



Sesiones en PHP

Definición y funcionamiento de las sesiones

Una sesión es un mecanismo de programación de las tecnologías de web que permite conservar información de un usuario al pasar de una página a otra.

Las sesiones se implementan mediante una cookie que almacena un valor que identifica al usuario en el servidor web cada vez que pasa de una página web a otra.

Pero a diferencia de una cookie, los datos asociados a una sesión se almacenan en el servidor y nunca en el cliente.

En el servidor web están almacenados todos los datos de la sesión y se puede acceder a ellos cuando que se pasa de un script a otro gracias al identificador almacenado en la cookie.

Identificador de sesión SID

- Cada usuario distinto de un sitio web tiene su propia información de sesión.
- Para distinguir una sesión de otra **los servidores usan los identificadores de sesión (SID).**
- Un SID es un atributo que se asigna a cada uno de los visitantes de un sitio web y lo identifica. Es único para cada usuario.
- De esta forma, si el servidor web conoce el SID de un usuario, puede relacionarlo con toda la información que posee sobre él, que se mantiene en la sesión del usuario.
- Esa información se almacena en el servidor web en ficheros, aunque también se pueden utilizar otros mecanismos de almacenamiento como bases de datos.

¿Dónde se almacena el SID?

Existen dos maneras de mantener el SID entre las páginas de un sitio web que visita el usuario:

Utilizando cookies

Propagando el SID en un parámetro de la URL. El SID se añade como una parte más de la URL. (**Esta opción no la usaremos nunca porque no es segura**)

Configuración

PHP incluye soporte de sesiones incorporado.

Sin embargo, antes de utilizar sesiones en tu sitio web, debes configurar correctamente PHP utilizando las siguientes directivas en el fichero **php.ini**

Directiva	Significado
<code>session.use_cookies</code>	Indica si se deben usar cookies (1) o propagación en la URL (0) para almacenar el SID.
<code>session.use_only_cookies</code>	Se debe activar (1) cuando utilizas cookies para almacenar los SID, y además no quieres que se reconozcan los SID que se puedan pasar como parte de la URL (este método se puede usar para usurpar el identificador de otro usuario).
<code>session.save_handler</code>	Se utiliza para indicar a PHP cómo debe almacenar los datos de la sesión del usuario. Existen cuatro opciones: en ficheros (<code>files</code>), en memoria (<code>mm</code>), en una base de datos SQLite (<code>sqlite</code>) o utilizando para ello funciones que debe definir el programador (<code>user</code>). El valor por defecto (<code>files</code>) funcionará sin problemas en la mayoría de los casos.
<code>session.name</code>	Determina el nombre de la cookie que se utilizará para guardar el SID. Su valor por defecto es <code>PHPSESSID</code> .
<code>session.auto_start</code>	Su valor por defecto es 0, y en este caso deberás usar la función <code>session_start</code> para gestionar el inicio de las sesiones. Si usas sesiones en el sitio web, puede ser buena idea cambiar su valor a 1 para que PHP active de forma automática el manejo de sesiones.
<code>session.cookie_lifetime</code>	Si utilizas la URL para propagar el SID, éste se perderá cuando cierres tu navegador. Sin embargo, si utilizas cookies, el SID se mantendrá mientras no se destruya la cookie. En su valor por defecto (0), las cookies se destruyen cuando se cierra el navegador. Si quieres que se mantenga el SID durante más tiempo, debes indicar en esta directiva ese tiempo en segundos.

¿Cómo se
realiza el inicio
de sesión?

El inicio de sesión puede
tener lugar de dos formas

De forma automática,
si se activa la
directiva
session.auto_start en
la configuración de
PHP.

Ejecutando la función
session_start() para
indicar a PHP que
inicie una nueva
sesión o reanude la
anterior

Funciones para utilizar sesiones

1. **session_start()**: Inicia una **nueva sesión o reanuda la sesión existente**.
2. **session_id()**: Obtiene o establece el identificador de sesión actual.
3. **session_name()**: Obtiene o establece el nombre de la cookie de sesión.
4. **session_destroy()**: Elimina todos los datos de la sesión y finaliza la sesión actual.
5. **session_status()**: Obtiene el estado actual de la sesión:
 - **PHP_SESSION_DISABLED**: Indica que las sesiones están deshabilitadas en la configuración de PHP.
 - **PHP_SESSION_NONE**: Indica que las sesiones están habilitadas en la configuración de PHP, pero **no se ha iniciado ninguna sesión** en el script actual.
 - **PHP_SESSION_ACTIVE**: Indica que las sesiones están habilitadas en la configuración de PHP y **una sesión está activa en el script actual**.
6. **session_get_cookie_params()**: Obtiene los parámetros de la cookie de sesión.
7. **session_set_cookie_params()**: Establece los parámetros de la cookie de sesión.
8. **session_regenerate_id(true)**: genera el nuevo ID de sesión, este se envía al cliente en forma de una nueva cookie de sesión, reemplazando la cookie anterior en el navegador del usuario. Esto es útil para mejorar la seguridad de la sesión y evitar ataques como la suplantación de sesión. El servidor

Variable superglobal \$_SESSION

\$_SESSION es **un array superglobal** que se utiliza para almacenar variables de sesión que persisten a lo largo de múltiples páginas o solicitudes **mientras una sesión de usuario está activa.**

Las variables almacenadas en \$_SESSION están disponibles para su uso en todas las páginas de un sitio web una vez que se ha iniciado una sesión, lo que permite mantener la información del usuario durante su visita a la aplicación web.

fichero1.php

```
<?php
// Iniciar sesión
session_start();
echo "<h1>Fichero 1</h1>";
// Obtener los valores del formulario si se han enviado
if (isset($_POST['nombre'])) {
    // Guardar información en $_SESSION para conservarla
    $_SESSION['nombre'] = $_POST['nombre'];
    $_SESSION['apellido'] = $_POST['apellido'];
}

include 'formulario1.php';

echo "<pre>Contenido de la sesión: <br>";
print_r($_SESSION);
echo "</pre>";

// Enlace a los otros ficheros
echo '<br><h3><a href="fichero2.php">Enlace al fichero 2</a></h3>';
?>
```

Nombre y Apellidos

Nombre:

Apellido:

Contenido de la sesión:

Array

(

[nombre] => Pepito

[apellido] => Pérez

)

fichero2.php

```
<?php
// Iniciar sesión
session_start();
echo "<h1>Fichero 2</h1>";
// Obtener los valores del formulario si se han enviado
if (isset($_POST['ciudad'])) {
    // Guardar información en $_SESSION para conservarla
    $_SESSION['ciudad'] = $_POST['ciudad'];
    $_SESSION['pais'] = $_POST['pais'];
}

include 'formulario2.php';

echo "<pre>Contenido de la sesión: <br>";
print_r($_SESSION);
echo "</pre>";

//Enlace a los otros ficheros
echo '<h3><a href="fichero1.php">Enlace al fichero 1</a></h3>';
```

Ciudad y País

Ciudad:

País:

Contenido de la sesión:

Array

(

[nombre] => Pepito

[apellido] => Pérez

[ciudad] => Salamanca

[pais] => España

)

Variable superglobal \$_SESSION

- La variable \$_SESSION aunque es superglobal **NO estará disponible hasta que inicie sesión.**
- Si un script intenta utilizarla antes de iniciar sesión obtendrá:

A screenshot of a PHP warning message. It features a red square icon with a white exclamation mark on the left. To its right, the text "Warning: Undefined global variable \$_SESSION" is displayed in white on a blue background. Further to the right, the text "in C:\xampp\htdocs\" is shown in white on an orange background.

(!) Warning: Undefined global variable \$_SESSION in C:\xampp\htdocs\

- Mientras la sesión permanece abierta, puedes utilizar la variable superglobal \$_SESSION para añadir información a la sesión del usuario o para acceder a la información almacenada en la sesión.

```
<?php
session_start();
var_dump($_SESSION);
```

Variable superglobal \$_SESSION

- Muy importante, \$_SESSION es una matriz asociativa y su índice **NO** puede ser un número.

```
<?php
session_start();
var_dump($_SESSION);
$_SESSION["1"] = "prueba";
var_dump($_SESSION);
```

(!) Warning: Unknown: Skipping numeric key 1 in Unknown on line 0

```
C:\xampp\htdocs\xClase\venancio\siones\prueba1.php:6:
array (size=1)
  1 => string 'prueba' (length=6)
```

Inicio de sesión

session_start()

- Si está activada la directiva **session.auto_start** en la configuración de PHP la sesión comenzará automáticamente en cuanto un usuario se conecte a un sitio web. **(No usaremos esta directiva)**
- En caso contrario, la sesión se inicia cuando un script ejecute la función **session_start()**.
- Si ese usuario había abierto una sesión con anterioridad, y ésta no se haya eliminado, en lugar de abrir una nueva sesión **se reanudará la anterior.**
- Para ello se utilizará el SID anterior, que estará almacenado en una cookie

Cookies de sesión

Cookies en uso

Permitido

Bloqueado

Las siguientes cookies se establecieron cuando visualizó esta página

localhost

Cookies

PHPSESSID

Nombre	PHPSESSID
Conteni...	i2rae1b03dl5f42pmsc7im37a3
Dominio	localhost
Ruta de...	/
Enviar p...	Solo conexiones del mismo sitio
Creado	lunes, 6 de noviembre de 2023, 11:37:36
Expira	Cuando finalice la sesión de exploración

Bloquear

Quitar

Listo

Valores por defecto de una sesión.

Funciones de configuración

- **session_id():** asigna un identificador a una sesión SID (se recomienda que contenga sólo caracteres alfanuméricos).
Si no asignamos un identificador, se asignará uno automáticamente.
En caso de ser usada esta función, **debe estar antes** de session_start().
- **session_name():** asigna el nombre a una sesión (se recomienda que sea corto y que contenga sólo caracteres alfanuméricos).
Si no indicamos un nombre, se usará por defecto el definido en el archivo de configuración de PHP (por defecto es **PHPSESSID**).
IMPORTANTE: en caso de usarse deberá estar definida en cada archivo .php que quiera usar la sesión y situada antes de session_start().
- Por defecto, permite que todas las aplicaciones del dominio accedan a la cookie. Establece **path= ' / '**

```

<?php
echo "<h3>Primera página</h3>";
// Establecer el nombre de la cookie de sesión personalizado
session_name('mi_sesion');

// Iniciar la sesión
session_start();

// Obtener el ID de sesión
$session_id = session_id();

echo "SID de sesión: <b>$session_id</b><br><br>";
// Obtener el nombre de la sesión
echo "Nombre de la sesión: <b>".session_name(). "</b><br>";

echo '<hr><pre>Contenido de $_COOKIE: <br>';
print_r($_COOKIE);
echo "</pre><hr>";

// Enlace a la segunda página
echo '<br><a href="archivo2.php">Enlace a la página 2</a>';
?>

```

Primera página

SID de sesión: **ntpnsei3c4gpd37c11g8vurek1**

Nombre de la sesión: **mi_sesion**

Contenido de \$_COOKIE:

```

Array
(
    [mi_sesion] => ntpnsei3c4gpd37c11g8vurek1
)

```

[Enlace a la página 2](#)

```

<?php
echo "<h3>Segunda página</h3>";
// Establecer el nombre de la cookie de sesión personalizado
session_name('mi_sesion');

// Iniciar la sesión
session_start();

// Obtener el ID de sesión
$session_id = session_id();

echo "SID de sesión: <b>$session_id</b><br><br>";
// Obtener el nombre de la sesión
echo "Nombre de la sesión: <b>".session_name(). "</b><br>";

echo '<hr><pre>Contenido de $_COOKIE: <br>';
print_r($_COOKIE);
echo "</pre><hr>";

// Enlace de regreso a la primera página
echo '<br><a href="archivo1.php">Enlace a la página 1</a>';
?>

```

Segunda página

SID de sesión: **ntpnsei3c4gpd37c11g8vurek1**

Nombre de la sesión: **mi_sesion**

Contenido de \$_COOKIE:

```

Array
(
    [mi_sesion] => ntpnsei3c4gpd37c11g8vurek1
)

```

[Enlace a la página 1](#)

Cookies en uso

Permitido Bloqueado

Las siguientes cookies se establecieron cuando visualizó esta página

▲ localhost

▲ Cookies

mi_sesion

Nombre	mi_sesion
Conteni...	ntpnsei3c4gpd37c11g8vurek1
Dominio	localhost
Ruta de...	/
Enviar p...	Solo conexiones del mismo sitio
Creado	lunes, 6 de noviembre de 2023, 11:46:48
Expira	Cuando finalice la sesión de exploración

Bloquear

Quitar

Listo

Iniciar una sesión
cambiando los
parámetros por
defecto

`session_set_cookie_params()`

o

`session_start(array $options = []): bool`

Establecer los parámetros de la cookie de sesión

```
session_set_cookie_params(  
    int $lifetime_or_options,  
    ?string $path = null,  
    ?string $domain = null,  
    ?bool $secure = null,  
    ?bool $httponly = null  
): bool
```

```
// Asignar un nombre a la sesión  
session_name('mi_sesion_ruta_admin');  
session_set_cookie_params([  
    'lifetime' => 7200,  
    'path' => '/xclase/asd/sesiones/admin/', //Ruta donde será válida, debe  
    ser absoluta y comenzar con /  
    'domain' => 'localhost',  
    'secure' => false, //Solo se enviará por HTTPS si es true.  
    'httponly' => true //No accesible desde JavaScript.  
]);  
// Iniciar la sesión  
session_start();
```

Iniciar una sesión
cambiando los
parámetros por
defecto

```
<?php
// Cookie que durará 2 horas, será válida solo en /admin y no será accesible
desde JavaScript.
// Asignar un nombre a la sesión
session_name('mi_sesion_ruta_admin');
session_set_cookie_params([
    'lifetime' => 7200,
    'path' => '/xclase/asd/sesiones/admin/',
    'domain' => 'localhost',
    'secure' => false,
    'httponly' => true    // No accesible desde JavaScript.
]);
session_start();
var_dump($_COOKIE);
```

Cookies en uso

Permitido Bloqueado

Las siguientes cookies se establecieron cuando visualizó esta página

localhost

Cookies

mi_sesion_ruta_admin

Nombre	mi_sesion_ruta_admin
Conteni...	l90hs1u0j07e5o21jg0qu7p77q
Dominio	localhost
Ruta de...	/xclase/asd/sesiones/admin/
Enviar p...	Solo conexiones del mismo sitio
Creado	miércoles, 16 de octubre de 2024, 17:57:35
Expira	miércoles, 16 de octubre de 2024, 19:57:35

Bloquear

Quitar

Listo

Iniciar una sesión cambiando los parámetros por defecto

`session_set_cookie_params()`

o

`session_start(array $options = []): bool`

Establecer los parámetros de la cookie de sesión

```
// Asignar un nombre a la sesión
session_name('mi_sesion_personalizada2');

// Configurar los parámetros de la cookie de sesión
$options = [
    'cookie_lifetime' => 3600,           // La cookie durará 1 hora.
    'cookie_path' => '/xclase/asd/sesiones/admin/', //válida en /admin
    'cookie_domain' => 'localhost', // Válida en todo el dominio.
    'cookie_secure' => false,           // SE envía por HTTP.
    'cookie_httponly' => true,           // No accesible vía JavaScript.
    'use_strict_mode' => true,           // NO aceptará cookies de sesiones con ID
    // enviadas por GET o POST. Solo por cookies.
];

// Iniciar la sesión con las opciones configuradas
session_start($options);
```

Cookies en uso

Permitido Bloqueado

Las siguientes cookies se establecieron cuando visualizó esta página

localhost

Cookies

mi_sesion_personalizada2

Nombre	mi_sesion_personalizada2
Conteni...	bl0kjde5j0bhocmv3pl8knhefe
Dominio	localhost
Ruta de...	/xclase/asd/sesiones/admin/
Enviar p...	Solo conexiones del mismo sitio
Creado	viernes, 18 de octubre de 2024, 12:27:49
Expira	viernes, 18 de octubre de 2024, 13:27:49

Inicio de sesión con la función `session_start()`

La función `session_start()` le indica a PHP que inicie una nueva sesión o reanude la anterior.

A partir de la versión 5.3.0 de PHP **devuelve false** en caso de no poder iniciar o restaurar la sesión.

Todas las páginas que necesiten utilizar la información almacenada en la sesión deberán ejecutar la función `session_start()`.

Inicio de sesión con la función `session_start()`

El inicio de sesión requiere utilizar cookies y éstas se transmiten en los encabezados HTTP.

Para poder iniciar una sesión utilizando `session_start()`, se debería realizar la llamada a esta función antes de que la página web muestre información en el navegador.

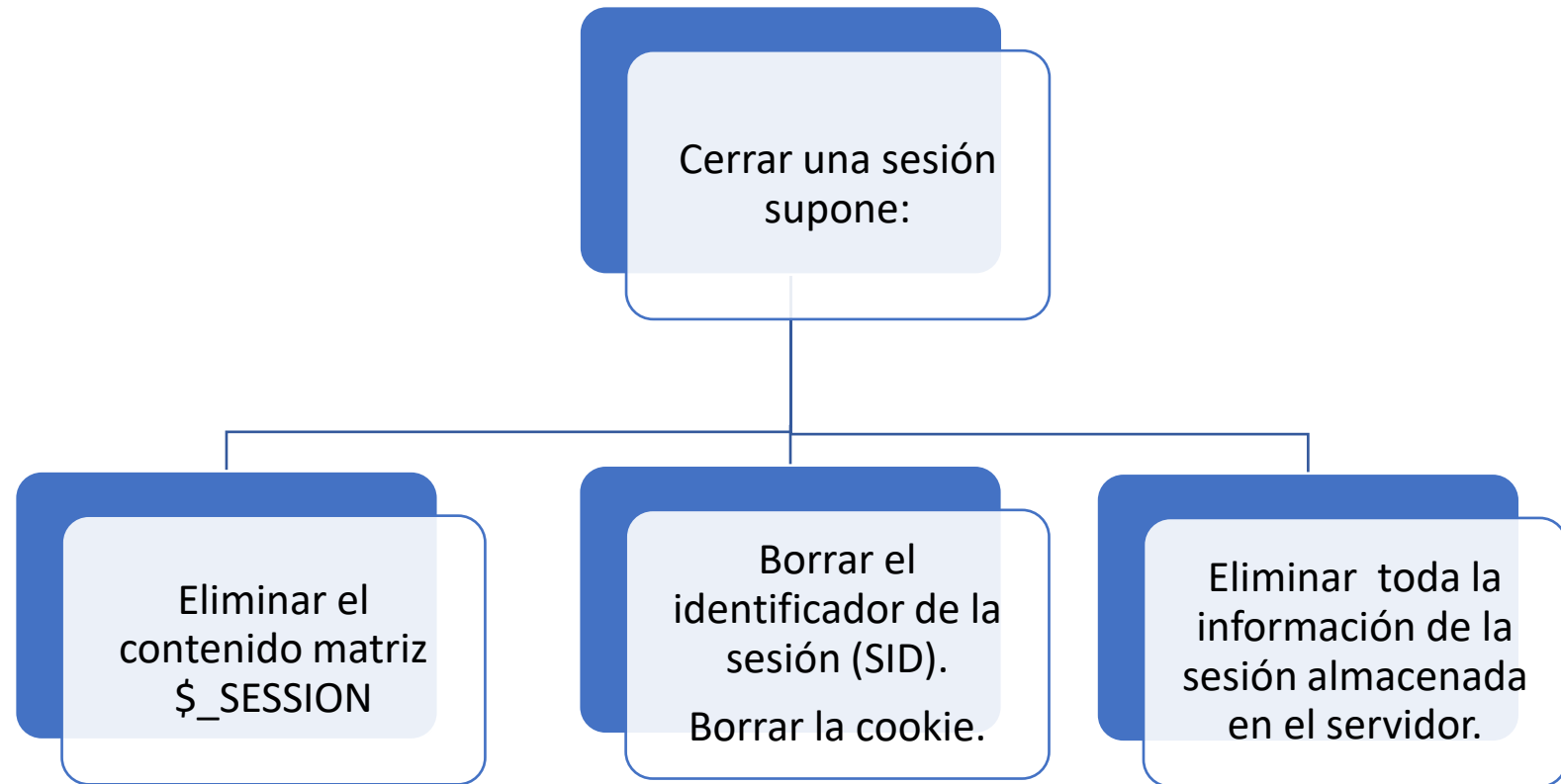
Si está activado el búfer de salida disponemos de un margen de 4096 bytes (según configuración).

Siempre que sea posible, se recomienda, **realizar el inicio de sesión al principio el script.**

¿Quién puede cerrar las sesiones?

1. El usuario: puede cerrar la sesión simplemente cerrando el navegador.
2. Un script: puede cerrar la sesión mediante la función `session_destroy()`.
3. El servidor: puede cerrar la sesión cuando ha pasado el tiempo indicado en segundos en la directiva `session.gc_maxlifetime`.

Cerrar la sesión



Destruir toda la información de una sesión

`session_destroy()`

Cuando se destruye una sesión, el script 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.

1. `session_destroy()`. Elimina completamente la información de la sesión **en el servidor**.
2. ¿Qué pasa con la cookie de sesión guardada en el cliente(navegador)?

Borrar todas las variables de sesión

En PHP tienes dos funciones para eliminar la información almacenada en la sesión:

- `$_SESSION = array();` **recomendado por el manual**
- `session_unset()`. Elimina todas las variables almacenadas en la sesión actual.

Nota del manual de PHP

Utilizar sólo `session_unset()` para códigos antiguos que no usan `$_SESSION`.

Borrar información asociada a la sesión

- **session_destroy()** borra todos los datos asociados con la sesión actual que se encuentran en el servidor, liberando la memoria utilizada por esos datos.
- No destruye ninguna de las variables globales asociadas con la sesión, ni destruye la cookie de sesión.
- Si se usa una cookie para propagar el id de sesión (comportamiento por defecto), **entonces la cookie de sesión se debe borrar.**
- setcookie() se debe usar para borrar la cookie de sesión en el cliente.
- Recuerda que para **borrar una cookie** debes utilizar la función setcookie() con el mismo nombre y **elegir un tiempo pasado para el campo expires**, ejemplo, time()-60

Destruir completamente una sesión

```
<?php
// 1-Iniciar la sesión
// 2-Borrar cookie de sesión
// 3-Eliminar datos del $_SESSION
// 4-Destruir la sesión (borrar archivo en el servidor).

session_start();

setcookie(session_name(), '', time()-42000);

$_SESSION = array();

session_destroy();

?>
```

Borrar variables
del array
\$_SESSION

Muy importante, nunca debes ejecutar:

`unset($_SESSION);`

Esta instrucción desactiva el registro de las variables de sesión a través de la matriz superglobal \$_SESSION.

Borrar una variable de sesión almacenada del array \$_SESSION

La forma correcta de hacerlo:

```
unset($_SESSION[ 'nombre' ] );
```

Eliminará esa variable del array de sesión.