



**INSTITUTO FEDERAL**

Sul-rio-grandense  
Câmpus Charqueadas

# Aula 7

- Manipulação de arquivos

**Desenvolvimento Back-end I**

**Prof. Sérgio Yoshimitsu Fujii**

# Manipulação de arquivos

## Aplicações

- Substituir um SGBD por arquivos texto em sites pequenos;
- Salvar informações de acesso em um arquivo de log;

## Por que?

- Pouco custoso computacionalmente em relação a um SGBD para operações pequenas.

# Manipulação de arquivos

## Funções

- **fopen()** – ‘Abre’ o arquivo para que possa ser manipulado.
- **fgets()** – Pega uma linha do arquivo até o máximo de 1024bytes.
- **feof()** – Durante a leitura de um arquivo, avisa se chegou ao final.
- **file\_get\_contents()** – Pega todo conteúdo do arquivo aberto como uma string.
- **ftruncate()** – Reduz o tamanho do arquivo. Usado para apagar seu conteúdo.
- **fwrite()** – Escreve no arquivo.

# Manipulação de arquivos

## Criar arquivos

- Função **fopen()**.

```
1  <?php
2
3      fopen($nomedoarquivo, $modo);
4
5  ?>
```

- **\$nomedoarquivo** – O nome do arquivo pode conter o caminho absoluto onde você deseja criar o seu arquivo. Por exemplo, 'www/dir/arquivo.txt'.
- **\$modo** – O modo indica as permissões de acesso para o arquivo que foi criado. Por exemplo, ele pode ser criado apenas para leitura (consulta de dados) ou para escrita e leitura.

# Manipulação de arquivos

## Modos de arquivos

Valor	Descrição
<b>w</b>	Abre o arquivo para escrita. Coloca o ponteiro no começo do arquivo e diminui( <i>trunca</i> ) o tamanho do arquivo para zero. Se o arquivo não existe, tenta criá-lo.
<b>w+</b>	Abre o arquivo para leitura e escrita. Coloca o ponteiro do arquivo no começo, diminuindo (truncando) o tamanho do arquivo para zero. Se o arquivo não existe, tenta criá-lo.
<b>r</b>	Abre o arquivo para leitura. Coloca o ponteiro de escrita no começo do arquivo. [Retorna um erro caso o arquivo não exista e não tenta criá-lo].
<b>r+</b>	Abre para leitura e escrita; Coloca o ponteiro no começo do arquivo. [Retorna um erro caso o arquivo não exista e não tenta criá-lo].

# Manipulação de arquivos

## Modos de arquivos

Valor	Descrição
<b>a</b>	Abre para escrita somente. Coloca o ponteiro do arquivo no final. Se o arquivo não existir, tenta criá-lo.
<b>a+</b>	Abre o arquivo para leitura e escrita. Coloca o ponteiro do arquivo no final. Se o arquivo não existir, tenta criá-lo.
<b>x</b>	Cria e abre o arquivo para escrita somente. Coloca o ponteiro no início do arquivo. Se o arquivo já existe, a chamada a <code>fopen()</code> irá falhar, retornando <code>FALSE</code> , gerando um erro nível <code>E_WARNING</code> . Se o arquivo não existe, tenta criá-lo.
<b>x+</b>	Cria e abre um arquivo para escrita e leitura. Coloca o ponteiro do arquivo no início. Se o arquivo já existe, a chamada a <code>fopen()</code> irá falhar, retornando <code>FALSE</code> , gerando um erro nível <code>E_WARNING</code> . Se o arquivo não existe, tenta criá-lo.

# Manipulação de arquivos

## Criar um novo arquivo

- Função **fopen()**.

```
1  <?php
2
3      $arquivo = fopen('meuarquivo.txt','w');
4      if ($arquivo == false) die('Não foi possível criar o arquivo.');
```

Tenha cuidado pois o modo 'w' coloca o ponteiro de escrita no início do arquivo. Se o arquivo já existir, ele deletará todo o conteúdo;

Se a função funcionar ele retorna o ponteiro do arquivo. Mas se a função falhar por qualquer motivo, a função retorna false.

# Manipulação de arquivos

## Lendo arquivos

- Função **fgets()**.

```
1  <?php
2      fgets($ponteiro, $tamanho);
3  ?>
```

A função *fgets()* é usada para ler um arquivo linha por linha. Ela aceita dois tipos de parâmetros:

- **\$ponteiro** – O ponteiro aqui utilizado, está armazenado na variável que você usou para abrir ou criar o arquivo.
- **\$tamanho** – É o tamanho em número de bytes que deve ser lida a linha do arquivo. Se não for especificado, o programa irá ler até o final da linha (até um número máximo de 1024 caracteres).



# Manipulação de arquivos

## Lendo arquivos

- Função **fgets()**. Ler a primeira linha

```
1  <?php
2      $arquivo = fopen('meuarquivo.txt','r');
3      if ($arquivo == false) die('Não foi possível abrir o arquivo.');
```

```
4      $linha = fgets($arquivo);
5      echo $linha;
6      fclose($arquivo);
7  ?>
```

- O arquivo foi aberto com o modo 'r' que posiciona o ponteiro no início do arquivo.
- O comando fgets() foi usado sem o parâmetro de tamanho. Isso quer dizer que toda primeira linha do arquivo está sendo lida.
- A função **fclose()** fecha o arquivo que está aberto.

# Manipulação de arquivos

## Lendo arquivos

- Função **fgets()**. Ler todas as linhas

```
1  <?php
2      $arquivo = fopen('meuarquivo.txt','r');
3      if ($arquivo == false) die('Não foi possível abrir o arquivo.');
```

```
4      while(true) {
5          $linha = fgets($arquivo);
6          if ($linha==null) break;
7          echo $linha;
8      }
9      fclose($arquivo);
10 ?>
```

- “E se meu arquivo tiver uma linha vazia?” Não existe linha vazia entre o início e o final de um documento. Mesmo que uma linha não tenha nada escrito, ela contém um marcador de quebra de linha.

# Manipulação de arquivos

## Lendo arquivos

- Função **fgets()**. Ler todas as linhas - **feof()**

```
1  <?php
2      $arquivo = fopen('meuarquivo.txt','r');
3      if ($arquivo == false) die('Não foi possível abrir o arquivo.');
```

// imprime linha por linha ate detectar o final

```
5      while(!feof($arquivo)) {
6          echo fgets($arquivo). '<br />';
7      }
8      fclose($arquivo);
9  ?>
```

- Outra forma mais interessante para ler um arquivo linha por linha (somente ler) é usando a função **feof()**.

# Manipulação de arquivos

## Lendo arquivos

- Função **file\_get\_contents()**. Ler todo o arquivo de uma vez

```
1  <?php
2      $arquivo = fopen('meuarquivo.txt','r');
3      $string = file_get_contents('meuarquivo.txt');
4      echo $string;
5      fclose($arquivo);
6  ?>
```

- A função **file\_get\_contents()** retorna todo o conteúdo de determinado arquivo;
- Não é recomendado ler arquivos grandes de uma só vez, recomenda-se ler em partes (linhas).

# Manipulação de arquivos

## Escrevendo em arquivos

- Função **fwrite()**.

```
1  <?php
2      fwrite($ponteiro, $string, $tamanho);
3  ?>
```

- **\$ponteiro** – O ponteiro aqui utilizado, está armazenado na variável que você usou para abrir ou criar o arquivo.
- **\$string** – O conteúdo textual que você deseja escrever em seu arquivo.
- **\$tamanho** – É o tamanho em número de bytes que deve ser gravado no arquivo. Opcional.

# Manipulação de arquivos

## Escrevendo em arquivos

- Função **fwrite()**.

```
1  <?php
2      $arquivo = fopen('meuarquivo.txt','w+');
3      if ($arquivo) {
4          if (!fwrite($arquivo, 'Novo conteúdo'))
5              die('Não foi possível atualizar o arquivo.');
```

- É importante observar que o modo 'w+' cria arquivos mas também trunca o arquivo caso ele exista, ou seja, deleta todo seu conteúdo assim que ele é aberto.

# Manipulação de arquivos

## Alterar conteúdo

```
1  <?php
2
3      $arquivo = fopen('meuarquivo.txt','r+');
4      $string = null;
5      if($arquivo) {
6          while(true) {
7              $linha = fgets($arquivo);
8              if($linha==null)
9                  break;
10
11              // busca na linha atual o conteúdo que vai ser alterado
12              if(preg_match("/Sérgio Fujii/", $linha)) {
13                  $string .= str_replace("Sérgio Fujii: 28 anos", "Sérgio Fujii: 38 anos", $linha);
14              } else {
15                  // vai colocando tudo numa nova string
16                  $string .= $linha;
17              }
18          }
19          // move o ponteiro para o início pois o ftruncate() não fará isso
20          rewind($arquivo);
21          // trunca o arquivo apagando tudo dentro dele
22          ftruncate($arquivo, 0);
23          // reescreve o conteúdo dentro do arquivo
24
25          if (!fwrite($arquivo, $string))
26              die('Não foi possível atualizar o arquivo.');
```

```
27      echo 'Arquivo atualizado com sucesso.';
28      fclose($arquivo);
29  }
30  ?>
```

# Manipulação de arquivos

## Funções úteis

- **explode()** - divide uma *string*.
- **opendir()** - abre um diretório.
- **readdir()** - lê os arquivos de um diretório.
- **closedir()** - fecha um diretório.
- **file\_exists()** - verifica se um arquivo existe.



# Manipulação de arquivos

## Manual PHP

- Funções para manipulação de arquivos:

[https://www.php.net/manual/pt\\_BR/book.filesystem.php](https://www.php.net/manual/pt_BR/book.filesystem.php)

- Fopen:

[https://www.php.net/manual/pt\\_BR/function.fopen.php](https://www.php.net/manual/pt_BR/function.fopen.php)