



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

PHP

Aula 4

Professora Marla Cristina da Silva Sopeña



Upload de Arquivos

Consiste em enviar um ou mais arquivos de um computador local para um computador remoto.

Por que fazer upload de arquivos?

Enviar fotos: Loja Virtual – Sites de Currículos - Sites de relacionamento .

Sites com espaços para compartilhamento de arquivos.de arquivos

Protocolos HTTP e FTP

Protocolos utilizados para realizar transferência de arquivos

HTTP (HyperText Transfer Protocol) – ocorre por meio de uma página web com formulário HTML com um campo de entrada tipo file, onde o usuário poderá selecionar um arquivo para envio

FTP (File Transfer Protocol) – ocorre por meio de algum software que possua as funcionalidades de um cliente FTP.

Quando o arquivo é recebido pelo programa PHP localizado no servidor, podemos definir seu destino destas duas formas:

- 1 . apenas movendo para o diretório destino. Neste caso, utilizamos transferência por HTTP.
2. usando as funções do PHP relativas ao protocolo FTP para conectar-se a um servidor FTP e transferir esse arquivo.

Upload utilizando PHP e HTTP

Formulário HTML – A linguagem HTML disponibiliza entre seus tipos de campo o tipo file para o envio de arquivos.

upload.html

Somente método POST permite o envio de arquivos

O tipo de encoding de dados, enctype = "multipart/form-data">, DEVE ser especificado

```
<form method="POST" action="executa_upload.php" enctype="multipart/form-data">
```

```
<input type="hidden" name="MAX_FILE_SIZE" value="200000">
```

Foto: <input type="file" name="imagem">

```
<input type="submit" name="acao" value="Enviar Arquivo">
```

```
</form>
```

Campo tipo file que possibilita a escolha e o envio de arquivos

O campo oculto (input type="hidden") MAX_FILE_SIZE (medido em bytes) deve preceder o campo de input de arquivo (input type="file"), e contém o tamanho limite do arquivo a ser enviado.

Considerações sobre o formulário para envio de arquivos

Campo do tipo file – exibe uma caixa de texto e um botão para a seleção de um arquivo.

Na opção **action** da tag **form** especifica-se o nome do programa PHP que será ativado pelo formulário e que será responsável pela manipulação do arquivo enviado.

A opção **enctype** com o valor **multipart/form-data** – indica que irá ocorrer o upload de um arquivo para o servidor.

O campo oculto que define o parâmetro **MAX_FILE_SIZE** é utilizado pelo programa navegador para definir um tamanho máximo em bytes para o arquivo a ser enviado.

Essa é uma validação de tamanho de arquivo feita no lado cliente. Para validar o tamanho do arquivo no lado do servidor, o arquivo deve primeiro chegar lá. Portanto, o arquivo será primeiramente transferido para o servidor, para somente depois o programa detectar que seu tamanho é inválido.

➤ É importante que sejam feitas as duas validações, tanto no lado do cliente como no lado do servidor.

\$_FILES

No PHP, a variável global **\$_FILES** é uma matriz que conterà toda a informação do arquivo enviado. Esta variável é identificada pela propriedade **name** do input., que será utilizado como o primeiro índice da matriz. (obs: No formulário do exemplo, o **name** do campo file é “**imagem**” definindo a identificação da matriz **\$_FILES['imagem']**). O segundo índice diz respeito a algum tipo de informação sobre o arquivo: nome, tamanho, tipo, nome temporário em que o arquivo é recebido no servidor, código de erro.

\$_FILES['imagem']['name']

O nome original do arquivo na máquina do cliente.

\$_FILES ['imagem'] ['type']

O tipo mime do arquivo, se o navegador fornecer essa informação. Um exemplo poderia ser "image/gif". O tipo mime no entanto não é verificado pelo PHP portanto não considere que esse valor será concedido.

\$_FILES ['imagem'] ['size']

O tamanho, em bytes, do arquivo enviado.

\$_FILES ['imagem'] ['tmp_name']

O nome temporário com o qual o arquivo enviado foi armazenado no servidor.

\$_FILES ['imagem'] ['error']

O código de erro associado a esse upload de arquivo.



\$_FILES

Duas dimensões: \$_FILES[identificador do arquivo][propriedade]

Identificador do arquivo: nome do campo tipo file definido no formulário HTML

Propriedades:

name- o nome original do arquivo no computador do usuário

type – contém o tipo do arquivo. Por exemplo, para uma imagem GIF temos o valor image/gif.

size – tamanho do arquivo em bytes.

tmp_name – nome temporário que o servidor utilizou para armazenar o arquivo

error – armazena o código do erro em caso de falha na operação de upload do arquivo.

diferença entre o vetor \$_POST e a matriz \$_FILES

Portanto, ao enviar um arquivo pelo formulário podemos utilizar pelo php uma nova variável (além da \$_POST) chamada \$_FILES. Essa variável funciona de forma semelhante ao \$_POST. A diferença é que o \$_FILES traz várias informações sobre o arquivo enviado, sendo uma matriz com duas dimensões.

```
<?php
```

```
$login = $_POST['login'], // <input type='text' name='login'>
```

```
$nome_arquivo = $_FILES['imagem']['name']; // <input type='file' name='imagem'>
```

```
$tamanho_arquivo=$_FILES['imagem']['size'];
```

```
$arquivo_temporario=$_FILES['imagem']['tmp_name']
```

```
$erro_upload = $_FILES['imagem']['error'];
```

```
var_dump($_FILES) // comando que mostra todo o conteúdo da matriz
```

```
?>
```


Recebendo e movendo o arquivo

```
<?php
$nome_arquivo=$_FILES['imagem']['name'];
$tamanho_arquivo=$_FILES['imagem']['size'];
$arquivo_temporario=$_FILES['imagem']['tmp_name'];
if (!empty($nome_arquivo)) {

    if (move_uploaded_file($arquivo_temporario, "imagens/$nome_arquivo")) {

        echo " Upload do arquivo: “. $nome_arquivo.”concluído com sucesso”; }

    else {

        die( “Falha no envio do arquivo”); }
    }
else{
    die(“Selecione o arquivo a ser enviado”); }
?>
```

executa_upload.php - exemplos_aula4/exemplo_1

Função `move_uploaded_file()`

A função `move_uploaded_file()` usa dois argumentos: o primeiro é o nome do arquivo temporário e o segundo é local onde o arquivo será salvo (incluindo seu novo nome). No caso usamos o nome original do arquivo.

O retorno desta função terá um valor lógico (`true` ou `false`) que indica se o arquivo foi movido com sucesso ou não.

Um erro que ocorre com frequência é o de quando o sistema operacional não consegue gravar o arquivo no local de destino devido à ausência de permissão de escrita no diretório.

Exemplo de parâmetros de configuração para o upload

```
<?php
```

```
// limitar as extensões? (sim ou não)
```

```
$limitar_ext="sim";
```

```
//extensões autorizadas
```

```
$extensoes_validas=array(".gif",".jpg",".jpeg",".bmp");
```

```
// caminho onde os arquivos serão armazenados
```

```
$caminho="imagens";
```

```
// limitar o tamanho do arquivo? (sim ou não)
```

```
$limitar_tamanho="sim";
```

```
//tamanho limite do arquivo em bytes
```

```
$tamanho_bytes="200000";
```

```
// se já existir o arquivo indica se ele deve ser sobrescrito (sim ou não)
```

```
$sobrescrever="não";
```

```
?>
```

Validação dos parâmetros de configuração

<?php

//código a ser incluído antes do move_uploaded_file do “executa_upload.php”

```
include “config_upload.php”;
```

```
if($sobrescrever==“não” && file_exists(“$caminho/$nome_arquivo”))  
    die(“Arquivo já existe”);
```

```
if($limitar_tamanho==“sim” && ($tamanho_arquivo > $tamanho_bytes) )  
    die(“Arquivo deve ter o no máximo $tamanho_bytes bytes”);
```

```
$ext = strrchr($nome_arquivo , '.');
```

```
if (($limitar_ext == “sim”) && !in_array($ext,$extensoes_validas))  
    die(“Extensão de arquivo inválida para upload”);
```

?>

Exemplo upload com validação - exemplos_aula4/exemplo_2

Exemplo upload com banco de dados - exemplos_aula4/exemplo_3

Funções utilizadas na validação

```
<?php
```

```
//será interpretado como true se $var estiver vazia
```

```
if (empty($var))  
{  
    echo "A variável está vazia";  
}
```

```
//será interpretado como true se $var estiver definida (vazia ou não)
```

```
if (isset($var))  
{  
    echo "A variável está definida";  
}  
?>
```

Funções utilizadas na validação

file_exists — Checa se um arquivo ou diretório existe

```
<?php
```

//será interpretado como true se o arquivo existir na pasta apontada na variável \$caminho

```
$caminho="fotos";  
$nome_arquivo=$_FILES['imagem']['name'];
```

```
file_exists("$caminho/$nome_arquivo");
```

```
?>
```

Funções utilizadas na validação

```
<?php
```

```
$nome_arquivo="geladeira.jpg";
```

// extrai parte de uma string a partir do caracter disposto no segundo argumento da função
– no exemplo a partir do ponto ('.').

```
$ext = strrchr($nome_arquivo, '.');
```

```
echo $ext; // mostra jpg
```

```
?>
```

Funções utilizadas na validação

`in_array` — Checa se um valor existe em um array

Exemplo:

```
<?php
```

```
$vetor=array('laranja','maça','uva');
```

```
// retorna true se o primeiro argumento da função existir em $vetor
```

```
in_array('laranja',$vetor); // retorna true
```

```
in_array('melão',$vetor); // retorna false
```

```
?>
```