



## SISTEMAS DISTRIBUIDOS

### INFORME DE AEC3: SERVLETS Y HTML EN SERVICIOS WEBS

Antonio Luis Ojeda Soto

#### 1.Explicación del proceso de la práctica.

Después de haber leído y estudiado la unidad 9, haber probado los códigos y entenderlos, haber leído el enunciado de la práctica y el vídeo de explicación comienzo con la misma.

Para empezar, creo una página web simple para la compra de componentes para ordenadores, donde el cliente deberá ir añadiendo a la cesta los componentes y la cantidad que quiere comprar de cada uno de ellos.

El mecanismo elegido para almacenar las peticiones serán **variables de servlet**. Con estas variables globales y el motor de servlets, el cliente podrá hacer todas las peticiones que quiera, las cuales se irán almacenando y podrán ser visualizadas cada vez que se haga una, persistiendo hasta que se cierre la página, cosa que con los scripts de CGI no hubiese sido posible.

El método elegido será **doPost**, para que cuando se hagan las peticiones, no se visualice en la url del navegador los componentes y cantidades, siendo de esta forma el pedido más seguro.

Me hubiese gustado hacer más en la práctica, como probar el **objeto sesión** o las **cookies**, pero por falta de tiempo con trabajos de otras asignaturas no me es posible.

## 2.Documentación del código implementado.

En el fichero **miPágina.html** es donde va a ir la página web principal. Principalmente estará compuesta de dos cajas de texto, donde el usuario introducirá el nombre del componente y en la otra caja el número de unidades que pide. Una vez hecha la petición, la página web también incorpora un botón **submit** llamado “Añadir a cesta” el cual redireccionará las peticiones a una página dinámica donde se irá incorporando la información de los pedidos actualizándose en cada petición.

El servlet estará compuesto por la clase **Variables**, que llevará incorporada dos variables: **contador** de tipo entero, y **nombre** de tipo String, las cuales recogerán la información introducida por el cliente en la página web, la procesarán, y actualizarán los datos en la página web dinámica. Además de esto, también incorpora como estructura de datos una **HashMap<String, Integer>** llamada **pares**, para relacionar a cada nombre de componente una cantidad peticionada con las variables anteriormente mencionadas. Para manejar las variables y la estructura de datos, dispondremos de dos métodos: **doPost()**, el cual hay que sobrescribir de obligado cumplimiento por ser un servlet, y que simplemente recogerá la petición y la redireccionará al método **hacerPeticiones()**. Este método recibe los datos mandados por **doPost()**, los introduce en las variables de servlet, casteando la información de la caja de texto cantidad a entero. En el primer for se irá comprobando si el componente había sido introducido previamente, si es así se sumará la cantidad que tenía previamente con la nueva. Después de esto, se introducirá la nueva información actualizada en la **HashMap pares**. En el segundo for se irá imprimiendo toda la información en la página dinámica para que el cliente vaya viendo, de forma actualizada, todas las peticiones que va añadiendo a su cesta de compras.