



Escuela: Universidad Tecnológica de Tijuana

Carrera: TI Desarrollo de Software Multiplataforma

Asignatura: Aplicaciones Web

Trabajo: Superglobales GET & POST

Integrantes:

- Miramontes Gutierrez Rodolfo Manuel
-

Docente: Ray Brunnet Parra Galaviz

Grupo: 3C

Los métodos GET y POST se utilizan comúnmente en el desarrollo web para enviar datos de formulario a un servidor. Ambos métodos sirven para enviar datos, pero difieren en cómo el servidor transmite y maneja los datos. Por tradición los get son para obtener información y los posts para enviar información.

El método GET se usa para recuperar datos del servidor agregando los datos del formulario a la URL como parámetros de consulta. Cuando se envía un formulario con el método GET, los datos del formulario se codifican y se agregan a la URL como pares clave-valor. Esto hace que los datos sean visibles en la URL, que los navegadores pueden marcar, compartir o almacenar en caché. El método GET es adecuado para situaciones en las que los datos del formulario no son confidenciales o cuando el envío del formulario es idempotente, lo que significa que no causa ningún efecto secundario en el servidor. Por ejemplo, un formulario de búsqueda que recupera resultados de búsqueda puede usar el método GET.

Por otro lado, el método POST se usa para enviar datos al servidor en el cuerpo de la solicitud HTTP. Cuando se envía un formulario mediante el método POST, los datos del formulario se incluyen en el cuerpo de la solicitud, que no es visible en la URL. Esto proporciona una forma más segura de transmitir datos confidenciales, como contraseñas o información de tarjetas de crédito, ya que no se exponen en la URL ni los navegadores los almacenan en caché. El método POST también es adecuado para situaciones en las que el envío del formulario provoca efectos secundarios en el servidor, como la creación de un nuevo registro en una base de datos o la actualización de datos existentes.

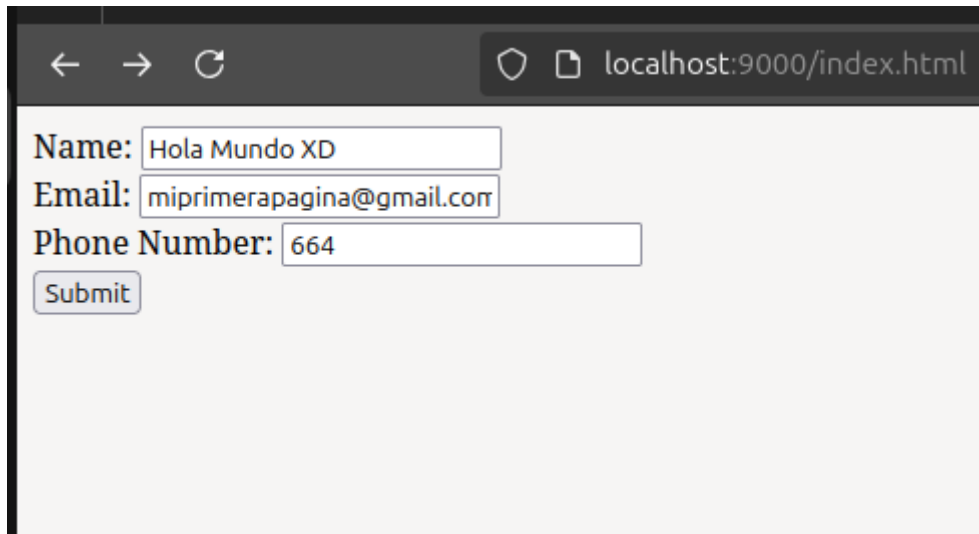
Implementación de el método POST

La utilización de estos dos métodos es bastante similar, prácticamente igual, por eso para este ejemplo utilicé el método POST, enviando los datos que se ingresan en un index.html hacía otro archivo llamado index.php, que es el que recibe esta información.

```
1 <form action="index.php" method="post">
2   <div>
3     <label for="name">Name: </label>
4     <input type="text" id="name" name="name">
5   </div>
6
7   <div>
8     <label for="email">Email: </label>
9     <input type="text" id="email" name="email">
10  </div>
11
12  <div>
13    <label for="phone">Phone Number: </label>
14    <input type="text" id="phone" name="phone">
15  </div>
16
17  <input type="submit" value="Submit">
18 </form>
```

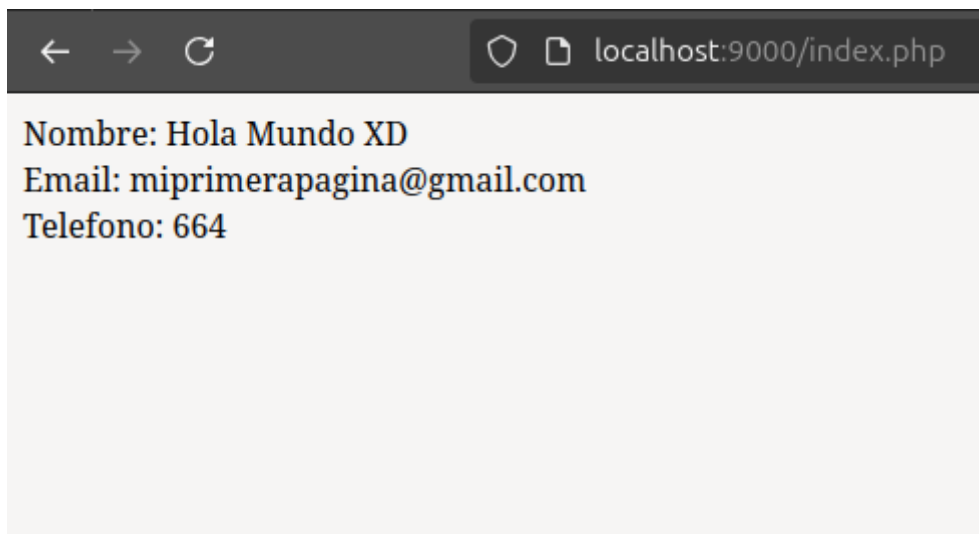
```
1 <?php
2
3   $name = ($_POST['name']);
4   $email = ($_POST['email']);
5   $phone = ($_POST['phone']);
6
7   echo "Nombre: " . $name . "<br />";
8   echo "Email: " . $email . "<br />";
9   echo "Telefono: " . $phone . "<br />"
10
11 ?>
```

Así es como se vería desde un navegador el formulario HTML:



A screenshot of a web browser window. The address bar shows 'localhost:9000/index.html'. The page content includes a form with three input fields and a submit button. The first field is labeled 'Name:' and contains the text 'Hola Mundo XD'. The second field is labeled 'Email:' and contains 'miprimera pagina@gmail.com'. The third field is labeled 'Phone Number:' and contains '664'. Below the fields is a button labeled 'Submit'.

Y así se vería en PHP, recibiendo la información que ingresemos después de apretar el botón submit:



A screenshot of a web browser window. The address bar shows 'localhost:9000/index.php'. The page content displays the data submitted in the form: 'Nombre: Hola Mundo XD', 'Email: miprimera pagina@gmail.com', and 'Telefono: 664'.

