

## SUMÁRIO

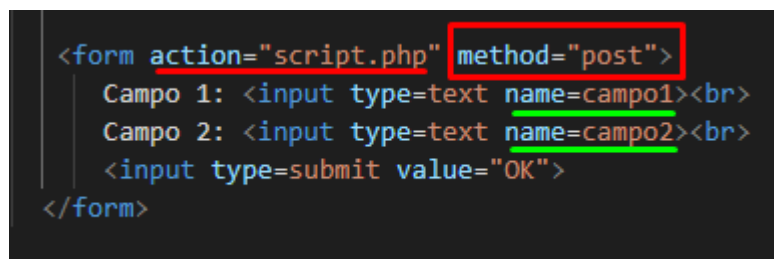
<b>1. INTRODUÇÃO AO FORMULÁRIO PHP .....</b>	<b>1</b>
<b>2. RECUPERANDO OS DADOS .....</b>	<b>1</b>
<b>3. BOAS PRÁTICAS: SEGURANÇA DO SISTEMA .....</b>	<b>2</b>
<b>3.1. SEGURANÇA NA RECUPERAÇÃO DOS DADOS .....</b>	<b>2</b>
<b>4. FONTES.....</b>	<b>3</b>

## 1. INTRODUÇÃO AO FORMULÁRIO PHP

Um formulário HTML é apenas uma interface de entrada de dados “enfeitada”, para onde os usuários poderão inserir informações que serão interpretados de alguma maneira por algum script do lado do servidor. E no nosso caso, esse script é um script PHP.

## 2. RECUPERANDO OS DADOS

Existem 2 métodos como as informações podem ser passadas: **GET e POST**. O recomendável sempre, para todos os formulários é usar o método POST, onde os dados enviados não são visíveis nas URLs, ocultando possíveis importantes informações e permitindo o envio de longas informações. O GET é totalmente o contrário disso.



```
<form action="script.php" method="post">
  Campo 1: <input type="text" name="campo1"><br>
  Campo 2: <input type="text" name="campo2"><br>
  <input type="submit" value="OK">
</form>
```

Na imagem acima, o **campo action** é utilizado para definir o local de envio dos dados, o **campo method** é utilizado para definir o método de envio dos dados, e o **campo name** utilizado para definir o nome do espaço alocado, para cada valor.

Assim, quando clicado em submit, os dados serão enviados para a aplicação, existente no servidor, “script.php”, que será carregada. Além disso, nesta – na aplicação script.php-, a URL visível na barra de endereço do navegador não terá, como no caso do GET, a descrição dos valores recebidos, embora tenham sido enviados por meio da própria.

Por fim, para que os valores recebidos sejam recuperados no código, deve-se utilizar além do nome respectivo, definido em name, na aplicação HTML mensageira, funcionalidades próprias do PHP para este fim. Para o método POST, deve-se escrever **\$\_POST['nome']**, e para o GET deve-se utilizar **\$\_GET["nome"]**.

Estas variáveis para recuperação dos dados enviados via HTTP, apesar de garantirem acesso aos valores, em qualquer operação, são atribuídas a outras criadas e definidas na própria aplicação, por via de regra. Por exemplo, **\$nome = \$\_POST['nome']**.

A partir do entendimento desta introdução básica, se pode entender e derivar a sua aplicação nas mais diversas situações. A exemplo, ao invés de um só dado, poderiam ter sido

enviados diversos, e assim, seriam recuperados, um por um, pela adequada variável de recuperação HTTP, e atribuídos a variáveis criadas para o resto da operacionalidade do sistema.

### 3. BOAS PRÁTICAS: SEGURANÇA DO SISTEMA

É possível, e bastante provável, que alguns usuários não insiram os dados esperados pelos desenvolvedores, nos formulários, ou que tentem burlar, de alguma forma, o sistema a partir da exploração de falhas possíveis.

Neste sentido, torna-se clara a necessidade de se criar um sistema robusto no que tange à segurança, para que saiba lidar com as mais diversas possibilidades que fujam do “normalmente esperado”.

#### 3.1. SEGURANÇA NA RECUPERAÇÃO DOS DADOS

Para evitar ataques do tipo XSS e SQL Injection, na recuperação de dados externos, é imprescindível que sejam verificados e validados. Para esta tarefa, pode-se usar as validações que vêm nativamente no PHP, como Isset e Empty. Veja abaixo:

```
if(isset($_GET["username"]) && !empty($_GET["username"]) && isset($_GET["senha"]) && !empty($_GET["senha"])){  
    //Se as variáveis forem diferentes de null e de vazio
```

Existe uma grande variedade de validações disponível nativamente no PHP, pra diversas situações. Abaixo, fontes com mais material sobre:

<https://imasters.com.br/back-end/seguranca-em-aplicacoes-web-com-php>

<https://canaltech.com.br/software/Blinde-sua-aplicacao-web-PHP-com-estas-15-dicas-de-seguranca/>

<http://www.devwilliam.com.br/php/aprenda-como-validar-dados-com-php>

[http://www.w3bai.com/pt/php/php\\_filter.html](http://www.w3bai.com/pt/php/php_filter.html)

#### **4. FONTES**

<https://www.devmedia.com.br/php-forms-manipulando-dados-de-formularios/29392>

<https://www.homehost.com.br/blog/tutoriais/php/php-foreach/>

