TEMA 6 - **Método de envio com GET e POST**

Dentro das linguagens de programação precisamos ter um maneira para enviar os dados entre a própria linguagem e para a base de dados, para enviar os dados que se encontra no formulário do html, vamos utilizar o método post e get, que colocamos dentro da tag form no parâmetro method.

6.1 Introdução sobre o conceito de envio GET e POST com HTML 5

Os métodos que vamos ver agora são usados na sintaxe de estrutura do PHP, **sendo o get meio de envio pela url e o post por maneira de aquisição de dados**, ambos seguros mais com maneiras de envios diferentes. Os dois métodos que vamos estudar agora precisam do formulário do html.

6.2 Método POST e GET

Método GET

Nesse método os dados são enviados juntamente com o nome da página, no caso pela url, método muito usado por se tratar de uma maneira rápida de envio ele não suporta dados com um poder muito grande. **Esse método é considerado o padrão do HTML, caso não especificado no method, ele que sempre vai estar disponível**.

Formulário do HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Aula de método</title>
</head>
<body>
  <form action="pag.php" method="get">
    Nome: <input type="text" name="cxnome">
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

Form é a tag usado no html para criar um formulário
Action é para onde os dados vão quando clicar no botão enviar no caso vai para a pagina "pag.php".
Name é o parâmetro usado para rotular a caixa do input.
Method é o método que vamos usar para enviar os dados no caso agora vai ser o GET.
Pagina que vai receber os dados

pag.php

```
<?php
$nome = $_GET["cxnome"];
echo "Nome: " . $nome;
?>
```

$nome é a variável que está recebendo o nome pela caixa input dado o **name de cxnome.**

$_GET[“ ”] é o parâmetro que sempre vamos usar para receber os dados do formulário, dentro dos colchetes colocamos o nome do input.

Método POST

No **método POST, não é possível passar dados no caso parâmetros pela URL**, somente pelo formulário, por aquisição de informação do usuário. Ele também vai seguir a mesma sintaxe usada no GET, considerado um **método mais seguro por não mostrar na url para onde os dados estão sendo enviados**. Método **muito usado para guardar dados no servidor de dados**, com maior poder de gerenciar dados, para criar um CRUD, esse método é o mais recomendado.

Formulário do HTML

```
<!DOCTYPE html>
<html>
<head>
 <title>Aula de método</title>
</head>
<body>
 <form action="pag.php" method="post">
  Nome: <input type="text" name="cxnome">
  <input type="submit" value="Enviar">
 </form>
</body>
</html>
```

Seguindo o mesmo exemplo do método get, só mudamos para POST no parâmetro method.

Pagina que vai receber os dados

```
<?php
$nome = $_POST["cxnome"];
echo "Nome: " . $nome;
?>
```

Seguindo a mesma ideia do método get, agora a variável muda para $_POST[“ ”], para receber do formulário que está usando o método post.

6.3 Include e Require suas diferenças e particularidades

Vamos agora incluir arquivos em páginas PHP, usando os comandos require() e include(), que são responsáveis por realizar a leitura das tags PHP e executar dentro delas, por via de código.
A grande diferença entre esses dois elementos que vamos falar agora é que o **include(), trabalha em tempo real**, exemplo podemos **usar ele nos laços de repetição**, já o require() não tem esse time real.

Include_once e require_once

Estes dois comandos também chamados de expressões trabalham de maneira idêntica ao include e require padrão, a única diferença entre eles quando usamos o _once, o arquivo será incluído uma única vez ou seja executado só um de cada vez, por isso que vamos usar esses dois comando nas conexão de banco de dados.

Veja o exemplo: Arquivo: pagina.php

pagina.php

```
<?php
        $escola = "Proz - Educação Profissional";
?>
```

pagina2.php

```
<?php
        include "pagina.php";
        echo "Escola " . $escola;
?>
```

As páginas que vão receber os arquivos, no cabeçalho da página precisamos digitar o comando include ou require, aí tudo que temos nessa página podemos reutilizar.

Ambos os exemplos incluem o arquivo arquivo_incluido.php, que define uma variável $mensagem. A diferença é como eles lidam com a falta desse arquivo:

Se o arquivo arquivo_incluido.php não for encontrado, o exemplo usando include emitirá apenas um aviso, permitindo que o script continue a ser executado, embora a variável $mensagem não seja definida.

base.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de Include</title>
</head>
<body>
  <h1>Exemplo de Include</h1>
  <?php
  // Incluindo o arquivo arquivo_incluido.php
  include 'arquivo_incluido.php';

  // Usando a variável definida no arquivo incluído
  echo "<p>$mensagem</p>";
  echo "<h1>Final script</h1>";
  ?>
</body>
</html>
```

arquivo_incluido.php

```
<?php
// Arquivo: arquivo_incluido.php
```

```
$mensagem = "Este é um arquivo incluído usando include.";
?>
```

Por outro lado, se o arquivo arquivo_incluido.php não for encontrado no exemplo usando require, o PHP emitirá um erro fatal e interromperá a execução do script imediatamente.

base.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de Require</title>
</head>
<body>
  <h1>Exemplo de Require</h1>
  <?php
  // Incluindo o arquivo arquivo_incluido.php
  require 'arquivo_incluido.php';

  // Usando a variável definida no arquivo incluído
  echo "<p>$mensagem</p>";
        echo "<h1>Final script</h1>";
  ?>
</body>
</html>
```

arquivo_incluido.php

```
<?php
// Arquivo: arquivo_incluido.php
$mensagem = "Este é um arquivo incluído usando require.";
?>
```

Resumo:

Neste capítulo, abordamos os métodos de envio GET e POST, fundamentais para a comunicação entre linguagens de programação e bases de dados. Ambos os métodos são utilizados em conjunto com formulários HTML, permitindo a transferência de dados.
No método GET, os dados são enviados pela URL, tornando-o rápido e fácil de implementar.
É o método padrão do HTML e amplamente disponível. Explicamos a estrutura de um formulário HTML e como os dados são recebidos no PHP usando $_GET.
Por outro lado, o método POST é mais seguro, pois não exibe os dados na URL. É recomendado para operações que envolvem o armazenamento de dados no servidor, como a criação de um CRUD. Demonstramos a configuração de um formulário HTML para usar o método POST e como os dados são recebidos no PHP usando $_POST.
Além disso, discutimos as diferenças entre os comandos require() e include() para incluir arquivos PHP em páginas, destacando que o include() funciona em tempo real, enquanto o require() não. Também apresentamos as variantes include_once e require_once, que garantem a inclusão única de arquivos, adequadas para conexões de banco de dados.Esses conceitos são essenciais para entender

como transmitir e manipular dados em desenvolvimento backend, permitindo a interação eficaz com bancos de dados e outros componentes do sistema.

ATIVIDADES:

1. Qual é a principal diferença entre os métodos GET e POST em termos de visibilidade dos dados enviados? Dê um exemplo de quando a visibilidade dos dados na URL pode ser um problema.

2. Liste pelo menos três atributos essenciais da tag <form> em HTML e explique a função de cada um deles.
3. Se você deseja incluir um arquivo PHP em uma página, qual comando PHP você usaria e por quê: require() ou include()? Explique.

4. Qual é a principal vantagem de usar o require_once em vez do require ao incluir arquivos em uma página PHP? Dê um exemplo de quando essa vantagem seria útil.

5. Crie um exemplo de formulário HTML que colete informações de um usuário, como nome, idade e endereço de e-mail, e utilize o método POST para enviar esses dados para uma página PHP. Descreva como acessar esses dados na página PHP.

6. Explique sobre essa linha de comando <form action=”pag.php” method=”POST”>

7. Qual é o método usado para enviar dados via URL?

8. Qual é a principal diferença entre include e require?

9.Desenvolva um cenário hipotético em que o uso do método GET para enviar dados de um formulário seria inadequado devido à natureza dos dados. Justifique sua escolha pelo método POST.

10. Compare e contraste os métodos de envio GET e POST em relação à segurança e aos tipos de dados que podem ser transmitidos. Dê exemplos de situações em que você usaria um método em vez do outro.

Respostas:

1. A principal diferença entre os métodos GET e POST em termos de visibilidade dos dados enviados é que no método GET, os dados são anexados à URL, tornando-os visíveis na barra de endereços do navegador, enquanto no método POST, os dados são enviados no corpo da requisição HTTP e não são visíveis na URL. Um exemplo de quando a visibilidade dos dados na URL pode ser um problema é ao enviar informações sensíveis, como senhas, através do método GET, pois esses dados podem ser facilmente interceptados e visualizados por terceiros.

2. Três atributos essenciais da tag `<form>` em HTML são:
- `action`: Especifica para onde os dados do formulário serão enviados quando o formulário for submetido.
- `method`: Define o método HTTP a ser usado para enviar os dados do formulário (GET ou POST).
- `name`: Um nome opcional para o formulário, que pode ser usado para referenciá-lo em scripts ou estilizações.

3. Se deseja incluir um arquivo PHP em uma página, o comando PHP que você usaria depende da necessidade. Se o arquivo PHP incluído for essencial para o funcionamento da página, é

recomendado usar `require()` porque ele interromperá o script caso o arquivo não seja encontrado, garantindo assim que a página não funcione corretamente sem o arquivo. Se o arquivo PHP incluído for opcional e não crítico para o funcionamento da página, você pode usar `include()` porque ele não interromperá o script caso o arquivo não seja encontrado, permitindo que a página continue funcionando.

4. A principal vantagem de usar `require_once` em vez de `require` ao incluir arquivos em uma página PHP é que `require_once` garante que o arquivo seja incluído apenas uma vez, mesmo que seja referenciado em várias partes do código. Isso evita erros de redeclaração de funções ou variáveis. Por exemplo, ao incluir um arquivo de configuração que define constantes ou funções globais, usando `require_once` garante que esses elementos não sejam definidos mais de uma vez, mantendo a integridade do código.

5. Exemplo de formulário HTML usando método POST:
```html
<!DOCTYPE html>
<html>
<head>
  <title>Formulário de Exemplo</title>
</head>
<body>
  <form action="processa_dados.php" method="post">
    Nome: <input type="text" name="nome"><br>
    Idade: <input type="number" name="idade"><br>
    E-mail: <input type="email" name="email"><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```
Na página PHP `processa_dados.php`, você pode acessar os dados enviados pelo formulário usando `$_POST`. Por exemplo:
```php
<?php
$nome = $_POST["nome"];
$idade = $_POST["idade"];
$email = $_POST["email"];

echo "Nome: $nome<br>";
echo "Idade: $idade<br>";
echo "E-mail: $email<br>";
?>
```

6. `<form action=”pag.php” method=”POST”>`: Esta linha de comando define um formulário HTML que enviará os dados para uma página PHP chamada `pag.php` usando o método POST quando o formulário for submetido.

7. O método usado para enviar dados via URL é o método GET.

8. A principal diferença entre include e require é que o include permite que o script continue a ser

executado mesmo se o arquivo incluído não for encontrado, enquanto o require interrompe a execução do script com um erro fatal caso o arquivo não seja encontrado.

9. Um cenário hipotético em que o uso do método GET para enviar dados de um formulário seria inadequado devido à natureza dos dados é ao lidar com informações sensíveis, como senhas ou dados pessoais. Como o método GET anexa os dados à URL, eles podem ser facilmente interceptados e visualizados por terceiros, o que representa um risco de segurança. Portanto, o método POST seria mais adequado nestes casos, já que os dados são enviados no corpo da requisição e não são visíveis na URL.

10. Comparando os métodos de envio GET e POST:
- Segurança: O método POST é considerado mais seguro do que o método GET, especialmente para dados sensíveis, pois os dados não são expostos na URL.
- Tipos de dados: Ambos os métodos podem transmitir uma ampla variedade de tipos de dados, incluindo texto, números, arquivos e outros. No entanto, o método POST é mais adequado para enviar grandes quantidades de dados ou dados sensíveis, devido à sua natureza de não expor os dados na URL.
- Exemplos de uso: O método GET é frequentemente utilizado para solicitações simples, como pesquisas ou navegação entre páginas, onde a visibilidade dos parâmetros na URL não é um problema. O método POST é mais comumente utilizado para enviar dados de formulários, especialmente quando se trata de informações sensíveis ou grandes volumes de dados.