Manipulando arquivos em PHP

- Manipular arquivos em PHP permite salvar informações para que as mesmas possam ser tratadas posteriormente;
- As informações são guardadas em disco, no servidor, sem a utilização de banco de dados;
- Operações sobre arquivos:
  - ✓ **Criação;**
  - ✓ **Abertura**
  - ✓ **Leitura;**
  - ✓ **Escrita;**
  - ✓ **Fechamento;**
  - ✓ **Exclusão.**

Quando usar arquivos em PHP

- Quando o volume de dados a ser armazenado é pequeno;
- Quando o servidor de banco de dados está armazenado em um host remoto;
- **Algumas considerações:**
  - ✓ Acessos a arquivos são feitos de forma mais rápida que acesso a banco de dados;
  - ✓ Acesso a arquivos exige menor complexidade de código do que acesso a banco de dados.

Exemplos de uso de arquivos em PHP

- Contadores de acesso;
- Gerenciador de publicidade online;
- Fóruns de discussão;
- Livros de visitas.

Funções comuns com arquivos em PHP

- Copiar um arquivo – **copy("arq_origem", "arq_destino");**
- Excluir um arquivo – **unlink("nome_arquivo");**
- Testar se o ponteiro chegou ao final do arquivo – **feof($apontador);**
- Testar se um arquivo existe – **file_exists("nome_arquivo");**
- Retornar o tamanho do arquivo em bytes – **filesize("nome_arquivo");**
- Criar um diretório – **mkdir("nome_diretorio");**
- Excluir um diretório – **rmdir("nome_diretorio");**
- Renomear um arquivo – **rename("nome_antigo", "nome_novo");**
- Abrir um arquivo – **fopen("nome_arquivo", "modo");**

Abertura de um arquivo

- **fopen** – todo arquivo em PHP precisa ser aberto antes de ser maniplado. Sintaxe:

```
fopen("nome_arquivo", "modo");
```

- **nome_arquivo** pode ser um arquivo local ou remoto;
- **modo** diz respeito à forma como o arquivo será aberto;
- Exemplos:

A)fopen("http://locaweb.com.br/arquivos/acesso.txt", "r+");

B)fopen("./arquivos/tarefas/contadores/contador.txt", "w+");

Modos de abertura

- **"r"** – abre somente para leitura. Posiciona o ponteiro no início do arquivo;
- **"r+"** – abre para leitura e escrita. Posiciona o ponteiro no início do arquivo;
- **"w"** – abre somente para escrita. Posiciona o apontar no início do arquivo. Se o arquivo já existir, apaga todo o seu conteúdo. Se ainda não existir, tenta criá-lo;
- **"w+"** – abre para leitura e escrita. Posiciona o apontador no início do arquivo. Se o arquivo já existir, o mesmo é apagado. Se ainda não existir, tenta criá-lo.
- **"a"** – abre somente para escrita. Vai para o final do arquivo. Se o arquivo não existir, tenta criá-lo ("a" vem da palavra **a**ppend);
- **"a+"** – abre para leitura e escrita, posicionando o apontador de registros no final do arquivo. Se o arquivo não existir, tenta criá-lo;

Fechamento de um arquivo

- Após encerradas as operações, todo arquivo em PHP precisa ser fechado. Algumas operações, também, exigem que o arquivo esteja fechado antes de as executarmos. Exemplos: *renomear, excluir, mover, etc...*
- A função abaixo retorna o valor true se conseguiu fechar o arquivo. False, caso contrário.

fclose($ponteiro);

- Exemplo:

```
<?php
$ponteiro = fopen("frases.txt", "r");
fclose($ponteiro);
?>
```

Lendo linhas de um arquivo

fgets($ponteiro);

- Todo conteúdo de um arquivo em PHP é gravado sendo separado por linhas. O marcador que separa as linhas é o "enter", que, no PHP, pode ser representado pela constante **PHP_EOL**;
- A função acima, sempre que for chamada, retorna uma única linha do arquivo. Se for chamada várias vezes, retorna uma linha a cada vez. Costuma ser usada dentro de um laço;
- Exemplo:

```
<php?
$ponteiro = fopen("teste.txt", "w+");
$linha = fgets($ponteiro);
echo("$linha");
fclose($ponteiro);
?>
```

Escrevendo dados em um arquivo

fwrite($ponteiro, $string);

- Escreve o conteúdo da variável **$string** no arquivo referenciado pelo endereço **$ponteiro**. Exemplo:

```
<php?
$conteudo = "Maria da Conceição Tavares“.PHP_EOL;
$ponteiro = fopen("nomes.txt", "w");
fwrite($ponteiro, $conteudo);
fclose($ponteiro);
?>
```

Escrevendo dados em um arquivo – PHP v5.0

file_put_contents($arquivo, $dados);

- Esta função abre o arquivo, escreve os dados nele e fecha o arquivo. Se o mesmo ainda não existir, ele é criado. Se já existir, o arquivo é zerado e reescrito com as novas informações de **$dados**. A variável $dados pode ser um string, um número ou um array unidimensional.
- Se você quiser preservar o conteúdo que já existe no arquivo e anexar as novas informações ao final, use **file_put_contents($arquivo, $dados, FILE_APPEND);**
- Equivale a chamar fopen(), fwrite() e fclose() em sequência.

Lendo dados de um arquivo – PHP v5.0

file_get_contents($arquivo);

- Esta função abre o arquivo passado como parâmetro e lê o conteúdo inteiro, retornando o dado no formato de uma string para a variável de retorno. Se a operação de leitura não puder ser feita, a função retorna ***false***. Exemplo:

```
$arquivo = "musicas.txt";

$conteudo = file_get_contents($arquivo);
```

- A variável **$conteudo** armazenará todas as músicas constantes do arquivo no formato de uma string ou **false**, se o mesmo não pôde ser lido. Equivale a chamar fopen(), fwrite() e fclose() em sequência.

Lendo dados de um arquivo – PHP v5.0

file($arquivo);

- Esta função abre o arquivo passado como parâmetro e lê cada uma das linhas do arquivo, armazenando-as como conteúdo de um vetor. A quebra de linha também é armazenada, mas pode ser removida. Há, também, um parâmetro para se ignorar linhas vazias. Se a operação de leitura não puder ser feita, a função retorna ***false***. Exemplo:

```
$arquivo = "musicas.txt";

$vetor = file($arquivo);
```

- A variável **$vetor** armazenará todas as músicas constantes do arquivo no formato de um vetor, com cada linha guardada em uma célula do mesmo. Equivale a chamar fopen(), fwrite() e fclose() em sequência.