

PHP

1. Gerenciando o *upload* de arquivos

1.1. Upload de arquivos com o método POST

Este recurso permite realizar *uploads* de arquivos de texto e binários. Adicionalmente, utilizam-se funções para autenticação e manipulação de arquivos no PHP.

Ex#1 - Formulário para *upload* de arquivo.

```
<form action="envia_imagem.php" method="post" enctype="multipart/form-data">
    <input type="hidden" name="MAX_FILE_SIZE" value="<?php echo 1024 * 1024; ?>" />
    <input type="file" name="arquivo" id="arquivo" /><br>
    <input type="submit" value="Enviar"/>
    <input type="reset" value="Apagar" />
</form>
```

O campo escondido (`input type="hidden"`) com `MAX_FILE_SIZE` (medido em bytes) deve preceder o campo de input de arquivo (`input type="file"`), e o seu valor é o tamanho limite aceito pelo PHP. Este elemento do formulário deve sempre ser usado para evitar que usuários tenham o problema de ter que esperar pela transferência de um arquivo para só então descobrir que ele era grande demais e a transferência falhar.

OBS.: O atributo `enctype="multipart/form-data"` deve ser acrescentado ao `form` para o envio de arquivo.

PHP

O arquivo PHP "envia_imagem.php" conterá o algoritmo responsável por receber o *upload*. Nele são utilizadas diversas funções PHP que auxiliam nesse processo. O exemplo a seguir processa o upload de um arquivo que vem do formulário anterior.

Ex#2 – Processa o upload de arquivo recebido do formulário.

```
<?php
    $pasta_destino = 'imagens/';
    $arquivo_destino = $pasta_destino . basename($_FILES['arquivo']['name']);
    $arquivo_temp = $_FILES['arquivo']['tmp_name'];
    if (move_uploaded_file($arquivo_temp, $arquivo_destino)):
        echo "<p>Arquivo enviado com sucesso.</p>";
    else:
        echo "<p>Não foi possível fazer o upload do arquivo!</p>";
    endif;
?>
```

A variável global \$_FILES conterá toda a informação do arquivo enviado por meio do name="arquivo" do exemplo #1. A seguir, temos alguns campos que a variável \$_FILES armazena.

- * \$_FILES['arquivo']['name'] - o nome original do arquivo na máquina do cliente.
- * \$_FILES['arquivo']['tmp_name'] - o nome temporário com o qual o arquivo enviado foi armazenado no servidor.
- * \$_FILES['arquivo']['size'] - o tamanho, em bytes, do arquivo enviado.

PHP

* `$_FILES['arquivo']['error']` - o código de erro associado a esse upload de arquivo.

Como suporte ao *upload* de arquivos em PHP duas funções para sistema de arquivo foram utilizadas: `move_uploaded_file()` e `basename()`. A primeira função é responsável por mover um arquivo enviado para uma nova localização. Já a segunda, devolve a parte correspondente ao nome do arquivo em um caminho, ou seja, dado uma *string* contendo o caminho completo de um arquivo, essa função irá retornar apenas a parte que corresponde ao nome do arquivo.

1.2. Mensagens de erro durante o *upload* de arquivo

O PHP retorna um código de erro apropriado dentro do arquivo submetido. Esse código pode ser encontrado em `$_FILES['arquivo']['error']` do *array* que é criado pelo PHP durante o upload do arquivo.

Tipo de erro	Valor	Descrição do erro
UPLOAD_ERR_OK	0	Não houve erro, o <i>upload</i> foi bem sucedido.
UPLOAD_ERR_INI_SIZE	1	O arquivo enviado excede o limite definido na diretiva <code>upload_max_filesize</code> do <i>php.ini</i> .
UPLOAD_ERR_FORM_SIZE	2	O arquivo excede o limite definido em <code>MAX_FILE_SIZE</code> no formulário HTML.
UPLOAD_ERR_PARTIAL	3	O <i>upload</i> do arquivo foi realizado parcialmente.
UPLOAD_ERR_NO_FILE	4	Nenhum arquivo foi enviado.

PHP

UPLOAD_ERR_NO_TMP_DIR	6	Pasta temporária ausente.
UPLOAD_ERR_CANT_WRITE	7	Falha em escrever o arquivo em disco.
UPLOAD_ERR_EXTENSION	8	Uma extensão do PHP interrompeu o <i>upload</i> do arquivo.