

Ejercicios de manejo de cookies y sesiones

Índice

1 Personalizar un sitio web (1 punto).....	2
2 Carro de la compra (1 punto).....	2
3 Mejoras para el carro de la compra (1 punto).....	3

1. Personalizar un sitio web (1 punto)

Una de las utilidades que se le dan a las cookies es la de personalizar sitios Web. La aplicación `cw-sesiones` contiene un formulario `form_pers.html` que le indica al usuario que introduzca su nombre, y elija un color de fondo. Dicho formulario llama después al servlet `es.ua.jtech.cw.ejercicios.PersonalizaServlet`. El formulario también tiene un enlace *Ir a página principal*, que internamente llama al servlet `es.ua.jtech.cw.ejercicios.PrincipalServlet`. Se pide:

- Que en el servlet `es.ua.jtech.cw.ejercicios.PersonalizaServlet` se guarde en dos cookies el nombre del usuario y el valor del color seleccionado. Después, redirigirá a `form_pers.html` de nuevo.
- Que el servlet `es.ua.jtech.cw.ejercicios.PrincipalServlet` (llamado desde el formulario anterior) tome las cookies que le envíe el usuario, y genere una página de bienvenida con el color de fondo que haya en la cookie con el color, y con un mensaje de saludo al nombre que haya en la cookie con el nombre. Para establecer el color de fondo, en el *body* podemos tener un atributo *bgcolor*, y ahí le colocaremos el valor que tenga la cookie.

```
<body bgcolor="red">
...
```

2. Carro de la compra (1 punto)

La aplicación `cw-sesiones` contiene también una página `form_carro.html` que tiene una lista de artículos para comprar. Para comprarlos, basta con pulsar el botón de "Añadir al carro" correspondiente. Dicho formulario llama al servlet `es.ua.jtech.cw.ejercicios.CarroServlet`. Se pide que dicho servlet almacene como objetos de sesión los objetos que se vayan comprando, y genere una página dinámica con:

- Cada uno de los objetos que se llevan comprados hasta ahora en la sesión.
- Precio total de la compra
- Un enlace al formulario `form_carro.html` para seguir comprando.

Nota

Para almacenar los objetos podemos utilizar cualquier estructura de datos que queramos (*ArrayList*, etc), y la guardaremos como un atributo de la sesión (es decir, dicha estructura ENTERA la guardaremos como UN UNICO atributo de sesión). Guardaremos, para cada artículo, su nombre y su precio unitario, para luego mostrar estos dos datos en la página que se genere. Para tomar el nombre del artículo y el precio, notar que cada artículo tiene asociado en la página un formulario con dos campos ocultos *articulo* y *precio*, con estos elementos.

Crear una clase interna en el propio *CarroServlet* (llamada *ObjetoCarro*, por ejemplo), que tenga como campos los valores que hay que guardar de cada artículo:

```
class ObjetoCarro
{
    String articulo;
    int precio;

    public ObjetoCarro(String articulo, int precio)
    {
        this.articulo = articulo;
        this.precio = precio;
    }

    public String getArticulo()
    {
        return articulo;
    }

    public int getPrecio()
    {
        return precio;
    }
}
```

Después, en el servlet *CarroServlet*, cada vez que el usuario compre un artículo, creáis un objeto de este tipo con los valores del artículo que haya enviado. Y lo añadís a la sesión, almacenando todos los objetos de tipo *ObjetoCarro* en una lista.

3. Mejoras para el carro de la compra (1 punto)

Sobre el ejercicio anterior:

- a) Añadid un enlace u opción para invalidar la sesión al llamar a `es.ua.jtech.cw.ejercicios.CarroServlet`. Comprobad que al pincharlo, y luego comprar un artículo, el carro aparecerá sólo con ese artículo.
- b) Si quisiéramos aplicar reescritura de URLs en el ejercicio anterior para prevenir que las cookies estén deshabilitadas, ¿qué cambios tendríamos que hacer? Dejadlos reflejados en el código.

