

# ÍNDICE

## COOKIES

1. Características de las cookies .....	3
---	---



### 1. Características de las cookies

Las cookies son pequeños trozos de información textual que la aplicación envía al navegador y que el navegador devuelve sin modificar cuando visita más tarde el mismo site o dominio. Dejando que el servidor lea la información enviada previamente al cliente, el site puede proporcionar a los visitantes un número de ventajas:

**Identificar a un usuario durante una sesión de comercio electrónico.** Muchas tiendas on-line usan una "carta de compra" a la que se añaden los artículos que selecciona el usuario, luego continúa comprando. Como la conexión HTTP se cierra después de enviar cada página, cuando el usuario selecciona un nuevo ítem para su carta, ¿cómo sabe la tienda que él es el mismo usuario que añadió el ítem anterior a su carta? Las cookies son una buena forma de conseguir esto.

**Evitar el nombre de usuario y la password.** Cuando un usuario se registra, se envía una cookie con un único ID de usuario. Cuando el cliente se vuelve a conectar más tarde, se devuelve el ID de usuario, el servidor lo busca, determina que pertenece a un usuario registrado, y no requiere explícitamente el nombre de usuario y la password.

**Personalizar un Site.** Muchos "portales" nos permiten personalizar el aspecto de la página principal. Usan cookies para recordar lo que queremos, para que obtengamos el mismo resultado inicial la próxima vez.

**Publicidad enfocada.** Si hacemos una búsqueda sobre "Java Servlets", un motor de búsqueda cobra más por un anuncio de un entorno de desarrollo de servlets que por el de una agencia de viajes on-line. El problema es que tienen que enseñarnos un anuncio aleatorio la primera vez que llegamos y no hemos realizado la búsqueda, así como cuando buscamos algo que no corresponde con las categorías de anuncios.

Añadir conveniencias al usuario y añadir valor al propietario del site es el propósito que hay detrás de las cookies. Las cookies nunca son interpretadas o ejecutadas de ninguna forma, y no pueden usarse para insertar virus o atacar nuestro sistema. Además, como los navegadores sólo aceptan 20 cookies por site y 300 cookies en total, cada cookie está limitada a 4Kb. Las cookies no se pueden usar para llenar el disco duro o lanzar otros ataques de denegación de servicio.

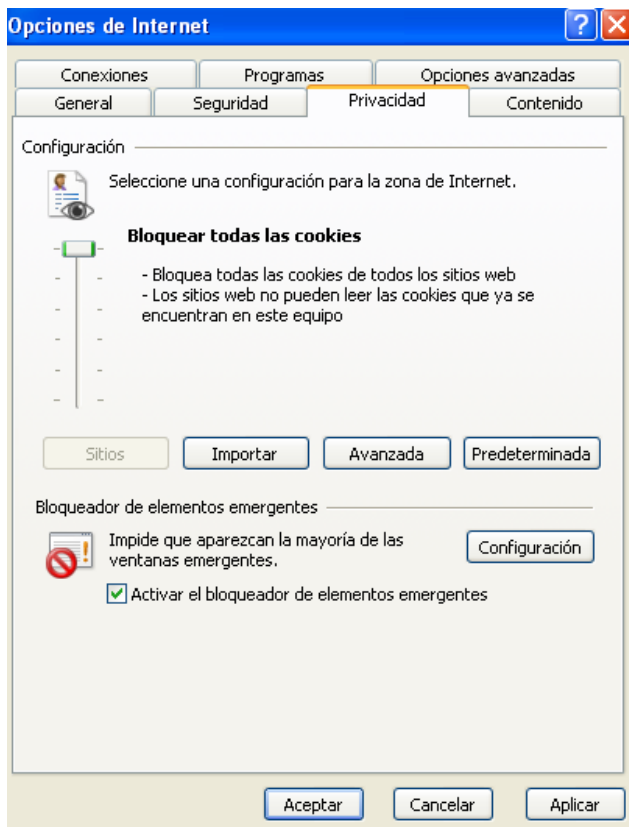
#### Registro de las cookies

Presentan un importante problema de privacidad, dos motores de búsqueda pueden compartir datos sobre usuarios cargando pequeñas imágenes de una tercera parte que usa las cookies y comparte los datos con los dos motores de búsqueda.

Por ejemplo, un site que podría tener un nivel de seguridad muy superior al estándar podría permitir a los usuarios saltarse el nombre y la password mediante cookies. Éste es el caso de algunas librerías on-line que usan cookies para recordar a sus usuarios y permite hacer pedidos sin tener que reintroducir nuestra información personal.



## Desactivar cookies



- Como autor de servlets que podemos usar cookies, no deberíamos confiar a las cookies información particularmente sensible, ya que el usuario podría correr el riesgo de que alguien accediera a su ordenador o a sus ficheros de cookies.
- Para enviar cookies al cliente, un servlet debería crear una o más cookies con los nombres y valores apropiados mediante `new Cookie(name, value)` seleccionar cualquier atributo opcional mediante `cookie.setXXX`, y añadir las cookies a la cabecera de respuesta mediante `response.addCookie(cookie)`.
- Para leer cookies entrantes, llamamos `request.getCookies()`, que devuelve un array de objetos `Cookie`, recorreremos el array hasta encontrar aquella cuyo nombre (`getName`) corresponda con el nombre que tenemos en mente, luego llamamos a `getValue` sobre esa.

## Crear cookies

Un objeto `Cookie` se crea llamando al constructor `Cookie`, que toma dos strings: el nombre y el valor de la cookie. Ni el nombre ni el valor deberían contener espacios en blanco y ninguno de estos caracteres: `[ ] ( ) = , " / ? @ ; ;`

Antes de añadir la cookie a la cabecera saliente, podemos configurar la cookie estableciendo alguno de sus atributos:

- `getDomain/setDomain`, obtiene o selecciona el dominio al que se aplica la cookie.
- `getMaxAge/setMaxAge`, obtiene o selecciona el tiempo (en segundos) que debe pasar hasta que expire la cookie.
- `getName/setName`, obtiene o selecciona el nombre de la cookie.



## Cookies

La cookie se añade a la cabecera de respuesta Set-Cookie por medio del método `addCookie` de `HttpServletResponse`. Aquí podemos ver un ejemplo:

```
Cookie userCookie = new Cookie ("user", "uid1234");
userCookie.setMaxAge (86400);
response.addCookie (userCookie);
```

### Recuperar Cookies

Desde el método `service ()` de un Servlet podemos recuperar las cookies enviadas por el navegador a través del método `getCookies ()` del objeto `HttpServletRequest`.

Este método devuelve un array de objetos `Cookie` con todas las cookies recibidas en la petición. Si buscamos una cookie en concreto habrá que recorrer el array:

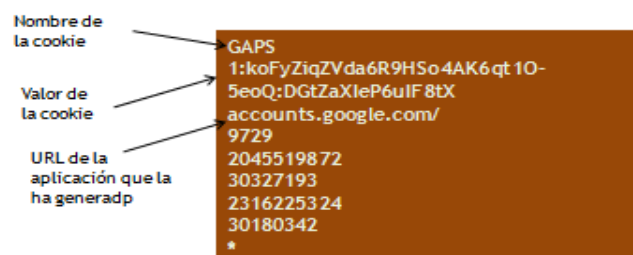
```
Cookie [] cookies= request.getCookies ();
String user=null;
if (cookies!=null){
    for(Cookie c:cookies){
        //si encuentra la cookie llamada user
        //recupera su valor
        if(c.getName.equals("user"){
            user=c.getValue();
        }
    }
}
```

### Estructura de una Cookie

Las cookies son almacenadas por el navegador en archivos `.txt` dentro del disco duro.

La localización de estos archivos depende del sistema operativo del cliente. En el caso de Windows 7, esta localización suele ser la carpeta `c:\usuarios\nombre_usuario`

A continuación, se muestra el contenido de uno de estos archivos:



### Cookies y atributos de sesión

Los servidores de aplicaciones utilizan cookies para almacenar los identificadores de sesión de los usuarios y así poder conocer que objeto HttpSession corresponde a cada usuario cuando este hace una llamada al método getSession().

A este tipo de cookie se les conoce como cookie de sesión y su nombre es jsessionid. La cookie jsessionid es gestionada automáticamente por el contenedor Web, el programador no tendrá que preocuparse de ella.

Aunque los navegadores permiten desactivar cookies de usuario, manteniendo el uso normal de cookies jsessionid, es posible que en un entorno de muy alta seguridad este tipo de cookies también estén desactivadas, lo que provocará un mal funcionamiento de la aplicación.

Existe una técnica que permite solucionar en parte ese problema y que se conoce como reescritura de URL.

### Reescritura de URL

Esta técnica consiste en añadir el identificador de sesión al final de cada URL que apunte a la aplicación. El servidor utilizará esta información para obtener el objeto HttpSession asociado al cliente.

Esta solución tiene la ventaja de que funciona con navegadores que no soportan cookies o cuando el usuario las ha desactivado. Sin embargo, tiene también algunos inconvenientes:

- Requiere gran cantidad de código extra en el Servlet para realizar la reescritura de todas las URL.
- Tenemos que ser muy cuidadosos con que cada URL que le devolvamos al usuario tiene añadida la información extra.
- Si el usuario accede a la aplicación mediante un bookmark o un enlace, la información de sesión puede perderse.