

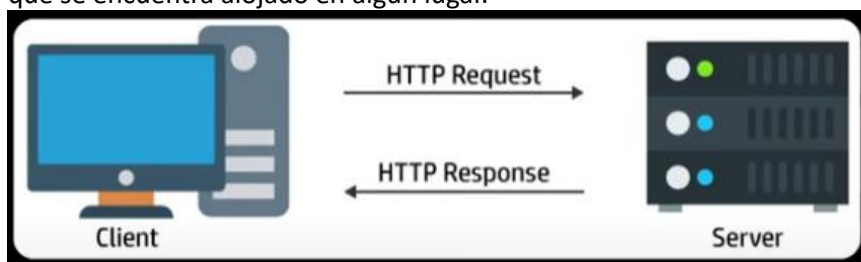
Laboratorios 5

Objetivos:

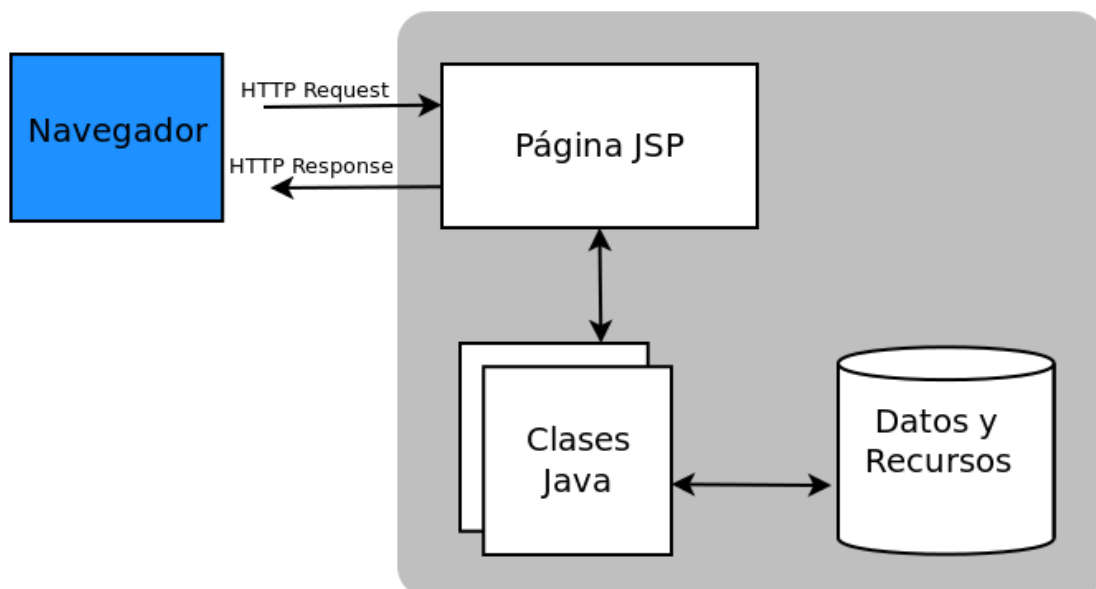
- ✓ Comprender el uso de JSP en la creación de aplicaciones web dinámicas.
- ✓ Aplicar los conceptos de JSP para desarrollar aplicaciones.

Recordemos que es y cómo trabaja la web.

- ✓ La "Web" se base en peticiones o consultas que hace un cliente hacia un servidor Web que se encuentra alojado en algún lugar.



- ✓ Ese cliente se comunica con un servidor, mediante un protocolo que se llama HTTP o Protocolo de transferencia de hipertexto.
- ✓ Mediante este protocolo, el cliente realiza consultas (Requests) y recibe respuestas (Responses).





JSP (Java Server Pages)

- Es una tecnología que permite crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos.
- Java Server Pages (JSP), requiere de un servidor web que sea compatible con contenedores servlet como, por ejemplo: Apache Tomcat o Glassfish.
- Si bien JSP no es una tecnología nueva, tiene la ventaja de la portabilidad de Java y permite comprender los fundamentos de la programación web.

Ejemplo de JSP

- Cada JSP va a contener mayoritariamente etiquetas HTML y CSS, como todo frontend, sin embargo, se puede utilizar etiquetas especiales para especificar porciones de código Java en donde sea necesario.

```
1  <!--
2      Document    : index
3      Created on  : 1 may. 2024, 13:13:00
4      Author     : Cpalacios
5  -->
6  <%@page contentType="text/html" pageEncoding="UTF-8"%>
7  <!DOCTYPE html>
8  <html>
9      <head>
10         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11         <title>JSP Page - Ejemplo Lab05 - 2024</title>
12     </head>
13
14     <body>
15         <H1>Ejemplos de expresiones</H1>
16         <!-- Mostrar la fecha y hora actual -->
17         <%= new java.util.Date().toString() %><br>
18
19         <!-- Convertir a mayúsculas un String -->
20         <%= "Texto a mayúsculas".toUpperCase() %><br>
21
22         <!-- Resultado de una expresión aritmética -->
23         <%= (5+2)/(float)3 %><br>
24
25         <!-- Generar un número aleatorio -->
26         <%= new java.util.Random().nextInt(100) %>
27     </body>
28
29 </html>
30
31
```



Etiquetas JSP

`<%-- --%>`

Apertura y cierre para realizar comentarios.

Por ejemplo: `<%-- esto es un comentario --%>`

`<%@ %>`

Apertura y cierre para directivas/atributos de configuración de JSP.

Por ejemplo: `<%@ page language='java' contentType='text/html' %>`

`<% %>`

Apertura y cierre para inclusión de sentencias o código Java en general. Esto no es visto/percibido por el usuario.

Por ejemplo: `<% if (numero > numero2) { ... } %>`

`<%= %>`

Apertura y cierre para mostrar el resultado de una expresión o contenido de una variable. Lo que se indique aquí será visualizado por el usuario en el apartado de HTML dentro del JSP que se indique.

Por ejemplo: `<%= nombre %>`

`<%! %>`

Apertura y cierre para hacer uso exclusivo de declaración de variables y métodos de instancia. Que se compartirán entre varios JSP asociados al mismo servlet.

Nota: Para declarar variables locales usar `<% %>`

Actividad



1- Crea un Proyecto y trabaja con JSP

Crea un proyecto webAppLab05, con Maven y te tipo Web Application.

Group Id: py.una.pol.par

Servidor: Apache Tomcat

Nota sobre estructura de archivos.

En NetBeans, un proyecto de tipo "Web Application" en la categoría "Java with Maven" sigue una estructura de directorios estándar que sigue las convenciones de Maven y las especificaciones de Java EE para el desarrollo de aplicaciones web.

1. src/main/java

Este directorio contiene el código fuente Java de la aplicación web. Aquí se encuentran las clases Java que implementan la lógica de negocio, controladores de servlets, filtros, etc.

2. src/main/resources



En este directorio se almacenan los recursos no Java de la aplicación, como archivos de configuración (por ejemplo, archivos XML de configuración de Spring), archivos de propiedades, archivos de configuración de bases de datos, etc.

3. **src/main/webapp**

Este es el directorio principal de la aplicación web. Contiene todos los recursos web, como páginas HTML, archivos CSS, JavaScript, imágenes, archivos JSP, plantillas de vistas, archivos de configuración de servlets (web.xml), entre otros.

4. **src/main/webapp/WEB-INF**

Este subdirectorio contiene archivos de configuración específicos de la aplicación web, como el archivo web.xml que define la configuración de los servlets, filtros, listeners y otros componentes de la aplicación web.

5. **target**

Este directorio es generado automáticamente por Maven y contiene los archivos compilados de la aplicación, así como los archivos JAR y WAR generados durante el proceso de compilación y empaquetado.

6. **pom.xml**

Este es el archivo de configuración principal de Maven. Contiene la configuración del proyecto, las dependencias de Maven, los plugins utilizados y otra información relacionada con la construcción del proyecto.

Que debe hacer la aplicación:

1. **index.jsp** – Implementar el ejemplo inicial y probar

```
<%--
Document: index
Created on: 1 may. 2024, 03:13:00
Author: Cpalacios
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page - Ejemplo Lab05 - 2024</title>
  </head>
  <body>
    <H1>Ejemplos de expresiones</H1>
    <%-- Mostrar la fecha y hora actual --%>
    <%= new java.util.Date().toString() %><br>

    <%-- Convertir a mayúsculas un String --%>
    <%= "Texto a mayúsculas".toUpperCase() %><br>

    <%-- Resultado de una expresión aritmética --%>
    <%= (5+2)/(float)3 %><br>

    <%-- Generar un número aleatorio --%>
    <%= new java.util.Random().nextInt(100) %>
  </body>
</html>
```



2. factorial.jsp – Debe calcular la factorial de un número recibido, aquí un ejemplo:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Cálculo del factorial</title>
</head>
<body>
  <h1>Cálculo del factorial</h1>

  <!-- Formulario que solicita al usuario el número a usar en el cálculo -->
  <form action="factorial.jsp" method="get">
    <p>Número: <input type="text" name="numero">
    <input type="submit" value="Calcular"></p>
  </form>

  <%
    // A través del objeto request se obtiene el valor pasado por el formulario
    String numeroGet = request.getParameter("numero");
    if(numeroGet!=null) {
      int numero=0;
      double factorial=1;
      boolean error = false;
      try {
        numero = Integer.valueOf(numeroGet);
        if(numero<1) {
          error = true;
        } else {
          for(int i=numero; i>1; i--) {
            factorial *= i;
          }
        }
      } catch(NumberFormatException e) {
        error = true;
      }
      if(error) {
        out.println("<p>Debe indicar un número entero mayor que 0</p>");
      } else {
        // Mostrar el resultado en la página usando el objeto out
        out.println("<p>Resultado: "+numero + "! = " + factorial+"</p>");
      }
    }

    // Uso del objeto session para contar las veces que se ejecuta la aplicación
    Integer contador = (Integer)session.getAttribute("contadorVisitas");
    if(contador!=null) {
      contador = Integer.valueOf(contador);
    } else {
      contador = 0;
    }
    if(contador!=0) {
      out.println("<p>Ejecuciones de la aplicación en esta sesión: "+contador+
"</p>");
    }
    contador++;
    session.setAttribute("contadorVisitas", contador);
  <%>
</body>
</html>
```

Notas:

En este ejemplo se utilizan los objetos predefinidos para ser usados en JSP:

request: permite obtener datos de la petición HTTP que se realiza. Por ejemplo, los parámetros GET y POST que son pasados a la página.

session: contiene los datos de la sesión, permitiendo guardar y recuperar información durante la vigencia de la sesión.

out: posibilita enviar información a la salida HTML que se genera.



3. Crea **galeria.jsp** un front que despliegue dinámicamente todo el contenido de un directorio de imágenes. (Implementación propia)

Entregas:

Lab05 – Entrega por:

1. **Plataforma EDUCA:** Documento basado en plantilla de informe de laboratorio (evidencias), en formato pdf.

2- **Repositorio:**

Dentro del repositorio creado en GitHub, con tu usuario y la descripción: **par2024**

Dentro de **laboratorios/Lab05**

Levantar la implementación realizada.

Permitir el acceso a: cnpalacios@pol.una.py.

Los códigos que no pueden ser probado, no serán considerados.