



Open  
TO  
Inspiration



# Sesiones y Cookies

PHP – Sesiones y Cookies

DAW – Desarrollo Web de Entorno Servido

Fernando Díaz-Alonso Dorado



# Sesiones y Cookies

## Generalidad

Una de las funcionalidades básicas de cualquier servidor de HTML es la de recoger el máximo de datos que pueda ayudar al usuario. Pero esto choca con una de las características del protocolo HTTP que es el stateless, es decir que no se conserva ningún estado de las peticiones que se recibe.

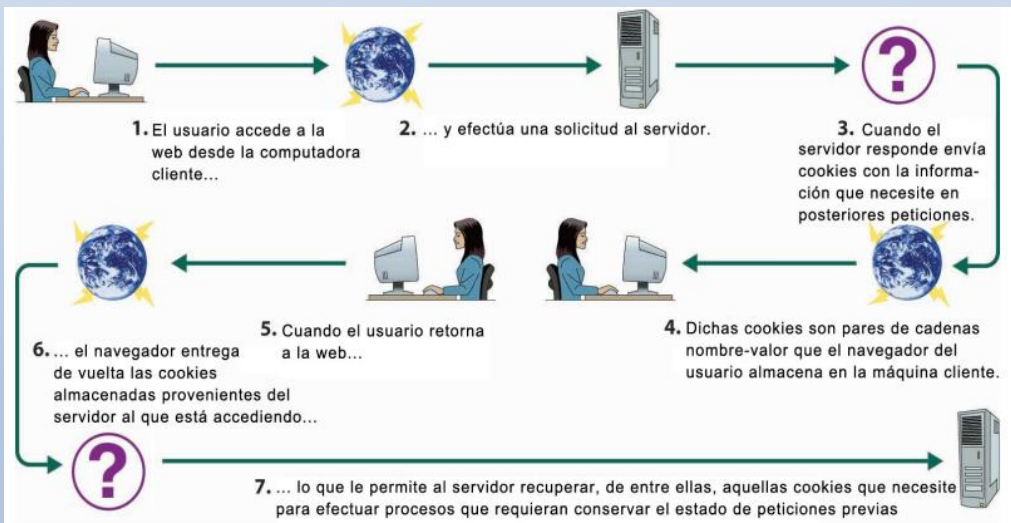
Por lo tanto, si nuestro servidor no tiene memoria, deberemos poder implementar algún método que en determinadas situaciones nos permita no tener que estar autenticándonos en cada una de las páginas que visitamos y dentro de sus subpáginas. Al final, para conseguir facilitarnos la vida se han creado las cookies y las sesiones.

Son mecanismos que nos permiten almacenar la información de nuestros usuarios, carritos de compra, preferencias, etc... para ofrecernos una experiencia de usuario más adecuada.

S

# Cookies

- Una cookie es un pequeño archivo que el servidor introduce en el ordenador del usuario.
- Cada vez que el usuario solicita una página desde un navegador, enviará la cookie también.



# Cookies II

## Usos y consideraciones

### Usos:

- Facilitar el mantenimiento del estado (PHP es stateless)
- Guardar datos con las preferencias de los usuarios.
- Posibilitar el uso de variables de sesión (Server-side)

### Consideraciones:

- El navegador debe permitir el uso de cookies.
- El tamaño de las cookies está limitado, por lo que guardan “poca” información.
- La información la guarda el navegador (Client-side)

<https://www.php.net/manual/es/features.cookies.php>

# Cookies III

## Creación, modificación

Creación o modificación:

***setcookies(nombre, valor, expira, path, dominio, segura, httpOnly);***

**<https://www.php.net/manual/es/function.setcookie.php>**

- Nombre → que identifica la cookie.
- Valor → contenido de la cookie.
- Expira → fecha en la que la cookie deja de ser válida.
- Path → que limita el uso de la cookie a una ruta concreta dentro del servidor.
- Dominio → que identifica a qué dominio pertenece la cookie.
- Segura → true o false, para indicar si es obligatorio usar https.
- HttpOnly → true o false, para indicar si forzamos a intercambiar la cookie solo a través de protocolo http/https

Eliminación: Mediante `setcookie()` con una fecha de caducidad en el pasado.

# Cookies IV

## Caducidad y obtener valor de la cookie

Caducidad:

- El valor de espiración es una marca de tiempo UNIX.
- Se establece en segundos.
- Para calcular la caducidad desde el momento actual:  
***time() + segundos\_has\_expiración.***
- Para calcular a fechas concretas → ***strtotime().***

Obtener el valor de la cookie introducida:

- Accedemos al array asociativo super global:  
***\$\_COOKIE["nombre\_cookie"]***
- Habitualmente se combina con ***isset()*** para evitar errores.  
***isset(\$\_COOKIE["nombre\_cookie"]) → nos devuelve un true o false***

# MySQL / MariaDB

## Insert, Update y Delete

Para realizar una sentencia insert, update y/o delete el proceso es igual que con una sentencia select. Usaremos:

***mysqli::query***

<https://www.php.net/manual/es/mysqli.query.php>

En este caso no se espera realmente datos de consulta, por lo que se puede recibir un true o false. En el caso del insert, existe la opción de recoger el valor de la id insertada mediante la función ***mysqli::insert id***.

<https://www.php.net/manual/en/mysqli.insert-id.php>

# Variables de sesión

## Permiten

- Acceder a la información de estado de un usuario desde cualquier parte de script de una aplicación web.
- Las variables de sesión se almacenan en el servidor a diferencia de las cookies que lo hacen en el cliente (navegador).
- Posee una marca temporal de caducidad, ***tienen asociada una cookie llamada PHPSESSID.***
- Establecer la caducidad, que se puede modificar por configuración o a través del script, pero suele estar establecida en 24 minutos.
- Que una vez se caduca la sesión se destruya la información.



# Variables de sesión II

## Definición

- Se crea un identificador único (para cada sesión, de forma particular) Este identificador es una cadena aleatoria compuesta por 32 caracteres en hexadecimal.
- Una cookie llamada ***PHPSESSID se envía automáticamente al cliente,*** para que guarde la cadena de identificación (antes generada) De esta forma el cliente queda asociado a la sesión y sus variables.
- Un archivo es creado en el servidor automáticamente, en la carpeta designada, y recibe como nombre el identificador pero con el prefijo ***sess\_***.
- Cuando PHP quiere acceder al valor de una variable de sesión, PHP automáticamente obtiene el “identificador único de sesión” desde la cookie PHPSESSID y luego busca en la carpeta designada el archivo asociado a este identificador.

# Variables de sesión III

## Habilitación y trabajo con una sesión

Para habilitar el acceso a una sesión:

***session\_start()***

**En cada script que haga uso de las variables de sesión.**

Para trabajar con las variables de sesión.

- Usaremos el array asociativo super global → `$_SESSION[]`  
`$_SESSION["nombre_variable"]` = valor → establecer, crear o modificarlo.  
`$_SESSION["nombre_variable"]` → para leer los datos.
- Para destruir la información de la sesión tenemos que usar:  
***session\_unset()*** → libera las variables de sesión, pero no elimina la sesión.  
***session\_destroy()*** → destruye la sesión.

Se suelen usar ambas funciones conjuntamente.

# Variables de sesión IV

## Parámetros de cookie de sesión

Para establecer los parámetros de la cookie de la sesión solemos usar la función:

***sesión\_set\_cookie\_params(nombre, valor, expira, path, dominio, segura, httpOnly);***

<https://www.php.net/manual/es/reserved.variables.session.php>

# Referencias

## Webgrafía

Todos los ejemplos se han sacado de la documentación oficial de PHP

- <https://www.php.net/>

## Bibliografía

Existen muchos libros de PHP, pero se recomienda:

Lopez, A. (2023). *Learning PHP 7*. Packt Publishing.