



SC-415 Programación III

Semana 1

Agenda

- Sobre el curso
- Introducción a xhtml,
- CSS,
- Javascript
- Trabajo con Netbeans
- Formularios simples e introducción a JSP's

Revisión del programa del curso

Evaluación	
Criterio	Porcentaje
Tareas	20%
Examen parcial	20%
Examen final	20%
Proyecto final	40%
Total	100%

Sobre las tareas

- Las tareas deben realizarse siempre “pensando” en su aplicación en el proyecto final del curso.
- Al inicio de cada clase el estudiante (o grupo) debe cargar su tarea y tenerla activa, el profesor tomará los primeros 20 minutos para validar la realización de la tarea, la realimentación y evaluación es inmediata.
- No se reponen tareas.

Sobre el proyecto final del curso (40%)

- El estudiante debe realizar los procesos del ciclo de vida de un proyecto de programación desde la determinación de requerimientos hasta la implantación de un sistema. El proceso permite crear una solución integral que se basa en lo desarrollado en las temáticas vistas en laboratorio y las tareas. Este proyecto presenta un sitio web transaccional, desarrollado con patrones de diseño, visualmente atractivo y útil, con una presentación de proyecto final a desarrollarse en semana 15 del curso. Elaboran un proyecto programado grupal (2 a 3 estudiantes).
- Semana 6. El grupo sube un RFP (Request for Propossal) basado en las recomendaciones de PMI sobre su contenido en un foro creado para este fin en la plataforma. Los participantes del curso deben hacer comentarios reflexivos sobre las diferentes propuestas.
- Semana 12. Esta entrega no es parcial, sino que es el mejor esfuerzo del grupo por entregar un producto terminado. Se deben presentar las incorporaciones de las observaciones que plantean los compañeros y el docente sobre el proyecto en proceso de desarrollo del grupo.
- Semana 15. Presentación: El grupo realiza una presentación ejecutiva del proyecto ante la clase.

El software de desarrollo

- Para este curso es esencial utilizar la versión de Netbeans 8.1 o superior.
- Se debe instalar la versión full.
- Se debe instalar Glassfish y Apache como servidores de aplicaciones.
- Se debe validar que el acceso a la base de datos Derby esté bien.



Introducción a HTML

Etiquetas para el explorador

- DTD: Document Type Definition

<!DOCTYPE html>

Se rige por reglas de HTML, bajo el protocolo html 5

Etiquetas generales

```
<html>  
  <head>  
    .  
    .  
  </head>  
  <body>  
    .  
    .  
  </body>  
</html>
```

Etiqueta <head>

Encabezado del documento html

Definiciones generales

<head>

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<link rel="stylesheet" type="text/css" href="/estilos.css">
```

```
<title>Progra III, Lec01</title>
```

```
<meta name="keywords" content="XHTML, Progra III, Ufidelitas" />
```

</head>

Detalles de <head>

- `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">`
 - Se utiliza para indicar el tipo de contenido de la página y el set de caracteres a utilizar
- `<link rel="stylesheet" type="text/css" href="/estilos.css">`
 - Se utiliza para incorporar hojas de estilos en el documento
- `<title>Progra III, Lec01</title>`
 - En el título de la pestaña (página) en el explorador
- `<meta name="keywords" content="XHTML, Progra III, UFidélitas" />`
 - Espacio fundamental para uso por los buscadores

Definición interna de estilos

```
<head>  
  <style type="text/css">  
    body { background:blue; color:white}  
    a:link { color: green }  
    a:visited { color: gray }  
    a:active { color: red }  
  </style>  
</head>
```

El background de la página será azul y la letra color blanca

Los links inicialmente son verdes, si se han visitado serán gris y si están activos serán rojos

Definición interna de estilos

```
<head>
```

```
  <style type="text/css">
```

```
    .redline { color:red; text-decoration:line-through;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Este es un párrafo de ejemplo para ver cómo se utiliza un estilo en una  
porción de la página <span class="redline">esta parte debiera salir en rojo y con  
línea intermedia</span></p>
```

```
</body>
```

Hoja de estilo externa

- Es un archivo de texto con las definiciones de estilos iguales a las vistas anteriormente

```
/*Archivo de estilo: estilos.css */  
.redline { color:red; text-decoration:line-through;}
```

<head>

<link rel="stylesheet" href="estilos.css" type="text/css">

</head>

<body>

<p>Este es un párrafo de ejemplo para ver cómo se utiliza un estilo en una porción de la página esta parte debiera salir en rojo y con línea intermedia</p>

</body>

Etiqueta <body>

- Es tan amplia la cantidad de etiquetas que se mencionan sólo algunas de ellas y se dividen en su rango de acción:
 - Párrafo
 - Carácter
 - Caracteres
 - Elementos de organización

Marcadores de párrafo

- `<p>`: Párrafos formateados
- `<h1> .. <h6>`: encabezados
- `<div>`: divisiones del documento
- `<center>`: alineación centrada
- `<blockquote>`: texto citado
- ``: lista no ordenada (requiere `` para cada elemento)
- ``: lista ordenada (requiere `` para cada elemento)

Dan un salto de línea, se utilizan para párrafos completos

Marcadores de caracter

- ``: negrita
- `<i>`: itálica
- `<big>`: texto grande
- `<small>`: texto pequeño
- ``: enfatizado
- ``: destacado
- `<tt>`: texto monoespaciado teletipo

El nivel más en detalle de marcadores, puede ser sólo para un caracter

Etiquetas abarcantes

- ``: aplica un estilo a una sección de texto

```
<head>
```

```
<style>
```

```
.especial {color=green; text-decoration:underline;font-weight:bold;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Esta es una prueba de un estilo</p><p><span class="especial">Acá debe aparecer un  
estilo diferente</span> este otro no...</p>
```

```
</body>
```

Elementos de organización

<code><table border="1"></code>	: Tabla
<code><thead></code>	: Encabezado de tabla
<code><tr></code>	: Fila del encabezado
<code><th></th> </tr></code>	: Celda del encabezado
<code></thead></code>	
<code><tbody></code>	: Cuerpo de la tabla
<code><tr></code>	: Fila del cuerpo
<code><td></td></code>	: Celda del cuerpo
<code></tr></code>	
<code></tbody></code>	
<code></table></code>	

Elementos de organización

```
<form action="prueba.jsp">  
  <input type="text" name="nombre" value="" size="30" />  
  <input type="submit" value="Enviar" />  
</form>
```


Referencias

- Enlace externo
 - `texto a desplegar`
- Enlace interno
 - `Parte2`
 - `Parte 2`

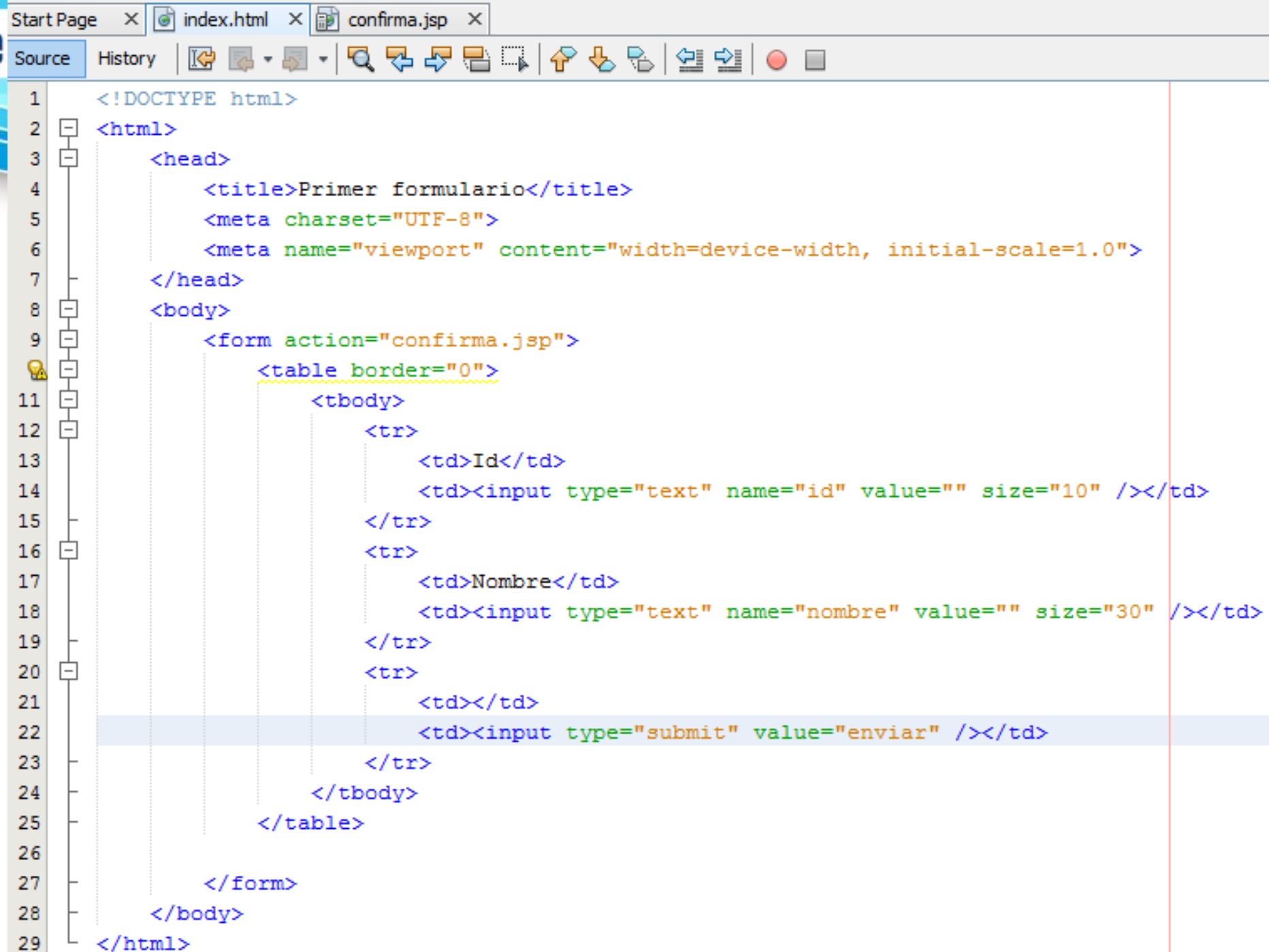
Imágenes

- ``

Uso de un formulario simple

- Vamos a ingresar a Netbeans creamos un nuevo proyecto:
 - Categoría: Java Web
 - Proyecto: Web Application
- Le llamamos Lec01
- Dejamos todo lo demás default
- Crearemos un formulario que pide un Id y un Nombre con un botón que al darle Click llamará a una página jsp “confirma.jsp” que simplemente presentará la información indicada a manera de respuesta
- La filmina siguiente muestra el contenido del index.html

Para crear la página index.html es más simple utilizar la ventana de paleta y arrastrar primero el formulario y luego la tabla.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Primer formulario</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   </head>
8   <body>
9     <form action="confirma.jsp">
10       <table border="0">
11         <tbody>
12           <tr>
13             <td>Id</td>
14             <td><input type="text" name="id" value="" size="10" /></td>
15           </tr>
16           <tr>
17             <td>Nombre</td>
18             <td><input type="text" name="nombre" value="" size="30" /></td>
19           </tr>
20           <tr>
21             <td></td>
22             <td><input type="submit" value="enviar" /></td>
23           </tr>
24         </tbody>
25       </table>
26     </form>
27   </body>
28 </html>
```


Crear una página jsp

1. Colocarse a la izquierda en la carpeta “Web Pages”
2. Dar botón derecho y seleccionar “JSP...”
3. Darle nombre “confirma”
4. Dejar el resto de las opciones default

Sobre el uso de código embebido

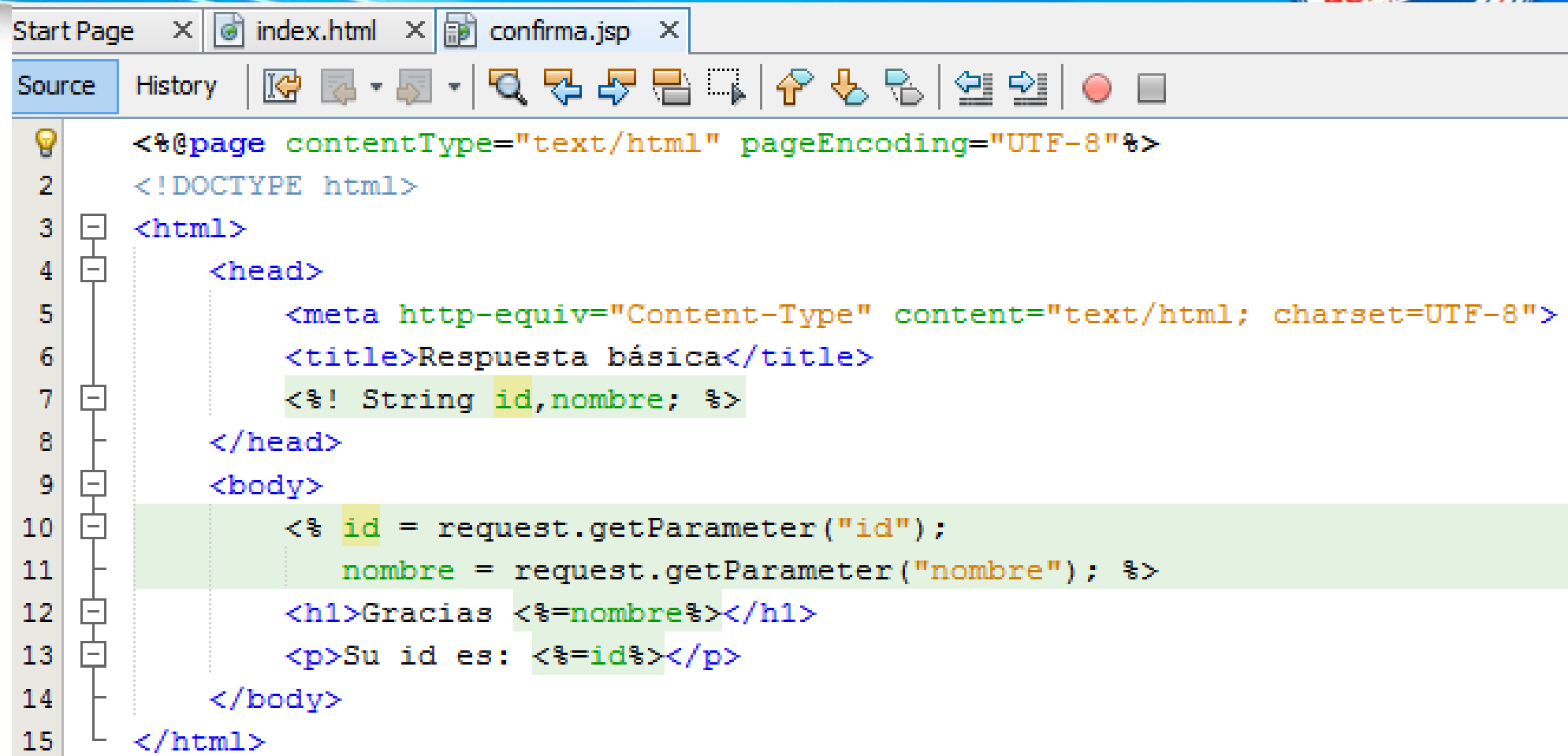
Recordemos el uso de:

`<%! %>` para usarlo en el head (definiendo variables persistentes).

`<% %>` para incrustar código java en el body

`<%=....%>` para reflejar valores de etiquetas en html

Definir el contenido de la página “confirma.jsp” como se muestra acá



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Respuesta básica</title>
    <%! String id,nombre; %>
  </head>
  <body>
    <% id = request.getParameter("id");
       nombre = request.getParameter("nombre"); %>
    <h1>Gracias <%=nombre%></h1>
    <p>Su id es: <%=id%></p>
  </body>
</html>
```

Ejecutamos el proyecto para
validar que todo esté bien!

Problemas sobre el uso de jsp como el anterior ejemplo

- A pesar que el ejemplo anterior funciona y podemos ver que hay muchos sitios comerciales que tienen este funcionamiento inclusive en otros lenguajes como PHP no esta dentro de las mejores prácticas de desarrollo de sistemas.
- Recargar los JSP con programación sobrecarga las página y se invierte tiempo en mantenimiento.
- No hay una separación por capas
- El creativo y el desarrollador deben utilizar el mismo archivo fuente
- Dado lo anterior es que se introduce el patrón de diseño MVC







Tarea #1

Desarrolle un sitio web en Netbeans que contenga al menos 4 páginas html, enlaces entre si, un formulario y todas las estructuras visuales html vistas en clase

Este sitio Web debiera ser un primer intento de su proyecto final del curso

Rúbrica para Tarea # 01

Vista de cuadrícula		Vista de lista		
		Principiante	Competente	Muy competente
Completitud 		7,5 (7,5%) Tiene menos de 3 hojas enlazadas	15 (15%) Tiene 3 hojas enlazadas o no todas son funcionales	30 (30%) Tiene 4 hojas completamente funcionales y enlazadas
Estilo de programación 		5 (5%) Hay páginas con nombres poco adecuados y falta documentación	10 (10%) Al menos una página tiene un nombre "default" o no tiene documentación	20 (20%) Los nombres de las páginas son adecuados para el estilo, hay documentación dentro de las páginas
Resolución del problema 		0 (0%) Se limitó a crear unas páginas con los botones y el contenido no es consistente	15 (15%) Si bien hay coesistencia en la mayoría de las página, existe al menos una que esta de "relleno"	30 (30%) Tódas la páginas creadas tienen sentido y estas adecuadamente creadas
Diseño visual 		0 (0%) Utiliza los fondos y tipos de letras "default"	10 (10%) Incoporta fondos interesantes, la fuente de letra es adecuada para la página	20 (20%) La interface se visualemnte atractiva, utiliza fondos y tipos de letra adecuados, investigó y aplicó como colocar una imagen de fondo.



Nos vemos en la siguiente clase

Temática de a revisar:

JSP's y Servlet's