

## MÓDULO 3

### EJERCICIOS RESUELTOS ATENDIENDO PETICIONES SERVLET

#### TEORIA 1

Explica el ciclo de vida de un Servlet.

- 1) La clase Java que representa el Servlet se despliega en el contenedor web.
- 2) El contenedor web crea una instancia de la clase Servlet y la carga en memoria.
- 3) El contenedor Web pasa seguidamente a la fase de inicialización del Servlet, llamando al metodo `init(ServletConfig)`. La implementación de dicho método permite, mediante el objeto `ServletConfig`, recuperar los parámetros de inicialización del Servlet, definidos en descriptor de despliegue (`web.xml`) de la aplicación.
- 4) El Servlet pasa a continuación a un estado llamado “en Servicio”, en espera de recepción de peticiones cliente. El contenedor web genera una cola de espera de threads, y ejecuta la instancia del servlet en un contexto multitarea. Cada thread llama al método `Service(...request,...response)` del servlet, quien se ocupa de ejecutar el método http correspondiente (`doPost()`, `doGet()`)
- 5) La instancia del servlet permanece cargada en memoria mientras se ejecute el contenedor web. Si éste se para, se llama al método `destroy()` sobre la instancia Servlet, indicando que ya no está en servicio.
- 6) Y, para acabar, como un Servlet es un objeto Java, la instancia del Servlet es borrada de la memoria por el garbage collector de Java.

#### TEORIA 2

Explica cómo funciona un Servlet a nivel interno (métodos `service` y `doXXX`)

Cuando un cliente llama a un servlet emitiendo una petición http, se llama al método `service()` sobre el contenedor Web. El método `service(...)` recibe dos parámetros, de tipo `javax.servlet.ServletRequest` y `javax.servlet.ServletResponse`.

El único proceso realizado por el método `Service(..)` es la invocación del propio método `service(..)` sobrecargado, que toma dos parámetros, del tipo `javax.servlet.http.HttpServletRequest` y `javax.servlet.http.HttpServletResponse`.

Además el método `service(..)` invoca por defecto al metodo correspondiente al método HTTP utilizado por el cliente (principalmente `doGet(...)`, `doPost(..)`, `doPut(..)`, `doDelete(..)`), transmitiéndole los parámetros de tipo `javax.servlet.http.HttpServletRequest` y `javax.servlet.http.HttpServletResponse`.

El desarrollador será quien se encargue de implementar entonces los procesos sobre la respuesta del método `doXXX(...)` o aquellos que utilicen la misma.

#### TEORIA 3

3. Comenta la relación que hay entre las clases Servlet, ServletRequest, ServletResponse, HTTPServlet, HTTPServletRequest y HTTPServletResponse.

Las clases Servlet, ServletRequest y ServletResponse estan contenidas en la API Servlet en el paquete `javax.servlet`.

Las clases HTTPServlet, HTTPServletRequest y HttpServletResponse están contenidas en la API Servlet en el paquete `javax.servlet.http`

ServletRequest y ServletResponse se utilizan como parámetros para la implementación del método `service` (`public void service(ServletRequest req, ServletResponse res)`), mientras que HTTPServletRequest y HttpServletResponse se utilizan para la implementación del método sobrecargado:

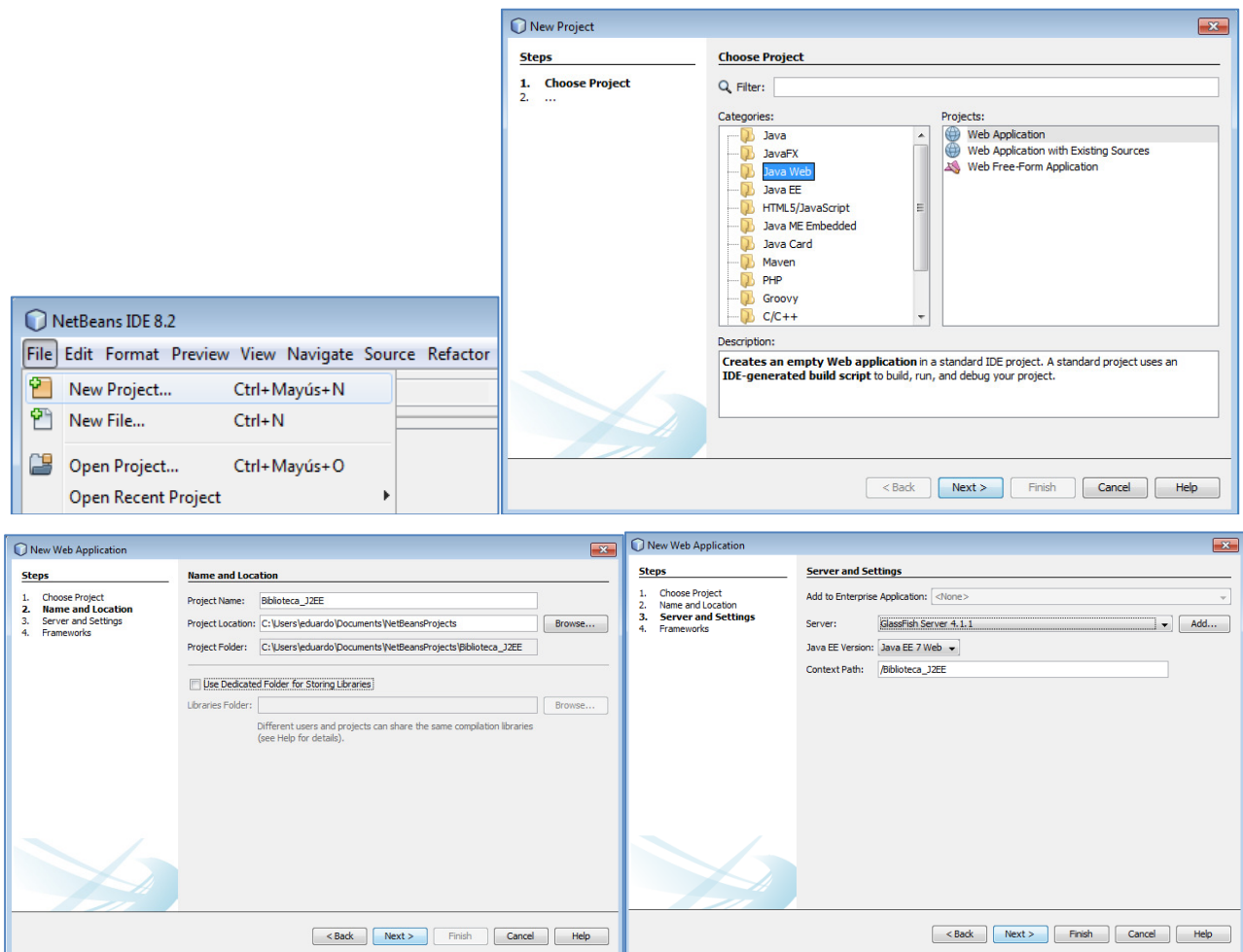
`service( public void service (que HttpServletRequest req, HttpServletResponse res))`

#### EJERCICIO 1

Crea un Proyecto J2EE al que llamaremos Biblioteca\_J2EE con NetBeans o Eclipse. Estos IDE's crean automáticamente la estructura del proyecto J2EE con sus carpetas (WEB-INF, clases...

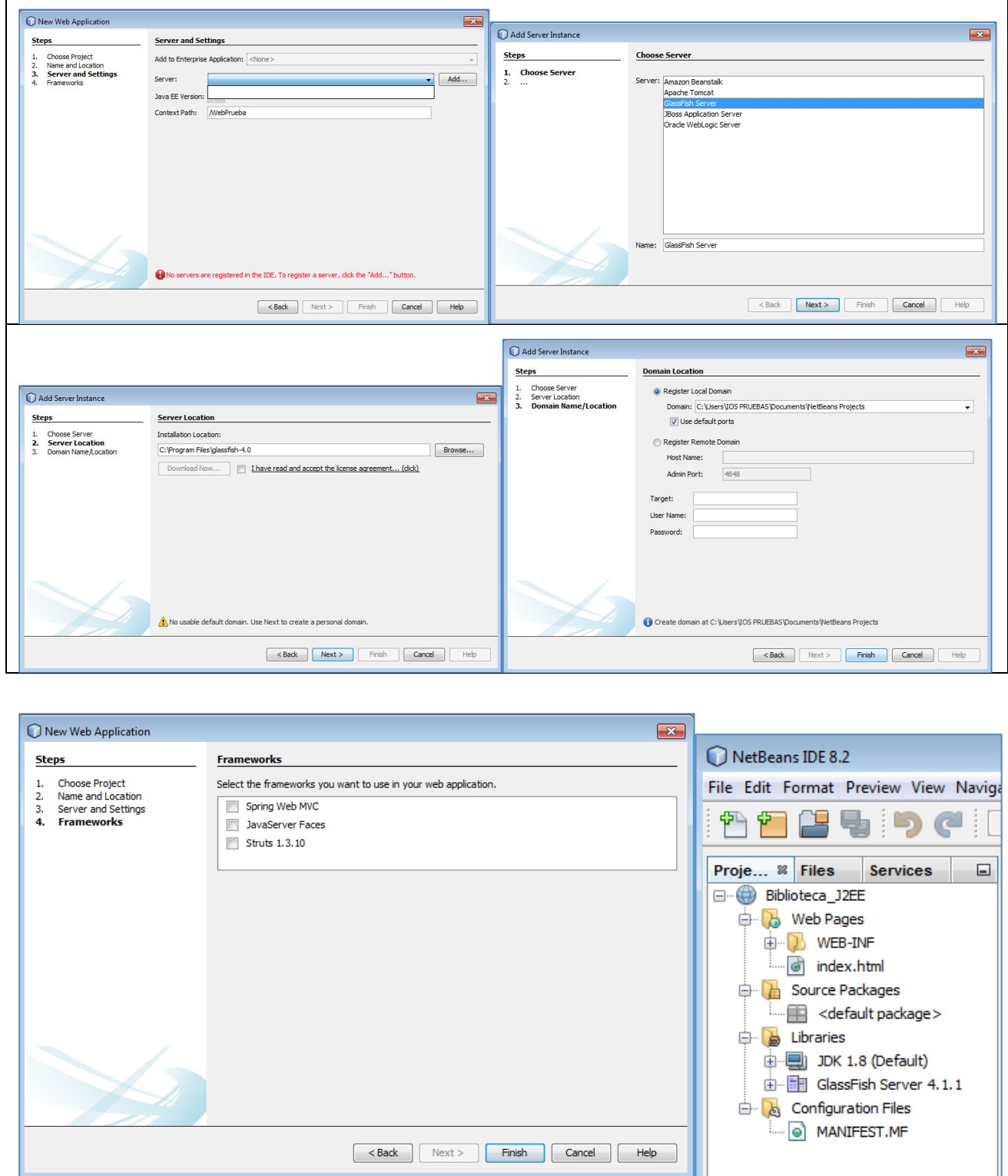
etc.) y su fichero de configuración web.xml. Si el IDE no tiene un servidor predefinido, agrégalo para que el proyecto lo tenga asignado. Esto lo podemos hacer en el momento de la creación del proyecto o posteriormente cuando ya está creado:

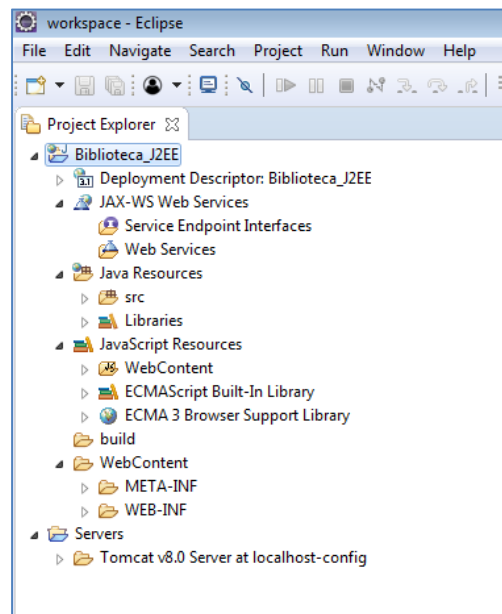
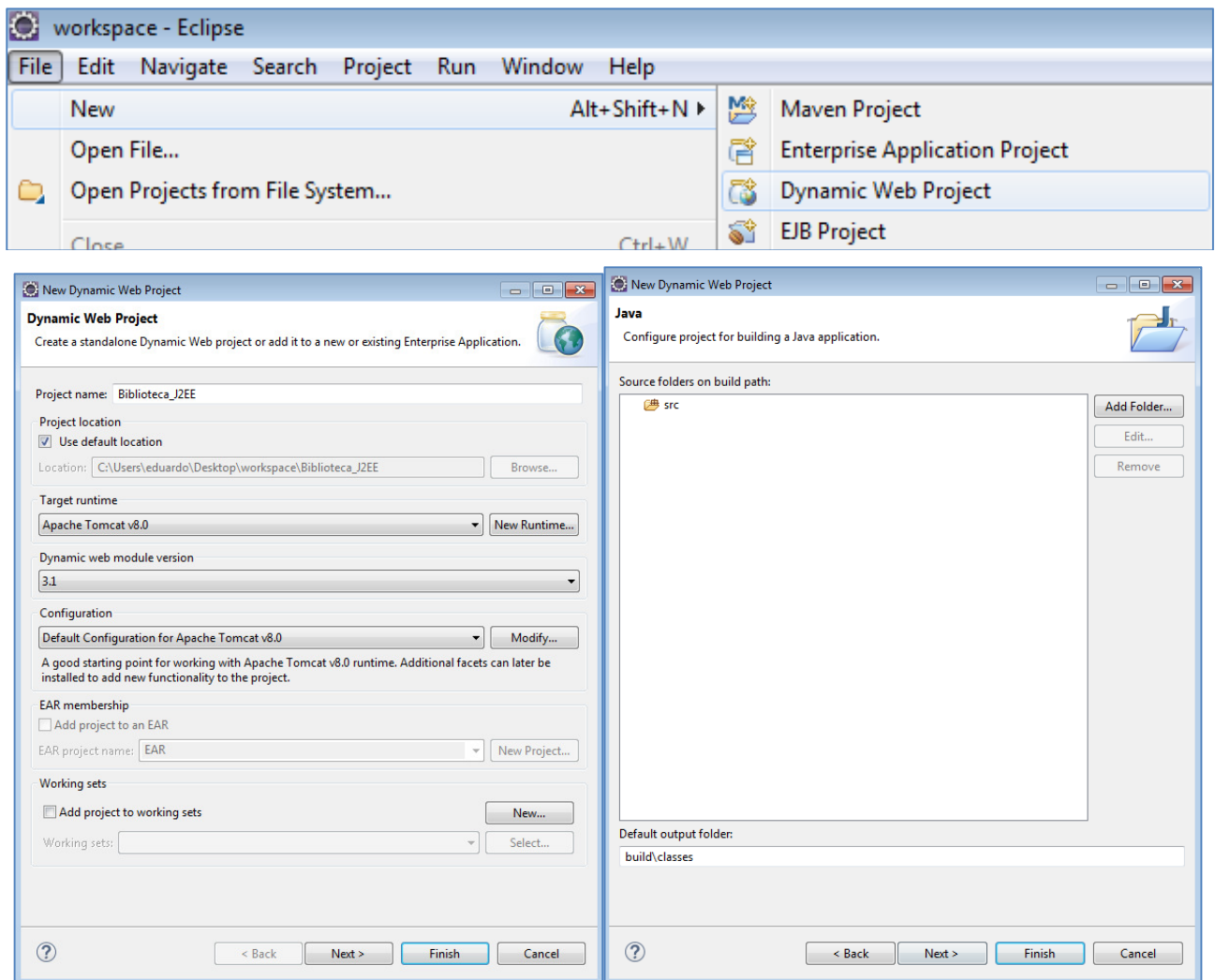
## NETBEANS



Si no tuviésemos definido ningún servidor de aplicaciones en Netbeans, lo podemos agregar en este mismo momento (durante la creación del proyecto) o posteriormente cuando ya está creado:

## Agregar servidor en el momento de creación del proyecto

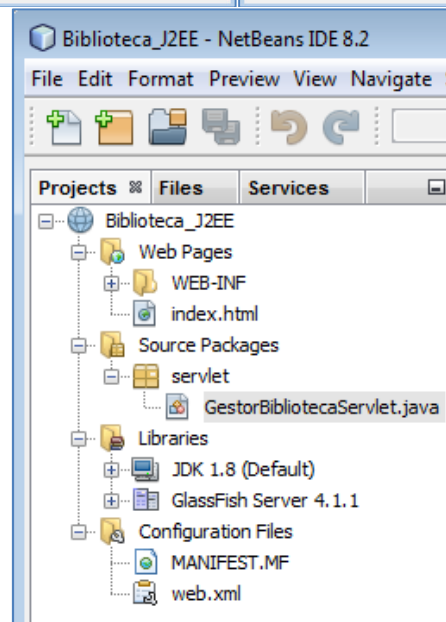
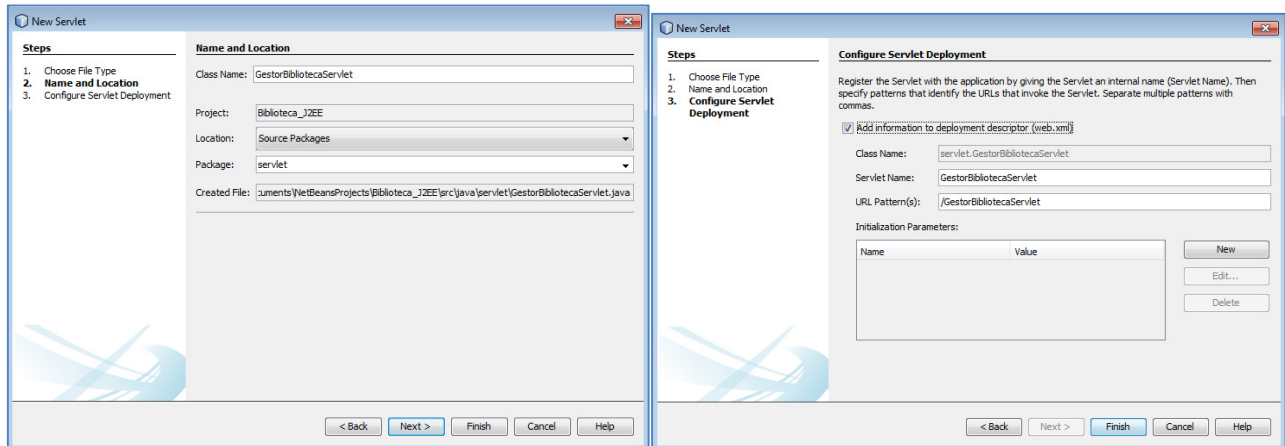
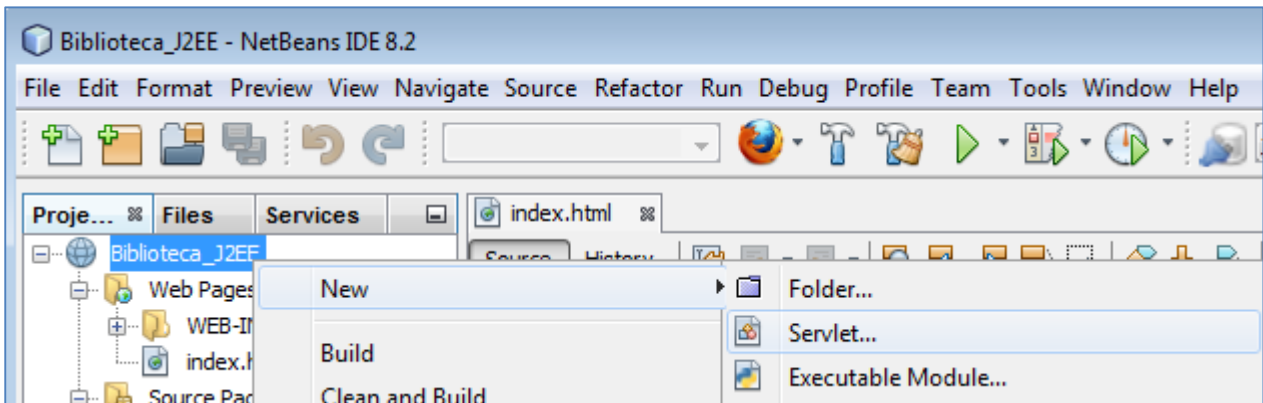




## EJERCICIO 2

Crea un Servlet al que llamaremos “GestorBibliotecaServlet”, dentro del package servlet.

## NETBEANS



```

public class GestorBibliotecaServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet GestorBibliotecaServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet GestorBibliotecaServlet at " +
                request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

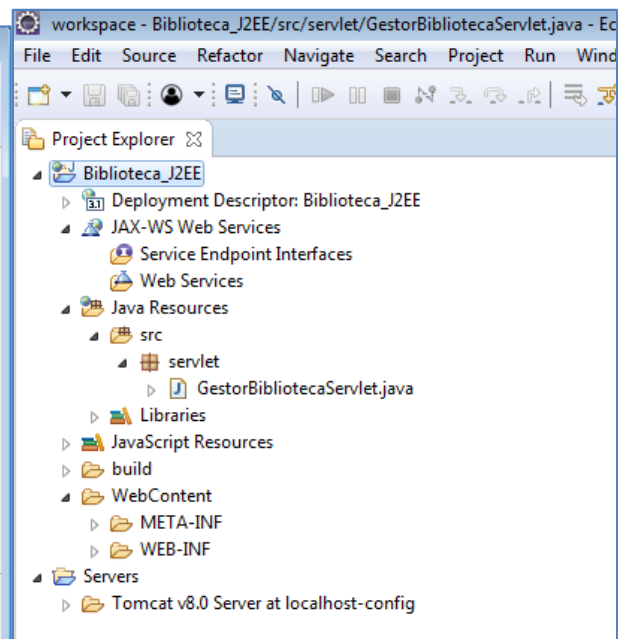
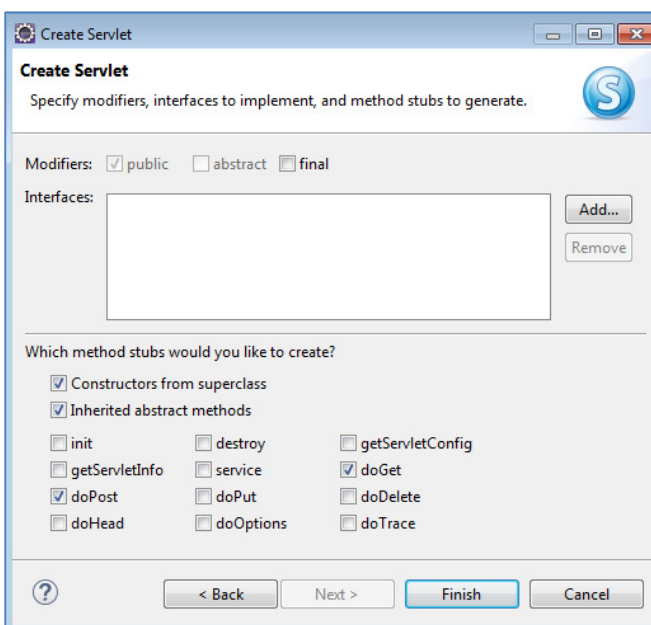
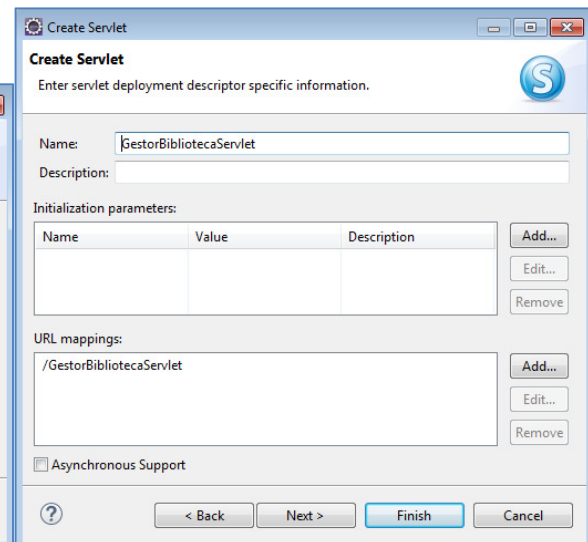
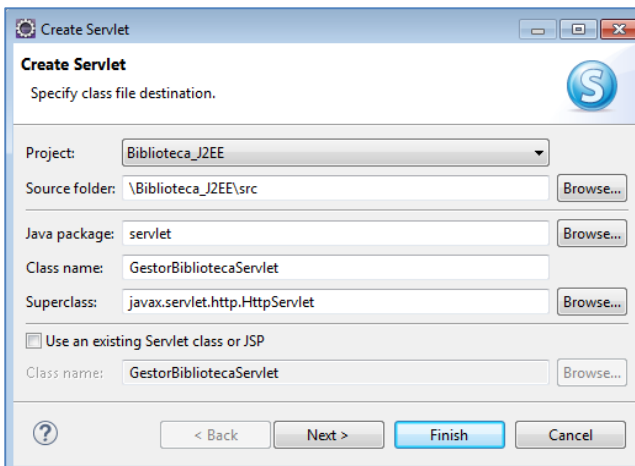
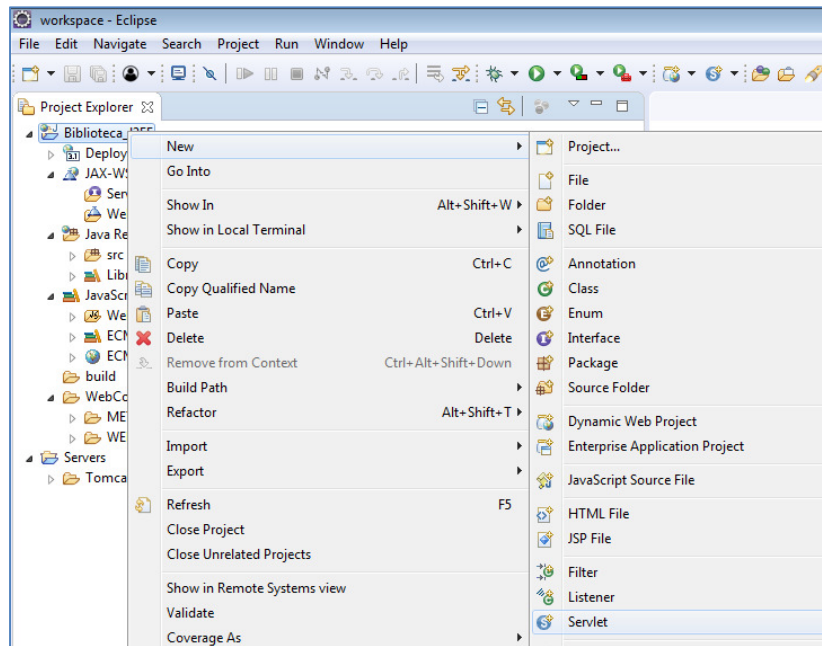
```

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```







```

1 package servlet;
2
3 import java.io.IOException;
4
5
6
7
8
9
10 /**
11  * Servlet implementation class GestorBibliotecaServlet
12  */
13 @WebServlet("/GestorBibliotecaServlet")
14 public class GestorBibliotecaServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * @see HttpServlet#HttpServlet()
19      */
20     public GestorBibliotecaServlet() {
21         super();
22         // TODO Auto-generated constructor stub
23     }
24
25     /**
26      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27      */
28     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29         // TODO Auto-generated method stub
30         response.getWriter().append("Served at: ").append(request.getContextPath());
31     }
32
33     /**
34      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35      */
36     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // TODO Auto-generated method stub
38         doGet(request, response);
39     }
40
41 }
42

```

### EJERCICIO 3

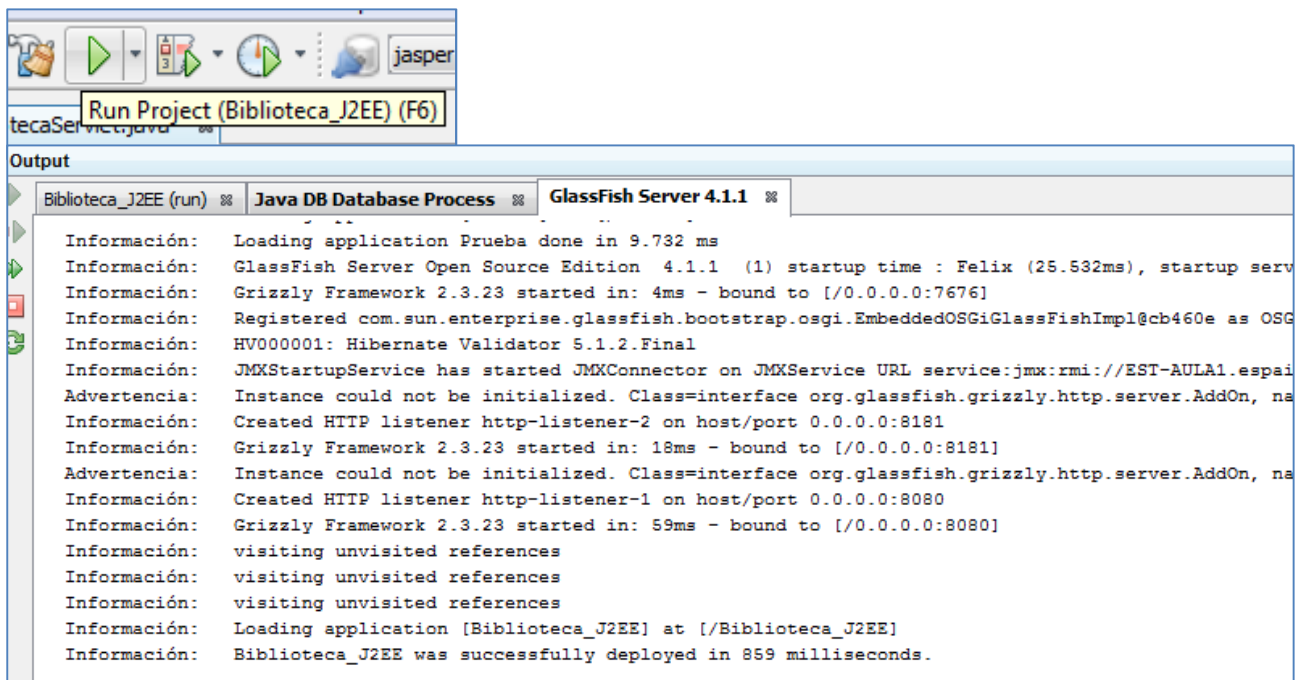
Ejecuta la aplicación web e inicia el servidor web. Desde un navegador pon la siguiente url:

[http://localhost:8080/Biblioteca\\_J2EE/GestorBibliotecaServlet](http://localhost:8080/Biblioteca_J2EE/GestorBibliotecaServlet)

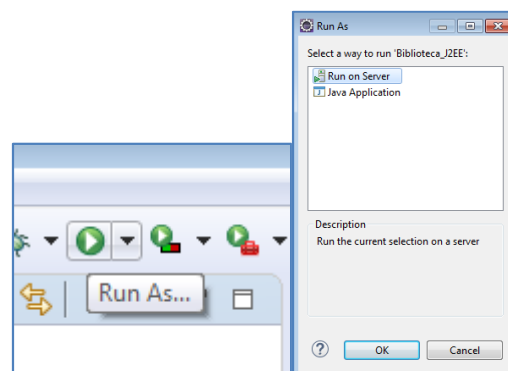
Podemos comprobar que el servlet queda configurado como un servicio web que está preparado para atender peticiones a través del protocolo HTTP.

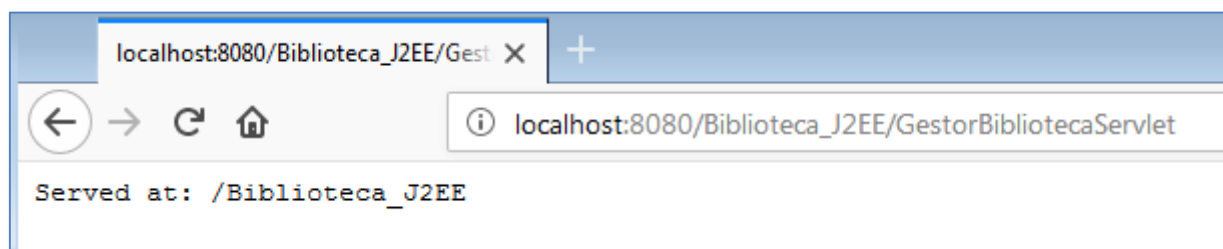
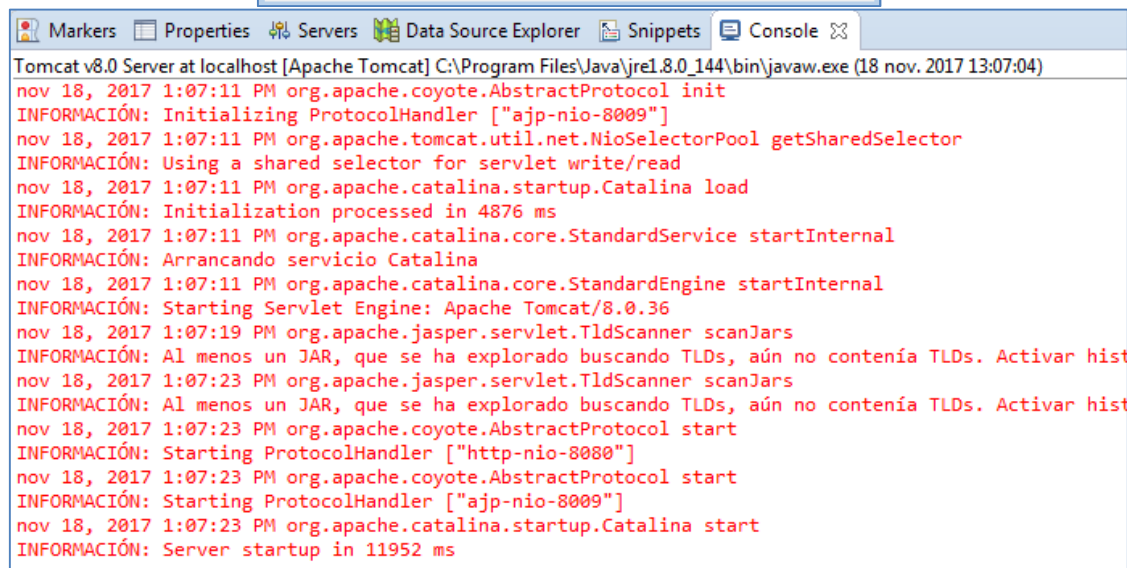
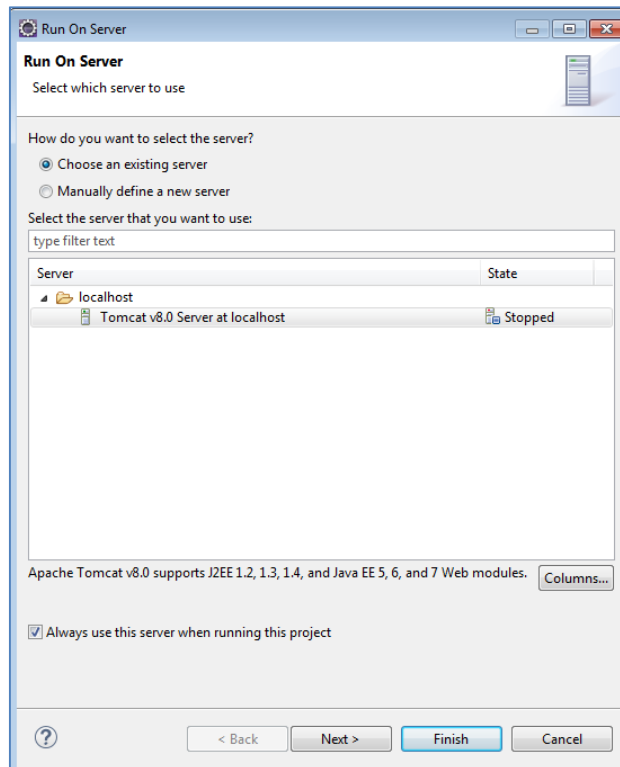
## NETBEANS





## ECLIPSE





## EJERCICIO 4

Modifica la respuesta del servlet GestorBibliotecaServlet para que pueda dar mensajes diferentes en función del tipo de petición:

- “Hola Mundo POST!!” si recibe una petición POST
- “Hola Mundo GET!!” si recibe una petición GET.

Sigue el modelo que utiliza Eclipse en la creación por defecto de la plantilla de un servlet, el cual hace las respuestas de doGet y de doPost independientes mediante la instrucción:

response.getWriter().append(mensaje).

### NETBEANS

```
public class GestorBibliotecaServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.getWriter().append("<h1>HOLA MUNDO GET</h1>");
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.getWriter().append("<h1>HOLA MUNDO POST</h1>");
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```

### ECLIPSE

```
10 /**
11  * Servlet implementation class GestorBibliotecaServlet
12  */
13 @WebServlet("/GestorBibliotecaServlet")
14 public class GestorBibliotecaServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * @see HttpServlet#HttpServlet()
19      */
20     public GestorBibliotecaServlet() {
21         super();
22         // TODO Auto-generated constructor stub
23     }
24
25     /**
26      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27      */
28     protected void doGet(HttpServletRequest request, HttpServletResponse response)
29         throws ServletException, IOException {
30         // TODO Auto-generated method stub
31         response.getWriter().append("<h1>HOLA MUNDO GET!!</h1>");
32     }
33
34     /**
35      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
36      */
37     protected void doPost(HttpServletRequest request, HttpServletResponse response)
38         throws ServletException, IOException {
39         // TODO Auto-generated method stub
40         response.getWriter().append("<h1>HOLA MUNDO POST!!</h1>");
41     }
42 }
```

## EJERCICIO 5

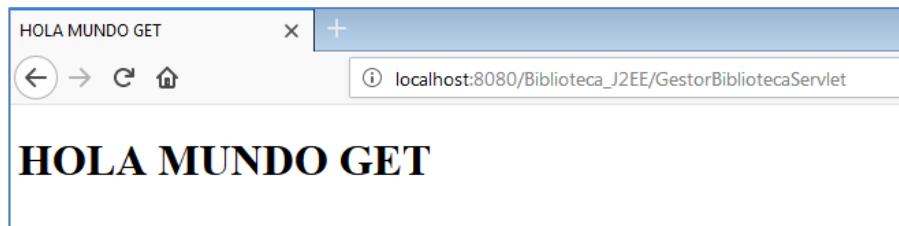
Repite el ejercicio 3. Inicia la aplicación web y desde un navegador pon la siguiente url:

[http://localhost:8080/Biblioteca\\_J2EE/GestorBibliotecaServlet](http://localhost:8080/Biblioteca_J2EE/GestorBibliotecaServlet)

¿Qué mensaje da el servlet?

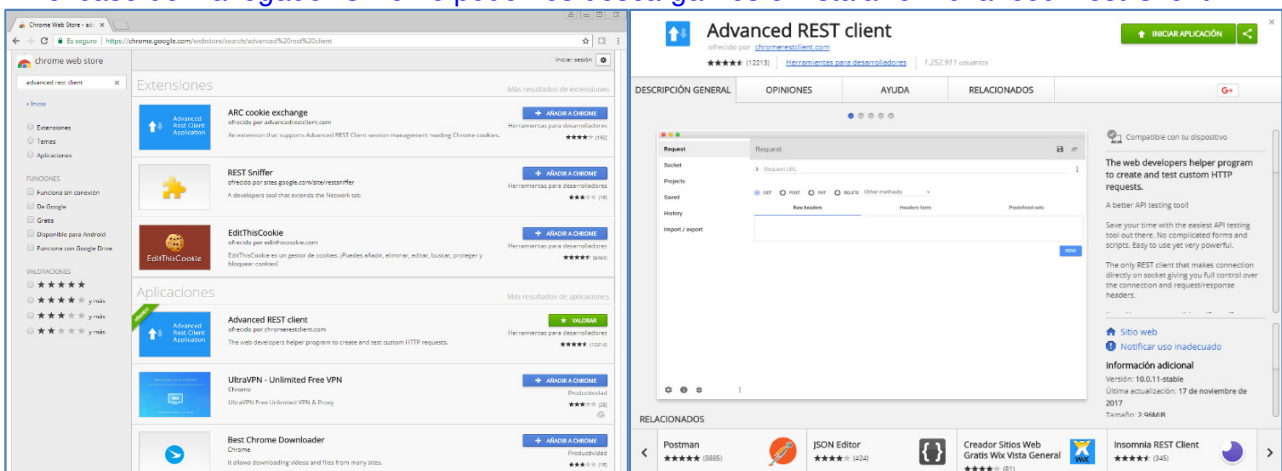
Indica dos sistemas que nos permitirían obtener los dos tipos de mensajes: GET y POST.

Idea: Mira el complemento Advanced Rest Client del navegador Chrome.

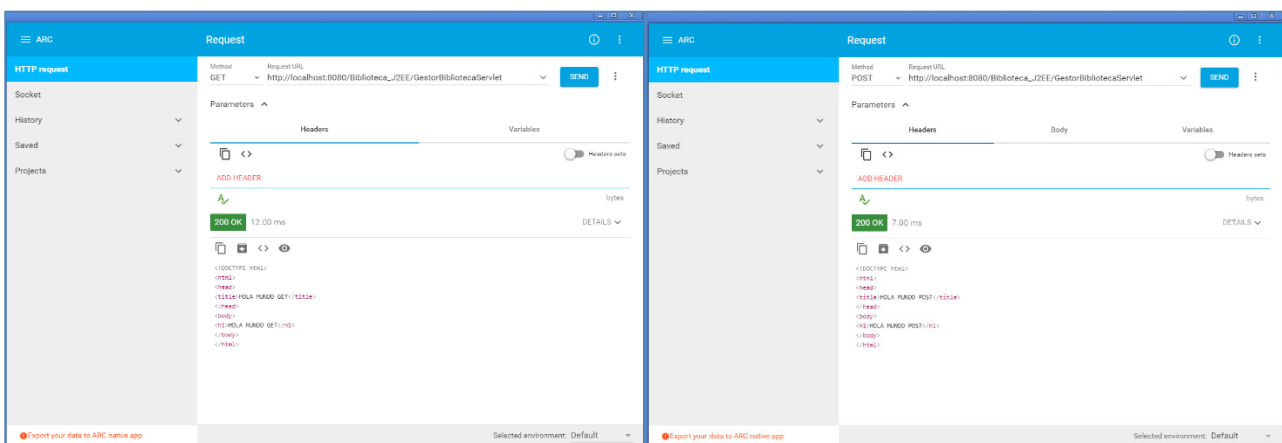


El mensaje que nos da el servlet es “Hola Mundo GET”, que es debido a que todas las llamadas realizadas directamente con una url desde un navegador producen un mensaje tipo GET hacia el servlet. Para conseguir realizar peticiones HTTP de los dos tipos, tanto GET como POST, se deben de hacer con formularios HTML o con algún plugin de un navegador.

En el caso del navegador Chrome podemos descargarnos e instalar el Advanced Rest Client:



Podemos generar peticiones tanto GET como POST gráficamente hacia nuestro servlet:



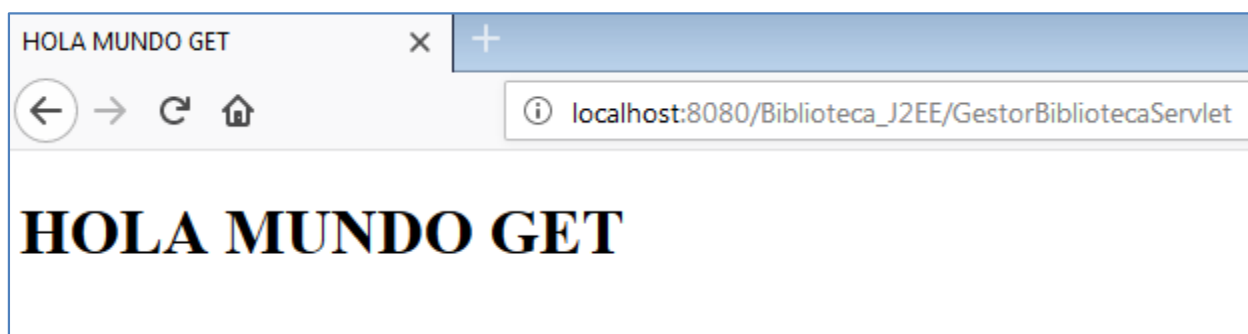
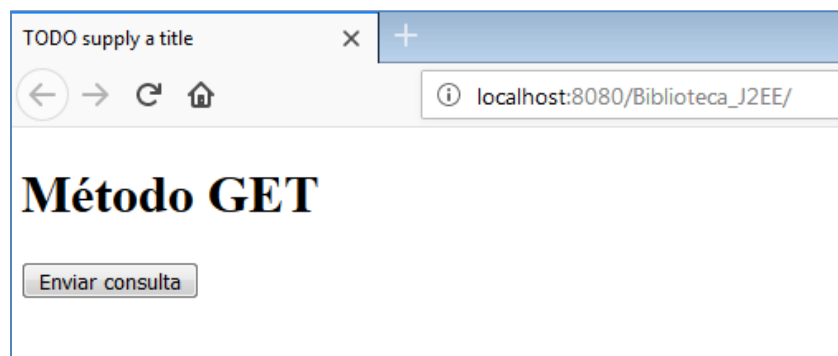
## EJERCICIO 6

Utiliza la página web index.html para crear un formulario que contenga un único botón de submit. Este formulario debe llamar al servlet GestorBibliotecaServlet a través del método GET.

Ejecuta la aplicación y llama al servlet desde el formulario. Indica cual es el mensaje de salida que se obtiene.

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div>TODO write content</div>
  </body>
</html>
```

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Método GET</h1>
    <form method="GET" action="GestorBibliotecaServlet">
      <input type="submit">
    </form>
  </body>
</html>
```



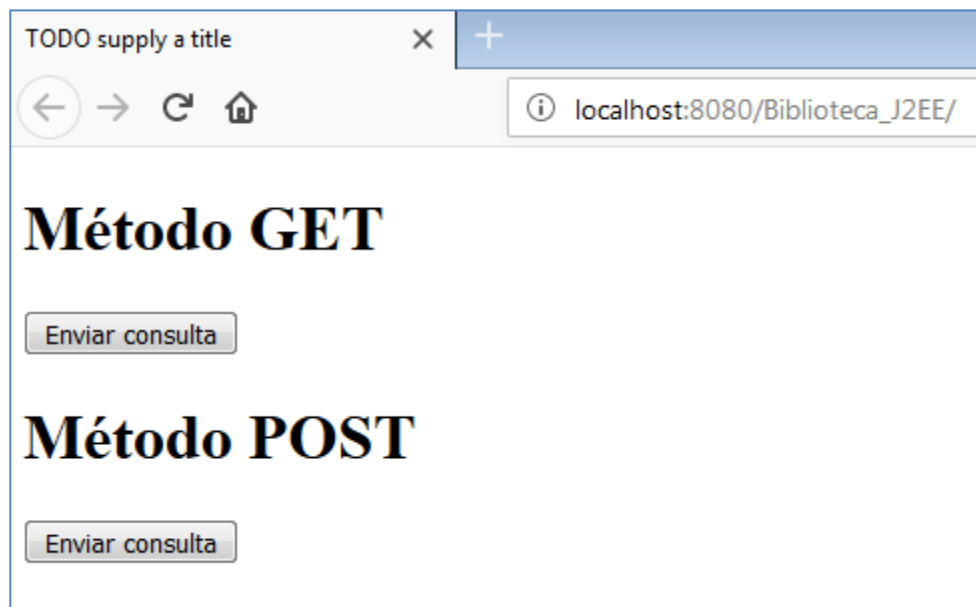
## EJERCICIO 7

Agrega otro formulario en el archivo index.html de manera que llame al servlet GestorBibliotecaServlet a través del método POST. Ejecuta la aplicación y mira el mensaje de salida que da el servlet.

```

<html>
  <head>
    <title>MÉTODOS GET-POST</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Método GET</h1>
    <form method="GET" action="GestorBibliotecaServlet">
      <input type="submit">
    </form>
    <h1>Método POST</h1>
    <form method="POST" action="GestorBibliotecaServlet">
      <input type="submit">
    </form>
  </body>
</html>

```



## EJERCICIO 8

Aprovechando la estructura de la función `processRequest`, que crea por defecto Netbeans cuando crea la plantilla de un servlet, haz que el servlet `GestorBibliotecaServlet` siga haciendo lo mismo, es decir dar un mensaje de "Hola Mundo POST!!" si recibe una petición POST y "Hola Mundo GET!!" si recibe una petición GET.

Idea: Para lograrlo se puede incluir un nuevo parámetro booleano dentro de la función `processRequest`.

```

public class GestorBibliotecaServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet NewServlet at " + request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```



```

public class GestorBibliotecaServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response,
        boolean methodGet) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            if (methodGet) {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>HOLA MUNDO GET</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>HOLA MUNDO GET</h1>");
                out.println("</body>");
                out.println("</html>");
            } else {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>HOLA MUNDO POST</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>HOLA MUNDO POST</h1>");
                out.println("</body>");
                out.println("</html>");
            }
        }
    }
}

```

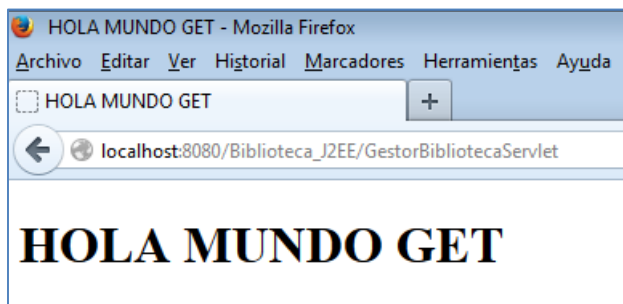
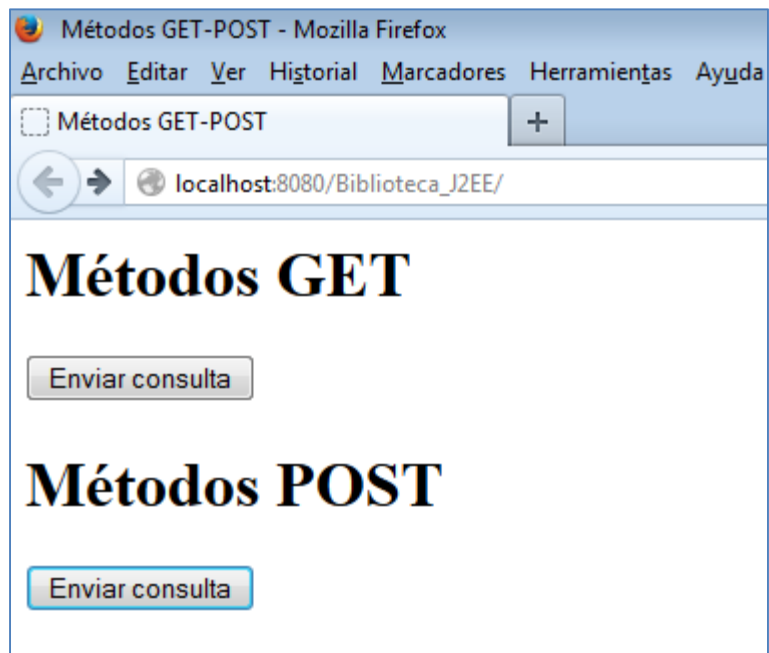
```

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response, true);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response, false);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```



## EJERCICIO 9

Realiza la misma operativa que en el ejercicio anterior, usando la estructura de la función `processRequest`, pero ahora utiliza una variable privada booleana para diferenciar entre llamadas GET y POST.

```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean methodGet;
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            if (methodGet) {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>HOLA MUNDO GET</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>HOLA MUNDO GET</h1>");
                out.println("</body>");
                out.println("</html>");
            } else {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>HOLA MUNDO POST</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>HOLA MUNDO POST</h1>");
                out.println("</body>");
                out.println("</html>");
            }
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        methodGet=true;
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        methodGet=false;
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

## EJERCICIO 10

Divide la función `processRequest` en dos partes, la parte GET y la parte POST, y asigna cada parte a las funciones `doGet` y `doPost`, de forma que funcionen de forma independiente y no se tenga que utilizar variables para diferenciar entre ambas llamadas.

```

public class GestorBibliotecaServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>HOLA MUNDO GET</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>HOLA MUNDO GET</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>HOLA MUNDO POST</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>HOLA MUNDO POST</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```

## EJERCICIO 11

Trataremos la conexión con la base de datos en capítulos posteriores, pero como “adelanto” debemos saber que ésta se iniciará en cuanto se cree el Servlet en memoria.

Modifica el ejercicio para que el usuario reciba el mensaje “Conectado a la BD” sólo la primera vez que acceda al Servlet.

Lo haremos con una variable booleana `YaIniciado`.

```

public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>HOLA MUNDO GET</title>");
            out.println("</head>");
            out.println("<body>");
            if (YaIniciado == false) {
                out.println("<h1>Conectado a la BD(llamada GET)</h1>");
                YaIniciado = true;
            }
            out.println("<h1>HOLA MUNDO GET</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

```

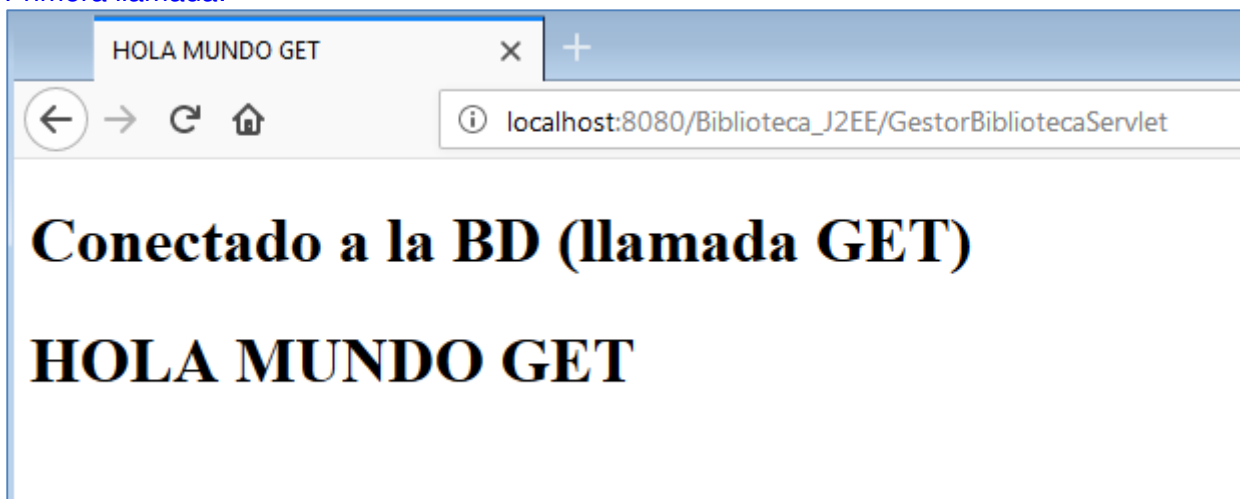
```

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>HOLA MUNDO POST</title>");
        out.println("</head>");
        out.println("<body>");
        if (YaIniciado == false) {
            out.println("<h1>Conectado a la BD(llamada POST)</h1>");
            YaIniciado = true;
        }
        out.println("<h1>HOLA MUNDO POST</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}

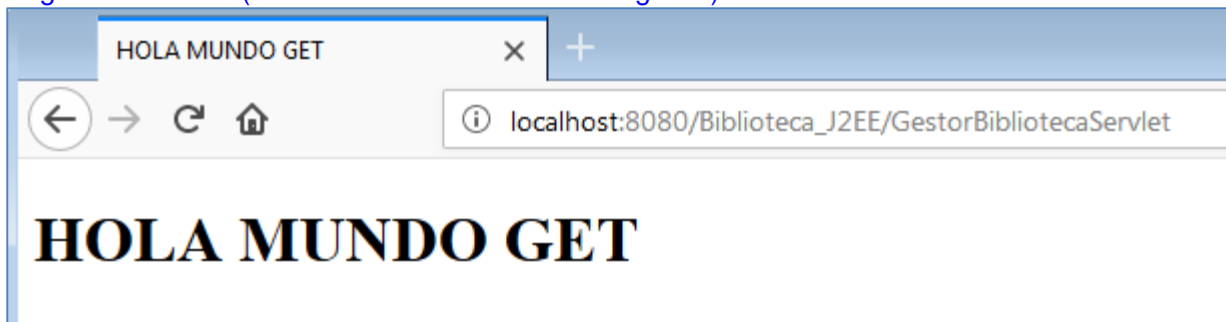
```



Primera llamada:



Segunda llamada (tirando hacia atrás en el navegador):



## EJERCICIO 12

Nos interesa saber quién se conecta a nuestra biblioteca. Para ello, pasaremos el usuario y su contraseña al servlet para que ésta valide si tenemos permisos para acceder.

Añade 2 campos a los formularios de la página web index.html que se llamarán usuario y password. El servlet tiene que contestar con “Bienvenido [usuario]” si se valida el acceso del usuario o “Acceso no permitido” en caso contrario.

```
<html>
  <head>
    <title>MÉTODOS GET-POST</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Método GET</h1>
    <form method="GET" action="GestorBibliotecaServlet">
      Usuario:<input type="text" name="usuario"><br><br>
      Password:<input type="password" name="password"><br><br>
      <input type="submit">
    </form>
    <h1>Método POST</h1>
    <form method="POST" action="GestorBibliotecaServlet">
      Usuario:<input type="text" name="usuario"><br><br>
      Password:<input type="password" name="password"><br><br>
      <input type="submit">
    </form>
  </body>
</html>
```



```

public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String usuario = request.getParameter("usuario");
        String password = request.getParameter("password");
        PrintWriter out = response.getWriter();
        if (usuario.toLowerCase().equals("eduard") &&
            password.toLowerCase().equals("eduard")){

            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>LOGIN</title></head>");
            out.println("<body>");
            if (YaIniciado==false){
                out.println("<h1>Conectado a la BD (llamada GET)</h1>");
                YaIniciado=true;
            }
            out.println("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");
            out.println("</body>");
            out.println("</html>");
        }else{
            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>LOGIN</title></head>");
            out.println("<body>");
            out.println("<h1>ACCESO NO PERMITIDO USUARIO "+usuario+"</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    String usuario = request.getParameter("usuario");
    String password = request.getParameter("password");
    PrintWriter out = response.getWriter();
    if (usuario.toLowerCase().equals("eduard") &&
        password.toLowerCase().equals("eduard")){

        out.println("<!DOCTYPE html>");
        out.println("<html><head><title>LOGIN</title></head>");
        out.println("<body>");
        if (YaIniciado==false){
            out.println("<h1>Conectado a la BD (llamada POST)</h1>");
            YaIniciado=true;
        }
        out.println("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");
        out.println("</body>");
        out.println("</html>");
    }else{
        out.println("<!DOCTYPE html>");
        out.println("<html><head><title>LOGIN</title></head>");
        out.println("<body>");
        out.println("<h1>ACCESO NO PERMITIDO USUARIO "+usuario+"</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
}

```

MÉTODOS GET-POST x +

localhost:8080/Biblioteca\_J2EE/

### Método GET

Usuario:

Password:

### Método POST

Usuario:

Password:

LOGIN x +

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet?usuario=EDUARD&password=EDUARD

**Conectado a la BD (llamada GET)**

**BIENVENIDO USUARIO EDUARD**

MÉTODOS GET-POST x +

localhost:8080/Biblioteca\_J2EE/

### Método GET

Usuario:

Password:

### Método POST

Usuario:

Password:

LOGIN x +

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet?usuario=PEPE&password=PEPE

**ACCESO NO PERMITIDO USUARIO PEPE**

MÉTODOS GET-POST x +

localhost:8080/Biblioteca\_J2EE/

### Método GET

Usuario:

Password:

### Método POST

Usuario:

Password:

LOGIN x +

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet

**Conectado a la BD (llamada POST)**

**BIENVENIDO USUARIO EDUARD**

MÉTODOS GET-POST x +

localhost:8080/Biblioteca\_J2EE/

### Método GET

Usuario:

Password:

### Método POST

Usuario:

Password:

LOGIN x +

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet

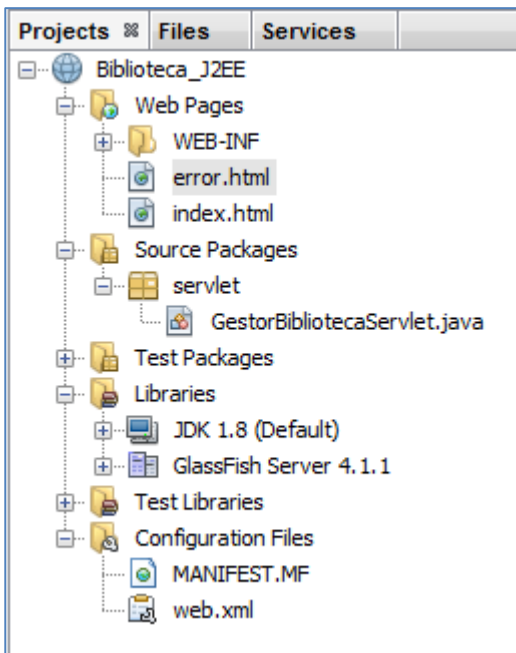
**ACCESO NO PERMITIDO USUARIO PEPE**

## EJERCICIO 13

Ahora crea el fichero error.html, el cual será redireccionado por el servlet cuando la identificación sea incorrecta. Por lo tanto el servlet ya no dibujará directamente el resultado de la identificación

errónea, sino que mostrará un fichero html donde se muestre el error. ¿Se le puede pasar un parámetro a un fichero html?

NOTA: Utiliza la función `response.sendRedirect("")`.



```
<html>
  <head>
    <title>ERROR AUTENTIFICACION</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1>ACCESO NO PERMITIDO USUARIO</h1>
  </body>
</html>
```

El problema es que no podemos pasarle a un fichero HTML un parámetro desde un servlet, o dicho de otra manera, un HTML no puede recoger parámetros HTTP. Si en lugar de un HTML fuera un JSP, entonces sí que podría.

```

public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String Usuario = request.getParameter("Usuario");
        String Password = request.getParameter("Password");

        if (Usuario.toLowerCase().equals("eduard") &&
            Password.toLowerCase().equals("eduard")) {
            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>LOGIN</title></head>");
            out.println("<body>");
            if (YaIniciado == false) {
                out.println("<h1>Conectado a la BD(llamada GET)</h1>");
                YaIniciado = true;
            }
            out.println("<h1>BIENVENIDO USUARIO "+Usuario+"</h1>");
            out.println("</body>");
            out.println("</html>");
        }else{
            response.sendRedirect("error.html");
        }
    }
}

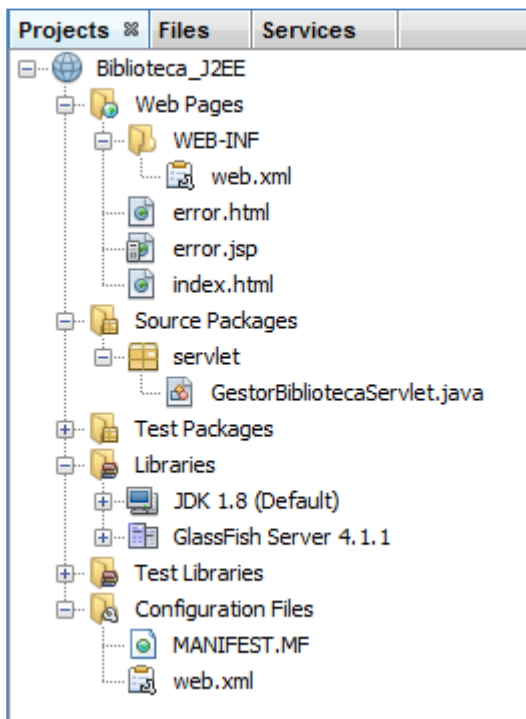
```

The screenshot shows two browser windows. The top window, titled 'MÉTODOS GET-POST', displays a login form at 'localhost:8080/Biblioteca\_J2EE/'. The form has two input fields: 'Usuario: PEPE' and 'Password: ....'. Below the fields is a button labeled 'Enviar consulta'. The bottom window, titled 'ERROR AUTENTIFICACIÓN', shows the 'error.html' page at 'localhost:8080/Biblioteca\_J2EE/error.html'. The page content is 'ACCESO NO PERMITIDO USUARIO'.

## EJERCICIO 14

Crea el fichero error.jsp, que sustituirá a error.html en el redireccionado por parte del servlet

GestorBibliotecaServlet cuando la identificación sea incorrecta. Le pasaremos el parámetro usuario para que lo muestre por pantalla, cosa que con un fichero html no se podía hacer.



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ERROR AUTENTIFICACIÓN</title>
  </head>
  <body>
    <h1>ACCESO NO PERMTIIDO USUARIO <%=request.getParameter("usuario")%></h1>
  </body>
</html>
```

```

public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String usuario = request.getParameter("usuario");
        String password = request.getParameter("password");
        PrintWriter out = response.getWriter();
        if (usuario.toLowerCase().equals("eduard") &&
            password.toLowerCase().equals("eduard")) {

            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>LOGIN</title></head>");
            out.println("<body>");
            if (YaIniciado==false){
                out.println("<h1>Conectado a la BD (llamada GET)</h1>");
                YaIniciado=true;
            }
            out.println("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");
            out.println("</body>");
            out.println("</html>");
        }else{
            response.sendRedirect("error.jsp?usuario="+usuario);
        }
    }
}

```

MÉTODOS GET-POST

localhost:8080/Biblioteca\_J2EE/

## Método GET

Usuario:

Password:

## Método POST

Usuario:

Password:

ERROR AUTENTIFICACIÓN

localhost:8080/Biblioteca\_J2EE/error.jsp?usuario=PEPE

## ACCESO NO PERMITIDO USUARIO PEPE


Una vez estemos conectados, nos interesa mostrar una página con un listado de libros disponibles. Para ello, en la respuesta de bienvenida del servlet `GestorBibliotecaServlet`, añadiremos un formulario con un campo de texto y un botón, que nos permitirá buscar una lista de libros por título. Este formulario debe apuntar a un nuevo servlet que llamaremos “`ConsultaLibrosServlet`”.

```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String usuario = request.getParameter("usuario");
        String password = request.getParameter("password");
        PrintWriter out = response.getWriter();
        if (usuario.toLowerCase().equals("eduard") &&
            password.toLowerCase().equals("eduard")){

            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>LOGIN</title></head>");
            out.println("<body>");
            if (YaIniciado==false){
                out.println("<h1>Conectado a la BD (llamada GET)</h1>");
                YaIniciado=true;
            }
            out.println("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");

            out.println("<form method=GET action=ConsultaLibrosServlet>");
            out.println("Selección de Libro: <input type='text' name='titulo'><br><br>");
            out.println("<input type=submit>");
            out.println("</form>");

            out.println("</body>");
            out.println("</html>");
        }else{
            response.sendRedirect("error.jsp?usuario="+usuario);
        }
    }
}
```



LOGIN

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet?usuario=EDUARD&password=EDUARD

## Conectado a la BD (llamada GET)

## BIENVENIDO USUARIO EDUARD

Selección de Libro:

Enviar consulta



```

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    String usuario = request.getParameter("usuario");
    String password = request.getParameter("password");
    PrintWriter out = response.getWriter();
    if (usuario.toLowerCase().equals("eduard") &&
        password.toLowerCase().equals("eduard")) {

        out.println("<!DOCTYPE html>");
        out.println("<html><head><title>LOGIN</title></head>");
        out.println("<body>");
        if (YaIniciado==false){
            out.println("<h1>Conectado a la BD (llamada POST)</h1>");
            YaIniciado=true;
        }
        out.println("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");

        out.println("<form method=POST action=ConsultaLibrosServlet>");
        out.println("Selección de Libro: <input type='text' name='titulo'><br><br>");
        out.println("<input type=submit>");
        out.println("</form>");

        out.println("</body>");
        out.println("</html>");
    }else{
        response.sendRedirect("error.jsp?usuario="+usuario);
    }
}

```

LOGIN

localhost:8080/Biblioteca\_J2EE/GestorBibliotecaServlet

## Conectado a la BD (llamada POST)

## BIENVENIDO USUARIO EDUARD

Selección de Libro:

Enviar consulta

Modifica el fichero GestorBibliotecaServlet para que redireccione y delegue su funcionalidad en caso de bienvenida hacia el archivo bienvenido.jsp. Este archivo debe garantizar exactamente la misma funcionalidad del ejercicio anterior. Utilizar sendRedirect con paso de parámetros.

```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String usuario = request.getParameter("usuario");
        String password = request.getParameter("password");
        if (usuario.toLowerCase().equals("eduard") &&
            password.toLowerCase().equals("eduard")) {
            boolean temp=YaIniciado;
            if (YaIniciado==false) YaIniciado=true;
            response.sendRedirect("consulta.jsp?usuario="+usuario+"&iniciado="+temp);
        }else{
            response.sendRedirect("error.jsp?usuario="+usuario);
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String usuario = request.getParameter("usuario");
        String password = request.getParameter("password");
        if (usuario.toLowerCase().equals("eduard") &&
            password.toLowerCase().equals("eduard")) {
            boolean temp=YaIniciado;
            if (YaIniciado==false) YaIniciado=true;
            response.sendRedirect("bienvenido.jsp?usuario="+usuario+"&iniciado="+temp);
        }else{
            response.sendRedirect("error.jsp?usuario="+usuario);
        }
    }
}
```

BIENVENIDA

localhost:8080/Biblioteca\_J2EE/bienvenido.jsp?usuario=EDUARD&iniciado=false

## Conectado a la BD (llamada GET)

## BIENVENIDO USUARIO EDUARD

Selección de Libro:

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>BIENVENIDA</title>
    </head>
    <body>
        <%String usuario=request.getParameter("usuario");%>
        <%String iniciado=request.getParameter("iniciado");%>

        <%if (iniciado.equals("false")){%>
            <h1>Conectado a la BD (llamada GET)</h1>
        <%}%>

        <h1>BIENVENIDO USUARIO <%=usuario%></h1>
        <form method=GET action=ConsultaLibrosServlet>
            Selección de Libro: <input type='text' name='titulo'><br><br>
            <input type=submit>
        </form>

    </body>
</html>

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>BIENVENIDA</title>
    </head>
    <body>
        <%
            String usuario = request.getParameter("usuario");
            String bd = request.getParameter("bd");
            StringBuilder pepe = new StringBuilder();
            if (bd.equals("false"))
                pepe.append("<h1>Conectado a la BD(llamada GET)</h1>");
            pepe.append("<h1>BIENVENIDO USUARIO "+usuario+"</h1>");
            pepe.append("<form method=GET action=ConsultaLibrosServlet>");
            pepe.append("Selección de Libro: <input type='text' name='titulo'><BR><BR>");
            pepe.append("<input type=submit>");
            pepe.append("</form>");
            out.println(pepe.toString());
        %>
    </body>
</html>

```

## EJERCICIO 17

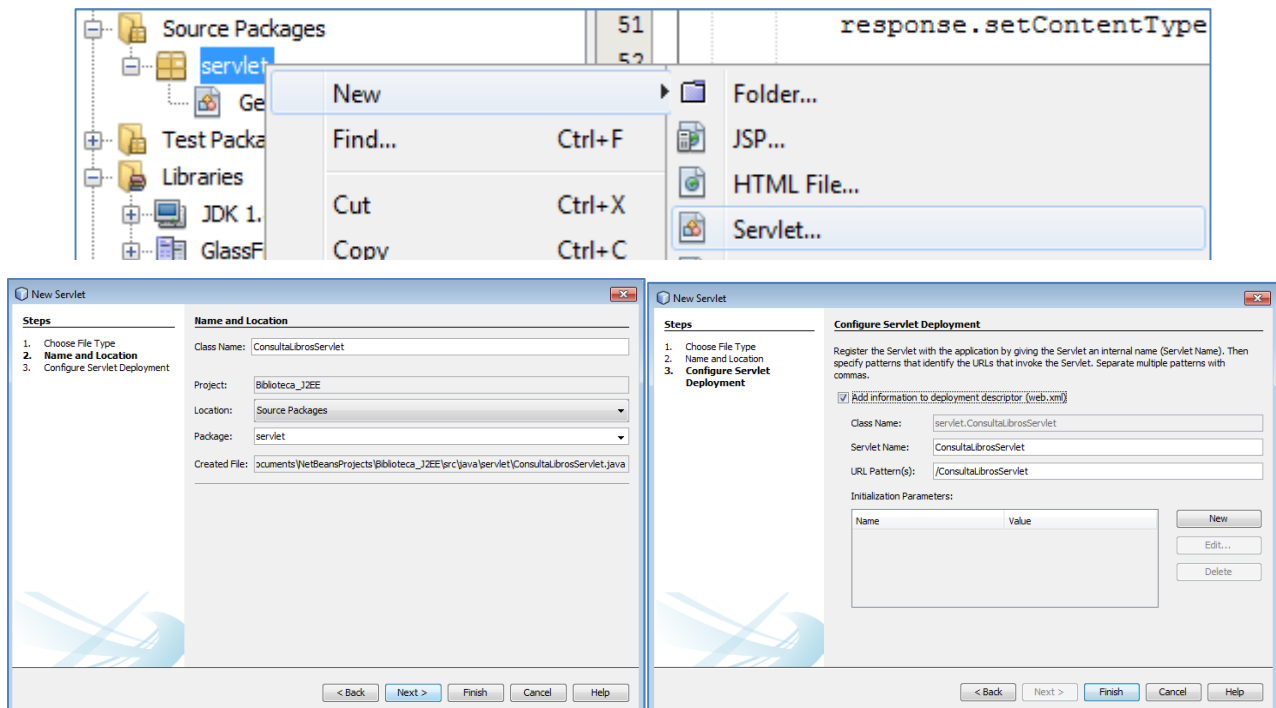
Crea el servlet ConsultaLibrosServlet. Estará preparado para recibir el título solicitado por el usuario, y buscará en el fichero de texto “libros.txt” (que se puede descargar del curso) todas las entradas que contengan dicha cadena de texto. Como respuesta, enviará al navegador un listado de libros separados por saltos de línea (etiqueta <br> de HTML). Considera los siguientes puntos:

- Utiliza el modelo processRequest para el servlet ConsultaLibrosServlet, para unificar los métodos doGet y doPost.
- Sitúa el fichero libros.txt en el directorio build/web de tu proyecto. Utiliza la api `getServletContext().getRealPath("/")` para obtener el path de este fichero

El contenido del fichero libros.txt es el siguiente:

```
libros.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
00000001; HARRY POTTER Y EL PRISIONERO DE AZKABÁN; J.K. ROWLING; SALAMANDRA; 26/9/2006 0:00:00; INFANTIL;
00000002; EL GRAN LABERINTO; FERNANDO SABATER PEREZ; ARIEL; 26/9/2006 0:00:00; FICCIÓN;
00000003; ROMEO Y JULIETA; WILLIAM SHAKESPEARE; SALAMANDRA; 26/9/2006 0:00:00; ROMANTICA;
00000004; LA CARTA ESFERICA; ARTURO PEREZ LOPEZ; SALAMANDRA; 29/9/2006 0:00:00; FICCIÓN;
00000005; CODIGO DA VINCI; DAN BROWN; ARIEL; 29/9/2006 0:00:00; FICCIÓN;
00000006; MUCHO RUIDO Y POCAS NUECES; WILLIAM SHAKESPEARE; SALAMANDRA; 29/9/2006 0:00:00; ROMANTICA;
00000007; PROTOCOLO; JOSE LOPEZ MURILLO; SALAMANDRA; 6/9/2006 0:00:00; SOCIAL;
00000008; LINUX; FERNANDO SABATER PEREZ; ARIEL; 6/9/2006 0:00:00; INFORMÁTICA;
00000009; EL TUMULTO; H.P. LOVECRAFT; DEBATE; 6/9/2006 0:00:00; CIENCIA;
00000010; PERSONAJES MITICOS; RICHARD HOLLIGHAM; DEBATE; 7/9/2006 0:00:00; ENTRETENIMIENTO;
00000011; EL TIEMPO; J.K. ROWLING; SALAMANDRA; 7/9/2006 0:00:00; CIENCIA;
00000012; DIETAS MEDITERRANEAS; ARTURO PEREZ LOPEZ; ARIEL; 16/9/2006 0:00:00; ASTRONOMIA;
00000013; ANGELES Y DEMONIOS; DAN BROWN; ARIEL; 17/9/2006 0:00:00; FICCIÓN;
00000014; FORTALEZA DIGITAL; DAN BROWN; ARIEL; 6/10/2006 0:00:00; FICCIÓN;
00000015; CAPITAN ALATRISTE; ARTURO PEREZ LOPEZ; ALFAGUARA; 9/10/2006 0:00:00; FICCIÓN;
00000016; PIEL DE TAMBOR; ARTURO PEREZ LOPEZ; ALFAGUARA; 16/10/2006 0:00:00; FICCIÓN;
00000017; TIEMPOS DE COLERA; GABRIEL GARCIA GARCIA; OVEJA NEGRA; 1/9/2006 0:00:00; OCIO;
00000018; NOTICIA DE UN SEQUESTRO; GABRIEL GARCIA GARCIA; ALFAGUARA; 7/12/2006 0:00:00; FICCIÓN;
```

Primero de todo creamos el servlet ConsultaLibrosServlet:



```

public class ConsultaLibrosServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String titulo = request.getParameter("titulo");
        String path = getServletContext().getRealPath("/");
        File f = new File(path + "libros.txt");
        BufferedReader entrada = new BufferedReader(new FileReader(f));

        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head><title>Servlet ConsultaLibrosServlet</title></head>");
        out.println("<body>");
        while (entrada.ready()){
            String linea=entrada.readLine();
            boolean presencia = linea.contains(titulo);
            if (presencia) out.println("<h3>"+linea+"</h3>");
        }
        out.println("</body>");
        out.println("</html>");
    }
}

```

LOGIN

←

→

↺

🏠

localhost:8080/Biblioteca\_J2EE/consulta.jsp?usuario=EDUARD&iniciado=false

## Conectado a la BD (llamada GET)

## BIENVENIDO USUARIO EDUARD

Selección de Libro:

LOGIN

Servlet ConsultaLibrosServlet

←

→

↺

🏠

localhost:8080/Biblioteca\_J2EE/ConsultaLibrosServlet?titulo=AL

...

🔒

☆

🔍 Buscar

00000001;HARRY POTTER Y EL PRISIONERO DE AZKABÁN;J.K ROWINS;SALAMANDRA;26/9/2006 0:00:00;INFANTIL;

00000003;ROMEO Y JULIETA;WILLIAM SHAKESPEARE;SALAMANDRA;26/9/2006 0:00:00;ROMANTICA;

00000004;LA CARTA ESFERICA;ARTURO PEREZ LOPEZ;SALAMANDRA;29/9/2006 0:00:00;FICCION;

00000006;MUCHO RUIDO Y POCAS NUECES;WILLIAM SHAKESPEARE;SALAMANDRA;29/9/2006 0:00:00;ROMANTICA;

00000007;PROTOCOLO;JOSE LOPEZ MURILLO;SALAMANDRA;6/9/2006 0:00:00;SOCIAL;

00000011;EL TIEMPO;J.K ROWINS;SALAMANDRA;7/9/2006 0:00:00;CIENCIA;

00000014;FORTALEZA DIGITAL;DAN BROWN;ARIEL;6/10/2006 0:00:00;FICCION;

00000015;CAPITAN ALATRISTE;ARTURO PEREZ LOPEZ;ALFAGUARA;9/10/2006 0:00:00;FICCION;

00000016;PIEL DE TAMBOR;ARTURO PEREZ LOPEZ;ALFAGUARA;16/10/2006 0:00:00;FICCION;

00000018;NOTICIA DE UN SECUESTRO;GABRIEL GARCIA GARCIA;ALFAGUARA;7/12/2006 0:00:00;FICCION;