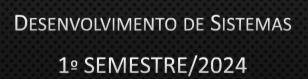
POST

Prof. Fábio Kravetz







GET X POST

- GET é o padrão de envio de dados
 - Caso não seja especificado o método GET é assumido pelo formulário HTML;
- Os dados são enviados juntamente com o nome da página para o envio de dados;
- http://localhost/cursophp/projeto01/index.php?total=0,50&opera cao=Divisão
 - >? representa o início da cadeia de variáveis;
 - > & identifica o início de uma nova variável;
 - > = separa as variáveis dos seus respectivos valores.



GET X POST

GET

- Método GET possui limitação de caracteres;
- ➤ Os dados enviados, via formulário, são visíveis na barra de endereço do navegador;
- > Pode ser utilizado para passagem de parâmetros por link;

POST

- Método mais seguro para envio de informações;
- Não há restrição ao tamanho dos dados e não são exibidos para o usuário;
- ➤ Realiza o envio de outros tipos de dados, não aceitos pelo GET, como imagens ou outros arquivos.



POST

Exemplo:

```
if(isset($_POST["nome"])){
    echo $_POST["nome"]." | ";
    echo $_POST["sobrenome"];
<form method="post">
    <label>Nome:</label>
    <input type="text" name="nome" ><br>
    <label>Sobreome:</label>
    <input type="text" name="sobrenome" ><br>
    <input type="submit" value="Salvar"><br>
</form>
```



- Ao permitir que os usuários insiram dados, é necessário validar os dados antes de usá-los;
- A validação pode ocorrer via HTML ou com a linguagem PHP;
- Existem formas de burlar validações feitas com HTML5;
- Tipos de validação
 - > Campos em branco;
 - Campos numéricos aceitando letras;
 - Endereços de e-mail inválidos (estrutura);
 - > Verificar quantidade de números de um CPF e afins (máscaras);
 - > Datas em formato incorreto (máscaras).



Tipos de validação

```
if($preçoVenda < $preçoCusto){
   echo "O Preço de Venda precisa ser
   maior do que o Preço de Custo";
}</pre>
```



```
if($campo == ""){
   echo "O Campo precisa ser preenchido.";
}
```

- Em alguns casos é necessário trabalhar com os dados no formato string para fazer a devida validação ou salvar no banco de dados;
- Funções nativas do PHP são utilizadas
 - > strlen(): Utilizada para saber a quantidade de caracteres que uma variável possui;
 - > str_replace(): Utilizada para substituir ou mais tipos de caracteres da variável;
 - > strstr(): Encontra a primeira ocorrência de uma string dentro de outra string (Case Sensitive);
 - > substr(): Retorna apenas parte de uma string.



Exemplos

```
$nome = "UP";
if(strlen($nome)<= 2){
   echo "O nome precisa ter mais de 2 dígitos.";
}</pre>
```



```
$CEP = "80365-000";
$CEP_formatado = str_replace("-", "", $CEP);
echo $CEP;
```

Exemplos

```
$var = "Hello world!";
if(strstr($var,"world")){
    echo strstr($var,"world");
}
```



```
echo substr("Hello world",10).
  "<br>";
echo substr("Hello world",1).
  "<br>";
```

Dúvidas?

