

## Índice

|   |   |
|---|---|
| 1. Clases relacionadas con el carro de la compra        | 1 |
| 2. Creación de un oyente de sesión                      | 2 |
| 3. Creación de un servlet                               | 2 |
| 4. Creación de una página jsp con el contenido estático | 4 |

---

### 1. Clases relacionadas con el carro de la compra

En la perspectiva JavaEE creamos un nuevo Dynamic Web Project que llamaremos tienda.

Vamos a crear en el directorio `src` una interface que representa elementos que se pueden añadir a un carro de la compra:

```
package edu.uv.aw.tallersesion;

public interface Comprable{
    public double getPrecio();
    public String getDescripcion();
    public int getID();
}
```

Ahora realizamos una clase que almacena objetos de este tipo y que representa el carro de la compra:

```
package edu.uv.aw.tallersesion;

import java.util.Vector;

public class CarroCompra<T extends Comprable>{
    private Vector<T> carro;

    public CarroCompra(){
        carro = new Vector<T>();
    }

    public void add(T el){
        carro.add(el);
    }

    public Vector<T> getCarro(){
        return carro;
    }

    public double getTotalCarro(){
        double valor = 0;
        for (T t:carro)
            valor += t.getPrecio();
        return valor;
    }
}
```

Vamos a suponer que los objetos que se van a vender en esta tienda son CDs:

```
package edu.uv.aw.tallersesion;

public class CD implements Comprable {
    private double precio;
    private String descripcion;
    private int id;
```

```
public CD(double p, String d, int id) {
    precio = p;
    descripcion = d;
    this.id = id;
}

@Override
public double getPrecio() {
    return precio;
}

@Override
public String getDescripcion() {
    return descripcion;
}

@Override
public int getID() {
    return id;
}
}
```

## 2. Creación de un oyente de sesión

Ahora vamos a crear una clase que escuchará los eventos asociados a la sesión: creación de una nueva sesión y finalización de la sesión. Para ello:

- Pulsamos con el botón derecho del ratón sobre el paquete.
- En el menú seleccionamos **New** y a continuación **Listener**.
- En la ventana que aparece proporcionamos el nombre de la clase: **OyenteSesion** y pulsamos sobre **Next**.
- En la ventana seleccionamos en el grupo HTTP **session events** la opción **Lifecycle** y pulsamos sobre **Next**.
- En la siguiente ventana pulsamos sobre **Finish**.

A continuación editamos el fichero que genera y ponemos el siguiente código:

```
package edu.uv.aw.tallersesion;

import javax.servlet.annotation.WebListener;
import javax.servlet.http.HttpSessionEvent;
import javax.servlet.http.HttpSessionListener;

@WebListener
public class OyenteSesion implements HttpSessionListener {

    public ListenerSesion() {
    }

    public void sessionCreated(HttpSessionEvent arg0) {
        CarroCompra<CD> c = new CarroCompra<CD>();
        arg0.getSession().setAttribute("comprado", c);
        System.out.println("Se ha creado una sesion");
    }

    public void sessionDestroyed(HttpSessionEvent arg0) {
        arg0.getSession().removeAttribute("comprado");
    }
}
```

### 3. Creación de un servlet

A continuación creamos un servlet en el proyecto.

```
package edu.uv.aw.tallersesion;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Vector;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet
 */
@WebServlet("/compra")
public class ServletCarroCompra extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private Vector<CD> cds;

    /**
     * En el constructor creamos los objetos CD de la tienda
     */
    public CarroCompra() {
        super();
        cds = new Vector<CD>();
        cds.add(new CD(20,"CD1",0));
        cds.add(new CD(23,"CD1",1));
        cds.add(new CD(10,"CD1",2));
        cds.add(new CD(15,"CD1",3));
        cds.add(new CD(16,"CD1",4));
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        // Obtenemos el atributo
        CarroCompra<CD> c = (CarroCompra<CD>)request.getSession().getAttribute("comprado");

        PrintWriter pw = response.getWriter();
        pw.println("<html>");
        pw.println("<body>");

        String accion = (String) request.getParameter("accion");

        // Si existe en parámetro accion
        if (accion!=null){
            if (accion.equals("comprar")){
                // Si accion=comprar leemos un segundo parametro id con el id del cd que compra
                int id = Integer.parseInt((String) request.getParameter("id"));
                c.add(cds.get(id));
                System.out.println("Comprado: " + id);
            }
        }

        // Pedimos que otro componente incluya su salida
        getServletContext().getRequestDispatcher("/items.jsp").include(request, response);

        // Mostramos el total de la compra:
        pw.println("El total de la compra es: ");
        pw.println(c.getTotalCarro());
        pw.println("<br/>");

        pw.println("</body>");
        pw.println("</html>");
    }

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

```
}
```

## 4. Creación de una página jsp con el contenido estático

Vemos que en el servlet se ha indicado que otro componente incluya el contenido estático.

Para ello:

- Pulsamos con el botón derecho sobre el proyecto.
- En el menú pulsamos sobre **New** y a continuación seleccionamos **JSP File**.
- En la ventana que aparece se especifica el nombre del fichero y se pulsa sobre **Finish**.

El fichero se crea por defecto en el directorio **WebContent**. Lo editamos y ponemos el siguiente código:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>

<h1>Pulsa en un producto para comprarlo</h1>
<a href="compra?accion=comprar&id=0">CD1</a><br/>
<a href="compra?accion=comprar&id=1">CD2</a><br/>
<a href="compra?accion=comprar&id=2">CD3</a><br/>
<a href="compra?accion=comprar&id=3">CD4</a><br/>
<a href="compra?accion=comprar&id=4">CD5</a><br/>
```

Como vemos, estamos mostrando una serie de enlaces al servlet y en cada enlace pasamos un argumento **id** diferente.

Ahora podemos instalar la aplicación en el servidor y comprobar si funciona correctamente.