MÓDULO 4

EJERCICIOS RESUELTOS COLABORACION ENTRE SERVLETS

TEORIA 1

Explica brevemente en qué consiste la arquitectura MVC.

TEORIA 2

Explica el funcionamiento, las ventajas y desventajas del uso de la interfaz SingleThreadModel.

Por cada Servlet el Servidor creará un Hebra. Según las implementaciones, un contenedor WEB puede gestionar una única instancia para cada Servlet, o bien un conjunto de ellas.

Este principio de funcionamiento puede generar problemas si el método de servicio trabaja con variables de instancia del Servlet, porque cada hebra puede modificar el valor de dichas variables, con independencia de la lógica de procesamiento de las otras hebras.

Es importante garantizar un funcionamiento aislado de cada hebra. Para ellos es necesario que la clase de Servlet implemente la iinterfaz java.servlet.SingleThreadModel.

Si el contenedor WEB supone una instancia por Servlet este generará una Cola para los distintos Servlets.

Si por lo contrario se ejecutan todos los Servlets al unisono esto generará un consumo de memoria enorme en el contenedor WEB.

La Ventaja es que se garantiza de que Cada instancia de un Servlet, pese a que sea el mismo se ejecute de forma independiente sin que cada uno interfiera en otro, pese a que sea el mismo Servlet.

La Desventaja, son básicamente dos:

- Consumo de recursos en el Servidor WEB, consumo de memoria.
- Acceso Recurrente a Datos, o inconsistencia en la creación de los mismos. Por ejemplo que se cree antes los Datos de un Usuario, pero el ID del Usuario no se haya creado con anterioridad.

EJERCICIO 1

Un punto débil de nuestra aplicación es que el nombre del fichero que contiene la información de libros está escrito dentro de nuestro código Java. En nuestro caso hemos utilizado la API getServletContext().getRealPath("/"), que nos proporciona el path real donde se encuentra nuestra aplicación y libros.txt. En el caso de que se cambiara la localización de este fichero, se tendría que recompilar el código.

Para evitar este problema, se puede definir una variable path en web.xml que leeremos en el init() de nuestro servlet ConsultaLibrosServlet a través de un objeto ServletConfig. De esta forma ante cualquier cambio de path, modificando el fichero web.xml, evitaría tener que volver a realizar una nueva recompilación del código.

Crea los cambios necesarios en web.xml y en el Servlet ConsultaLibrosServlet para que esto funcione tal y como se ha explicado.

```
<servlet>
    <servlet-name>ConsultaLibrosServlet</servlet-name>
    <servlet-class>servlet.ConsultaLibrosServlet</servlet-class>
    <init-param>
        <param-name>path</param-name>
        <param-value>c:\\libros.txt</param-value>
    </init-param>
</servlet>
```

```
public class ConsultaLibrosServlet extends HttpServlet {

    public String ruta_libros;

    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        ruta_libros = getInitParameter("path");
        System.out.println(ruta_libros);
    }

    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");
        File f = new File(ruta_libros);
        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>");
    }
```

EJERCICIO 2

En el problema anterior definíamos un parámetro de configuración que sólo era visible para el servlet ConsultaLibrosServlet. En el caso de que quisiéramos que este parámetro fuera visible por todos los servlets de la aplicación, se tendría que definir como un elemento <context-param> en web.xml en lugar de <init-param> y utiliza un objeto ServletContext en lugar de ServletConfig para recuperarlo.

Define este parámetro como de contexto e intenta leerlo desde GestorBibliotecaServlet.

```
    <context-param>
        <param-name>pathcontext</param-name>
        <param-value>c:\\libros.txt</param-value>
    </context-param>

</web-app>
```

```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;

    public String ruta_libros;
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        ServletContext context = getServletContext();
        ruta_libros = context.getInitParameter("pathcontext");
        System.out.println(ruta_libros);
    }
```

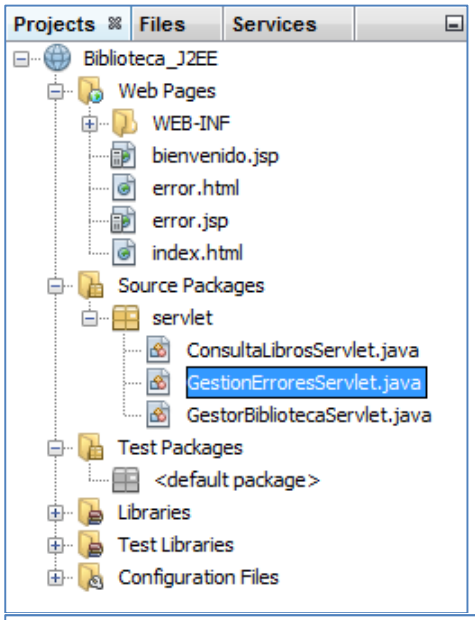
```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;

    public String ruta_libros;
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        ServletCon
        ruta_libros = context.getInitParameter("pathcontext");
        System.out.println(ruta_libros);
    }
```

ruta_libros = (java.lang.String) "c:\\libros.txt"

EJERCICIO 3

Vamos a crear un tercer Servlet, que llamaremos “GestionErroresServlet”. Este Servlet gestionará los mensajes de error de todos los servlets de la aplicación. Cuando en otro servlet se genere un error, se lo notificará al servlet GestionErroresServlet y será éste quien genere la página de error que será enviada al cliente. La página de error mostrada debe de tener la siguiente interficie:




```java
@WebServlet(name = "GestionErroresServlet", urlPatterns = {"/GestionErroresServlet"})
public class GestionErroresServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head><title>Servlet GestionErroresServlet</title></head>");
        out.println("<body>");
        out.println("<h1>SERVLET GESTION DE ERRORES</h1>");
        out.println("<h2>Usuario: </h2>");
        out.println("<h2>Password: </h2>");
        out.println("<h2>Mensaje: </h2>");
        out.println("</body>");
        out.println("</html>");
    }
```

EJERCICIO 4

Modifica el servlet GestorBibliotecaServlet, de forma que cuando el par usuario-contraseña no sea correcto redirija la salida hacia el servlet GestionErroresServlet, pasándole 3 parámetros:

- usuario
- password
- un mensaje de error, indicando la situación

Nota: Toda la comunicación entre los Servlets se realiza con un objeto RequestDispatcher, el cual se obtiene al utilizar el método ServletRequest.getRequestDispatcher. Después utiliza en este último el método “forward”.

```
public class GestorBibliotecaServlet extends HttpServlet {
    private boolean YaIniciado=false;

    public String ruta_libros;
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        ServletContext context = getServletContext();
        ruta_libros = context.getInitParameter("pathcontext");
        System.out.println(ruta_libros);
    }

    @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("bienvenido.jsp?usuario="+usuario+"&iniciado="+temp);
        }else{
            //response.sendRedirect("error.jsp?usuario="+usuario);
            String msg = "Usuario y/o password son erróneos";
            request.setAttribute("usuario", usuario);
            request.setAttribute("password", password);
            request.setAttribute("msg_error", msg);
            request.getRequestDispatcher("/GestionErroresServlet").forward(request, response);
        }
    }
```

EJERCICIO 5

Modifica el servlet GestionErroresServlet para que capture y muestre los atributos que anteriormente le ha pasado el servlet GestorBibliotecaServlet.

```java
@WebServlet(name = "GestionErroresServlet", urlPatterns = {"/GestionErroresServlet"})
public class GestionErroresServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head><title>Servlet GestionErroresServlet</title></head>");
        out.println("<body>");
        out.println("<h1>SERVLET GESTION DE ERRORES</h1>");
        out.println("<h2>Usuario: "+ request.getAttribute("usuario")+"</h2>");
        out.println("<h2>Password: "+request.getAttribute("password")+"</h2>");
        out.println("<h2>Mensaje: "+request.getAttribute("msg_error")+"</h2>");
        out.println("</body>");
        out.println("</html>");
    }
```

EJERCICIO 6

Modifica GestionErroresServlet para que envíe el mensaje en modo texto (text/plain en setContentType) y explica qué sucede.

EJERCICIO 7

Modifica el servlet GestionErroresServlet para que redirija la salida hacia el fichero msg_error.jsp, en lugar de mostrar el mensaje por pantalla directamente.

Ten en cuenta que msg_error.jsp recibirá los 3 atributos que le envía GestionErroresServlet, que a su vez los recibió de GestorBibliotecaServlet.

```jsp
<%@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</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

```java
@WebServlet(name = "GestionErroresServlet", urlPatterns = {"/GestionErroresServlet"})
public class GestionErroresServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        //response.setContentType("text/plain;charset=UTF-8");
        //response.setContentType("text/html;charset=UTF-8");
        /*PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head><title>Servlet GestionErroresServlet</title></head>");
        out.println("<body>");
        out.println("<h1>SERVLET GESTION DE ERRORES</h1>");
        out.println("<h2>Usuario: "+ request.getAttribute("usuario")+"</h2>");
        out.println("<h2>Password: "+request.getAttribute("password")+"</h2>");
        out.println("<h2>Mensaje: "+request.getAttribute("msg_error")+"</h2>");
        out.println("</body>");
        out.println("</html>");  */
        request.getRequestDispatcher("msg_error.jsp").forward(request, response);
    }
```

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP DE ERROR</title>
    </head>
    <body>
        <h1>JSP DE ERROR</h1>
        <h2>Usuario: <%=request.getAttribute("usuario")%> </h2>
        <h2>Password: <%=request.getAttribute("password")%></h2>
        <h2>Mensaje: <%=request.getAttribute("msg_error")%></h2>
    </body>
</html>
```

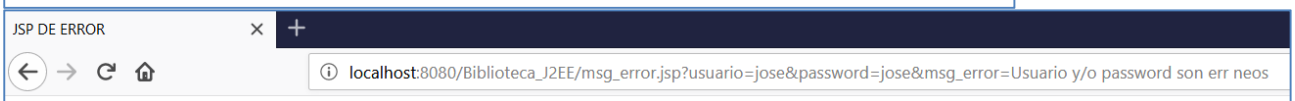
EJERCICIO 8

Investiga de qué forma el servlet GestionErroresServlet puede redirigir la salida hacia el fichero msg_error.jsp, con un retardo de 5 segundos, pasándole los parámetros ya definidos.

NOTA: Consulta los métodos getHeader/setHeader de ServletResponse.

```java
@WebServlet(name = "GestionErroresServlet", urlPatterns = {"/GestionErroresServlet"})
public class GestionErroresServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        //response.setContentType("text/plain;charset=UTF-8");
        //response.setContentType("text/html;charset=UTF-8");
        /*PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head><title>Servlet GestionErroresServlet</title></head>");
        out.println("<body>");
        out.println("<h1>SERVLET GESTION DE ERRORES</h1>");
        out.println("<h2>Usuario: "+ request.getAttribute("usuario")+"</h2>");
        out.println("<h2>Password: "+request.getAttribute("password")+"</h2>");
        out.println("<h2>Mensaje: "+request.getAttribute("msg_error")+"</h2>");
        out.println("</body>");
        out.println("</html>");  */
        //response.setHeader("Refresh", "10; URL=msg_error.jsp");
        String url="5;URL=msg_error.jsp?usuario="+request.getAttribute("usuario");
        url += "&password="+request.getAttribute("password");
        url += "&msg_error="+request.getAttribute("msg_error");
        response.setHeader("Refresh", url);
        //request.getRequestDispatcher("msg_error.jsp").forward(request, response);
    }
```

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP DE ERROR</title>
    </head>
    <body>
        <h1>JSP DE ERROR</h1>
        <h2>Usuario: <%=request.getParameter("usuario")%> </h2>
        <h2>Password: <%=request.getParameter("password")%></h2>
        <h2>Mensaje: <%=request.getParameter("msg_error")%></h2>
        <!--<h2>Usuario: <%=request.getAttribute("usuario")%> </h2>
        <h2>Password: <%=request.getAttribute("password")%></h2>
        <h2>Mensaje: <%=request.getAttribute("msg_error")%></h2>-->
    </body>
</html>
```



JSP DE ERROR

Usuario: jose

Password: jose

Mensaje: Usuario y/o password son err neos