

# Introducción Sesiones en PHP

Una de las limitaciones de las páginas web es que cada página web es un documento independiente. Eso hace que dos programas PHP no puedan, en principio, compartir información.

Aunque un formulario puede enviar información a un programa mediante controles ocultos , o una página puede redirigir a otra página enviando la información en la dirección, eso no puede considerarse compartir información, puesto que la página que recibe la información no puede saber si la información ha sido manipulada por el camino.

Por compartir información nos estamos refiriendo a utilizar variables comunes, sin que la información salga del servidor.

El ejemplo siguiente muestra dos páginas: en la primera se crea una variable, que se puede utilizar en esa misma página. La segunda página, que se encuentra en el mismo servidor, no tiene acceso a la variable creada en el primer programa y por eso se produce un aviso.

PHP puede superar esta limitación mediante las sesiones. Las sesiones permiten que páginas distintas puedan acceder a una variable común, la matriz `$_SESSION`.

El ejemplo siguiente muestra dos páginas: en la primera se crea la variable en la matriz `$_SESSION`, que se puede utilizar en esa misma página. La segunda página, que se encuentra en el mismo servidor, tiene acceso a la variable creada en el primer programa (si se ha ejecutado antes, obviamente).

Las sesiones no deben confundirse con las cookies. Las cookies es un método que permite guardar información en el disco duro del cliente para recuperarla en el futuro, mientras que en las sesiones la información se mantiene en el servidor hasta que se cierra la sesión (por intervención del usuario o por tiempo).

El trabajo con sesiones tiene tres partes:

- Creación o apertura de la sesión

Cuando alguna página crea una sesión utilizando la función correspondiente, el servidor asocia al navegador del usuario un identificador de usuario único. El identificador se guarda en el usuario en forma de cookie o, si el navegador del usuario no permite la creación de cookies, añadiendo el identificador en la dirección de la página.

- Utilización de la sesión

Una vez creada la sesión, las páginas solicitadas por el mismo navegador pueden guardar y recuperar información en el servidor, información que se asocia al identificador de usuario, por lo que no es accesible a otras página. La información se conserva hasta que el usuario o el servidor destruyan la sesión.

- Destrucción o cierre de la sesión

Tanto el usuario como el servidor pueden cerrar la sesión. El usuario puede destruir la sesión cerrando el navegador. El servidor puede destruir la sesión cuando alguna página utilice la función correspondiente o debido al tiempo transcurrido desde su creación.

## 1 Directiva session.save\_handler

Para utilizar sesiones mediante el mecanismo propio de PHP (es decir, sin necesidad de crear funciones propias), la directiva session.save\_handler del archivo de configuración php.ini debe tener el valor files.

```
session.save_handler = files           ; Valor recomendado en este módulo
```

Esta configuración es la más habitual, pero algunos gestores de contenidos (CMS) tienen sus propias funciones de gestión de sesiones y requieren que esta directiva tome el valor user.

```
session.save_handler = user           ; Valor definido en algunos servidores
```

Si se ha necesitado modificar php.ini, pero queremos ejecutar otros programas que no incluyen sus propias funciones de gestión de sesiones, se puede incluir en las páginas PHP la función [ini\\_set\(\)](#) antes de abrir la sesión. Ese cambio sólo afectará a la página que incluya la llamada a la función.

```
ini_set("session.save_handler", "files"); // Necesario únicamente cuando
                                           //session.save_handler = user en php.ini
session_start();
```

En este curso se supondrá que la directiva tiene el valor "files".

## 2 Crear la sesión

En PHP, las sesiones se crean mediante la función `session_start()`. Si la sesión no existía, esta función crea la sesión y le asocia un identificador de sesión único. Si la sesión ya existía, esta función permite que la página tenga acceso a la información vinculada a la sesión. Es decir, todas las páginas que quieran guardar datos en `$_SESSION` o leer datos de `$_SESSION` deben comenzar con la función `session_start()`.

La creación de sesiones requiere el envío de cabeceras HTTP, por lo que la función `session_start()` debe utilizarse antes de empezar a escribir el contenido de la página. En caso contrario PHP producirá un aviso y no se creará la sesión. El motivo es que el identificador de sesión se utiliza en las cabeceras de respuesta HTTP y las cabeceras se envían antes del texto de la página.

El ejemplo siguiente es incorrecto, ya que utiliza la función `session_start()` después de haber escrito texto.

```
<?php
// Este código es incorrecto, la sesión se crea después de crear texto
print "<p>Hola</p>\n";
session_start();
?>
```

En principio, cuando el navegador se conecta a un servidor, la sesión es única, es decir, todas las páginas del mismo dominio compartirán la misma matriz `$_SESSION`. La función [`session\_name\(\)`](#) permite establecer un nombre de sesión específico, de manera que todas las páginas que declaren el mismo nombre de sesión accederán a la misma matriz `$_SESSION` y las que tengan nombres de sesión distintos accederán a matrices `$_SESSION` diferentes.

En el ejemplo siguiente, los dos primeros programas crean la misma variable en `$_SESSION`, pero como se ha utilizado la función `session_name()` con nombres diferentes, realmente lo hacen en matrices `$_SESSION` distintas.

El nombre de sesión distingue entre minúsculas y mayúsculas, es decir, dos sesiones con el mismo nombre, pero uno en minúsculas y otro en mayúsculas, son dos sesiones distintas.

El nombre de la sesión no puede contener únicamente números, ni tampoco puede contener los caracteres punto (.), ampersand (&), más (+), corchete izquierdo ([) ni almohadilla (#).

**Nota:** No he encontrado una lista oficial de caracteres no admitidos. Los caracteres `.` `&` `+` `[` y `#` los he encontrado probando manualmente, pero es posible que haya otros caracteres problemáticos.

### 3 Utilizar la sesión

Cuando una página ha creado una sesión o ha accedido a una sesión ya existente mediante `session_start()`, la página tiene acceso a la matriz `$_SESSION` que contiene las variables de esa sesión.

La matriz `$_SESSION` es una matriz asociativa en la que se pueden definir valores como en cualquier otra matriz. La diferencia es que `$_SESSION` es accesible desde páginas diferentes (siempre que esas páginas tengan asociada la misma sesión), manteniéndose los valores de una página a otra.

El ejemplo siguiente muestra dos páginas. La primera página guarda información en `$_SESSION` y la segunda la utiliza.

Los nombres de los primeros índices de la matriz `$_SESSION` tienen que cumplir las mismas reglas que los nombres de las variables, es decir, que el primer carácter debe ser una letra o un guion bajo (`_`). En particular, no deben ser números ni contener caracteres no alfanuméricos.

### 4 Borrar elementos de la sesión

Los valores de `$_SESSION` se borran como en cualquier otra matriz mediante la función [`unset\(\)`](#).

Para borrar todos los valores de `$_SESSION` se pueden borrar uno a uno o utilizar la función [`session\_unset\(\)`](#). Pero no se debe utilizar [`unset\(\$\_SESSION\)`](#):

- [`unset\(\$\_SESSION\)`](#) impide que el resto de la página escriba o lea valores en `$_SESSION`, pero la sesión conserva los valores, por lo que otras páginas seguirían viendo esos valores.
- la función [`session\_unset\(\)`](#) borra todos los valores pero permite que el resto de la página (y otras páginas) escriba o lea valores en `$_SESSION`.

## 5 Cerrar la sesión

Cerrar una sesión es destruir la matriz `$_SESSION` y el identificador de la sesión.

Las sesiones se pueden cerrar de varias maneras:

- El usuario puede cerrar la sesión simplemente cerrando el navegador.
- Un programa puede cerrar la sesión mediante la función `session_destroy()`.
- El servidor puede cerrar la sesión cuando ha pasado el tiempo indicado en segundo en la directiva `session.gc_maxlifetime`.

Cuando se destruye una sesión, el programa que ha destruido la sesión sigue teniendo acceso a los valores de `$_SESSION` creados antes de la destrucción de la sesión, pero las páginas siguientes no. Si se ejecuta el primero de los ejemplos siguientes y después el segundo, se obtienen los resultados indicados: