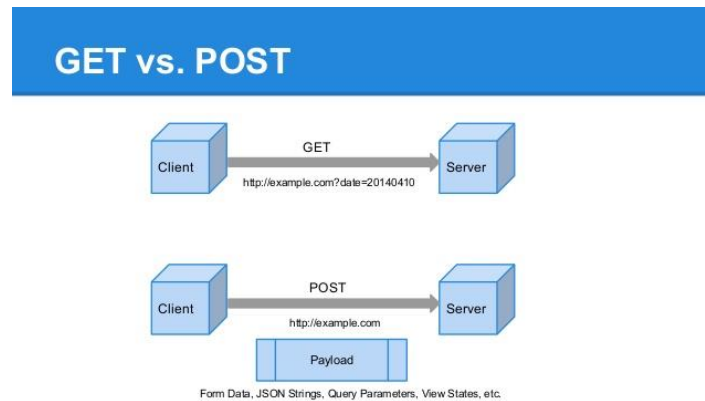


## GET vs. POST

Cuando trabajamos con formularios en HTML, los datos que los usuarios introducen se envían al servidor utilizando uno de dos métodos principales: **GET** o **POST**.

Ambos generan un array asociativo en PHP (por ejemplo, `array('clave1' => 'valor1', 'clave2' => 'valor2', ...)`), donde las claves corresponden a los nombres de los controles del formulario y los valores contienen los datos ingresados por el usuario.

Estos datos están disponibles en las superglobales `$_GET` y `$_POST`, que son arrays accesibles desde cualquier parte del script, sin importar su ámbito.



### GET

- Los datos del formulario se envían como parte de la URL en forma de una cadena de consulta (query string). Por ejemplo:  
`http://ejemplo.com/procesar.php?nombre=Juan&edad=30`
- Los datos enviados están disponibles en el array `$_GET`.
- **Características de GET:**
  1. La información enviada es visible públicamente en la URL.
  2. Existe un límite en la cantidad de datos que se pueden enviar (aproximadamente 2000 caracteres).
  3. Las URLs generadas son marcables y pueden compartirse fácilmente, lo que es útil para búsquedas, filtros o accesos directos.
- **¿Cuándo usarlo?**

Para datos no sensibles, como filtros o parámetros de búsqueda.

Cuando se necesita que la información esté disponible en la URL para su marcación o compartición.

- **Ejemplo práctico con GET:**

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

```
<label for="busqueda">Buscar:</label>
```

```
<input type="text" id="busqueda" name="q">
```

```
<button type="submit">Buscar</button>
```

```
</form>
```

Si el usuario introduce "PHP básico" en el campo de búsqueda y envía el formulario, la URL generada será:

<http://ejemplo.com/procesar.php?q=PHP+básico>

Esto es exactamente lo mismo que si escribieras manualmente el enlace con esos parámetros en un atributo href en HTML, como este ejemplo:

```
<a href="procesar.php?q=PHP+básico">Buscar "PHP básico"</a>
```

En el script procesar.php, podrías acceder al valor de la búsqueda así:

```
<?php
$busqueda = $_GET['q']; // Contendrá 'PHP básico'
echo "Resultados para: " . $busqueda;
?>
```

## POST

- Los datos del formulario se envían en el **cuerpo de la solicitud HTTP**, no en la URL.
- Los datos enviados están disponibles en el array `$_POST`.
- **Características:**
  1. La información enviada es **invisible públicamente**, ya que no aparece en la URL.
  2. No hay un límite práctico en la cantidad de datos que se pueden enviar.
  3. Permite funcionalidades avanzadas, como la carga de archivos mediante multipart/form-data.

- **¿Cuándo usarlo?**

Para datos sensibles, como contraseñas o información personal.

Cuando se envían grandes cantidades de datos o archivos al servidor.

Cuando no se necesita que los datos sean visibles en la URL.

- **Ejemplo práctico con POST:**

```
<form action="procesar.php" method="POST">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
```

```
<label for="mensaje">Mensaje:</label>

<textarea id="mensaje" name="mensaje"></textarea>

<button type="submit">Enviar</button>

</form>
```

Los datos no aparecerán en la URL, pero estarán disponibles en el servidor. En el script procesar.php, podrías manejarlos así:

```
<?php

$nombre = $_POST['nombre'];

$mensaje = $_POST['mensaje'];

echo "Gracias, " . $nombre . ", por tu mensaje: " . $mensaje;

?>
```