# Investigacion de REQUEST

## Metodo Get y Post

### Que son?:

El método GET como POST son protocolo HTPP el cual envía al servidor como petición (request) y recibe una respuesta a dicha solicitud (response).

### Como funciona?:

Cuando un usuario rellena un formulario en una página web los datos hay que enviarlos de alguna manera. Vamos a considerar las dos formas de envío de datos posibles: usando el método POST o usando el método GET.

Por ejemplo: <form action="http://www.aprenderaprogramar.com/prog/newuser" method ="get">

será el envío de los datos del formulario a la url especificada usando el método get.

Un ejemplo de uso del método post sería este:

<form action="http://www.aprenderaprogramar.com/prog/newuser" method ="post">

En el caso de un envío de datos usando el método POST, aunque estos datos también serán enviados, de una forma que podemos denominar “oculta”, no los podremos ver en la URL. Para poder recuperar los valores de los campos en el caso de un envío con el método POST necesitaríamos otras herramientas.

### Diferencias:

Por tipos de datos; el GET solo acepta caracteres ASCII mientras que el método POST acepta ASCII y datos binarios.

Por longitud de datos; el GET tiene un limite al máximo del url de 2048 caracteres mientras que el POST es ilimitado.

Por datos codificados; el GET añade datos codificados al URI además de servir para recuperar datos, en cambio el POST añade los datos al cuerpo y no al URI, sirviendo para almacenar y actualizar datos.

## Metodo PATCH

## Que es?:

Los parches de Composer permiten realizar modificaciones a librerías PHP para corregir posibles problemas de código o adaptar su funcionalidad a las necesidades de un proyecto concreto.  Es un archivo de texto que consta de una lista de diferencias y se produce ejecutando el programa de diferencias relacionado con el archivo original y actualizado como argumentos.

### Como funciona?:

#### Método 1:

Crear un parche con PHP Storm

Si utilizas el software **PHP Storm**, una de las formas más sencillas de generar parches es utilizar la funcionalidad del propio IDE, mediante los siguientes pasos:

1. Realizar los cambios deseados sobre el fichero original.

2. Acceder al historial de cambios desde File > Local history > Show history…

3. Seleccionar el item que incluye las modificaciones que se desean incluir en el parche y seleccionar la opción “Create patch” desde el menú contextual.

#### Metodo 2:

Crear un parche con Symplify Vendor Patches

Otro método sencillo de generar parches es utilizar la librería **[Symplify Vendor Patches](https://github.com/symplify/vendor-patches" \t "_blank)**, que permite generar parches para las librerías del vendor de un proyecto (librerías obtenidas desde Composer) mediante los siguientes pasos:

1. Instalar la librería**Symplify Vendor Patches**:

composer require symplify/vendor-patches --dev

Copy

2. Crear una copia del fichero a parchear, incluyendo el sufijo “.old” en su nombre.  
Por ejemplo: ***Product.php*** > ***Product.php.old***

3. Realizar las modificaciones necesarias en el fichero original.

4. Crear el parche lanzando el siguiente comando:

vendor/bin/vendor-patches generate

## Metodo DELETE

### Que es?:

  permite la eliminación condicional de los registros.

### Como funciona?:

La sentencia DELETE se suele usar para borrar unos registros de una tabla que cumplen una o varias condiones. Vamos a utilizar una sintaxis de este tipo:

mysqli\_query($link, " DELETE FROM nombreTabla WHERE columna (>, <, =, etc.) valorEspecificado ");

la cláusula WHERE que nos sirve para indicar una condición.

## Metodo PUT

### Que es?:

La **petición HTTP PUT** crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición.

La diferencia entre el método PUT y el método [POST](https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST) es que PUT es un método idempotente.

### Como Funciona:

Para enviar una solicitud PUT con PHP, utilice el método HTTP PUT y proporcione los datos en el cuerpo del mensaje PUT. El método de solicitud HTTP PUT crea un nuevo recurso o reemplaza un recurso existente en el servidor. El encabezado de solicitud Content-Type especifica el tipo de datos en el cuerpo del mensaje de solicitud PUT, y el encabezado de solicitud Content-Length especifica el tamaño de los datos en el cuerpo de solicitud PUT. A diferencia de una solicitud PATCH, que reemplaza parcialmente un recurso en el servidor y puede contener solo datos parciales, una solicitud PUT sobrescribe completamente el recurso y debe tener una copia completa de los datos. En este Ejemplo de Solicitud PHP PUT, enviamos JSON a la URL de eco de ReqBin. Haga clic en Enviar para ejecutar la solicitud PHP PUT en línea y vea los resultados. El código PHP se generó automáticamente para el ejemplo de solicitud PUT.