

## SESIONES EN PHP

Las sesiones en PHP son un mecanismo que permite mantener la información del usuario a lo largo de múltiples páginas web durante una visita a un sitio web específico. Las sesiones tienen una duración que puede estar determinada por la configuración del servidor o por el tiempo de inactividad del usuario.

Cuando se inicia una sesión, PHP genera un identificador único de sesión (conocido como **session\_id**) para el usuario. Este identificador se almacena en una cookie en el navegador del usuario, o se puede transmitir a través de la URL si las cookies están desactivadas.

Existen varias funciones con las que manejamos este comportamiento:

### **session\_start()**

La función **session\_start()** en PHP realiza dos tareas principales:

1. **Iniciar una nueva sesión:** Cuando se llama por primera vez en un script PHP, **session\_start()** inicia una nueva sesión si no existe una sesión activa para el usuario (entendiendo como usuario nuestro navegador). Esto implica la generación de un nuevo **session\_id** único para identificar la sesión del usuario y la creación del array **\$\_SESSION** para almacenar datos de sesión.  
  
**\$\_SESSION** es un array superglobal en PHP que se utiliza para almacenar variables de sesión que están disponibles para todas las páginas del sitio web mientras dure la sesión del usuario. Es una forma de mantener datos específicos del usuario a lo largo de múltiples páginas web durante su visita al sitio. Los datos en **\$\_SESSION** se almacenan en el servidor y están asociados con el **session\_id** único generado para el usuario. Vamos a poder añadir en dicho array toda la información que nos resulte útil mantener hasta que cerremos la sesión en nuestro website.
2. **Reanudar una sesión existente:** Si ya existe una sesión activa para el usuario, **session\_start()** reanuda esa sesión. En este caso, PHP recupera el **session\_id** del navegador del usuario, ya sea a través de una cookie de sesión (si está habilitada) o mediante otros medios, y recupera los datos asociados con ese **session\_id** del almacenamiento de sesiones en el servidor. Esto incluye restaurar el contenido del array **\$\_SESSION** para que esté disponible en el script actual.

### **session\_destroy()**

**session\_destroy()** es una función de PHP que destruye todos los datos asociados con la sesión actual y finaliza la sesión del usuario. Esto incluye eliminar el **session\_id** y los datos almacenados en **\$\_SESSION**. Es útil para limpiar la sesión cuando el usuario cierra sesión, abandona el sitio o realiza ciertas acciones que requieren el reinicio de la sesión.

### **exit**

Es una función de PHP que detiene la ejecución del script de PHP inmediatamente en el punto donde se llama

### header('Location: ....')

Es una función de PHP que redirige al usuario a otra página web o URL dentro del mismo sitio web. Por ejemplo, si ponemos **header('Location: index.php')**, el usuario del navegador será redirigido a la página principal del sitio, **index.php**. Esta función es comúnmente utilizada después de realizar ciertas acciones, como iniciar o cerrar sesión, o después de completar un proceso para llevar al usuario a una página específica.

### La cookie de sesión: el enlace entre el navegador y el servidor

Una cookie es un pequeño archivo de texto que un sitio web guarda en el navegador del usuario para almacenar información temporal —como identificadores de sesión, preferencias o datos de navegación— y poder reconocerlo en visitas posteriores.

Por ejemplo, cuando buscas un vuelo, la web puede guardar una cookie con tus búsquedas recientes; si vuelves a entrar más tarde, el sitio reconoce tu interés y puede mostrarte precios ligeramente más altos para generar sensación de urgencia o aprovechar la demanda detectada.

Cuando PHP crea una sesión con `session_start()`, genera un identificador único (`session_id`) que sirve para reconocer al usuario en cada petición.

Para recordar este identificador, PHP crea automáticamente una cookie en el navegador, normalmente llamada PHPSESSID. -> **¡Investiga cómo acceder a ella en tu navegador!**

Esta cookie no guarda los datos de sesión, solo el identificador.

En cada nueva petición, el navegador envía la cookie al servidor, y PHP sabe así qué archivo de sesión recuperar.

Los datos que el desarrollador ha ido metiendo en la aplicación (por ejemplo, `$_SESSION['usuario'] = "Carlos"`) se almacenan en un archivo dentro de la carpeta especificada por `session.save_path`, por defecto `/tmp`.

### Duración de la sesión: cómo PHP decide cuándo termina

Una sesión tiene **dos componentes con vida útil diferente**:

Qué controla	Parámetro	Dónde actúa	Explicación práctica
Cuánto tiempo el navegador recuerda al usuario	<code>session.cookie_lifetime</code>	En el navegador	Es el tiempo que dura la cookie de sesión antes de borrarse. Si está en 0, la cookie se elimina al cerrar el navegador (comportamiento más común).
Cuánto tiempo el servidor conserva los datos	<code>session.gc_maxlifetime</code>	En el servidor	Es el tiempo máximo que el archivo de sesión permanece guardado. Si pasa ese tiempo sin actividad, el servidor elimina los datos de esa sesión.

Para cambiar estos valores hay varias opciones, veremos dos de ellas:

#### **Opción 1: En el propio código de la aplicación:**

Debe hacerse antes de `session_start()`, ya que PHP no puede cambiar los valores de sesión una vez iniciada.

```
<?php
ini_set('session.gc_maxlifetime', 1800);
ini_set('session.cookie_lifetime', 1800);
session_start();
?>
```

#### **Opción 2: En el archivo de configuración php.ini:**

Es la forma más correcta y global de hacerlo. Afecta a todas las aplicaciones PHP del servidor (a menos que alguna lo sobrescriba).

Busca y edita estas líneas:

```
session.gc_maxlifetime = 1800
session.cookie_lifetime = 1800
```

Luego reinicia el servidor web (Apache, Nginx, etc.) para aplicar los cambios.

#### **Ejemplo práctico**

```
session.cookie_lifetime = 0 // La cookie dura hasta cerrar el navegador
```

```
session.gc_maxlifetime = 1800 // El servidor guarda los datos 30 minutos
```

Esto significa que, mientras el usuario no cierre el navegador y regrese antes de 30 minutos, su sesión seguirá activa.

Si cierra el navegador, perderá la cookie y el servidor ya no sabrá quién es.

Si pasan más de 30 minutos sin actividad, el servidor eliminará los datos de sesión aunque la cookie aún exista.

#### **Consejo práctico**

En la mayoría de los casos es recomendable **mantener ambos valores iguales**, para que navegador y servidor tengan el mismo límite de tiempo:

### Visualizar el estado actual de la sesión

Para ver los datos de la sesión en tiempo real, puedes crear un archivo llamado, por ejemplo, ver\_session.php, dentro de tu proyecto:

```
<?php
session_start();

echo "<h3>Datos actuales de la sesión:</h3>";
echo "<pre>";
var_dump($_SESSION);
echo "</pre>";

echo "<p>ID de sesión: " . session_id() . "</p>";
echo "<p>Ruta de guardado: " . session_save_path() . "</p>";

echo "<h3>Cookies:</h3><pre>";
var_dump($_COOKIE);
echo "</pre>";
?>
```

Este script muestra:

- Las variables almacenadas en \$\_SESSION.
- El identificador único de sesión (session\_id).
- La carpeta del servidor donde PHP guarda los archivos de sesión.
- Las cookies activas en el navegador (incluida PHPSESSID).

Así podrás comprobar visualmente cómo PHP gestiona y conserva la información de sesión entre tus páginas