

# Análisis de Sistemas

**Materia:**  
Programación Web II

**Docente contenidista:** ROLDÁN, Hernán

**Revisión:** Coordinación

# Contenido

Más sobre las variables de sesión y cookies .....	3
Funcionamiento de las sesiones.....	4
Paso a Paso para definir un variable de sesión.....	5
Destrucción de una sesión .....	7
Funcionamiento de las cookies .....	9
Paso a Paso para definir un variable de cookie .....	10
Comparativa: ventajas y desventajas.....	12
Bibliografía .....	14
Para ampliar la información .....	14

# Clase 6



¡Te damos la bienvenida a la materia  
**Programación Web II!**

**En esta clase vamos a ver los siguientes temas:**

- Más sobre las variables de sesión y cookies
- Funcionamiento de las sesiones
- Paso a Paso para definir un variable de sesión
- Destrucción de una sesión
- Funcionamiento de las cookies
- Paso a Paso para definir un variable de cookie
- Comparativa: ventajas y desventajas

# Más sobre las variables de sesión y cookies

La anterior clase estuvimos trabajando con las variables súper globales, entre ellas `$_SESSION` y `$_COOKIE` y aprendiendo sobre sus usos, pero dada la importancia que tienen estas variables en el desarrollo web, en esta oportunidad continuaremos desarrollando más en profundidad el tema en cuestión, identificando entre otras cosas cuáles son las principales diferencias entre ambas como así también cuáles son las ventajas y desventajas de su uso.

Hasta ahora sabemos que la variable superglobal `$_SESSION` en PHP se utiliza para almacenar y acceder a datos del usuario a lo largo de su sesión en el sitio web.

Esta variable proporciona una forma de mantener información específica del usuario en todas las páginas del sitio mientras dure su sesión.

Cada usuario tiene su propia sesión, y los datos almacenados en `$_SESSION` están disponibles solo para ese usuario.

Por otra parte, sabemos que la variable superglobal `$_COOKIE` se utiliza para almacenar y acceder a datos en el lado del cliente.

Los datos se almacenan en forma de *cookies*, que son pequeños archivos de texto almacenados en el navegador del usuario.

Las cookies se envían automáticamente con cada solicitud al servidor, lo que permite mantener la persistencia de los datos entre múltiples visitas del usuario al sitio web.

# Funcionamiento de las sesiones

Una sesión es una forma de almacenar información (en variables) para ser utilizada en múltiples páginas.

Cuando un usuario visita un sitio web y comienza una nueva sesión, el servidor crea un identificador de sesión único y lo almacena en una cookie en el equipo del usuario.

El servidor también crea un archivo en el servidor para almacenar las variables de sesión para ese usuario.

El identificador de sesión en la cookie se utiliza para identificar la sesión del usuario en el servidor.

Cuando el usuario navega a una página diferente en el sitio web, el identificador de sesión se envía de vuelta al servidor en una cookie, y el servidor recupera las variables de sesión correspondientes para ese usuario.

*Las sesiones son útiles para almacenar datos temporales que son específicos de un único usuario y una sola sesión del navegador. Por ejemplo, podrías utilizar una sesión para almacenar los elementos del carrito de compras de un usuario o el estado de inicio de sesión.*

# Paso a Paso para definir un variable de sesión

## Paso 1: Iniciar una sesión.

- Antes de poder definir una variable de sesión, tenés que iniciar la sesión en tu script PHP utilizando la función `session_start()`. Agregá esta función al comienzo de tu script, antes de cualquier otra salida al navegador.

```
<?php

session_start();

?>
```

## Paso 2: Definir una variable de sesión.

- Una vez que la sesión está iniciada, podés definir una variable de sesión asignándole un valor. Podés utilizar el nombre de la variable que desees y asignarle el valor deseado. Por ejemplo, vamos a definir una variable de sesión llamada "usuario" con el valor "Bart Simpson":

```
<?php

session_start();

$_SESSION['usuario'] = 'Bart Simpson';

?>
```

### **Paso 3:** Acceder a la variable de sesión.

- Una vez que la variable de sesión está definida, podés acceder a su valor en cualquier parte de tu script PHP. Para acceder a una variable de sesión, utilizá el mismo nombre de variable que utilizaste al definirla. Por ejemplo, para acceder al valor de la variable de sesión "usuario", hacé lo siguiente:

```
<?php

session_start();

// Acceder al valor de la variable de sesión

$usuario = $_SESSION['usuario'];

// Utilizar el valor de la variable de sesión

echo 'Bienvenido, ' . $usuario;

?>
```

*Recordá que, para acceder correctamente a una variable de sesión, la sesión debe estar iniciada en cada página o script en el que quieras utilizarla.*



# Destrucción de una sesión

La destrucción de una sesión se debe realizar en diferentes momentos según el contexto y los requisitos de tu aplicación.

Aquí tenés algunos escenarios comunes en los que podés considerar destruir una sesión:

- **Cierre de sesión:** Cuando un usuario decide cerrar sesión, es necesario destruir la sesión actual. Esto implica eliminar todos los datos de sesión y finalizar la sesión del usuario. Al destruir la sesión, se cierra el acceso a los datos y funcionalidades protegidas por la sesión.
- **Inactividad del usuario:** Si tenés requisitos de seguridad específicos y deseas cerrar la sesión de un usuario después de cierto período de inactividad, podés establecer un tiempo de expiración para la sesión. Si el usuario no realiza ninguna acción durante ese período, la sesión se considerará inactiva y se destruirá automáticamente.
- **Cambio de contexto o privilegios:** En algunas aplicaciones, puede ser necesario destruir y crear una nueva sesión cuando un usuario cambia de contexto o adquiere nuevos privilegios. Por ejemplo, si un usuario pasa de ser un visitante anónimo a iniciar sesión como usuario registrado, podés destruir la sesión de invitado y crear una nueva sesión con los datos de usuario registrados.
- **Reinicio de estado de la aplicación:** En ciertos escenarios, puede ser necesario reiniciar el estado de la aplicación y asegurarse de que todas las sesiones se destruyan. Esto puede ocurrir, por ejemplo, cuando se reinicia el servidor o cuando se realiza un reinicio controlado de la aplicación.



El siguiente es un ejemplo simplificado de cómo destruir una sesión.

```
<?php

session_start();

// Destruir la sesión actual
session_destroy();

// Redirigir a otra página
header('Location: otra_pagina.php');

exit();

?>
```

*Siguiendo estas recomendaciones, podrás destruir las sesiones de manera segura y efectiva en tu aplicación PHP, garantizando la protección de los datos del usuario.*

# Funcionamiento de las cookies

Una cookie es una pequeña pieza de datos que se almacena en el navegador web de un usuario.

Se puede utilizar para almacenar información como preferencias de usuario o información de inicio de sesión.

Cuando un usuario visita un sitio web, el servidor puede enviar una cookie al navegador del usuario, el cual la almacenará.

Cuando el usuario regresa al sitio web más tarde, el servidor puede acceder a la cookie y utilizar la información almacenada en ella.

Las cookies se almacenan como archivos en el equipo del usuario y pueden permanecer allí durante un tiempo especificado, a menos que el usuario decida eliminarlas.

Las cookies tienen un tamaño limitado, típicamente de 4KB o menos.

*Las cookies son útiles para almacenar datos a más largo plazo que deben persistir a lo largo de múltiples sesiones. Por ejemplo, podrías utilizar una cookie para almacenar el idioma o el tema preferido de un usuario, de manera que el usuario no tenga que configurar sus preferencias cada vez que visita el sitio web.*

# Paso a Paso para definir un variable de cookie

**Paso 1:** Definir una variable cookie.

- Para definir una variable cookie en PHP, utilizá la función *setcookie()*.

Esta función toma varios parámetros, incluido el nombre de la cookie, el valor, y opciones adicionales como la expiración, la ruta y el dominio.

```
<?php

// Definir una variable cookie

setcookie('nombre', 'Bart Simpson', time() + 3600, '/');

?>
```

En este ejemplo, estamos definiendo una cookie llamada "nombre" con el valor "Bart Simpson".

Utilizamos la función `time()` para obtener el tiempo actual y luego le sumamos 3600 segundos (1 hora) para establecer la expiración de la cookie.

El parámetro  `'/'`  indica que la cookie está disponible en todo el sitio web.

## Paso 2: Acceder a la variable cookie.

- Una vez que la cookie está definida, podés acceder a su valor en cualquier parte de tu script PHP utilizando la variable `$_COOKIE`. Esta variable es un array asociativo que contiene todas las cookies disponibles.

```
<?php

// Acceder al valor de la variable cookie

$nombre = $_COOKIE['nombre'];

// Utilizar el valor de la variable cookie

echo 'Hola, ' . $nombre;

?>
```

En este ejemplo, estamos accediendo al valor de la cookie "nombre" utilizando la variable `$_COOKIE['nombre']` y luego lo utilizamos para imprimir un mensaje.

*Para acceder correctamente a una cookie, asegurate de que la cookie esté definida y disponible en la página o script en el que deseas utilizarla.*

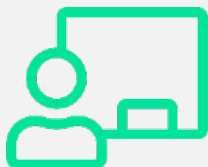
## Comparativa: ventajas y desventajas

	Sesiones	Cookies
<b>Alcance</b>	Solo accesibles dentro del script PHP que las creó.	Pueden ser accedidas por cualquier script en el dominio que las creó.
<b>Persistencia</b>	Almacenadas en la memoria del servidor y eliminadas cuando el usuario cierra su navegador.	Almacenadas como archivos en el equipo del usuario y pueden permanecer allí durante un tiempo especificado a menos que el usuario las elimine.
<b>Tamaño</b>	Pueden almacenar tanta información como pueda caber en el espacio de almacenamiento de sesión del usuario (generalmente varios megabytes).	Limitadas en tamaño, típicamente a 4KB o menos.
<b>Seguridad</b>	Más seguras porque se almacenan en el servidor y no son accesibles para el usuario.	Menos seguras porque se almacenan en el equipo del usuario y pueden ser accedidas o modificadas por el usuario.
<b>Casos de Uso</b>	Almacenamiento de datos temporales específicos de un único usuario y una sola sesión de navegador.	Almacenamiento de datos a más largo plazo que deben persistir a lo largo de múltiples sesiones.



Hemos llegado así al final de esta clase en la que vimos:

- Más sobre las variables de sesión y cookies
- Funcionamiento de las sesiones
- Paso a Paso para definir un variable de sesión
- Destrucción de una sesión
- Funcionamiento de las cookies
- Paso a Paso para definir un variable de cookie
- Comparativa: ventajas y desventajas



Te esperamos en la **clase en vivo** de esta semana.  
No olvides realizar el **desafío semanal**.

**¡Hasta la próxima clase!**

# Bibliografía

Documentación Oficial de PHP:

<https://www.php.net/manual/es/index.php>

W3Schools: PHP Tutorial:

<https://www.w3schools.com/php/default.asp>

Cabezas Granado, L., González Lozano, F., (2018). Desarrollo web con PHP y MySQL. Anaya Multimedia.

Heurtel, O., (2016). Desarrolle un sitio web dinámico e interactivo. Eni Ediciones.

Welling, L., Thompson, L., (2017). Desarrollo Web con PHP y MySQL. Quinta Edición. Editorial Anaya.

## Para ampliar la información

[PHP Sessions](#)

[PHP Cookies](#)

[PHP: The Right Way](#)

[Todo sobre PHP](#)

[PHP Programming Language Tutorial - Full Course](#)

[PHP Full Course for non-haters](#)

[CURSO de php desde cero](#)

[MDN: Introducción al lado Servidor](#)

[Guía de HTML](#)