

# DAWS – 2º

## PHP Métodos de entrada

JESÚS MOLINA HERNÁNDEZ

[jmolina@florida-uni.es](mailto:jmolina@florida-uni.es)

# Índice

1. **Introducción – métodos de entrada**
2. **GET - ¿Qué es una petición HTTP GET?**
3. **GET - Estructura de peticiones GET en PHP**
4. **GET - Gestión de la petición GET en PHP**
5. **POST - ¿Qué es una petición HTTP POST?**
6. **POST - Manejo de peticiones POST en PHP**
7. **POST - Enviar datos con formularios en HTML**

# Introducción – métodos de entrada

## ¿Qué son los métodos de entrada en PHP?

- Los **métodos de entrada** se refieren a las formas en que los datos son enviados desde el cliente (navegador) al servidor web.
- En PHP, las peticiones más comunes son enviadas a través de formularios HTML y pueden ser procesadas usando **métodos HTTP**.

## Principales métodos HTTP usados en PHP:

- **GET**: Envía los datos en la URL (query string). Usado para obtener información del servidor.
- **POST**: Envía los datos en el cuerpo de la solicitud. Usado para enviar datos al servidor, como formularios o archivos.

Hay otros métodos pero apenas se usan.

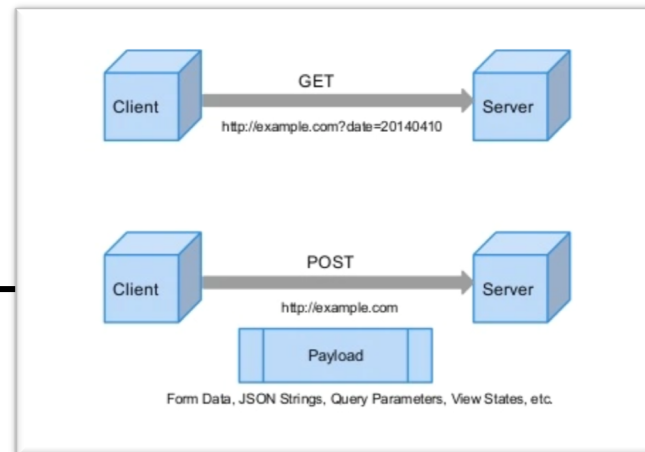
# Opciones de Métodos: GET vs POST

## GET

- **Forma de envío:** Datos se envían en la URL.
- **Casos de uso:** Navegación, búsquedas, obtener recursos (leer datos).
- **Ventajas:** Fácil de compartir enlaces y marcar páginas.
- **Desventajas:** Límite en el tamaño de los datos y menos seguro (visible en la URL).

## POST

- **Forma de envío:** Datos se envían en el cuerpo de la petición HTTP.
- **Casos de uso:** Formularios con información sensible (login, registros), subir archivos.
- **Ventajas:** Sin límite de tamaño para los datos, más seguro (no visible en la URL).
- **Desventajas:** No se puede compartir la URL directamente con los datos enviados.



# ¿Qué es una petición HTTP GET?

- Una **petición HTTP GET** es un método para enviar datos desde el cliente (navegador) al servidor.
- Cuando envías un formulario con el método GET, los datos se adjuntan a la URL como una **cadena de consulta** (query string).

## Características de GET:

- **Los datos se ven en la URL**, lo que los hace más accesibles y fáciles de compartir, pero menos seguros para información sensible.
- **Tamaño limitado**: La longitud máxima de la URL es limitada, lo que restringe la cantidad de datos que se pueden enviar.
- **Usado típicamente**: Para páginas que no alteran el estado del servidor, como consultas de búsqueda o visualización de información.

# Estructura de peticiones GET en PHP

- Estructura de la petición GET:

[www.paginaweb.com/index.php?nombre=jesus&apellido=molina](http://www.paginaweb.com/index.php?nombre=jesus&apellido=molina)

- La petición se realiza mediante la URL del navegador: Tras la URL del script añadimos el símbolo '?'
- Luego pasaremos tantas variables como necesitemos encadenándolas con el símbolo '&'
- Y cada variable que adjuntemos lo haremos mediante un par que está compuesto por el nombre de la variable, el símbolo '=' y posteriormente el valor que tenga esa variable.

# Gestión de la petición GET en PHP

- PHP procesa los datos enviados por GET utilizando la variable superglobal \$\_GET o \$\_REQUEST.
- Ejemplo básico

```
$variable1 = $_GET['nombre'];  
$variable2 = $_GET['apellido'];  
  
echo $variable1 . " " . $variable2 . "<br>";  
echo "Hola " . $variable1 . " " . $variable2;
```

# ¿Qué es una petición HTTP POST?

- Una **petición HTTP POST** es un método para enviar datos desde el cliente (navegador) al servidor.
- A diferencia de GET, que envía datos en la URL, POST envía los datos en el cuerpo de la solicitud.
- Se usa comúnmente en formularios para enviar información sensible o grandes cantidades de datos.



# Manejo de peticiones POST en PHP

- PHP procesa los datos enviados por POST utilizando la variable superglobal `$_POST`.
- Esta variable es un array asociativo que contiene los datos enviados en el cuerpo de la solicitud.
- Ejemplo básico

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    $nombre = $_POST['nombre'];  
    echo "Hola, $nombre";  
}
```

# Enviar datos con formularios en HTML

- Los formularios HTML permiten enviar datos al servidor utilizando el método POST.
- Campos como <input>, <textarea>, y <select> se usan para recoger información.
- Especifica el atributo method="POST" para asegurarte que los datos se envíen de forma segura.

## Ejemplo

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