EL).

Se denomina EL a un nuevo lenguaje aparecido con la especificación 2.0 del JSP.



JSP. Expression Lenguaje (EL).

Operador “.”.- utilizado sólo cuando el elemento de la derecha es la propiedad de un *Bean* o una clave de un *Mapa*.

Operador “[]”.- utilizando este operador, el elemento de la izquierda puede ser un *Bean*, un *Mapa*, un *Array* o una *Lista*. El elemento de la derecha sigue las reglas de Java para su acceso.

Ej.-

Servlet `String asignaturasMatriculado= {"PBI", "DAAI", "CASI", "ASO"}
Request.setAttribute(listaAsignaturas, asignaturasMatriculado)`

JSP `La primera asignatura es: $ { listaAsignaturas[0] }`

JSP. Expression Lenguaje (EL).

Para listas y mapas, el manejo es similar:

Ej.- Listas.

Servlet `ArrayList asignaturasMatriculado= new ArrayList();`
`asignaturasMatriculado.add("PBI");`
`...`
`Request.setAttribute(listaAsignaturas, asignaturasMatriculado)`

JSP La primera asignatura es: `$ { listaAsignaturas[0] }`

JSP. Expression Lenguaje (EL).

Para listas y mapas, el manejo es similar:

Ej.- Mapas.

Servlet `Map asignaturasMatriculado= new HashMap();`
`asignaturasMatriculado.put("primera", "PBI");`
`...`
`Request.setAttribute(listaAsignaturas, asignaturasMatriculado)`

JSP `La primera asignatura es: $ { listaAsignaturas[primera] } o`
`La primera asignatura es: $ { listaAsignaturas.primera }`

JSP. Expression Lenguaje (EL).

Los objetos implícitos en EL:

pageScope	}	Mapas del ámbito de los atributos
requestScope		
sessionScope		
applicationScope		

param	}	Mapas de los atributos de la solicitud (request)
paramValues		

header	}	Mapas de las cabeceras de la solicitud (request)
headerValues		

cookie	}	Mapa de las cookies
--------	---	---------------------

initParam	}	Mapa de los parámetros de inicio
-----------	---	----------------------------------

pageContext	}	Referencia al pageContext (No es un mapa)
-------------	---	---

JSP. Expression Language (EL).

Ej.-

HTML

```
<form action= "miBean.jsp">
  Nombre: <input type= "text" name= "name">
  ID#: <input type="text" name= "empID">
  Comida Favorita: <input type="text" name= "food">
  Segunda Comida Favorita: <input type="text" name= "food">
  <input type= "submit">
</form>
```

JSP

```
Nombre: ${param.name} </br>
ID: ${param.empID} </br>
Comida Favorita: ${param.food} </br>
Segunda Favorita: ${paramValues.food[1]} </br>
Esta es la ficha de ${paramValues.name[0]}
```



```
Nombre: Pablo
ID: 205678392P
Comida Favorita: Paella
Segunda Comida: Pescado
Esta es la ficha de Pablo
```

JSP. Expression Language (EL).

Cookies:

JSP scriptlet

```
<% Cookie[ ] cookies= request.getCookies( );  
for (int i= 0; i<cookies.length; i++) {  
    if ((cookies[i].getName( )).equals("asignatura"))  
        out.println(cookies[i].getValue( ));  
} %>
```

JSP EL

```
$ { cookie.asignatura.value }
```

JSP. Expression Language (EL).

Parámetros de contexto inicial:

XML DD

```
<context-param>  
  <param-name> miEmail </param-name>  
  <param-value> pablo.garcia@dec.usc.es </param-value>  
</context-param>
```

JSP scriptlet

```
Mi e-mail es: <%= application.getInitParameter("miEmail") %>
```

JSP EL

```
Mi e-mail es: $ { initParam.miEmail }
```