



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

AULA:

# Passagem de parâmetros em PHP (URL e Formulários)

Programação para Internet

**Alba Lopes, Profa.**

<http://docentes.ifrn.edu.br/albalopes>  
[alba.lopes@ifrn.edu.br](mailto:alba.lopes@ifrn.edu.br)

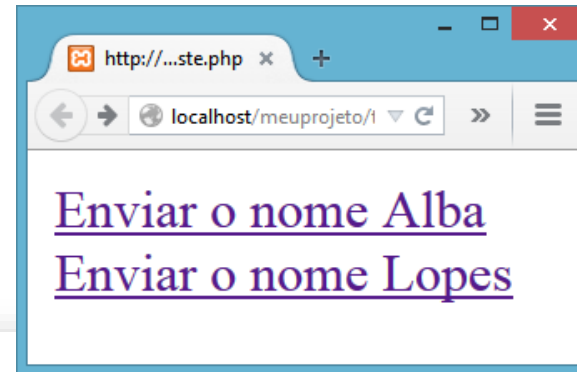
# Passagem de parâmetros

- ▶ Em PHP, há algumas formas de passar parâmetro de uma página para outra:
  - ▶ URL (via método GET)
  - ▶ Formulário (via método GET ou POST)
  - ▶ Cookies
  - ▶ Sessão



# Método GET

- ▶ É possível enviar informações através do método GET
  - ▶ Em formulários ou em URL.
- ▶ Exemplo de método GET em links:



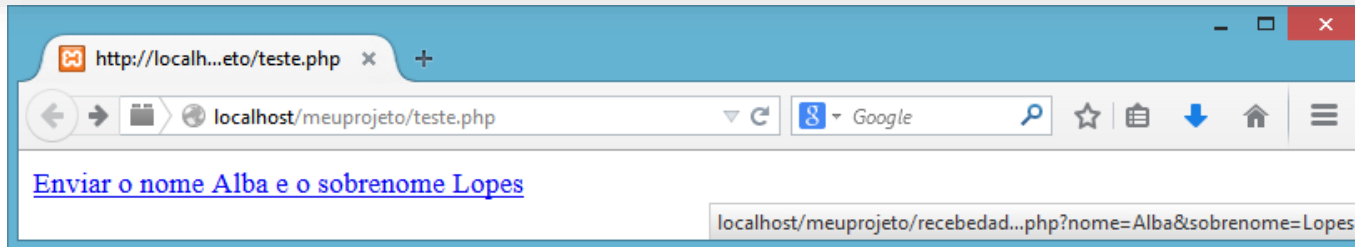
```
<body>
  <p>
    <a href="recebedados.php?nome=Alba" />Enviar o nome Alba</br>
    <a href="recebedados.php?nome=Lopes" />Enviar o nome Lopes
  </p>
</body>
```



# Método GET

- É possível enviar vários parâmetros, através da concatenação.
- No método GET, utiliza-se o padrão:
  - `url?par1=valor1&par2=valor2&par3=valor3`

```
<a href="recebedados.php?nome=Alba&sobrenome=Lopes" />  
Enviar o nome Alba e o sobrenome Lopes</a>
```



# Recuperando valores pelo método GET

- ▶ Para recuperar os valores enviados através do método GET, basta utilizar o array global `$_GET`, indexado pelo nome do parâmetro:

```
<?php  
  
echo $_GET["nome"];  
echo $_GET["sobrenome"];
```

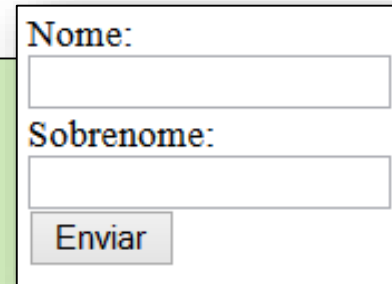
```
<?php  
  
echo "Olá" . $_GET["nome"] . " " . $_GET["sobrenome"] . "!";
```



# Como enviar dados via Formulário

- ▶ Há um atributo `action` a ser inserido na tag `<form>` quando o formulário é criado que deve indicar para onde os dados serão submetidos

```
<form action= "recebedados.php" >  
  <label for= "nome"> Nome: <br/>  
  <input type= "text" name="nome"><br/>  
  <label for= "sobrenome"> Sobrenome: <br/>  
  <input type= "text" name="sobrenome"><br/>  
  <input type= "submit" value="Enviar">  
</form>
```



Nome:

Sobrenome:

Enviar



# Como enviar dados via Formulário

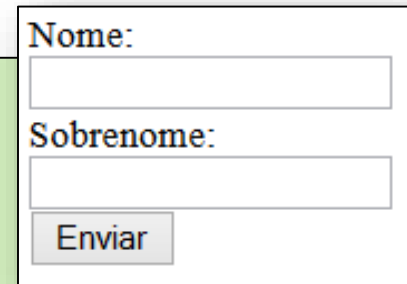
- ▶ O nome do parâmetro a ser enviado deve ser definido no atributo **name** dos campos do formulário!
- ▶ Por padrão, o método de envio dos dados pelo formulário é via método **GET**.
- ▶ Porém, é possível utilizar o método **GET** ou o método **POST**:
- ▶ Ambos cumprem a mesma tarefa, mas de maneiras diferentes
  - ▶ **POST**: empacota as variáveis e as envia de forma oculta para o servidor
  - ▶ **GET**: também empacota as variáveis, porém anexa os valores antes de enviá-las para o servidor



# Enviado dados de formulário via método GET

- ▶ Para especificar o tipo do método de envio, há um atributo `method` a ser inserido na tag `<form>` que indica o método de envio de dados. Para o caso do método GET, a utilização do atributo não faz diferença.:

```
<form action= "recebedados.php" method= "get" >  
  <label for= "nome"> Nome: <br/>  
  <input type= "text" name="nome"><br/>  
  <label for= "sobrenome"> Sobrenome: <br/>  
  <input type= "text" name="sobrenome"><br/>  
  <input type= "submit" value="Enviar">  
</form>
```



Nome:

Sobrenome:

Enviar

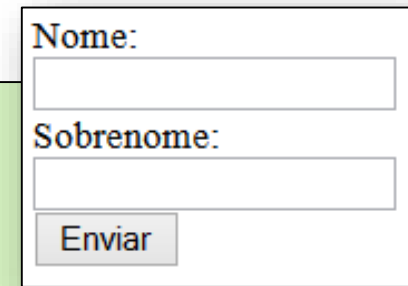




# Enviado dados de formulário via método POST

- ▶ Já no caso do método POST, é necessário indicar explicitamente o método do envio

```
<form action= "recebedados.php" method= "post" >  
  <label for= "nome" > Nome: <br/>  
  <input type= "text" name="nome"><br/>  
  <label for= "sobrenome" > Sobrenome: <br/>  
  <input type= "text" name="sobrenome"><br/>  
  <input type= "submit" value="Enviar" >  
</form>
```



Nome:

Sobrenome:

Enviar



# Recebendo dados de formulário via método POST

- ▶ Caso os dados tenham sido enviados pelo método POST, deve-se utilizar o array global `$_POST` do PHP, indexado pelo nome do campo:

```
<?php  
  
echo "Olá" . $_POST["nome"] . " " . $_POST["sobrenome"] . "!";
```

- ▶ Quando os dados são recebidos via método POST, eles vão ocultos. Não aparecem explicitamente para o usuário.



## Recebendo dados de GET ou POST

- ▶ Caso não faça diferença se os dados foram enviados via método GET ou método POST, pode-se utilizar o array global `$_REQUEST` do PHP, indexado pelo nome do campo:

```
<?php  
  
    echo "Olá" . $_REQUEST["nome"] . " " . $_REQUEST["sobrenome"] . "!";
```

- ▶ Entretanto, vale lembrar que em alguns casos não é desejado que o dados seja recebido por ambos os métodos. Por isso é importante utilizar os arrays específicos da origem (`$_GET` ou `$_POST`).



# Recuperar dados de Radio Button



Meu primeiro formulário

file:///C:/Users/Alba/Documer

Nome:

Sobrenome:

Masculino

Feminino

...

```
<label> Sexo: </label>
```

```
<input type="radio" value="M" name="sexo"> Masculino <br>
```

```
<input type="radio" value="F" name="sexo"> Feminino <br>
```

...



Botões de rádio com mesmo nome só permitem que um elemento seja selecionado. Ao recuperar um campo do tipo radio, no PHP, o value do objeto selecionado é que será enviado por parâmetro.



# Recuperando os dados de Checkbox no PHP

...

```
<label>Onde estuda:</label> <br>  
<input type="checkbox" name="onde_estuda[]" value="IFRN"> IFRN <br>  
<input type="checkbox" name="onde_estuda[]" value="UFRN"> UFRN <br>  
<input type="checkbox" name="onde_estuda[]" value="UERN"> UERN <br>  
<input type="checkbox" name="onde_estuda[]" value="Outro"> Outro...
```

...



No PHP, para que seja possível recuperar todas as opções selecionadas nos checkboxes de mesmo nome, pode-se incluir colchetes após o valor do atributo **name**



# Recuperando os dados de Checkbox no PHP

```
<?php

$sondeestuda = $_GET['onde_estuda'];

//imprimindo o array com print_r
print_r($sondeestuda);

//imprimindo cada elemento do array
foreach ($sondeestuda as $escola){
    echo "<br/> ".$escola;
}
```



Os dados serão recebidos no PHP como um array. Dessa forma, é possível utilizar qualquer mecanismo de manipulação de arrays



# Select

```
<select name= "cidade" >
```

```
<option value= "Nova Cruz" > Nova Cruz </option>
```

```
<option value= "Serrinha" > Serrinha </option>
```

```
<option value= "Montanhas" > Montanhas </option>
```

```
<option value= "Brejinho" > Brejinho</option>
```

```
<option value= "Monte Alegre" > Monte Alegre</option>
```

```
<option value= "Natal" > Natal</option>
```

```
</select>
```

...



Para que seja possível recuperar o valor selecionado no campo do tipo select, é importante que seja definido o atributo value de cada opção. O name é definido para o objeto select:



# Devolvendo um dado para o usuário via URL

- ▶ Vimos como usuário pode passar parâmetros para o servidor. Mas como passar parâmetros de volta para o usuário?
- ▶ Exemplo:
  - ▶ Fazer a soma de 2 números e colocar o resultado em um campo de texto na mesma página.

Número 1:	<input type="text"/>
Número 2:	<input type="text"/>
<input type="button" value="Somar"/>	
Resultado:	<input type="text"/>





# Devolvendo um dado para o usuário via URL

- ▶ Teríamos as seguintes páginas:

## form.php

```
<form action="calcular.php" method="post">
  Número 1: <input type="text" name="n1"><br/>
  Número 2: <input type="text" name="n2"><br/>
  <input type="submit" value="Somar"> <br/>
  Resultado: <input type="text" width="5"><br/>
</form>
```

## calcular.php

```
<?php
$n1 = $_POST["n1"];
$n2 = $_POST["n2"];
$resultado = $n1 + $n2;
echo "A soma é: ".$resultado;
```



# Devolvendo um dado para o usuário via URL

- ▶ Para que a página calcular.php devolva um valor para a página form.php, é possível fazer um redirecionamento após o cálculo do valor e passar o resultado pela URL, como vimos anteriormente:
  - ▶ O redirecionamento é feito através da função:

```
header("Location: url")
```

```
calcular.php
```

```
$n1 = $_POST["n1"];  
$n2 = $_POST["n2"];  
$resultado = $n1 + $n2;  
header("Location: form.php?res=$resultado");
```



# Devolvendo um dado para o usuário via URL

- Pode-se então, fazer um tratamento na página form.php para que o dado seja exibido quando receber um parâmetro via método GET:

```
Número 1: <input type="text" name="n1"><br/>
Número 2: <input type="text" name="n2"><br/>
<input type="submit" value="Somar"> <br/>
Resultado:
<input type="text" value="<?php echo (isset($_GET["res"]))?$_GET["res"]:"";?>"><br/>
```

- No exemplo acima foi usado o operador ternário para resumir o condicional *if*:

```
<?php
    if (isset($_GET["res"])) {
        echo $_GET["res"];
    }
?>
```



# Devolvendo um dado para o usuário via URL

- ▶ É possível fazer o mesmo para que os campos Número 1 e Número 2 também fiquem preenchidos, incluindo os parâmetros na URL no arquivo calcular.php :

```
<?php
$n1 = $_POST["n1"];
$n2 = $_POST["n2"];
$resultado = $n1 + $n2;
header("Location: form.php?res=$resultado&n1=$n1&n2=$n2");
```

- ▶ E recebendo de volta no arquivo form.php

```
Número 1: <input name="n1" value="<?php echo (isset($_GET["n1"])?$_GET["n1"]:"");?>"
Número 2: <input name="n2" value="<?php echo (isset($_GET["n2"])?$_GET["n2"]:"");?>"
```



# Devolvendo um dado para o usuário via URL

- ▶ Essa é uma forma de ser feita a passagem de parâmetro do PHP para o usuário
- ▶ Mas há formas mais elegantes de se fazer esse procedimento, que o caso da utilização de Sessão e Cookies.



# Referências

- ▶ Fábio Procópio; João Maria. Introdução a PHP. Notas de Aula. Disponível em: <https://sites.google.com/site/fabiooprocopio/programacao-web>
- ▶ Pedro Baesse. Manipulação e Tipos de Dados em PHP. Notas de Aula. Disponível em: <http://docente.ifrn.edu.br/pedrobaesse>
- ▶ Pedro Baesse. PHP e formularios em HTML. Notas de Aula. Disponível em: <http://docente.ifrn.edu.br/pedrobaesse>
- ▶ CITI. Curso PHP - Aula 01. Disponível em: [http://www.cin.ufpe.br/~vbom/arquivos/PHP/PHP%20-%20Aula\\_01.pdf](http://www.cin.ufpe.br/~vbom/arquivos/PHP/PHP%20-%20Aula_01.pdf)

