

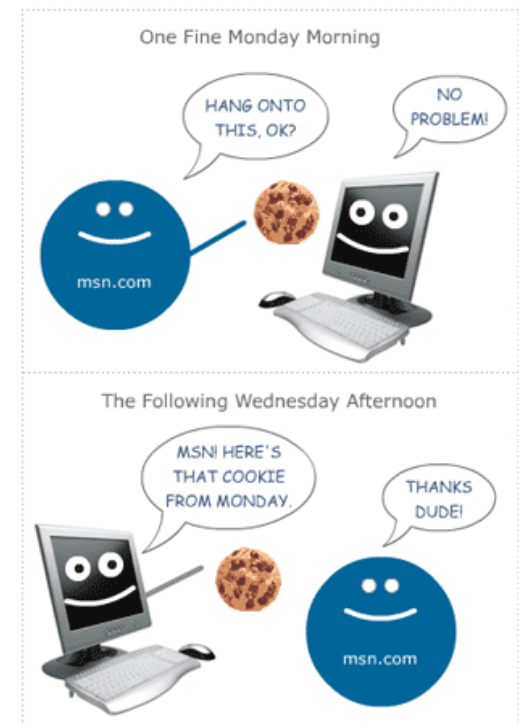
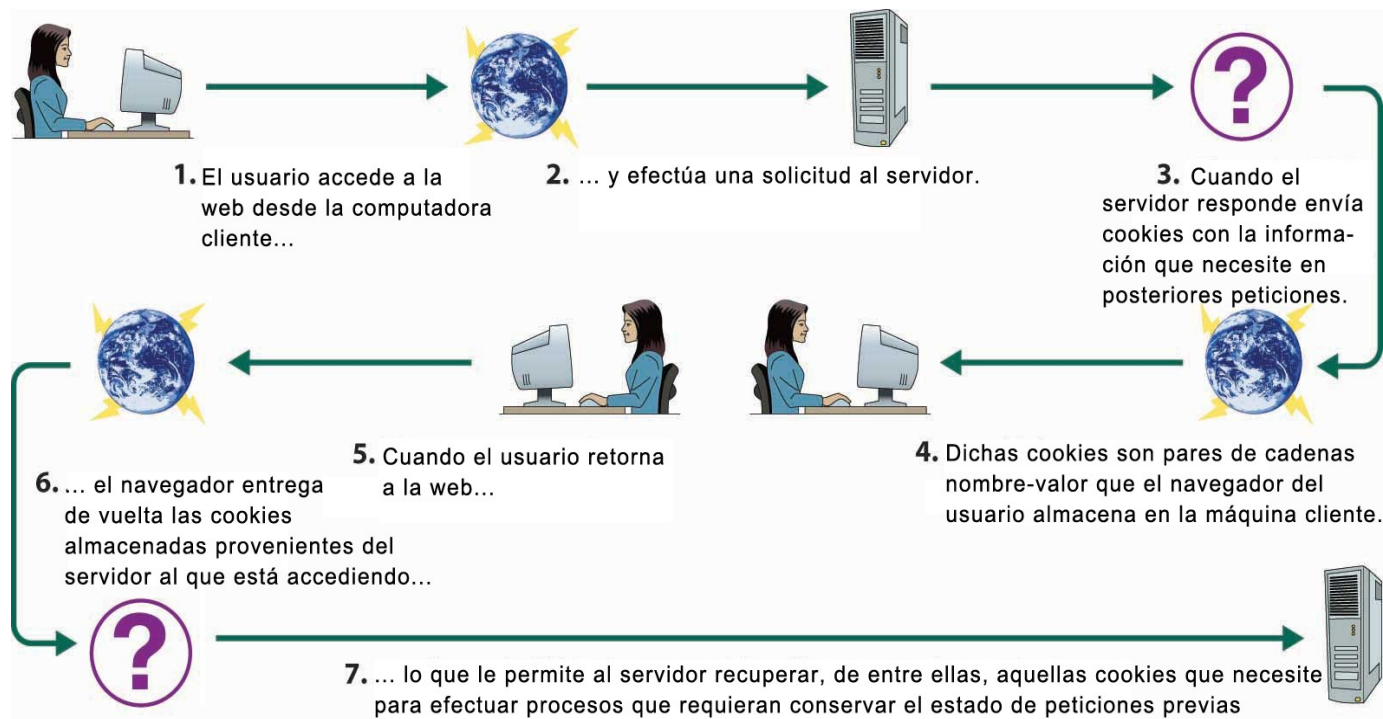
10.1. Sesiones y Cookies en PHP

1º DAW - Programación
David Soler Talens

Cookies (I)



- 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.





- **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)

Cookies (III)



- **Creación o modificación**

setcookie(nombre, valor, expira, path, dominio, segura, httpOnly);

- *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 | false, para indicar si es obligatorio usar https
- *HttpOnly* → true | false, para indicar si forzamos a intercambiar la cookie solo a través de protocolo http/https

- **Eliminar una cookie**

- *setcookie()* con una fecha de caducidad en el pasado

Cookies (IV)



- **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_hasta_expiración
- Para calcular a fechas concretas → *strtotime()*

- **Recuperar valor de cookie**

- Accedemos al array asociativo super global
\$_COOKIE["nombre_cookie"]
- Habitualmente se combina con *isset()* para evitar errores
isset(\$_COOKIE["nombre_cookie"]) → devuelve true|false

Variables de sesión (I)



- **Permiten**

- a una aplicación web acceder a *información de estado* de un usuario desde cualquiera Script de ésta.
- A diferencia de las cookies, las variables de sesión **se almacenan en el servidor**
- Una marca temporal de caducidad, dado que **tienen asociada una cookie llamada PHPSESSID**
- Establecer la **caducidad**, que se puede modificar por configuración o a través del Script, pero suele estar establecida 24 minutos
- Que se **destruya la información** de sesión una vez caducada la misma.

Variables de sesión (II)



- **La definición de una sesión comprende los siguientes pasos:**
 - 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*.
 - 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)



- **Habilitar el acceso a la sesión**

`session_start()` → en cada script que haga uso de las variables de sesión!!!

- **Trabajo con variables de sesión**

- Con el array asociativo super-global `$_SESSION`

`$_SESSION['nombre_variable'] = valor` → establecer/crea/modificar

`$_SESSION['nombre_variable']` → leer

- Destruir la información de sesión

`session_unset()` → libera las variables de sesión, aunque la sesión permanece activa

`session_destroy()` → destruye la sesión



Se suelen usar conjuntamente

Variables de sesión (IV)



- **Establecer los parámetros de la cookie de sesión**

session_set_cookie_params(nombre, valor, expira, path, dominio, segura, httpOnly);

- Referencia completa

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