



Nombre del docente:  
Parra Galaviz Ray Brunett

Materia:  
Desarrollo de aplicaciones web.

Alumno:  
Vázquez Jiménez Ángel Javier.

Actividad:  
Investigación - Súper Globales \$GET Y \$POST

Grado y grupo:  
3-D TM

Fecha de entrega:  
18 de octubre de 2024.

Carrera:  
TSU en TI área Desarrollo de Software Multiplataforma.

Algunas variables predefinidas en PHP son "superglobales", lo que significa que están disponibles en todos los ámbitos a lo largo del script. No es necesario emplear global \$variable; para acceder a ellas dentro de las funciones o métodos.

Las variables superglobals son:

\$GLOBALS

\$\_SERVER

**\$\_GET**

**\$\_POST**

\$\_FILES

\$\_COOKIE

\$\_SESSION

\$\_REQUEST

\$\_ENV

Las súper globales en PHP, como \$\_GET y \$\_POST, son variables predefinidas que permiten acceder a la información enviada a través de formularios web o parámetros en las URL. Son ampliamente utilizadas para manejar datos en aplicaciones web dinámicas.

## 1. \$\_GET:

Utilizada para acceder a los datos enviados en la URL (método GET).

Los datos se envían como parte de la URL, y se pueden ver en la barra de direcciones del navegador.

Sintaxis:

```
$nombre = $_GET['nombre'];
```

```
$edad = $_GET['edad'];
```

La variable \$\_GET se utiliza para obtener datos que se envían a través de la URL. Una de las principales ventajas de este método es su simplicidad y transparencia: los parámetros se agregan directamente en la barra de direcciones del navegador, lo que hace que sea muy fácil compartir enlaces con información preestablecida. Además, esto resulta útil para pruebas rápidas y para funciones como la paginación o los filtros de búsqueda en una aplicación web. Sin embargo, una de sus principales desventajas es la limitación en el tamaño de los datos que se pueden enviar, ya que las URL tienen una longitud máxima establecida por los navegadores.

Otra desventaja importante es la falta de privacidad, ya que los datos enviados a través de \$\_GET son visibles en la URL, lo que hace que no sea adecuado para transmitir información confidencial, como contraseñas o datos personales.

## 2. \$\_POST

Post se utiliza para acceder a los datos enviados mediante el cuerpo de una solicitud. Estos datos, NO son visibles en la URL y generalmente post se utiliza cuando se trabaja con formularios, ya que lo que hace es enviar datos como contraseñas, nombres, números, etc.

Sintaxis:

```
$usuario = $_POST['usuario'];
```

```
$clave = $_POST['clave'];
```

\$\_POST permite enviar datos de una forma más segura y eficiente. Al no incluir la información en la URL, los datos se transmiten en el cuerpo de la solicitud HTTP, lo que garantiza que no sean visibles en la barra de direcciones del navegador. Este método es ideal para el envío de grandes cantidades de datos o información sensible, como credenciales de acceso o detalles de una transacción. Además, no existe un límite práctico en el tamaño de los datos que pueden ser enviados mediante \$\_POST, lo que lo hace más flexible en comparación con \$\_GET. Sin embargo, una desventaja es que, al no poder verse los datos en la URL, no es tan fácil compartir un enlace o realizar pruebas simples como con \$\_GET. Además, si bien es más seguro que \$\_GET, la información enviada a través de \$\_POST no es completamente segura a menos que se utilice en conjunto con HTTPS, que cifra la transmisión de datos entre el navegador y el servidor.

Ejemplo utilizando post y get:

```
<form action="procesar.php" method="GET">
```

```
  Nombre: <input type="text" name="nombre">
```

```
  <input type="submit" value="Enviar">
```

```
</form>
```

```
<form action="procesar.php" method="POST">
```

```
  Usuario: <input type="text" name="usuario">
```

```
  Contraseña: <input type="password" name="clave">
```

```
  <input type="submit" value="Enviar">
```

```
</form>
```

## Referencias bibliográficas.

*PHP: Hypertext Preprocessor.* (s. f.).

<https://www.php.net/manual/es/reserved.variables.globals.php>

*CICS Transaction Server for z/OS 5.6.* (s. f.). <https://www.ibm.com/docs/es/cics-ts/5.6?topic=support-php-superglobals>

Soni, S. (2021, 29 marzo). Explicación de las variables superglobales en PHP: con una hoja de referencia. Code Envato Tuts+. <https://code.tutsplus.com/es/php-superglobals-explained-with-cheatsheet--cms-36598t>