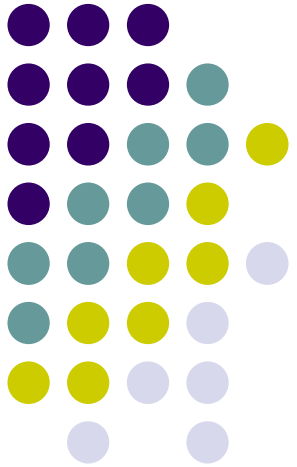
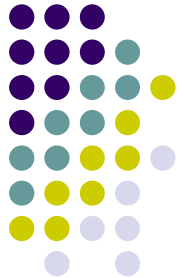
Java



Manejo de Forms, inclusión y delegación de peticiones

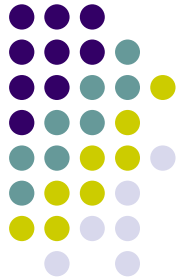
Lic. Claudio Zamoszczyk
claudio@honou.com.ar

Contenidos



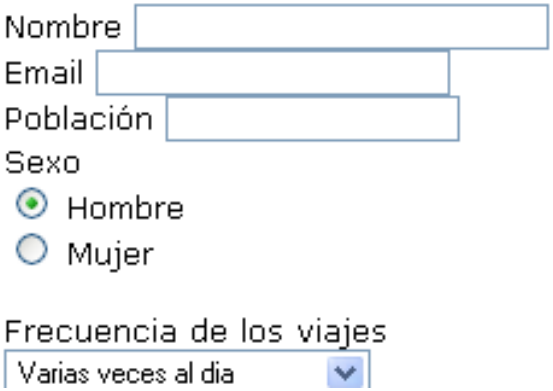
- Introducción
- HTML Form
- Formularios y JSP.
- Inclusión y delegación de peticiones

HTML– Introducción Formularios

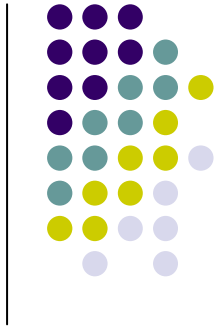


Hasta ahora hemos visto la forma en la que el HTML gestiona y muestra la información, esencialmente mediante texto, imágenes y enlaces. Nos queda por ver de qué forma podemos intercambiar información entre el servidor y el cliente.

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Mediante los mismo podemos permitir al usuario cargar datos para luego estos sean enviados al servidor y sean procesados o almacenados en una base de datos.



HTML - Formularios



Texto corto

Las cajas de texto son colocadas por medio de la etiqueta <input>. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: **type y name.**

La etiqueta es de la siguiente forma:

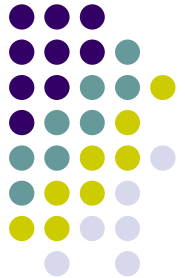
Hola

<input type="text" name="nombre">

De este modo expresamos nuestro deseo de crear una caja de texto cuyo contenido será llamado nombre (por ejemplo).

El nombre del elemento del formulario es de gran importancia para poder identificarlo en nuestro programa de procesamiento. Por otra parte, es importante indicar el atributo type, ya que, como veremos, existen otras modalidades de formulario que usan esta misma etiqueta.
El empleo de estas cajas esta fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta.

HTML - Formularios



Además de estos dos atributos, esenciales para el correcto funcionamiento de nuestra etiqueta, existen otra serie de atributos que pueden resultarnos de utilidad pero que no son imprescindibles:

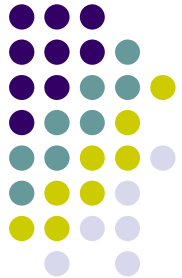
Size: Define el tamaño de la caja en número de caracteres.

Maxlength: Indica el tamaño máximo del texto que puede ser tomado por el formulario.

Value: En algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo value.

```
<input type="text" name="nombre" value="Perico Juan">
```

HTTP - Formularios



Texto oculto

Podemos esconder el texto escrito por medio asteriscos de manera a aportar una cierta confidencialidad. Este tipo de campos son análogos a los de texto con una sola diferencia: remplazamos el atributo type="text" por type="password":

<input type="password" name="nombre">

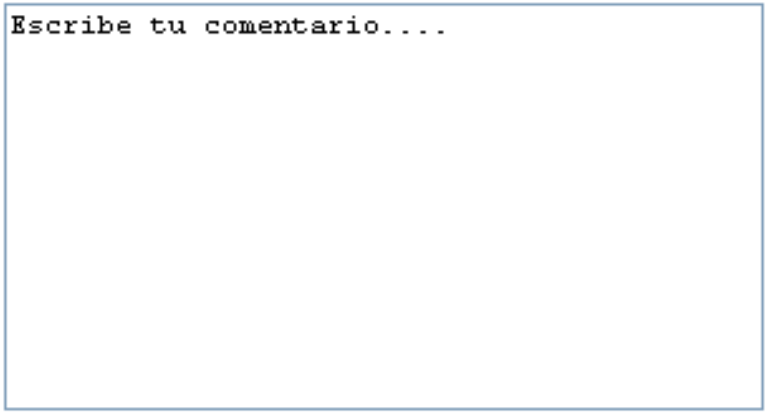
****

HTTP - Formularios

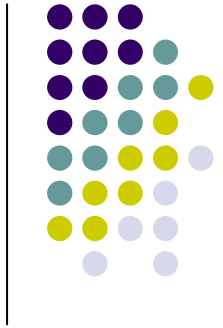
Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: <textarea>.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. Dentro de la etiqueta textarea deberemos indicar, el atributo name. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:



HTTP - Formularios



Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: <textarea>.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. Dentro de la etiqueta textarea deberemos indicar, el atributo name. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

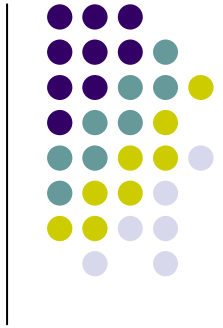
Rows: Define el número de líneas del campo de texto.

Cols: Define el número de columnas del campo de texto.

La etiqueta queda por tanto de esta forma:

```
<textarea name="comentario" rows="10" cols="40">comentario..</textarea>
```

HTML - Formularios



Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta con su respectivo cierre: <select>

Dentro de esta etiqueta definiremos su nombre por medio del atributo name. Cada opción será incluida en una línea precedida de la etiqueta <option>. Podemos ver, a partir de estas directivas, la forma más típica y sencilla de esta etiqueta:

```
<select name="estacion">
  <option>Primavera</option>
  <option>Verano</option>
  <option>Otoño</option>
  <option>Invierno</option>
</select>
```

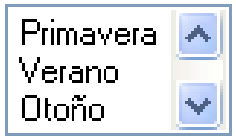
HTML - Formularios

Esta estructura puede verse modificada principalmente a partir de otros dos atributos:

Size: Indica el número de valores mostrados de la lista. El resto pueden ser vistos por medio de la barra lateral de desplazamiento.

Multiple: Permite la selección de más varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas ctrl o shift. Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

```
<select name="estacion" multiple>
  <option>Primavera</option>
  <option>Verano</option>
  <option>Otoño</option>
  <option>Invierno</option>
</select>
```



HTML - Formularios

La etiqueta **<option>** puede asimismo selecionada por medio de otros atributos

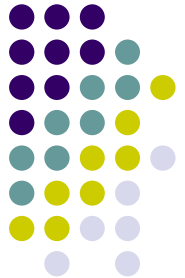
selected: Del mismo modo que multiple, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta esta elegida por defecto.

Así, si cambiamos la línea del código anterior:
<option>Otoño</option>

por:


```
<select name="estacion" multiple>
  <option>Primavera</option>
  <option>Verano</option>
  <option selected>Otoño</option>
  <option>Invierno</option>
</select>
```

HTML - Formularios



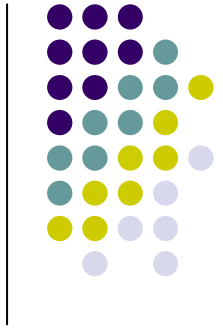
Value: Define el valor de la opción que será enviado al programa si el usuario elige esa opción. Este atributo puede resultar muy útil si el formulario es enviado a un programa puesto que a cada opción se le puede asociar un número o letra, lo cual es más fácilmente manipulable que una palabra o texto.

<option value="1">Primavera</option>

De este modo, si el usuario elige primavera, lo que le llegara al programa es una variable llamada estacion que tendrá como valor 1.

estacion=1

HTML - Formularios

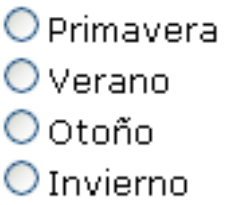


Botones de radio

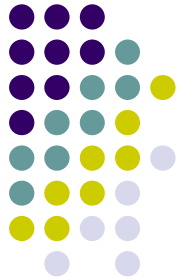
Existe otra alternativa para plantear una elección, en este caso, obligamos al usuario a elegir únicamente una de las opciones que se le proponen.

La etiqueta empleada en este caso es <input> en la cual tendremos el atributo type ha de tomar el valor radio. Veamos un ejemplo:

```
<input type="radio" name="estacion" value="1">Primavera
<br>
<input type="radio" name="estacion" value="2">Verano
<br>
<input type="radio" name="estacion" value="3">Otoño
<br>
<input type="radio" name="estacion" value="4">Invierno
```



HTML - Formularios



Cabe señalar que es posible preseleccionar por defecto una de las opciones. Esto puede ser conseguido por medio del atributo **checked:**

```
<input type="radio" name="estacion" value="2" checked>Verano
```

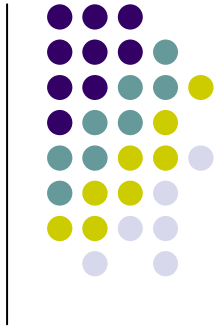
HTML - Formularios

Cajas de validación

Este tipo de elementos pueden ser activados o desactivados por el usuario por un simple clic sobre la caja en cuestión. La sintaxis utilizada es muy similar a las vistas anteriormente:

<input type="checkbox" name=“comida“ value=“4507">Me gusta la paella

HTML - Formularios



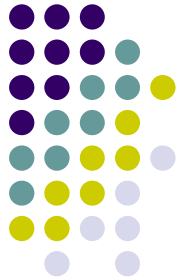
Datos ocultos

En algunos casos, aparte de los propios datos enviados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario. Este tipo de datos, que no se muestran en la página pero si pueden ser detectados solicitando el código fuente.

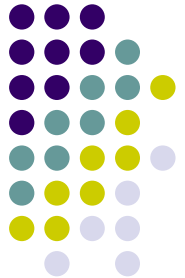
<input type="hidden" name=“idPedido" value=“451774568">

Esta etiqueta, incluida dentro de nuestro formulario, enviara un dato adicional al programa encargado de la gestión del formulario. podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo (una redirección por ejemplo, un id de usuario, etc).

¿Preguntas?



HTML – Formularios (agrupación)

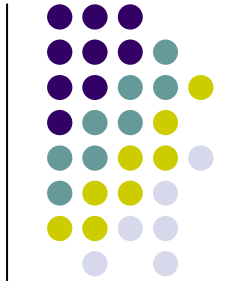


Los formularios son definidos por medio de las etiquetas <form> y </form>. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta <form> debemos especificar algunos atributos:

Action: Define el tipo de acción a llevar a cabo con el formulario, y hacia donde se desea enviar los datos existentes dentro del formulario. Todo lo que se encuentre dentro del tag form será enviado al servidor. Si lo que queremos es que el formulario sea procesado por una pagina jsp o un servlet, tenemos q especificar la url donde se encuentra el recurso

<form action="/clientes/alta.jsp" ...>

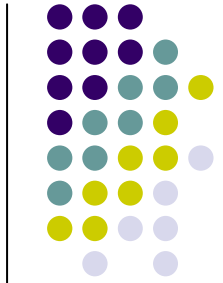
HTML - Formularios



Method: **Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar esta atributo son post y get.**

```
<form action="/clientes/alta.jsp" method= "post" >
   Nombre:<input type="text" name="nombre"><br>
   Edad:<input type="text" name="edad"><br>
</form>
```

HTML - Formularios

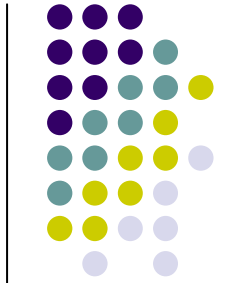


botón de envío

Para dar por finalizado el proceso de relleno de un formulario y hacerlo llegar al servidor, el navegante ha de enviar los datos mediante botón previsto a tal efecto. La construcción de dicho botón no reviste ninguna dificultad una vez familiarizados con las etiquetas input ya vistas

```
<form action="/clientes/alta.jsp" method= "post" >
   Nombre:<input type="text" name="nombre"><br>
   Edad:<input type="text" name="edad"><br>
   <input type="submit" value="Enviar">
</form>
```

Ejercicio



Nombre
Email
Población
Sexo
- Hombre
- Mujer

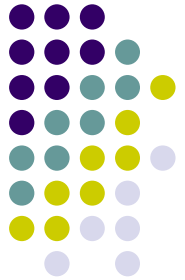
Frecuencia de los viajes
Varias veces al dia

Comentarios sobre su satisfacción personal

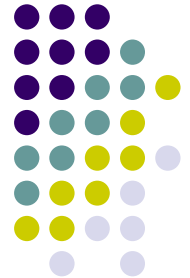
Deseo recibir notificación de las novedades en las líneas de autobuses.

Enviar formulario

¿Preguntas?



Formularios y JSP



Como vimos anteriormente el objeto request contiene información enviada al servidor a traves de una pagina web por parte del cliente.
El método *getParameter()* recoge valores enviados por medio de un formulario.
Así podemos recuperar los valores de los campos de un fórmulario.

```
<html>
  <body>
   <form method= "post">
      <input type="text" name=“nombre"/>
      <input type="submit"/>
   </form>
   <br>
    Valor del nombre:<%= request.getParameter(“nombre") %>
</body>
</html>
```

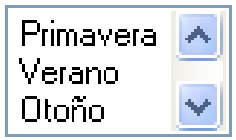
Formularios y JSP

En el caso que un campo pueda contener multiples valores, es necesario utilizar el método getParameterValues(“xxxx”), que retornada un array del tipo String con cada uno de los valores.

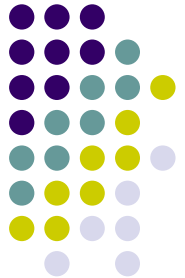
String comidas[] = request.getParameterValues(“comida");

```
<input type="checkbox" name=“comida“ value=“4507">Pollo<br>
<input type="checkbox" name=“comida“ value=“4508">Pescado<br>
<input type="checkbox" name=“comida“ value=“4509">Pastas<br>
```

```
<select name="estacion" multiple>
  <option>Primavera</option>
  <option>Verano</option>
  <option>Otoño</option>
  <option>Invierno</option>
</select>
```

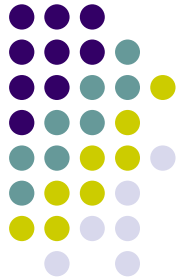


Ejercicio

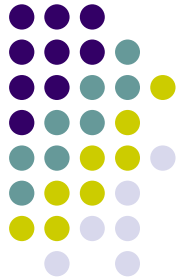


Utilizando el formulario del ejercicio anterior, realizar una pagina JSP que recupere el valor de los elementos del formulario y los muestre por pantalla.

¿Preguntas?



Inclusión de páginas estáticas

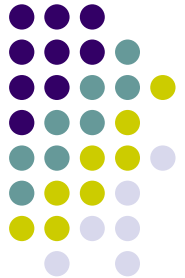


Permite incluir un archivo en el lugar donde se especifique, al contrario que con la acción <jsp:include> que veremos más adelante, la directiva include simplemente copia el contenido del archivo byte a byte, siendo el resultado similar a si copiáramos el texto del archivo incluido y lo pegáramos en el JSP.

Ejemplo:

```
<html>
  <head>
     <%@ include file="titulo.txt"%>
   </head>
<body>
     <%@ include file="cuerpoPagina.html"%>
</body>
</html>
```

Inclusión de páginas dinamicas

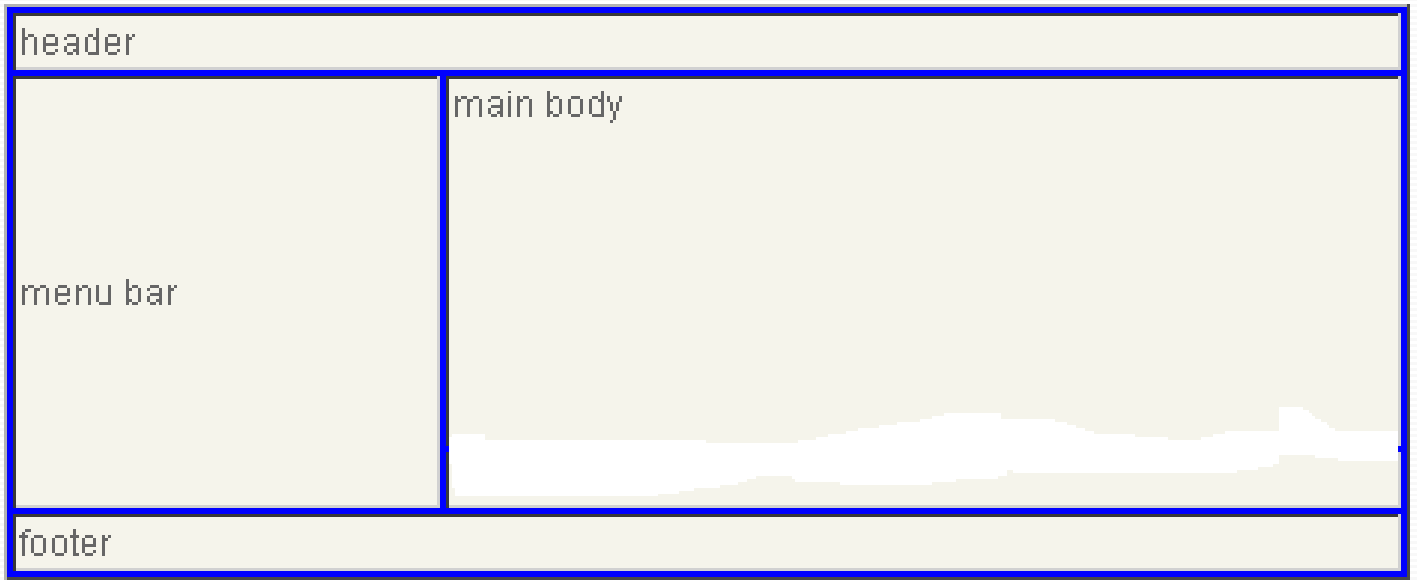


Se realiza con la acción <jsp:include page="pagina.jsp">. Incluye la salida de otra página JSP en la actual, al contrario que con la directiva <%@include file="fichero.ext"%> la página incluida se ejecuta y su salida se inserta en la página que la incluye, con la directiva se incluye el contenido del archivo (no su salida) y se ejecuta conjuntamente con la página principal.

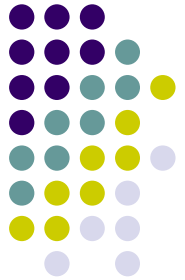
Ejemplo:

```
<html>
  <head>
    <jsp:include page="cabecera.jsp"/>
  </head>
<body>
    <jsp:include page="cuerpo.jsp“ />
</body>
</html>
```

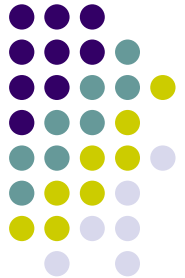
Patrón Inclusión de páginas



¿Preguntas?



Delegación de peticiones



En el esquema de trabajo típico de las paginas jsp y de los servlets es recibir una petición http, ejecutar un código determinado, generar la respuesta html y enviarla al cliente.

En muchas ocasiones y por cuestiones de diseño, estos no generan la respuesta sino que actúan como gestores de la petición mediante la ejecución de código de control. En función del bloque condicional que se ejecute y de las reglas de negocio que existan se delega la generación de respuesta a otros recursos tales como páginas html, páginas jsp, etc.

La delegacion se realiza mediante la etiqueta

<jsp:forward page="destinoxxxx.jsp" />.

Delegación de peticiones Ejemplo