



**INSTITUTO
FEDERAL**

Sudeste de
Minas Gerais

Campus
Manhuaçu

Desenvolvimento Web III

Upload de arquivos

Prof. Leonardo C. R. Soares - leonardo.soares@ifsudestemg.edu.br

Instituto Federal do Sudeste de Minas Gerais

22 de janeiro de 2025





Upload de arquivos

Para que possamos enviar arquivos (textos ou binários) utilizando formulários html, faz-se necessário que o formulário possua o atributo `enctype="multipart/form-data"`.





Upload de arquivos

Para que possamos enviar arquivos (textos ou binários) utilizando formulários html, faz-se necessário que o formulário possua o atributo `enctype="multipart/form-data"`.

O formulário pode conter um campo para limitar o tamanho máximo do arquivo que será enviado. Este campo deve ser nomeado como `MAX_FILE_SIZE` e deve preceder o campo *file*. O valor deste campo é dados em *bytes*. Esta configuração é apenas para o lado cliente. Do lado do servidor, esta configuração é feita no arquivo *php.ini*





Formulário para cadastro e upload

```
1 <form enctype="multipart/form-data" action="up.php" method="POST">
2     Nome <input type="text" name="nome" /><br>
3     <!-- MAX_FILE_SIZE deve preceder o campo de seleção do arquivo -->
4     <input type="hidden" name="MAX_FILE_SIZE" value="50000" />
5     Foto: <input name="foto" type="file" /><br>
6     <button type="submit">Cadastrar</button>
7 </form>
```





\$ _FILES

O vetor super global \$ _FILES contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.





\$ _FILES

O vetor super global \$ _FILES contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.

- ▶ \$ _FILES['userfile']['name']: Nome original do arquivo na máquina cliente.





\$ _FILES

O vetor super global \$ _FILES contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.

- ▶ \$ _FILES['userfile']['name']: Nome original do arquivo na máquina cliente.
- ▶ \$ _FILES['userfile']['type']: O tipo *mime* do arquivo (se o browser fornecer esta informação). Um exemplo de conteúdo poderia ser "image/gif".





\$ _FILES

O vetor super global `$ _FILES` contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.

- ▶ `$ _FILES['userfile']['name']`: Nome original do arquivo na máquina cliente.
- ▶ `$ _FILES['userfile']['type']`: O tipo *mime* do arquivo (se o browser fornecer esta informação). Um exemplo de conteúdo poderia ser "image/gif".
- ▶ `$ _FILES['userfile']['size']`: O tamanho do arquivo em *bytes*.





\$ _FILES

O vetor super global \$ _FILES contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.

- ▶ \$ _FILES['userfile']['name']: Nome original do arquivo na máquina cliente.
- ▶ \$ _FILES['userfile']['type']: O tipo *mime* do arquivo (se o browser fornecer esta informação). Um exemplo de conteúdo poderia ser "image/gif".
- ▶ \$ _FILES['userfile']['size']: O tamanho do arquivo em *bytes*.
- ▶ \$ _FILES['userfile']['tmp_name']: O nome do arquivo temporário no qual o arquivo foi armazenado no servidor.





\$ _FILES

O vetor super global `$ _FILES` contém todas as informações do arquivo enviado. O nome do objeto *file* enviado pelo formulário é utilizado como identificador do primeiro índice do vetor.

- ▶ `$ _FILES['userfile']['name']`: Nome original do arquivo na máquina cliente.
- ▶ `$ _FILES['userfile']['type']`: O tipo *mime* do arquivo (se o browser fornecer esta informação). Um exemplo de conteúdo poderia ser "image/gif".
- ▶ `$ _FILES['userfile']['size']`: O tamanho do arquivo em *bytes*.
- ▶ `$ _FILES['userfile']['tmp_name']`: O nome do arquivo temporário no qual o arquivo foi armazenado no servidor.
- ▶ `$ _FILES['userfile']['error']`: O código de erro associado ao arquivo enviado.





`$_FILES`

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:





`$_FILES`

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:

- ▶ 0: Não há erro.





\$_FILES

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:

- ▶ 0: Não há erro.
- ▶ 1: O arquivo excede o tamanho máximo definido na diretiva `upload_max_filesize` do `php.ini`





`$_FILES`

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:

- ▶ 0: Não há erro.
- ▶ 1: O arquivo excede o tamanho máximo definido na diretiva `upload_max_filesize` do `php.ini`
- ▶ 2: O arquivo excede o tamanho definido no formulário HTML.





`$_FILES`

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:

- ▶ 0: Não há erro.
- ▶ 1: O arquivo excede o tamanho máximo definido na diretiva `upload_max_filesize` do `php.ini`
- ▶ 2: O arquivo excede o tamanho definido no formulário HTML.
- ▶ 3: O upload do arquivo foi feito parcialmente.





\$_FILES

Os principais códigos de erro (`$_FILES['userfile']['error']`) são:

- ▶ 0: Não há erro.
- ▶ 1: O arquivo excede o tamanho máximo definido na diretiva `upload_max_filesize` do `php.ini`
- ▶ 2: O arquivo excede o tamanho definido no formulário HTML.
- ▶ 3: O upload do arquivo foi feito parcialmente.
- ▶ 4: Não foi feito o upload do arquivo.





Cadastro com foto

Primeiramente, vamos conferir se ocorreu algum erro ao mover o arquivo temporariamente para o servidor.

```
1 <?php
2 // Vetor com os tipos de erros de upload do PHP
3 $_ERRO[0] = 'Não houve erro';
4 $_ERRO[1] = 'O arquivo no upload é maior do que o limite do PHP';
5 $_ERRO[2] = 'O arquivo ultrapassa o limite de tamanho especificado no HTML';
6 $_ERRO[3] = 'O upload do arquivo foi feito parcialmente';
7 $_ERRO[4] = 'Não foi feito o upload do arquivo';
8 if ($_FILES['foto']['error'] != 0)
9 |     die("Não foi possível fazer o upload. Erro: <b>" . $_ERRO[$_FILES['foto']['error']] . "</b>");
```





Cadastro com foto

Para evitar arquivos potencialmente perigosos, vamos limitar as extensões que podem ser enviadas. Note o uso do operador idêntico na comparação com a função `array_search`. Ele é necessário para se evitar que o valor zero (o primeiro índice de um vetor) possa ser entendido como `false`.

```
10 // Vetor com as extensões permitidas
11 $extensoes = array('jpg', 'png', 'gif');
12 $pedacos = explode('.', $_FILES['foto']['name']);
13 $extensao = strtolower(end($pedacos));
14 if (array_search($extensao, $extensoes) === false) {
15     echo "Por favor, envie arquivos com as seguintes extensões: ";
16     for ($i=0;$i<count($extensoes);$i++)
17         echo $extensoes[$i]. " ";
18     die();
19 }
```





Banco de dados e tabela

```
mysql> use cadastro;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
{figures/up3.png}
Database changed
mysql> desc contatos;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra        |
+-----+-----+-----+-----+-----+-----+
| codigo| int           | NO   | PRI | NULL    | auto_increment |
| nome  | varchar(200)  | YES  |     | NULL    |               |
| foto  | varchar(200)  | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,01 sec)
```





Cadastro com foto

Feito as verificações, primeiramente vamos salvar o cadastro no banco de dados. Esta etapa tem que vir antes do *upload* pois usaremos o código gerado para renomear o arquivo.

```
20 try{
21     $con = new PDO("mysql:host=localhost;dbname=cadastro","root","dft001");
22     $con->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
23     $pst = $con->prepare("insert into contatos (nome) values (?)");
24     $pst->bindValue(1,$_POST['nome']);
25     $pst->execute();
26     $codigo = $con->lastInsertId();
27 }catch(PDOException $p){
28     echo $p->getMessage();
29 }
```





Cadastro com foto

Após o cadastro, tentaremos mover o arquivo do local temporário para o definitivo. Se tudo ocorrer bem, atualizaremos o registro na tabela com o nome do arquivo.

```
30 $uploadfile = "./images/" . $codigo . basename($_FILES['foto']['name']);
31 if (move_uploaded_file($_FILES['foto']['tmp_name'], $uploadfile)) {
32     try{
33         $pstmt = $con->prepare("update contatos set foto=? where codigo=?");
34         $pstmt->bindValue(1, $codigo . basename($_FILES['foto']['name']));
35         $pstmt->bindValue(2, $codigo);
36         $pstmt->execute();
37         echo "Cadastro efetuado com sucesso.";
38     }catch(PDOException $p){
39         echo $p->getMessage();
40     }
41 } else {
```





Cadastro com foto

Se algo der errado, avisaremos o usuário e o excluiremos da tabela.

```
41 } else {  
42     echo "Não foi possível enviar o arquivo. O cadastro não foi efetuado.";  
43     try{  
44         $pst = $con->prepare("delete from contatos where codigo=?");  
45         $pst->bindValue(1, $codigo);  
46         $pst->execute();  
47     }catch(PDOException $p){  
48         echo $p->getMessage();  
49     }  
50 }  
51 ?>
```





Exibindo

Com os arquivos salvos em um diretório conhecido e, com o nome dos mesmos salvos junto a cada registro, exibir uma listagem com as fotos é simples. No exemplo abaixo considere que `$dados` é um vetor de objetos contendo todos os registros da tabela:

```
<table>
  <?php foreach($dados as $dado){ ?>
    <tr><td>
      <b><?=$dado->nome;?></b><br>
      <figure>
        nome;?>" />
      </figure>
    <?php } ?>
</table>
```





Bibliografia complementar

- ▶ Documentação oficial
- ▶ W3School: PHP File Upload
- ▶ Linha de código: Upload de arquivos em PHP





Exercício

Você foi escolhido para desenvolver o mural da turma. Para isso, cria uma aplicação PHP que permita a cada aluno se cadastrar, informando, seu nome, email, foto e um pequeno texto com suas perspectivas para o futuro. Em uma página separada, monte o mural da turma. Exiba todos os alunos cadastrados, incluindo suas fotos. Seu sistema deverá ser orientado à objetos.



