**Códigos PHP**

<?php

// Define o nome do arquivo que será usado

$filename = "teste.txt";

// Verifica se o arquivo "teste.txt" já existe

if(!file\_exists("teste.txt")){

// Se o arquivo não existir, abre o arquivo para escrita, criando-o se necessário e apagando qualquer conteúdo existente

$handle = fopen("teste.txt", "w");

} else {

// Se o arquivo já existir, abre o arquivo para adicionar novos dados ao final do conteúdo existente

$handle = fopen("teste.txt", "a");

}

// Escreve a string "Conteúdo\n" no arquivo

fwrite($handle, "Conteúdo\n");

// Garante que todos os dados foram gravados no arquivo

fflush($handle);

// Fecha o arquivo após a escrita

fclose($handle);

// Reabre o arquivo "teste.txt" para leitura

$handle = fopen("teste.txt", "r");

// Lê e exibe o conteúdo do arquivo linha por linha

while (!feof($handle)) {

// Lê uma linha do arquivo

$line = fgets($handle);

// Exibe a linha lida, com uma quebra de linha HTML

echo $line . "<br>";

}

// Fecha o arquivo após a leitura

fclose($handle);

?>

Este código adiciona a linha "Conteúdo" ao final do arquivo "teste.txt" e, em seguida, lê o arquivo e exibe seu conteúdo no navegador. Na tela, você verá a linha "Conteúdo" exibida com uma quebra de linha HTML.

**GET x POST**

### **Método GET**

**O que é:**

* GET é um método de solicitação HTTP usado para obter dados de um servidor. Quando você usa GET, os dados são enviados como parte da URL da solicitação.

**Exemplo:**

* Se você estiver pesquisando por um livro em um site e preencher um formulário de pesquisa, ao enviar o formulário, a URL pode parecer algo como site.com/pesquisar?livro=programacao. Aqui, livro=programacao é o dado enviado.

**Diferença Principal:**

* **Visibilidade:** Os dados são visíveis na URL, o que pode ser útil para compartilhar ou salvar URLs de pesquisa, mas não é adequado para informações sensíveis.
* **Segurança:** Menos seguro para dados confidenciais, pois a URL pode ser vista e registrada.
* **Uso Ideal:** Ideal para operações que não alteram o estado do servidor, como pesquisas e filtros onde os dados não são sensíveis.

### **Método POST**

**O que é:**

* POST é um método de solicitação HTTP usado para enviar dados ao servidor para criar ou atualizar recursos. Os dados são enviados no corpo da solicitação, não na URL.

**Exemplo:**

* Se você estiver preenchendo um formulário de registro de usuário com seu nome e senha, e enviar o formulário, os dados serão enviados para o servidor sem serem visíveis na URL. A solicitação é feita de forma que o nome e a senha não aparecem na barra de endereços do navegador.

**Diferença Principal:**

* **Visibilidade:** Os dados não são visíveis na URL, o que melhora a segurança ao lidar com informações sensíveis.
* **Segurança:** Mais seguro para informações confidenciais, pois os dados são incluídos no corpo da solicitação e não são expostos na URL.
* **Uso Ideal:** Ideal para enviar dados sensíveis, como informações de login, ou quando os dados a serem enviados são grandes ou complexos, como registros de usuários ou formulários de pagamento.

### **Diferenças entre GET e POST**

1. **Visibilidade dos Dados:**
   * **GET:** Dados são visíveis na URL.
   * **POST:** Dados não são visíveis na URL; são enviados no corpo da solicitação.
2. **Segurança:**
   * **GET:** Menos seguro para dados sensíveis, pois a URL pode ser visualizada e registrada.
   * **POST:** Mais seguro para dados sensíveis, pois os dados são enviados de forma oculta no corpo da solicitação.
3. **Uso de Dados:**
   * **GET:** Melhor para solicitações que não alteram o estado do servidor e onde a visibilidade dos dados é aceitável, como filtros e pesquisas.
   * **POST:** Melhor para operações que alteram o estado do servidor, como criar ou atualizar recursos, e quando os dados são sensíveis ou extensos.
4. **Limitação de Tamanho:**
   * **GET:** Limitado pelo tamanho máximo da URL.
   * **POST:** Capacidade de envio de dados muito maior, sem limitações significativas de tamanho.

Em resumo, GET é adequado para solicitações simples e não sensíveis que podem ser exibidas na URL, enquanto POST é apropriado para enviar informações que precisam ser protegidas ou são grandes e complexas.