

# Desarrollo de Aplicaciones Web

**JSP. Bibliotecas de Etiquetas.**

# JSP. Bibliotecas de etiquetas.

Una biblioteca de etiquetas a medida es un conjunto especial de etiquetas que puede utilizarse dentro de una JSP.

Se componen de dos elementos principales:

- **Clases de manipuladores** de etiquetas.- clases que proporcionan la funcionalidad de las etiquetas y contienen el código que las ejecuta.
- **Descriptor** de la biblioteca de etiquetas (TLD).- documento XML que describe las etiquetas, haciendo coincidir etiquetas con clases manipuladoras de las mismas.

# JSP. Bibliotecas de etiquetas.

Uso de bibliotecas de etiquetas:

```
<%@ taglib uri="ejemploTags.tld" prefix="daw" %>
<html>
  <head>
    Ejemplo de Tags
  </head>
  <body>
    Programación Básica en Internet. La hora actual es: </daw:time>
  </body>
</html>
```

taglib → permite declarar la biblioteca de etiquetas personalizadas y el prefijo a utilizar

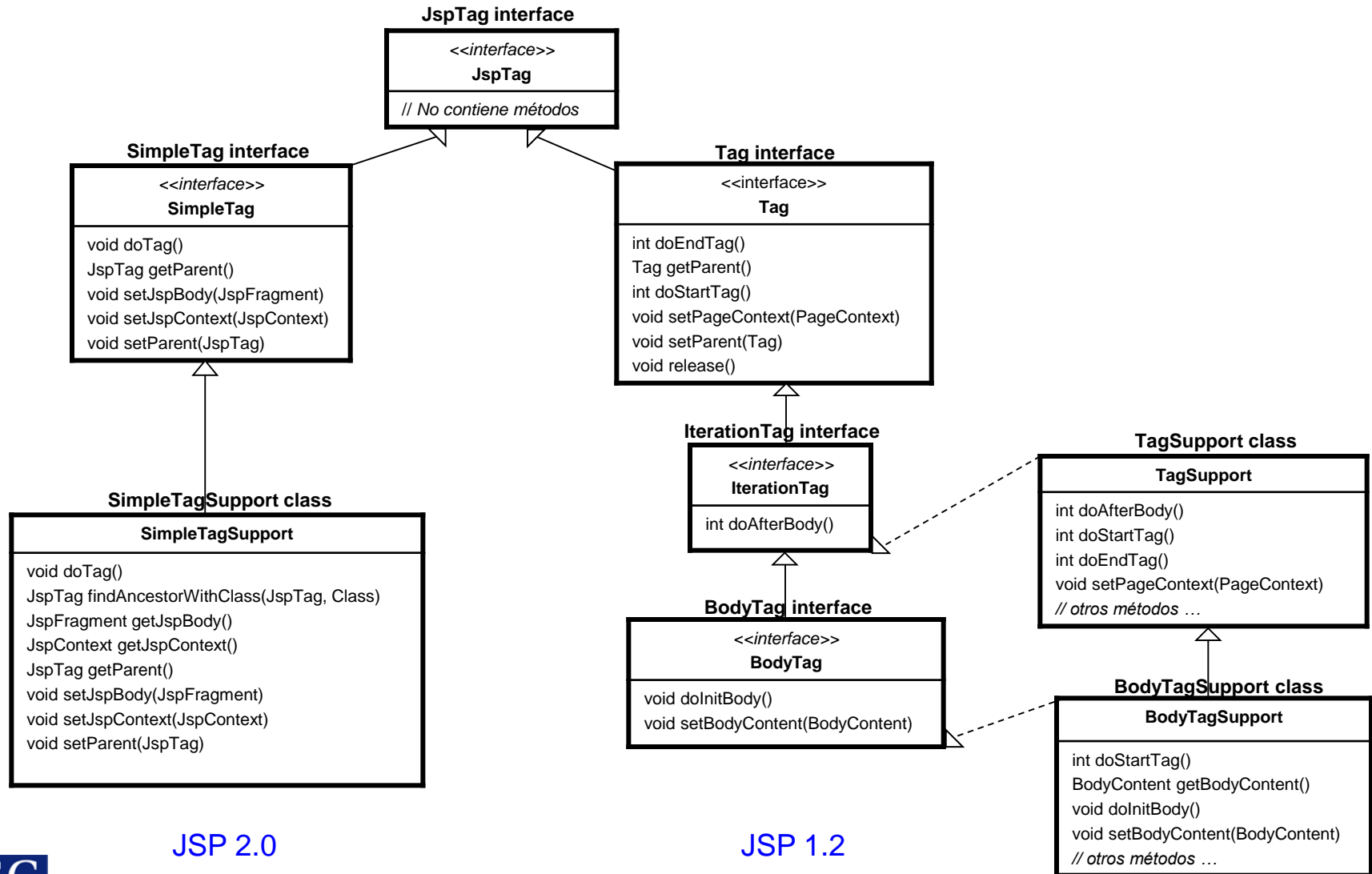
# JSP. Bibliotecas de etiquetas.

Construcción de una etiqueta a medida. Dos etapas:

- Crear la clase del manipulador de la etiqueta. Mediante el uso de la interfaz JspTag, siendo los dos métodos más utilizados:
  - doStartTag()
  - doEndTag()
- Presentar dicha clase a la etiqueta correspondiente, mediante el archivo descriptor de biblioteca de etiquetas (.tld).

Existe una clase de ayuda TagSupport que implementa la interfaz Tag. Esta clase es la que se extiende para crear la etiqueta correspondiente.

# JSP. Bibliotecas de etiquetas.



# JSP. Bibliotecas de etiquetas.

## Manipuladores de etiquetas:

- Clases java que soportan ciertos métodos, entre los que están *doStartTag()* y *doEndTag()*:

- *doStartTag()*.- método llamado por el contenedor al leer la etiqueta de apertura (*<daw:time>*). Devuelve:
  - EVAL\_BODY\_INCLUDE.- hace que se evalúe el cuerpo de la etiqueta, haciendo que esté disponible para su uso por el manipulador.
  - SKIP\_BODY.- se salta el cuerpo de la etiqueta.
- *doEndTag()*.- método llamado por el contenedor al leer la etiqueta de cierre (*</daw:time>*). Devuelve:
  - EVAL\_PAGE.- la ejecución de la página JSP continua normalmente.
  - SKIP\_PAGE.- la página JSP se detiene, después de evaluar la etiqueta.

*Nota.- en general la creación de un manipulador de etiquetas consiste en definir un código personalizado para que sea ejecutado por los métodos *doStartTag()* y *doEndTag()*.*

# JSP. Bibliotecas de etiquetas.

## Ejemplo de manipulador de etiquetas:

```
package daw;

import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.text.SimpleDateFormat;

public class timeTag extends TagSupport {
    public int doEndTag() throws JspException {
        SimpleDateFormat sdf;
        sdf= new SimpleDateFormat("HH:mm:ss");
        String time= sdf.format(new java.util.Date());
        try {
            pageContext.getOut().print(time);
        }
        catch (Exception e) {
            throw new JspException(e.toString());
        }
        return EVAL_PAGE;
    }
}
```

# JSP. Bibliotecas de etiquetas.

Descriptor de biblioteca de etiquetas (TLD).- es un archivo **xml** con la forma aproximada:

```
<?xml versión="1.0" encoding="ISO-8859-1"?>

<taglib>
    .....
    <tag>
    .....
    </tag>
    <tag>
    .....
    </tag>
    .....
    .....
    .....
</taglib>
```



# JSP. Bibliotecas de etiquetas.

Etiqueta `<taglib>`

Elemento	Descripción
<code>&lt;tlib-versión&gt;</code>	Versión de la implementación de la biblioteca, asignada por el desarrollador.
<code>&lt;jsp-version&gt;</code>	Versión de la especificación JSP.
<code>&lt;short-name&gt;</code>	Nombre corto para la biblioteca
<code>&lt;uri&gt;</code>	URI que identifica de forma única a la biblioteca
<code>&lt;display-name&gt;</code>	Nombre corto, que será mostrado por las herramientas
<code>&lt;tag&gt;</code>	Zona de inclusión de información para cada etiqueta
<code>&lt;description&gt;</code>	Descripción del propósito de la biblioteca

# JSP. Bibliotecas de etiquetas.

Etiqueta `<tag>`

Elemento	Requerido	Descripción
<code>&lt;name&gt;</code>	Sí	Nombre único para la etiqueta. Será el que aparezca en la página JSP.
<code>&lt;tag-class&gt;</code>	Sí	Clase de manipuladores que implementa: <i>Tag</i> , <i>IterationTag</i> , <i>BodyTag</i> , <i>SimpleTag</i> .
<code>&lt;tei-class&gt;</code>	No	Clase opcional de información extra de etiquetas (Tag Extra Info).
<code>&lt;description&gt;</code>	No	Descripción del propósito de la etiqueta.
<code>&lt;attribute&gt;</code>	No	Información sobre un atributo utilizado por la etiqueta.
<code>&lt;variable&gt;</code>	No	Permite el uso de variables <i>scripting</i> por parte de la etiqueta.

# JSP. Bibliotecas de etiquetas.

Ejemplo de manipulador de TLD *EjemploTags.tld*:

```
<?xml versión="1.0" encoding="ISO-8859-1"?>

<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>EjemploTags</short-name>
  <description> Conjunto de ejemplos de Tags.</description>
  <tag>
    <name>time</name>
    <tag-class>daw.timeTag</tag-class>
  </tag>
</taglib>
```

# JSP. Bibliotecas de etiquetas.

Indicamos la localización del fichero de descripción, a través del fichero de despliegue web.xml

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

<web-app>
    <jsp-config>
        <taglib>
            <taglib-uri>mytags</taglib-uri>
            <taglib-location>/WEB-INF/ejemploTags.tld</taglib-location>
        </taglib>
    </jsp-config>
</web-app>
```

# ***JSTL. JavaServer Page Standard Tag.***

Para utilizar librerías personalizadas, es necesario incluir el fichero descriptor en la aplicación:

