

# PRACTICA 9



PHP

***FORMULARIOS***

php

*Por Ramon Abramo*

## EJERCICIO 1 FORMULARIO CON GET

Realizamos el siguiente formulario en el ejercicio1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <form action="ejercicio1Destino.php" method="get">
      Nombre: <input type="text" name="nombre"><br>
      Email: <input type="text" name="email"><br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```

Nombre:

Email:

Como ya sabemos cuando un usuario rellena los datos y hace click en enviar, los datos del formulario se envían para ser procesados al archivo ejercicio1Destino.php. Los datos se envían a través de HTTP con el método HTTP GET (viajan en la URL)

Vamos a realizar el ejercicio1Destino.php que va a mostrar los datos recibidos.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    Hola <?php echo $_GET["nombre"]; ?><br>
    Tu email es: <?php echo $_GET["email"]; ?>
  </body>
</html>
```

Nombre: 

Email: 


Hola Ramon

Tu email es: ramon@alpe.es

Podemos mostrar los datos utilizando el método recortado de Php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <!--
      Metodo largo
    -->
    Hola <?php echo $_GET["nombre"]; ?><br>
    Tu email es: <?php echo $_GET["email"]; ?>

    <br><br>
    <!--
      Metodo recortado
    -->
    Hola <?= $_GET["nombre"] ?><br>
    Tu email es: <?= $_GET["email"] ?>

  </body>
</html>
```

## EJERCICIO 2 FORMULARIO CON POST

Como podemos ver cuando el formulario envia los datos al servidor podemos utilizar las variables superglobales \$\_GET, \$\_POST.

\$\_GET y \$\_POST son variables superglobales que forman arrays de keys y values, donde los keys son los nombres del formulario (atributo "name") y los values son los datos de entrada de los usuarios. El hecho de que sean superglobals hace que sean accesibles desde el script independientemente del ámbito.

- \$\_GET es un array de variables que se pasan al script a través de los parámetros de URL. La información que se envía es visible para todo el mundo y tiene limitada la cantidad de información que se puede enviar a 2000 caracteres. Las URLs con los datos enviados pueden guardarse en marcadores, lo que puede ser útil en ciertos casos. \$\_GET sólo se emplea para el envío de información reducida y no sensible.

Nombre:

Email:

Hola Ramon

Tu email es: ramon@alpe.es

```
/practica8/ejercicio1Destino.php?nombre=Ramon&email=ramon%40alpe.es
```

- \$\_POST es un array de variables que se pasan al script a través del método HTTP POST. La información que se envía no es visible para los demás ya que los nombres y variables van embebidas en el body del HTTP request. No tiene límites en la cantidad de información a enviar. Soporta datos de entrada multi-part binary a la hora de subir archivos al servidor (lo que permite subir múltiples archivos a la vez).

Vamos a crear el mismo formulario, pero utilizando la etiqueta label para las etiquetas.

```
<form action="ejercicio2Destino.php" method="post">
  <div>
    <label for="nombre">
      Nombre:
    </label>
    <input type="text" name="nombre" id="nombre" placeholder="Introduce tu nombre">
  </div>
  <br>
  <div>
    <label for="email">
      Email:
    </label>
    <input type="email" id="correo" name="email" placeholder="Introduce tu correo">
  </div>
  <br>
  <div>
    <input type="submit" value="Enviar">
  </div>
</form>
```

Nombre:

Email:

Enviar

Vamos a realizar la página ejercicio2Destino.php que recibe los datos.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    Hola <?= $_POST["nombre"] ?><br>
    Tu email es: <?= $_POST["email"] ?>
  </body>
</html>
```

Probamos el funcionamiento

Nombre:

Email:

Enviar

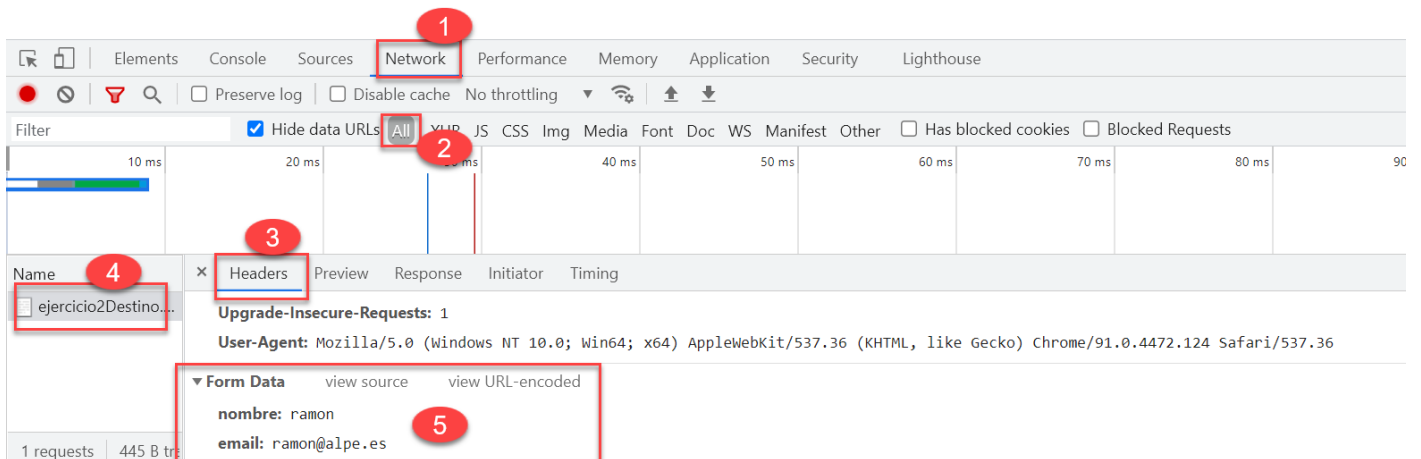
[ejercicio2.php](#)

Hola ramon  
Tu email es: ramon@alpe.es

[ejercicio2Destino.php](#)

Podemos ver los datos enviados por POST desde las herramientas de desarrolladores (F12).





### EJERCICIO 3 FORMULARIO Y REQUEST

`$_REQUEST` es un array asociativo que reagrupa los métodos `$_GET`, `$_POST`. Esta tabla también es superglobal; por lo tanto, es accesible en todos los scripts PHP.

Si los datos se envían con POST, GET se pueden recuperar con `$_REQUEST`.

Creamos el siguiente formulario

Nombre:

Email:

Telefono:

**ENVIAR**

Le he realizado igual con la diferencia que he metido el elemento de HTML5 button.

```
<form action="ejercicio3Destino.php" method="get">
  <div>
    <label for="nombre">
      Nombre:
    </label>
    <input type="text" name="nombre" id="nombre" placeholder="Introduce tu nombre">
  </div>
  <br>
  <div>
    <label for="email">
      Email:
    </label>
    <input type="email" id="correo" name="email" placeholder="Introduce tu correo">
  </div>
  <br>
  <div>
    <label for="telefono">
      Telefono:
    </label>
    <input type="text" id="telefono" name="telefono" placeholder="Introduce tu telefono">
  </div>
  <br>
  <div>
    <button>ENVIAR</button>
  </div>
</form>
```

Cuando se quiere probar que datos me llegan del formulario es muy usado la función var\_dump para realizar pruebas.

```
<?php
    var_dump($_GET);
    var_dump($_REQUEST);
?>
```



Nombre:

Email:

Telefono:

ENVIAR

```
C:\xampp\htdocs\2021\poo\php\practicas\practica8\ejercicio3Destino.php:9:
array (size=3)
```

```
  'nombre' => string 'Ramon' (length=5)
  'email' => string 'ramon@alpe.es' (length=13)
  'telefono' => string '614565656' (length=9)
```

```
C:\xampp\htdocs\2021\poo\php\practicas\practica8\ejercicio3Destino.php:10:
array (size=3)
```

```
  'nombre' => string 'Ramon' (length=5)
  'email' => string 'ramon@alpe.es' (length=13)
  'telefono' => string '614565656' (length=9)
```

El método REQUEST no es un método de envío propiamente dicho, es decir, no podemos poner en un formulario HTML que el method sea REQUEST. Con la variable \$\_REQUEST recuperaremos los datos de los formularios enviados tanto por GET como por POST.

La desventaja principal es que no queda claro de donde procede esa información si por GET o por POST y eso podría llevarnos a confusiones o problemas de seguridad.

## EJERCICIO 4 FORMULARIO CON LOS DOS METODOS

Vamos a crear un formulario con dos botones. Cada uno de ellos puede enviar los datos utilizando un método diferente.

```
<form action="ejercicio4Destino.php">
  <div>
    <label for="nombre">
      Nombre:
    </label>
    <input type="text" name="nombre" id="nombre" placeholder="Introduce tu nombre">
  </div>
  <br>
  <div>
    <label for="email">
      Email:
    </label>
    <input type="email" id="correo" name="email" placeholder="Introduce tu correo">
  </div>
  <br>
  <div>
    <button formmethod="GET">Enviar por GET</button>
    <button formmethod="POST">Enviar por POST</button>
  </div>
</form>
```

Nombre:

Email:

La página que recoge los datos :

```
<?php
echo "Enviado por GET:";
var_dump($_GET);
echo "Enviado por POST:";
var_dump($_POST);
?>
```

Enviado por GET:

```
C:\xampp\htdocs\2021\poo\php\practicas\practica8\ejercicio4Destino.php:11:
array (size=2)
  'nombre' => string 'ramon' (length=5)
  'email' => string 'ramon@alpe.es' (length=13)
```

HEMOS PULSADO ENVIAR  
POR GET

Enviado por POST

```
C:\xampp\htdocs\2021\poo\php\practicas\practica8\ejercicio4Destino.php:13:
array (size=0)
  empty
```

## EJERCICIO 5 FORMULARIO

Crear un formulario que nos pida:

- Nombre

- Apellidos
- Dirección
- Código Postal
- Teléfono
- Correo

Nombre

Apellidos

Codigo Postal

Telefono

Correo

Enviar

---

Colocar todos los campos como requeridos utilizando el atributo required en cada input.

---

En otra página php realizar la recogida de datos y mostrarlo en una serie de divs:

Nombre  Nombre: ramon

Apellidos  Apellidos: abramo

Codigo Postal  Codigo Postal: 39009

Telefono  Telefono: 123123123

Correo  Correo: ramon@alpe.es