

# CC5002 - Desarrollo de Aplicaciones Web

## Clase 8: Java Server Pages

# Plan de clases proyectado (1era Mitad)

1. Intro
2. HTTP y HTML
3. Servlets y Tomcat
4. JSP y Threads
5. Sesiones y Cookies
6. Bases de datos
7. JavaBeans
8. ->JSP
9. MVC
10. JSTL
11. Seguridad
12. CSS
13. Javascript
14. AJAX
15. Control

# Clase de Hoy

- Temas Administrativos
- Preguntas
- Java Server Pages

# Bonus Tarea 1

- El lunes se entregarán los bonus.
- Todos tienen bonus!

# Motivación: HTML Estático

- Publicar Información
- Estructura de la Información en el HTML
- Estilo especificado separadamente
  - ▶ Hojas de Estilo CSS especifican como presentar la información
- Se pueden crear usando editores HTML
  - Dreamweaver y otros
  - Incluso Word

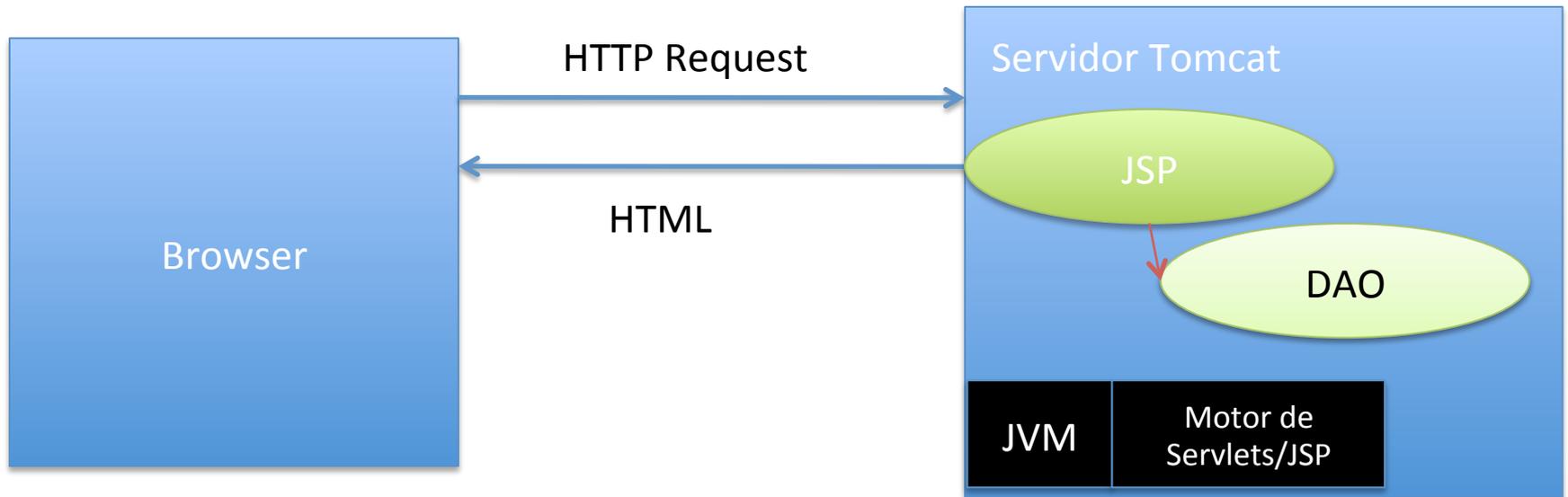
# Motivación: Lógica

- La lógica se maneja con un lenguaje de programación poderoso
  - Ambiente de programación poderoso
  - Eficiente
  - Que soporte reusabilidad

# Como combinamos los dos?

- Todo se puede hacer en un servlet
  - Pero requiere una gran cantidad de lógica repetitiva para enviar datos al browser.
- La meta es combinar lo mejor del HTML estático y de Java dinámico en un solo concepto de programación
  - Java Server Pages (JSPs)
  - Microsoft tiene un concepto similar con sus Active Server Pages
  - Además, también deben conocer PHP

# Java Server Pages- Arquitectura 1



# Java Server Pages - Archivos

- Contienen
  - HTML
  - Código
    - Llamadas Java o expresiones Java
- El código JSP es compilado en un Servlet, y luego es ejecutado

# Ejemplo: HolaMundo.jsp

```
<html>
<head>
<title>Hola, Mundo Ejemplo JSP</
  title>
</head>
<body>
<h2>Hola, Mundo desde un JSP!</
  h2>
</body>
```

# Pero ahora pueden insertar código Java

- Colocar sentencias entre los tags `<% y %>`
- Poner las expresiones entre los tags `<%= y %>`
- Hay variables estándar ya definidas como:
  - `HttpServletRequest` **request**
  - `HttpServletResponse` **response**
  - `JspWriter` **out** (Funciona como `PrintWriter`)
  - `HttpSession` **session**

# Recuerden HolaX.jsp

```
<html>
<head><title>Ejemplo HolaX JSP</title></head>
<body>
<%
    String name = request.getParameter("name");
    if (name == null) {
%>
    <form method="GET" action="HolaX.jsp">
        <font size="+2">Cual es tu nombre?</font><br/>
        <input type="text" name="name" /><br/>
        <input type="submit" />
    </form>
<%
    } else {
        out.println("<h2>Hola, " + name + "!</h2>");
    }
%>
</body></html>
```

# Recuerden Time.jsp

```
<html>
<head>
  <title>Ejemplo Time en JSP</title>
</head>
<body>
  <h1>
    La hora actual en milisegundos es
    <%= System.currentTimeMillis() %>
  </h1>
  <hr/>
  <h1>
    La fecha actual es <%= new
    java.util.Date() %>
  </h1>
</body>
</html>
```

# Como Funciona Esto?

- El archivo .jsp es compilado en un servlet
  - Se genera una estructura de servlet
  - Todo el HTML es insertado como Strings escapados, dentro de sentencias .write()
  - Todo el código entre los tags `<%` y `%>` es insertado en el servlet tal cual.
- El código lo pueden encontrar en
- `apache-tomcat\work`

# Ejemplo: HolaUChile.jsp

- Editado usando una herramienta
- Con fotos sacadas de la web
- En las partes donde necesito código dinámico solamente lo incluyo entre los tags `<%= y %>`

# Que Tan Lejos Podemos Llegar?

- Podemos colocar definiciones de funciones completas dentro de los tags `<%! Y %>`
- Pueden importar usando
  - `<%@ page import="..." %>`
- También hay tags para comentarios
  - `<%-- --%>`

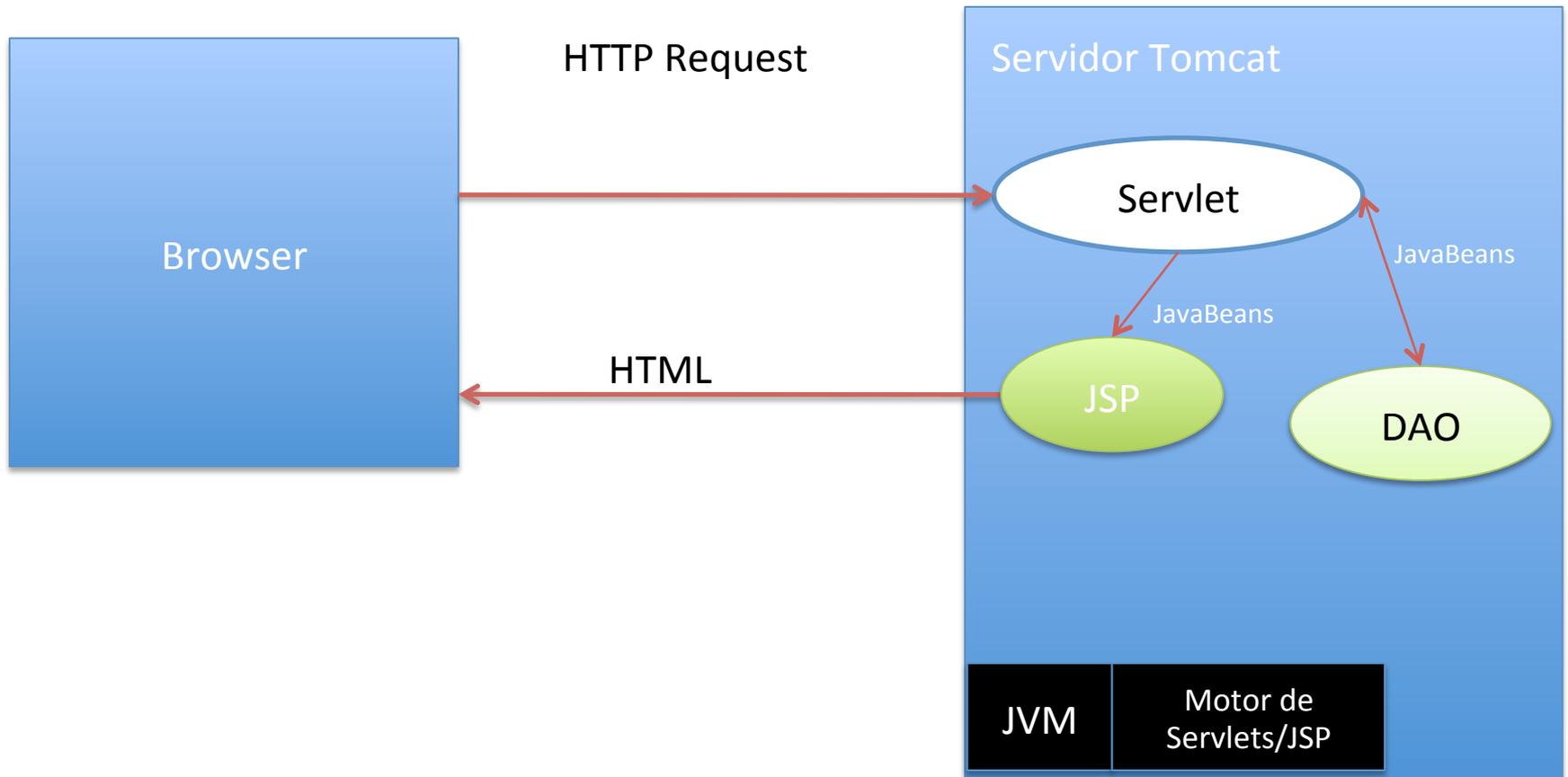
# Ejemplos Usando Beans

- Entry.java
- EntryDAO.java
- AddrBookLookup1.java

# Otras Formas de Usar JSPs

- Arquitectura 2

# Java Server Pages- Arquitectura 2



# Ejemplos

- AddrBookLookup2.jsp
- AddrBookLookupForm.html
- AddrBookError.jsp
- AddrBookEntry.jsp
- AddrBookList.jsp
- NameSearch.java
- IdSearch.java

# EL – Expression Language de JSP

- Un lenguaje de scripting que incluye JSP 2.0
- Colocas tus expresiones adentro de `${ }`
- Las variables pueden referirse a JavaBeans
  - (U otros objetos)
- El operador `.` (punto) se traduce como como un accesor (`get`) de un bean

# Las Variables en JSP EL

- Las variables son encontradas buscando en un orden
  - Contexto de la pagina
    - Variables locales dentro de la pagina
    - Tiempo de vida: Un JSP
  - Contexto del request
    - Usadas para pasarle atributos del servlet al JSP
    - Tiempo de vida: Un request
  - Contexto de la sesión – Atributos de sesión
    - Usado para almacenar info de la sesión. Ej: Quien esta logueado, carro de compras, etc
    - Tiempo de vida: La sesión
  - Contexto de la aplicación
    - Usado para almacenar info sobre la aplicación. Ej: Parámetros de inicio

# Como Escribir tu Código en JSPs?

1. `<% aquí va tu código %>`
2. `<%= expresión de una línea %>`
3. Arquitectura 2: Dispatch del servlet al JSP

# Yo prefiero la arquitectura 2

- Te permite procesar los inputs usando un ambiente Java completo, y luego enviar a los respectivos JSPs para que ellos formateen la respuesta.

# Modelo-Vista-Controlador

- La próxima clase veremos en extenso el uso formal de la arquitectura 2 conocida como Modelo-Vista-Controlador
- Ya han visto la mayoría de sus partes
  - Un controlador: Ejemplo AccessController
  - Modelo: DAOs
  - Vista: Java Server Pages