

## Manejo de Cookies

<b>Manejo de Cookies</b>	<b>1</b>
Competencias	2
Introducción	2
Entender funcionamiento de cookies	3
Armar cuerpo de cookies para su recuperación	5



**¡Comencemos!**

## Competencias

- Entender funcionamiento de cookies
- Aplicar métodos básicos para manejo de cookies
- Armar cuerpo de cookies para su creación

## Introducción

En la sección anterior vimos como trabajar con las variables de sesión en un *servlet*. En este capítulo vamos a trabajar con *cookies*, que es una tecnología que también se usa para administrar valores en una aplicación *java*. Las cookies por definición son pequeñas piezas de software que se encargan de persistir los valores que viajan en los *request* (las peticiones de los clientes).

Una *cookie* se compone de un nombre, un valor y atributos adicionales como comentarios, path y direcciones de dominio, además de un tiempo de expiración y números de versión.

El proyecto completo lo puedes encontrar en el archivo Material Apoyo Lectura, en el proyecto *manejoCookies*.

## Entender funcionamiento de cookies

Vamos a crear un nuevo web *dinamic proyect* en eclipse, para luego crear dos *servlet* los cuales tendrán las siguientes labores:

1. Primer *servlet*: Se encargará de generar una *cookie* para enviarla al navegador.
2. Segundo *servlet*: Su misión será la de recuperar la *cookie* mediante el *request* del cliente, para luego mostrar por pantalla su *id* y su *valor*. El primer *servlet* tendrá el nombre de *ManejoCookie*.

Aplicar métodos básicos para manejo de cookies

```
@WebServlet("/generadorCookie")
public class ManejoCookies extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException, ServletException{
        Cookie cookie = new Cookie("userjorge","viernes2019");
        cookie.setMaxAge(9000);
        cookie.setComment("Cookie guardada con propositos
educativos");
        response.addCookie(cookie);
        PrintWriter print = response.getWriter();
        print.println("<html><body>");
        print.println("<h2><i>Cookie guardada
correctamente:</i></h2><br>");
        print.println("Valor de la cookie: <strong>"+
cookie.getValue() + "</strong>");
        print.println("<br>");
        print.println("Tiempo de la duración de la cookie: <strong>"
+ cookie.getMaxAge()+"</strong>");
        print.println("<br>");
        print.println("Comentario: <strong>" + cookie.getComment() +
"</strong>");
        print.println("</body></html>");
    }
}
```

Para manejar cookies, debemos generar un constructor:

```
Public Cookie(java.lang.String name,java.lang.String value)
```

Este constructor especifica un *nombre* y un *valor*. El constructor acepta valores de la clase String, pero por norma sólo puede recibir valores alfanuméricos: no se acepta comas, ni comillas, ni caracteres especiales ni espacios.

El objeto *cookie* creado anteriormente, cuenta con varios métodos que dan funcionalidad y permiten generar y obtener valores de la cookie creada. En el código utilizamos los siguientes:

- **setMaxAge(int expiración)**: Setea el tiempo máximo de expiración de la *cookie* en segundos.
- **setComment(String comentario)**: Añade un comentario que representa el propósito de la *cookie*.
- **addCookie(Cookie cookie)**: Añade la *cookie* al response *http*, para que el *servlet* la envíe al navegador. Puede ser llamada varias veces para guardar múltiples cookies.
- **getValue()**: Retorna el valor de la *cookie*.
- **getComment()**: Retorna el valor del comentario.
- **getMaxAge()**: Retorna el tiempo máximo de expiración. Si retorna -1 indica que la *cookie* se destruirá al cerrarse el navegador.

Para más detalles de los métodos restantes, puedes visitar la API de [javax](https://www.javatpoint.com/java-api/java/net/cookie).

## Armar cuerpo de cookies para su recuperación

El segundo *servlet*, se encargará de obtener los valores de la *cookie* que guardamos anteriormente, el código es el siguiente:

Podemos ver un ciclo iterativo, esto es a causa de que el *request* guarda las *cookies* en una estructura de arreglo, por lo cual para acceder a sus valores hay que utilizar su índice. Dicho esto, continuemos analizando.

```
@WebServlet("/recuperaCookie")
public class RecuperaCookie extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException, ServletException{
        Cookie[] cookies = request.getCookies();
        PrintWriter print = response.getWriter();
        print.println("<html><body>");
        print.println("<h2><i>Cookie recuperada
correctamente:</i></h2><br>");
        for(int i = 0; i < cookies.length; i++) {
            print.println("Id de la cookie: <strong>"+
cookies[i].getValue() + "</strong>");
            print.println("<br>");
            print.println("Valor de la cookie: <strong>" +
cookies[i].getName()+"</strong>");
            print.println("<br>");
        }
        print.println("</body></html>");
    }
}
```

Se genera un objeto de la clase *Cookie*, el cual es un arreglo y le asignamos el valor depositado en el *request*:

```
Cookie[] cookies = request.getCookies();
```

Al tener el objeto con el contenido de la *cookie* del *request*, utilizando el ciclo iterativo accedemos a los valores mediante su índice.

```
[i].getValue();
```

```
[i].getName();
```

El resto del código es *html* incrustado, que será generado en la página web tal como lo hemos hecho en los ejemplos anteriores.

Para probar el funcionamiento del proyecto, ejecutar primero el *servlet* *ManejoCookies.java* mediante la dirección URL:

<http://localhost:8080/ManejoCookie/generaCookie>

y la ejecución mostrará los datos generados en un *html*:



Imagen 1. Ejecución del código.  
Fuente: Desafío Latam

En este estado de la aplicación, las *cookies* guardadas ya están guardadas en el computador del usuario, y para recuperarlas es necesario ejecutar el segundo *servlet* de nombre *RecuperaCookie.java* mediante la url:

<http://localhost:8080/ManejoCookie/recuperaCookie>

Y verás cómo recuperamos los valores de la *cookie*:



Imagen 2. Recuperando el valor de la *cookie*.  
Fuente: Desafío Latam

Con este ejemplo funcionando podemos verificar como mediante *servlets* podemos manipular las *cookies* de una manera bastante sencilla.