

Sesiones

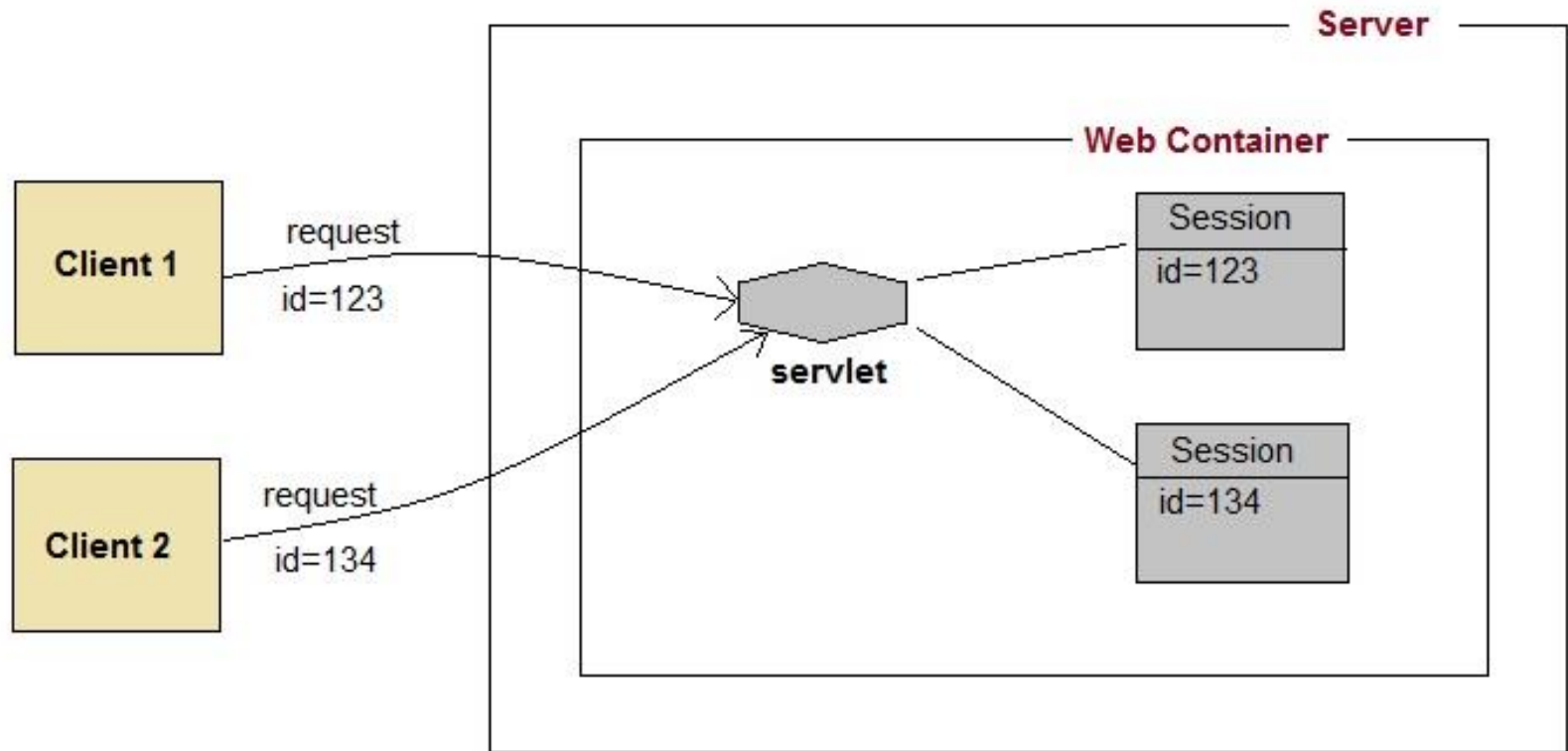
Carrera: COMPUTACIÓN E INFORMÁTICA

Semestre: 2017 - II

Nombre de Unidad Didáctica: TALLER DE PROGRAMACION CONCURRENTE



1. Introducción



- El protocolo HTTP permite acceder a páginas web y enviar datos de un formulario pero tiene una limitación: es un protocolo sin estado.
- Cuando se solicita una página independientemente del tipo que sea, el servidor abre una conexión por la que envía los datos y luego ésta es cerrada una vez que ha terminado

3. ¿Qué es una sesión?

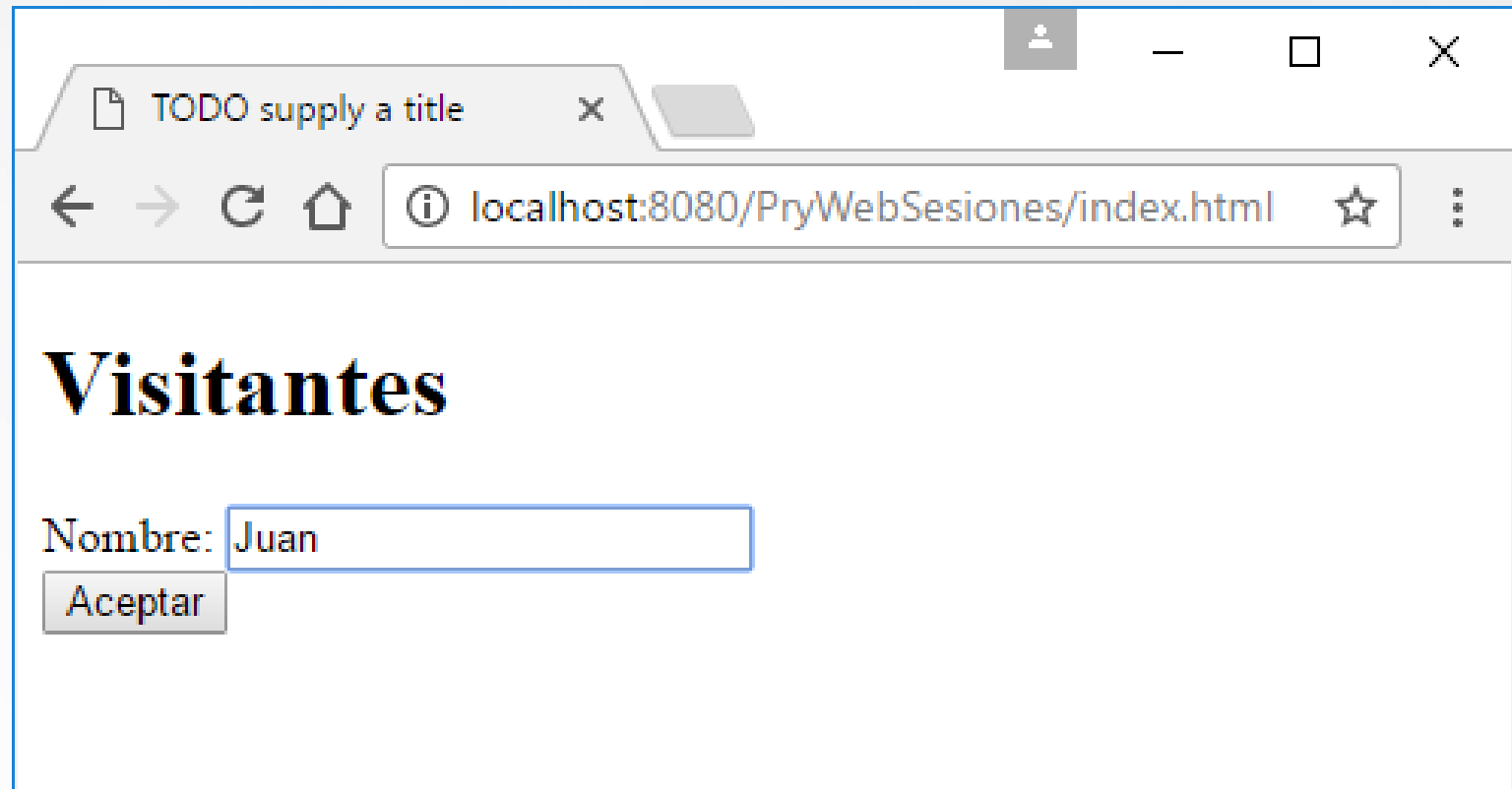
- Una sesión es una serie de comunicaciones entre un cliente y un servidor en la que se realiza un intercambio de información.
- Por medio de una sesión se puede hacer un seguimiento de un usuario a través de la aplicación.
- Tiempo de vida de una sesión comienza cuando un usuario se conecta por primera vez a un sitio web y su finalización puede darse cuando:
 - Se abandona el sitio web.
 - Se alcanza un tiempo de inactividad
 - Se ha cerrado o reiniciado el servidor

- Acceder al objeto sesión
 - Mediante el método `request.getSession()` se obtiene un objeto `HttpSession`
- Obtener información asociada con la sesión
 - Mediante el método `getAttribute` del objeto `HttpSession`
- Guardar información en una sesión
 - Usar el método `setAttribute` con un nombre y un valor
- Descargar datos de la sesión
 - Método `removeAttribute()`
 - Método `invalidate`, descarga una sesión entera

5. Métodos de la clase HttpSession

- `getCreationTime()`
 - Devuelve la hora de creación de la sesión en milisegundos desde las 00:00 del 01/01/1970
- `getId()`
 - Devuelve el identificador asignado a la sesión
- `getLastAccessedTime()`
 - Devuelve la hora de la última petición al servlet con el identificador asignado a esa sesión
- `getSessionContext()`
 - Devuelve el contexto al que está asociada la sesión
- `getValue(String)`
 - Devuelve el objeto asociado al nombre dado

6. Ejemplo



A screenshot of a web browser window. The address bar shows 'localhost:8080/PryWebSesiones/index.html'. The page title is 'TODO supply a title'. The main content area displays the word 'Visitantes' in a large, bold, black serif font. Below this, there is a form with the label 'Nombre:' followed by a text input field containing the name 'Juan'. Below the input field is a button labeled 'Aceptar'.

Visitantes

Nombre:

```
<body>  
  <h1>Visitantes</h1>  
  <form action="SesionServlet">  
    Nombre: <input type="text" name="nombre" /><br>  
    <input type="submit" value="Aceptar" />  
  </form>  
</body>
```


6. Ejemplo

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    //Buscamos el listado en la sesión
    Vector listado = (Vector) request.getSession().getAttribute("listado");
    if (listado == null) {
        listado = new Vector();
    }
    //Añadimos el visitante al listado en caso de que hayamos recibido su nombre
    if ((String) request.getParameter("nombre") != null) {
        listado.addElement((String) request.getParameter("nombre"));
    }
    //Para el caso de que el listado no estuviera en sesión lo añadimos
    request.getSession().setAttribute("listado", listado);
    //Mostramos lista de visitante
    out.println("<h1>Visitantes</h1>");
    for (int i = 0; i < listado.size(); i++) {
        out.println((String) listado.elementAt(i)+"<br>");
    }
    out.println("<br><a href=\"index.html\">volver</a><br>");
}
```

6. Ejemplo

