# Inclusión de páginas

#### Que es una inclusión

Un inclusión se utiliza cuando existe cierta pagina ya creada y es necesaria agregarla como parte de otras páginas

Es muy común agregar banners en distintas páginas, o los pie de página con los datos del dueño de la página

Existen dos forma de incluir en JSP, a través de la directiva include y a través del tag <jsp:include>

#### Directiva include

Representa la inclusión de la página a través de la directiva <% @include

La directiva se procesa cuando el JSP se traduce al servlet (tiempo de traducción), es decir que el código de la página incluida queda como código del servlet

Se utiliza para incluir página estáticas como banners o pie de páginas

Si la página incluida cambia, es necesario recompilar la página original

La forma de utilizarlo es:

<% @ include file="banner.html" %>

#### El tag <jsp:include>

Representa la inclusión de la página a través del tag <jsp:include

El tag se procesa cuando el JSP se ejecuta, es decir en tiempo de ejecución

Si el recurso incluido es estático (por ejemplo, banner.html), el código es insertado dentro del JSP que lo llamo

Si el recurso incluido es dinámico (por ejemplo, procesar.jsp), se le envía el request para que lo pueda utilizar

La forma de utilizarlo es:

<jsp:include page="procesar.jsp"/>

# Redireccionamiento

#### Que es un redireccionamiento

El redireccionamiento es la forma de enviar el flujo de la ejecución hacia otro destino

Transfiere el control a otro componente Web

Típicamente en entornos Web resulta necesario cambiar de página JSP o Servlet para continuar con el

#### procesamiento

El redireccionamiento permite a otro pagina continuar con la ejecución de la página original

El Web container se encarga de que la pagina destino reciba el request de la página original

# El tag <jsp:forward>

Es el tag utilizado para hacer un redireccionamiento

Este mecanismo NO envía una respuesta al cliente sino que lo resuelve internamente

La página destino recibe el request de la página original

La forma de uso es la siguiente:

%>

```
<jsp:forward page="paginaDestino.jsp" />
Un ejemplo de utilización
<%
boolean usuarioEsValido = true;
if( usuarioEsValido )
{
%>
    <jsp:forward page="bienvenido.jsp" />
<%
} else
{
%>
    <jsp:forward page="error.jsp" />
<%
}
```

Es análogo a realizar lo siguiente sin tags jsp:

RequestDispatcher rd = request.getRequestDispatcher("bienvenido.jsp");

rd.forward(request, response);

#### El método response.sendRedirect()

El objeto response contiene un método que permite redireccionar

Este mecanismo envía una respuesta al cliente solicitando que haga un nuevo pedido con la página destino

La página destino NO recibe el request de la página original

La forma de uso es la siguiente:

<%

response.sendRedirect("destino.jsp");

%>

# Manejo de Errores

#### **Excepciones**

Las páginas JSP pueden contener errores en tiempo de ejecución

Dichos errores se manifiestan a través de excepciones

Resulta necesario capturar las excepciones para que el usuario final pueda continuar con el uso de la aplicación sin recibir paginas sin formato con los datos de la excepción lanzada por el web container

#### **Errores HTTP**

Otro tipo de errores muy común son las páginas que representan a los errores de HTTP

Por ejemplo, el error 404 que representa que una página no ha sido encontrada

Entre los más conocidos están el 401=Unauthorized,500=Internal Server Error, etc.

Los Web containers deben proveer un mecanismo para capturar y procesar estos errores

# Configuración de errores en web.xml

Una de las posibilidades para la captura de errores es especificar las páginas que procesarán los errores en el descriptor de despliegue

Se pueden especificar tanto excepciones como errores HTTP

En el caso de las excepciones, siempre conviene utilizar superclases para agrupar errores

Las Web components a los que puede ser redireccionado en caso de error pueden ser páginas html estáticas, jsp o servlets

La forma de utilizar una página html como receptora del error es la siguiente:

```
<error-page>
<exception-type>java.lang.NullPointerException
</exception-type>
<location>/pagina-error-estatica.html</location>
</error-page>
```

La forma de utilizar una página JSP como receptora del error es la siguiente:

```
<error-page>
<exception-type>java.lang.NullPointerException
</exception-type>
<location>/pagina-error-dinamica.jsp</location>
</error-page>
```

La forma de utilizar una página JSP como receptora del error es la siguiente:

```
<error-page>
<exception-type>java.lang.NullPointerException
```

```
</exception-type>
<location>/ServletError</location>
</error-page>
```

Par poder utilizar el servlet como procesador de error, este servlet deberá poseer el mapeo correspondiente

La forma de utilizar una página JSP como receptora del error 404 HTTP es la siguiente:

```
<error-page>
<error-code>404</error-code>
<location>/pagina-error-dinamica.jsp</location>
</error-page>
```

# Configuración de errores en JSP

Es posible configurar en cada página donde direccionar en caso de error

Para esto resulta necesario agregar la siguiente línea al principio de cada página JSP:

```
<%@page errorPage="pagina-error.jsp"%>
```

En caso de tener configurado – para una misma excepción – tanto el descriptor de despliegue como dentro de la página, el Web container le entrega prioridad a la configuración de la página

#### La pagina de error

Dentro de la página de error es posible detectar que excepción ha sido lanzada, junto con información adicional

Dicha excepción se presenta como un objeto implícito denominado exception

Para poder contar con dicho objeto, es necesario agregar el siguiente código como primera línea del archivo JSP:

```
<%@page isErrorPage="true" %>
```