

# Lab 2 – Desarrollar un Servlet sencillo

## Objetivos

- Al finalizar el laboratorio debes ser capaz de desarrollar y publicar un Servlet sencillo.

## Preparación

- Este laboratorio asume que ya está instalada la versión de NetBeans que incluye un contenedor Web Glassfish atendiendo (*listening*) por el puerto 8080.

## Ejercicio

En este ejercicio realizarás lo siguiente:

1. Crear un nuevo proyecto Web en NetBeans.
2. Escribir una página Web simple que envíe un parámetro al Servlet.
3. Desarrollar un Servlet que lea el parámetro y responda con un mensaje de saludo.

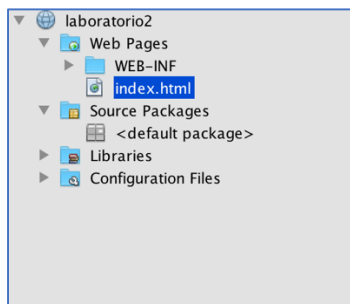
Este Servlet genera una respuesta HTTP (HTTP response) después de que el usuario envía su nombre desde el navegador Web.

### Actividad 1. Crear un nuevo proyecto Web en NetBeans.

Primero debes crear un proyecto en NetBeans donde posteriormente agregarás el archivo HTML y el Servlet de Java.

1. Entra al menú **File – New Project**.
2. Selecciona la categoría **Java Web** y el tipo de proyecto **Web Application**.
3. Asígnale al proyecto el nombre **laboratorio2**.
4. Toma nota de la ruta que se indica en **Project Location**, pues es la ruta donde se guardará el proyecto. Si quieres puedes especificar una ruta diferente, pero es importante que tomes nota de dónde lo estás guardando, pues usarás esa ruta para enviar el reporte de laboratorio. Da clic en *Next*.
5. Asegúrate de que esté seleccionado el Server **GlassFish**. Si solo tuviera la opción *Tomcat* no hay problema, quiere decir que en la instalación no se incluyó *GlassFish*, pero puede funcionar con Tomcat.
6. En la siguiente ventana NO seleccionar ningún framework.

Al crear el proyecto deben tener una estructura similar a esta:

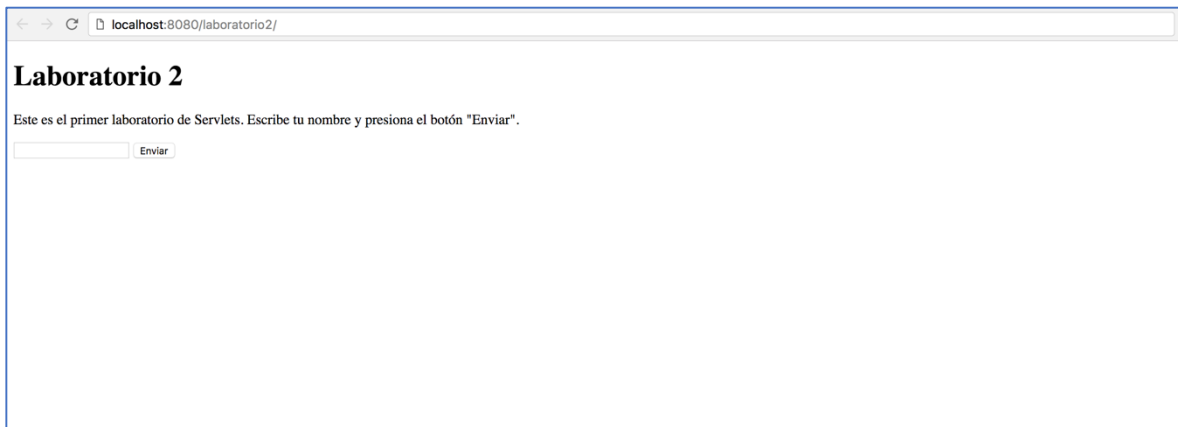


## Actividad 2. Escribir una página Web simple.

El siguiente paso es crear una página Web. Puedes utilizar la página **index.html** que se genera automáticamente en el proyecto o crear una nueva. Para este ejercicio usaremos la que ya está en el proyecto para modificarla.

1. En la sección de `<body>` agrega un encabezado con el título **Laboratorio 2**.
2. Después agrega un párrafo con una descripción breve de la página.
3. A continuación agrega un formulario **<form>** con un atributo **action** que indique qué acción se realizará. En nuestro caso será un servlet que desarrollarás en el siguiente paso, pero puedes agregarlo ahora con el valor **SaludoServlet**.
4. En la nueva sección de `<form>` agrega dos controles: un **input** de tipo **text** que se llame **"nombre"**, y un **input** de tipo **submit** con el valor **"Enviar"**.
5. Al final, fuera del formulario, agrega un encabezado pequeño con tu nombre.

El resultado debe ser una página similar a esta:



The screenshot shows a web browser window with the address bar displaying `localhost:8080/laboratorio2/`. The page title is **Laboratorio 2**. Below the title, there is a paragraph of text: "Este es el primer laboratorio de Servlets. Escribe tu nombre y presiona el botón "Enviar".". Below this text is a form consisting of a text input field and a submit button labeled "Enviar".

Puedes ejecutar la aplicación dando clic con el botón derecho en el proyecto y escogiendo la opción **Run**, lo cual abrirá una ventana de tu navegador con la página `index.html`, o bien abriendo directamente la página en tu navegador escribiendo el URL:

**`http://localhost:8080/laboratorio2/`**

Recuerda que, dado que **index.html** es la página principal, puedes no escribirla.

## Actividad 3. Desarrollar un Servlet.

Por último crearemos el Servlet de Java que recibirá el dato proveniente del formulario, lo procesará y enviará de regreso la respuesta al usuario.

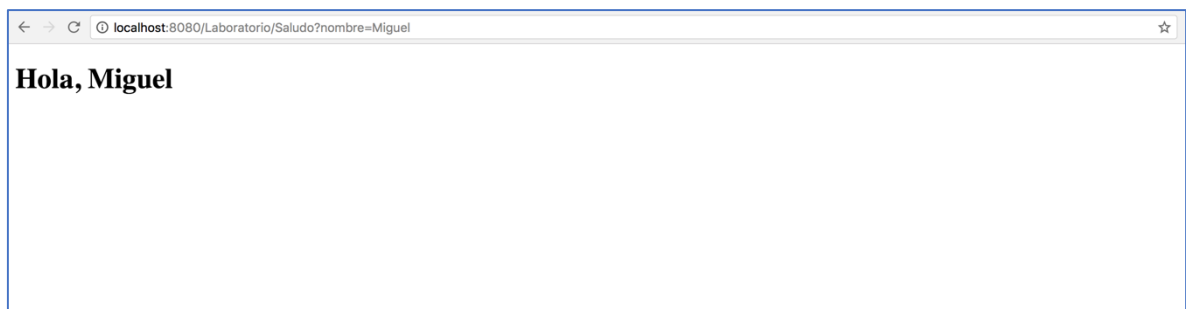
1. En el proyecto **laboratorio2**, hacer clic derecho en el folder **Source Packages** y en la opción **New** seleccionar **Servlet...**
2. El nombre del Servlet será **SaludoServlet**. Deja las demás opciones como están y da clic en **Next**.
3. Marca la opción "Add information to deployment descriptor (web.xml)".
4. Observa que el URL Pattern será **/SaludoServlet**. No lo cambies por ahora, aunque puede modificarse para que diga algo como "saludo" solamente, sin revelar el nombre de la clase de Java. Por lo general si una aplicación dice "Servlet" en el URL, es fácil saber que está hecha en Java y un atacante puede buscar vulnerabilidades específicas más fácilmente.

5. Modifica la línea que está antes de la declaración de clase donde dice **@author** y escribe tu nombre completo a continuación, seguido de tu número de matrícula.
6. En el código del nuevo Servlet, en el método **processRequest()** antes del primer **out.println()** agregar una variable de tipo **String** llamada **nombre** a la que le asignarás el dato que viene del formulario usando el método **getParameter()** del objeto **request**. La línea debe ser similar a:

```
String nombre = request.getParameter("nombre");
```

Es importante que el nombre que está entre comillas coincida con el nombre del parámetro que quieres leer y que se envía desde el formulario HTML.

7. Haz las modificaciones necesarias en las líneas de **out.println()** en el Servlet de manera que al ejecutar el Servlet en el paso siguiente se obtenga algo similar a esto:



- a. El título del HTML de respuesta debe decir "Saludo".
  - b. El mensaje debe ser un encabezado que agregue el nombre almacenado en la variable **nombre** del paso anterior. En Java puedes concatenar cadenas usando el operador **+** de esta manera: "**cadena 1**" + **variable** + "**cadena 2**"
8. Para ejecutar el Servlet haz lo siguiente:
  - a. Abre la página **index.html** en el navegador.
  - b. Escribe un nombre y presiona el botón "Enviar". El Servlet deberá responder con un mensaje de saludo.

## Preguntas de Reflexión

1. ¿Qué ocurriría si, en lugar de escribir su nombre, el usuario agregara algo diferente, por ejemplo: **<script>alert("prueba")</script>**?
  - a. ¿Qué comportamiento tiene el navegador?
  - b. Desde el punto de vista de seguridad, ¿qué implicaciones o posibles riesgos tiene esto?
  - c. ¿Cómo te imaginas que podría evitarse esta situación?
2. ¿Qué ocurre si cambias el valor al final de el URL en la respuesta del Servlet?
  - a. Desde el punto de vista de seguridad, ¿qué implicaciones o posibles riesgos tiene esto?
  - b. ¿Cómo te imaginas que podría evitarse esta situación?

## Especificaciones de la Entrega del Reporte de Laboratorio

1. El reporte debe incluir una **portada** con tus datos al principio.

2. El reporte debe contener la descripción de los pasos realizados para llevar a cabo la práctica del laboratorio. Cada paso debe contar con un fragmento de código o imagen que ilustre lo descrito. Piense en el reporte como una explicación para alguien ajeno al tema y detalle los puntos técnicos que sean necesarios.
3. Contesta las preguntas mencionadas en la sección “Preguntas de reflexión”.
4. Comprime en un archivo **.zip** el directorio raíz de la práctica, que en este ejemplo se menciona en la actividad 1 como “**laboratorio2**” (obviamente se incluirá todo su contenido).
5. El envío de la tarea debe incluir **dos** archivos:
  - a. El reporte de laboratorio en formato **PDF**.
  - b. El archivo .zip con el código fuente del proyecto.