Instituto Politécnico Nacional

Escuela Superior de Cómputo

**Servlets con método POST y GET**

**Alumna:** Hernández Cruz Dulce

**Materia:** Web Application Development

**Grupo:** 3CM3

**Profesor:** Alejandro Cifuentes**Objetivo:** Realizar un formulario que calcule la suma de dos números con el método GET y realice la resta con el método POST utilizando servlets y los métodos de doGET y doPost respectivamente.

**Introducción**

Todo servlet hereda de HttpServlet y, al igual que un applet, contiene una serie de métodos predefinidos, que son invocados en respuesta a eventos:

**Inicialización**. Se carga en memoria una instancia del servlet. La instanciación se produce la primera vez que el servlet es cargado en memoria por parte del servidor de aplicaciones. Como consecuencia de este evento se ejecuta el siguiente método:

public void init(ServletConfig config) throws ServletException

**Petición:** el cliente realiza la petición al servlet. Dentro de la versión 1.1 y en función del método de invocación usado en HTML (DO o GET) se ejecutará uno de los siguientes:

public void doGet(HttpServletRequest peticion, HttpServletResponse respuesta) throws ServletException, IOException

public void doPost(HttpServletRequest peticion, HttpServletResponse respuesta) throws ServletException, IOException

El objeto 'peticion' es enviado por la JVM del servidor de aplicaciones y representa (encapsula) información de la petición HTTP. Más adelante veremos para que puede servir el objeto 'respuesta', por de pronto nos sirve decir que en este objeto se encapsulan los servicios para generar la salida. También tiene la posibilidad de usar:

public void service(HttpServletRequest peticion, HttpServletResponse respuesta) throws ServletException, IOException

**Destrucción del servlet:** la contraimagen de init(), el servlet va a ser destruido (normalmente por la detención del servidor). No se invoca cuando se termina la petición, sino cuando se va a descargar el servlet de memoria:

public void destroy()

**Desarrollo**

1. Crear un proyecto de aplicación web en NetBeans e iniciar el servidor Tomcat que viene incluido.
2. Dentro del proyecto se agrega un nuevo elemento, en este un Servlet.

En el archivo que se agrega a proyecto: ServletBasico.java, vamos a definir las funciones doGET y doPost (en el primero que haga la suma y el otro la resta).

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String a, b;

int x, y, resultado;

a = request.getParameter("a");

b = request.getParameter("b");

try {

x = Integer.parseInt(a);

y = Integer.parseInt(b);

resultado = x + y;

} catch (Exception e) {

e.printStackTrace();

out.println("Error al recibir parámetros con GET");

out.print(e);

out.close();

return;

}

out.println("<html>");

out.println("<head><title>Un servlet básico</title></head>");

out.println("<body>");

out.println("<h1>Esta es una prueba de un servlet con GET</h1>");

out.println("<h1>La resta de " + x + " + " + y + " = " + resultado + "</h1>");

out.println("</body></html>");

out.close();

}

Los números a sumar (a y b), se obtienen del formulario de la página html.

Como los datos que se reciben son String, se necesita convertirlos a enteros.

Después el resultado se imprime en la página HTML.

Lo mismo se repite para el método doPost(), pero con la operación de resta.

En el action del formulario se coloca el nombre del servlet: ServletBasico y en method GET o POST.

<form action="ServletBasico" method="GET">

<p>Numero a = <input type="text" name="a"></p>

<p>Numero b = <input type="text" name="b"></p>

<p><input type="submit" value="Hacer suma"></p>

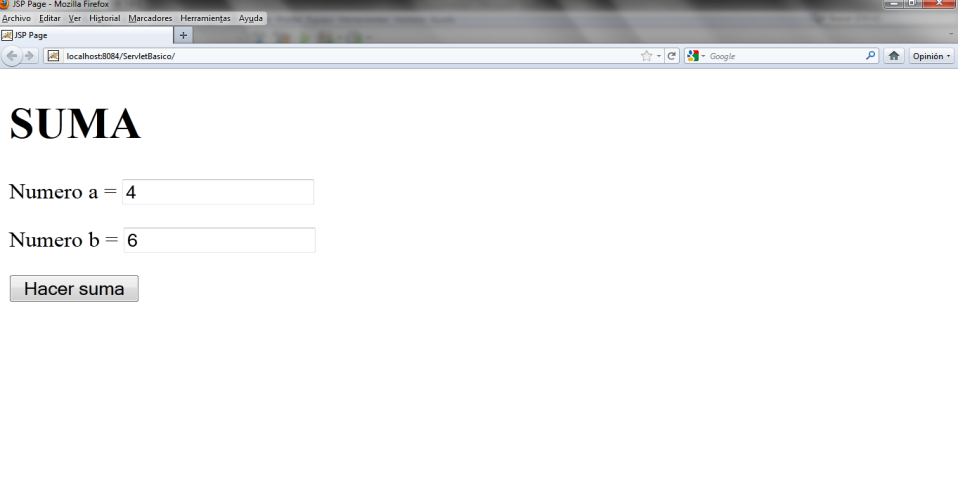
</form>

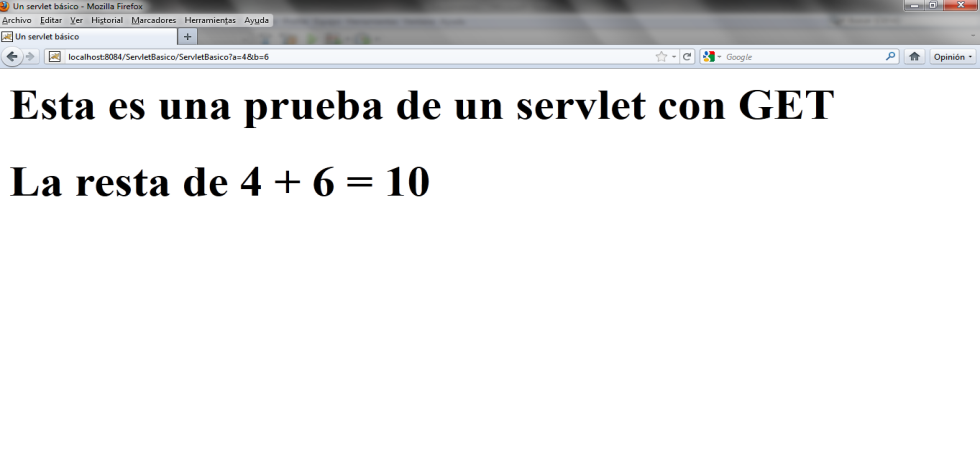
1. Archivo XML

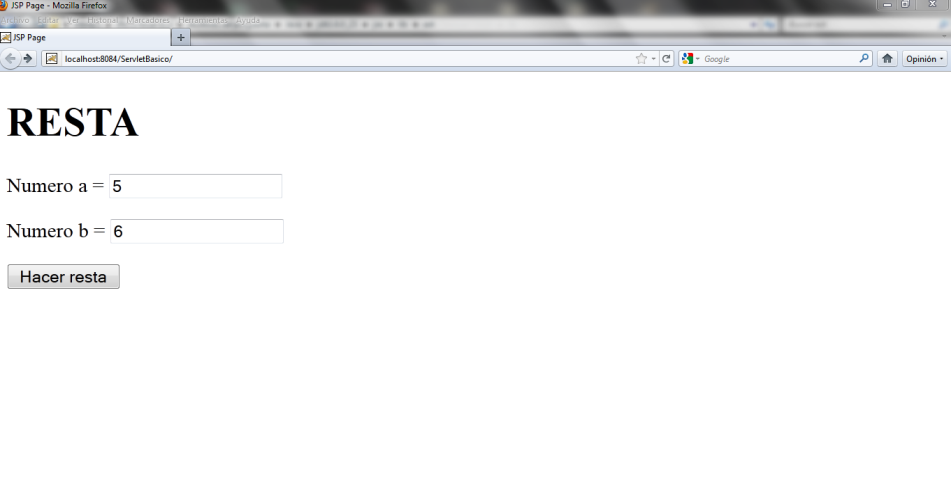
Si se utiliza NetBeans, no se debe crear manualmente el archivo web.xml, ya que de hacerlo asi, al momento de ejecutar el proyecto marca error. Lo que se debe hacer es dar Deploy, y automáticamente se genera.

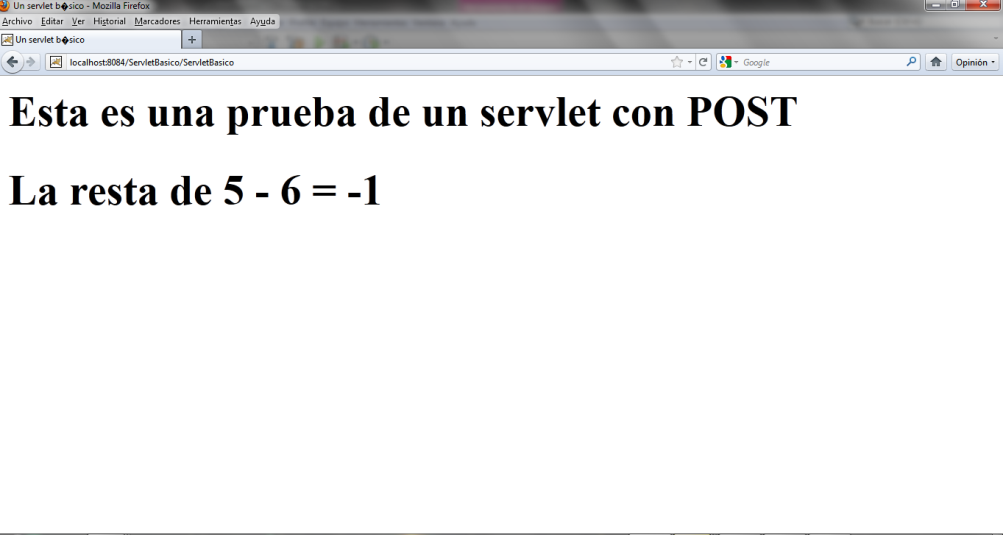
1. Finalmente se ejecuta el proyecto.

Antes de ejecutarlo, se debe verificar que Tomcat se inicie.









Observación: una de las diferencias que se pueden ver entre GET y POST es la siguiente:

GET <http://localhost:8084/ServletBasico/ServletBasico?a=3&b=4>

POST <http://localhost:8084/ServletBasico/ServletBasico>

En GET, la ruta muestra los valores que se están enviando.

respuesta.setContentType("text/html");

String actual = peticion.getParameter("actual");

String nacio = peticion.getParameter("nacio");

double act = Double.parseDouble(actual);

double nac = Double.parseDouble(nacio);

double result = act - nac;

if (result < 18) {

out.println("<html>");

out.println("<head>");

out.println("<title> Procesamiento de peticiones get con datos </title>");

out.println("</head>");

out.println("<body>");

out.println("Usted es mayor de edad");

out.println("</body>");

} else {

out.println("<html>");

out.println("<body>");

out.println("Usted es menor de edad");

out.println("</body>");

out.println("</html>");

out.close();

}