



**Universidad Tecnológica de Tijuana**

***Tecnologías de la Información***

***Área Desarrollo de Software Multiplataforma.***

## **\$\_GET & \$\_POST**

***Aplicaciones Web***

***Ray Brunett Parra Galaviz***

### **Alumnos:**

Mayo Ramos Angel David

**Grupo 3C**

**Fecha de Entrega:**

17/10/24

En php existen distintas variables superglobales que permiten realizar distintas peticiones y procesos desde el lado del backend, el uso de estas variables es bastante útil a la hora de desarrollar una aplicación web.

Una variable superglobal es una variable incorporada en el lenguaje que está disponible en todos los ámbitos, es decir, puede ser accedida desde cualquier parte del script, ya sea en funciones, clases, o scripts independientes, sin necesidad de declarar su acceso global.

Entre las distintas variables superglobales que existen nos encontramos con dos especialmente útiles para manejar solicitudes de un usuario y recibirlas del lado del servidor: GET y POST. Su utilidad es similar pero la forma en la que funciona no, así que es importante identificar sus diferencias para saber cuál utilizar en función al contexto.

La variable GET nos permite manejar solicitudes ligándose a la URL de la página. Esto tiene ciertas ventajas como: facilita compartir enlaces y es útil para búsquedas. A pesar de sus ventajas esta cuenta con bastantes desventajas como su nula confidencialidad y la limitante que solo podemos enviar caracteres ASCII mediante este método, además de tener un límite de máximo 2,000 caracteres.

Ejemplo:

```
<form action = "/search.php" method = "GET">  
  Search Keyword: <input type = "text" name = "keyword" />  
  <input type = "submit" value="Search"/>  
</form>
```

La variable POST maneja las solicitudes de manera más sutil incluyendo en el cuerpo de la solicitud, esto permite que la información delicada no se comprometa y también tiene la ventaja de que podemos enviar información binaria con este metodo, asi como imagenes y audio. A pesar de que parezca que el método POST es mejor, no siempre es así, manejar información en URL es útil en ciertos momentos para el usuario ya que así el puede guardar filtros de búsqueda y visualizar lo que está pasando.

Ejemplo:

```
<form action="/login.php" method="POST">  
  Username: <input type="text" name="username" />  
  Password: <input type="password" name="password" />  
  <input type="submit" value="Login"/>  
</form>
```

En conclusión, es importante saber cuándo estamos manejando información sensible para utilizar el método **POST**, y cuándo podemos priorizar la conveniencia del usuario utilizando el método **GET**. Cada uno tiene su lugar según el contexto de la aplicación web

### References

IONOS. (2020, August 11). *GET vs. POST – Differences between the two HTTP*

*requests*. IONOS. Retrieved October 17, 2024, from

<https://www.ionos.com/digitalguide/websites/web-development/get-vs-post/>

Lázaro, D. (2018). *GET y POST en PHP*. Diego Lázaro. Retrieved October 17, 2024,

from <https://www.diego.com.es/get-y-post-en-php>

Soni, S. (2021, March 29). *Explicación de las variables superglobales en PHP: con*

*una hoja de referencia | Envato Tuts+*. Code. Retrieved October 17, 2024, from

<https://code.tutsplus.com/es/php-superglobals-explained-with-cheatsheet--cms-36598t>