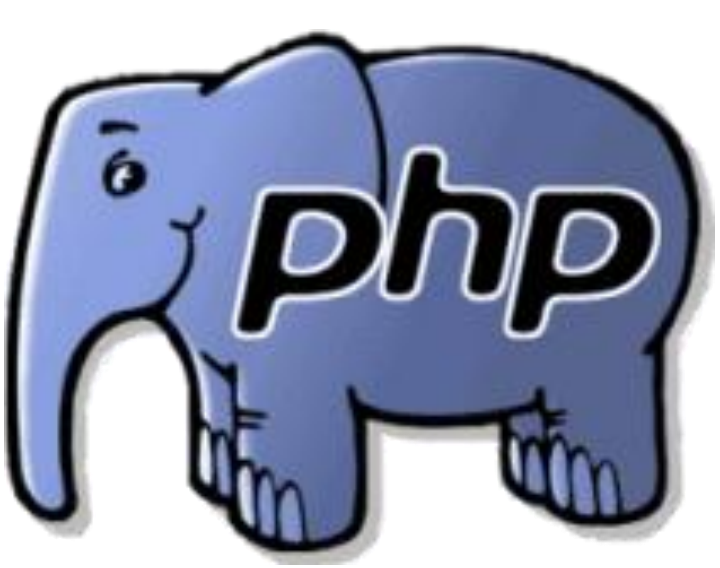


BuriFormação do Programador PHP
PHP Intermediário| Aula 1 – Upload de arquivos

Instrutor: Eder Martins Franco

v1.0

Previously...

- Curso de PHP Básico
  1. Introdução e conceitos básicos;
  2. Configurações de servidor e ambiente;
  3. Variáveis e constantes;
  4. Tipos de dados;
  5. Operadores;
  6. Estruturas de controle;
  7. Funções;
  8. Trabalhando com formulários;
  9. Enviando e-mails;
  10. Sessões;
  11. Cookies;
  12. Banco de dados (mysql);
- Atividade final;

Agenda

- **1. Upload de arquivos**
  - Introdução
  - Formulário HTML para upload
  - Diretivas do php.ini para upload
  - A super global $_FILES
  - Manipulação dos arquivos recebidos
  - Enviando múltiplos arquivos
  - Problemas comuns
  - Exercícios
  - Referências

Introdução

- O PHP é capaz de trabalhar com upload de arquivos de texto e binários, por meio de formulários HTML, em qualquer browser atual;
- O PHP também suporta o upload de arquivos por meio do método HTTP PUT. Nós estudaremos isso com mais detalhes em nossa aula sobre APIs REST.
- Existem algumas diretivas no php.ini relacionadas ao upload de arquivos, que devem ser verificadas durante a configuração do servidor;
- Normalmente, o upload de arquivos já vem habilitado no php.ini, sem precisar de configurações adicionais;

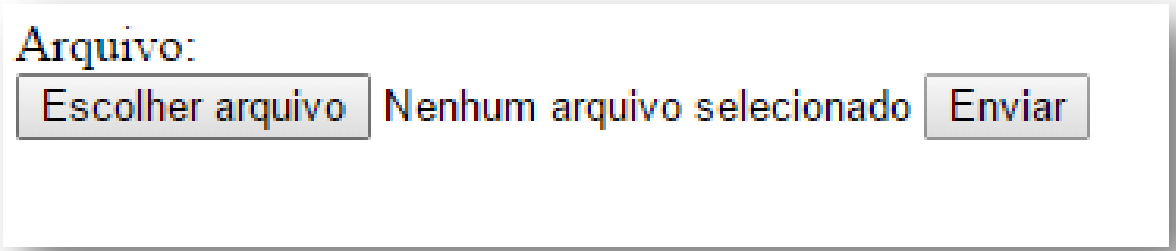
Formulário HTML para upload

- Já vimos anteriormente que um formulário HTML deve possuir uma script de destino (action) e um método HTTP;
- Um formulário HTML para upload de arquivos deve utilizar o método POST e além disso deve possuir, no mínimo, os seguintes itens:
  - Na tag <form>, deve ser adicionado o atributo enctype com o valor "multipart/form-data", para que o browser entenda que você deseja realizar um upload de arquivo no formulário atual;
  - Um input do tipo file, para que o browser possa criar um campo com um botão para abertura de uma janela do seu explorador de sistemas de arquivo, para identificação do arquivo que deverá ser enviado;

Formulário HTML para upload

- Exemplo:

```
1 <form enctype="multipart/form-data" action="enviar.php" method="post">
2     Arquivo:<br/>
3     <input name="userfile" type="file" />
4     <input type="hidden" name="MAX_FILE_SIZE" value="2048" />
5     <input type="submit" value="Enviar" />
6 </form>
```



Formulário HTML para upload

- No PHP Manual, é exemplificado que você pode criar um input do tipo hidden com o nome MAX_FILE_SIZE, cujo valor deve ser o tamanho máximo em MB que o arquivo carregado por este campo poderá ter;
- Isso ajuda o browser a controlar o tamanho máximo do arquivo, mas por se tratar de algo contido no HTML, pode ser facilmente contornado por usuários mais experientes;
- Uma tratativa de segurança deve ser a definição dos limites e configurações relacionados ao upload de arquivo no php.ini, como veremos a seguir.

Diretivas do php.ini para upload

- file_uploads
  - Habilita ou desabilita o upload de arquivos no PHP;
  - Vem habilitado por padrão;
- upload_tmp_dir
  - É o diretório padrão para onde o PHP envia os arquivos quando são recebidos via post;
  - Deve ser um diretório onde o servidor web tenha permissão para escrita;
  - Se não for definido, o PHP utilizará o diretório padrão, cujo local varia de acordo com a instalação do PHP;
  - No Wamp, por exemplo, este diretório geralmente é "C:/wamp/tmp";
  - Obs.: Este local não precisa ser necessariamente "conhecido" pelo desenvolvedor, pois pode ser acessado por meio de uma variável pré-definida do PHP;
- upload_max_filesize
  - É o tamanho máximo que um arquivo pode ter ao ser enviado por upload pelo PHP;
  - O valor padrão é de 2M;

Diretivas do php.ini para upload

- Outras diretivas que podem influenciar no upload de arquivos são as seguintes:

- post_max_size
  - O tamanho máximo de informações que podem ser enviadas por post, incluindo arquivos e textos;
  - O valor padrão é 8M;

- max_input_time
  - O tempo máximo que o PHP poderá levar para processar informações enviadas por POST;
  - Isso deve ser observado atentamente quando você permite que seu usuários envie arquivos muito grandes, o que pode levar bastante tempo;
  - O tempo padrão é de 60 segundos;
  - Recomenda-se também verificar as diretivas memory_limit e max_execution, que devem ser modificadas para arquivos muito grandes.

A super global $_FILES

- Já vimos antes que as informações enviadas por POST para um script PHP podem ser manipuladas por meio da super global $_POST;
- Para manipulação de arquivos, existe uma superglobal específica chamada $_FILES;
- A super global $_FILES é um *array associativo* que contém variáveis que nos fornecem informações sobre o nome do arquivo, tipo, o diretório temporário para onde foi carregado, seu tamanho em bytes e uma variável para controle de erros;
- A seguir, veremos a especificação de cada uma delas.

A super global $_FILES

- Primeiro, vejamos um exemplo no qual enviamos um arquivo de imagem por meio de um formulário HTML:

```
1 <form enctype="multipart/form-data" action="enviar.php" method="post">
2     Arquivo:<br/>
3     <input name="arquivo" type="file" />
4     <input type="submit" value="Enviar" />
5 </form>
```

Arquivo:
Escolher arquivo minha_imagem_batuta.jpg Enviar

A super global $_FILES

- Ao utilizarmos o var_dump() na super global $_FILES na página enviar.php, veremos o seguinte:

```
array (size=1)
  'arquivo' =>
    array (size=5)
      'name' => string 'minha_imagem_batuta.png' (length=23)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\php6C0B.tmp' (length=23)
      'error' => int 0
      'size' => int 408452
```

- Vamos especificar e entender cada uma das variáveis contidas no array recebido.

A super global $_FILES

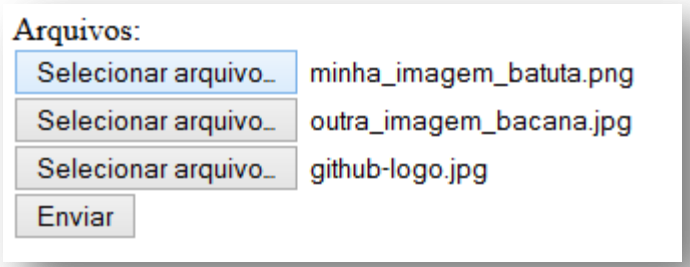
- $_FILES['userfile']
  - É o valor do atributo name do input do tipo file que foi definido no formulário HTML (o nome do campo);
  - Este primeiro índice do array $_FILES contém um novo array, com as informações do arquivo recebido;

```
1 <form enctype="multipart/form-data" action="enviar.php" method="post">
2     Arquivos:<br/>
3     <input name="arquivo" type="file" />
4     <input type="submit" value="Enviar" />
5 </form>
```

```
array (size=1)
  'arquivo' =>
    array (size=5)
      'name' => string 'minha_imagem_batuta.png' (length=23)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\php6C0B.tmp' (length=23)
      'error' => int 0
      'size' => int 408452
```

A super global $_FILES

- $_FILES['userfile']
  - Se tivermos vários inputs de arquivos no HTML, teremos vários índices com mesmo nome no array $_FILES, cada um com informações sobre um dos arquivos enviados:



A super global $_FILES

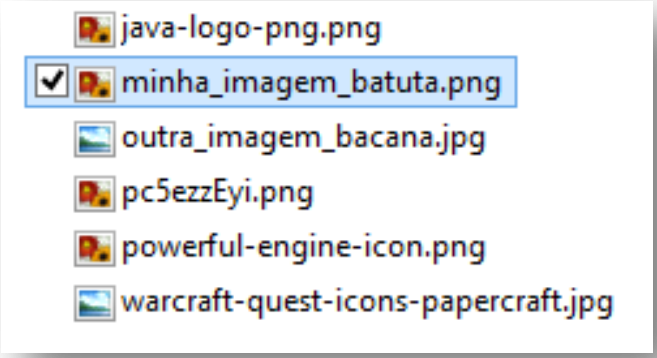
- $_FILES['userfile']
  - O resultado do envio de vários arquivos:

```html
<form enctype="multipart/form-data" action="enviar.php" method="post">
    Arquivos:<br/>
    <input name="arquivo" type="file" /><br/>
    <input name="arquivo2" type="file" /><br/>
    <input name="arquivo3" type="file" /><br/>
    <input type="submit" value="Enviar" />
</form>
```

```
array (size=3)
  'arquivo' =>
    array (size=5)
      'name' => string 'minha_imagem_batuta.png' (length=23)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\php2260.tmp' (length=23)
      'error' => int 0
      'size' => int 408452
  'arquivo2' =>
    array (size=5)
      'name' => string 'outra_imagem_bacana.jpg' (length=23)
      'type' => string 'image/jpeg' (length=10)
      'tmp_name' => string 'C:\wamp\tmp\php2261.tmp' (length=23)
      'error' => int 0
      'size' => int 79702
  'arquivo3' =>
    array (size=5)
      'name' => string 'github-logo.jpg' (length=15)
      'type' => string 'image/jpeg' (length=10)
      'tmp_name' => string 'C:\wamp\tmp\php2272.tmp' (length=23)
      'error' => int 0
      'size' => int 38128
```

A super global $_FILES

- $_FILES['userfile']['name']
  - É o nome original do arquivo no computador do usuário




```
array (size=3)
  'arquivo' =>
    'name' => string 'minha_imagem_batuta.png' (length=23)
    'tmp_name' => string 'C:\wamp\tmp\php2260.tmp' (length=23)
    'error' => int 0
    'size' => int 408452
```

A super global $_FILES

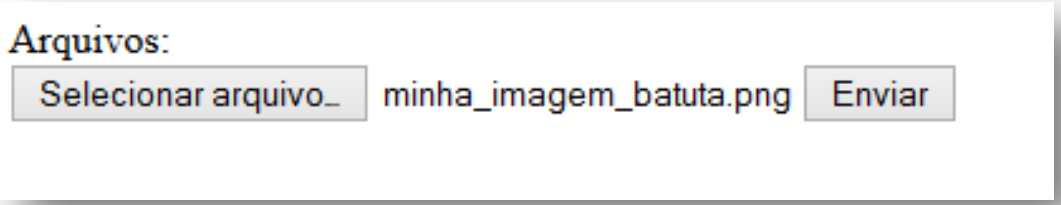
- $_FILES['userfile']['type']
  - Representa o mime type do arquivo enviado. Ou seja: o tipo de conteúdo do arquivo;
  - Neste caso, por se tratar de uma imagem, o mime type é image/jpeg;
- $_FILES['userfile']['size']
  - Contém o tamanho, em bytes, do arquivo;
- $_FILES['userfile']['tmp_name']
  - Contém o caminho completo para o local temporário para onde o arquivo foi enviado;
  - O arquivo neste local é removido ao sair do script, ou se o arquivo for movido para outro local, utilizando move_uploaded_file();

A super global $_FILES

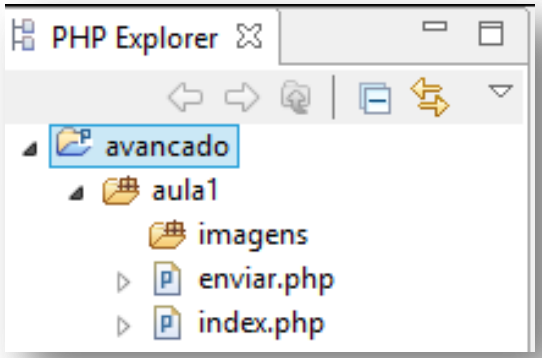
- $_FILES['userfile']['error']
  - Contém um código de erros que podem ocorrer durante o upload do arquivo, representados (a partir do PHP 4.3.0) pelas constantes abaixo:
    - UPLOAD_ERR_OK
      - Valor: 0; não houve erro, o upload foi bem sucedido.
    - UPLOAD_ERR_INI_SIZE
      - Valor 1; O arquivo no upload é maior do que o limite definido em upload_max_filesize no php.ini.
    - UPLOAD_ERR_FORM_SIZE
      - Valor: 2; O arquivo ultrapassa o limite de tamanho em MAX_FILE_SIZE que foi especificado no formulário HTML.
    - UPLOAD_ERR_PARTIAL
      - Valor: 3; o upload do arquivo foi feito parcialmente.
    - UPLOAD_ERR_NO_FILE
      - Valor: 4; Não foi feito o upload do arquivo.

Manipulação dos arquivos recebidos

- A função move_uploaded_file() nos permite mover o arquivo recebido para um novo diretório dentro da nosso projeto, recebendo como parâmetros o path do arquivo temporário recebido e um path para arquivo de destino:




```
array (size=3)
  'arquivo' =>
    array (size=5)
      'name' => string 'minha_imagem_batuta.png' (length=23)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\php2260.tmp' (length=23)
      'error' => int 0
      'size' => int 408452
```

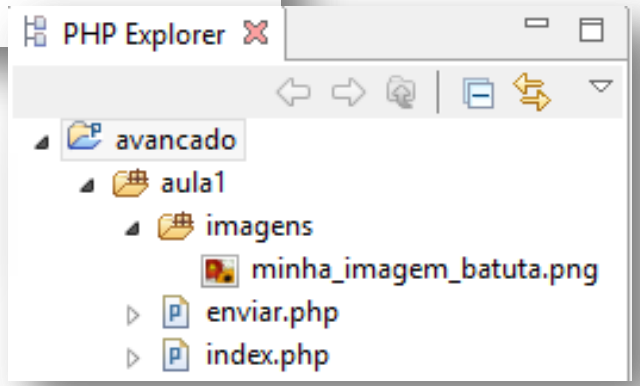


Manipulação dos arquivos recebidos

- Movendo o arquivo:

```php
<?php
//Este é o caminho completo para o arquivo temporário que o PHP criou
$arquivo_temporario = $_FILES["arquivo"]['tmp_name'];

//Definindo aqui que o arquivo irá para o diretório imagens
$diretorio_destino = "imagens/";

/* Aqui estou preservando o nome original do arquivo, mas
eu poderia sem problemas definir um novo nome */
$nome_arquivo= $_FILES["arquivo"]['name'];

//O arquivo de destino deve conter o path e o nome completos
$arquivo_destino = $diretorio_destino.$nome_arquivo;

//Movendo o arquivo
$retorno = move_uploaded_file($arquivo_temporario, $arquivo_destino);

var_dump($retorno);
```

```
boolean true
```

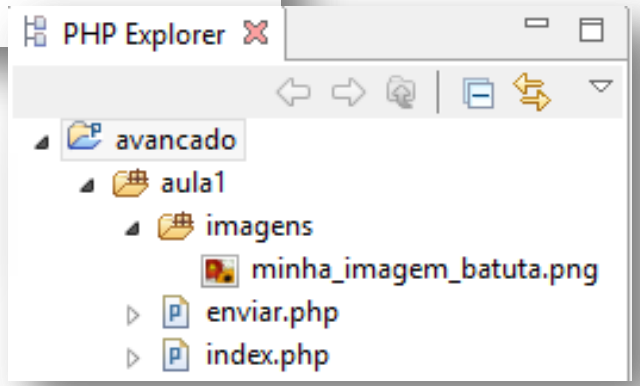


Manipulação dos arquivos recebidos

- Movendo o arquivo:

```php
<?php
//Este é o caminho completo para o arquivo temporário que o PHP criou
$arquivo_temporario = $_FILES["arquivo"]['tmp_name'];

//Definindo aqui que o arquivo irá para o diretório imagens
$diretorio_destino = "imagens/";

/* Aqui estou preservando o nome original do arquivo, mas
eu poderia sem problemas definir um novo nome */
$nome_arquivo= $_FILES["arquivo"]['name'];

//O arquivo de destino deve conter o path e o nome completos
$arquivo_destino = $diretorio_destino.$nome_arquivo;

//Movendo o arquivo
$retorno = move_uploaded_file($arquivo_temporario, $arquivo_destino);

var_dump($retorno);
```

```
boolean true
```



Manipulação dos arquivos recebidos

- A função move_uploaded_file retornará TRUE se o arquivo foi movido com sucesso;
- Se o path temporário passado para a função for inválido, a função retornará FALSE; Isso pode ser verificado com a função is_uploaded_file();
- Para verificar se qualquer path é válido (o de destino, por exemplo), você pode utilizar a função is_dir();
- Para verificar se o arquivo foi movido com sucesso e existe no local de destino indicado por você, as funções is_file() ou file_exists() podem ser utilizadas;

Manipulação dos arquivos recebidos

- Exemplos:

```php
<?php
//Este é o caminho completo para o arquivo temporário que o PHP criou
$arquivo_temporario = $_FILES["arquivo"]['tmp_name'];

echo "É um path válido do arquivo enviado?";
var_dump(is_uploaded_file($arquivo_temporario));

//Definindo aqui que o arquivo irá para o diretório imagens
$diretorio_destino = "imagens/";

echo "O diretório de destino é válido?";
var_dump(is_dir($diretorio_destino));

/* Aqui estou preservando o nome original do arquivo, mas
eu poderia sem problemas definir um novo nome */
$nome_arquivo= $_FILES["arquivo"]['name'];

//O arquivo de destino deve conter o path e o nome completos
$arquivo_destino = $diretorio_destino.$nome_arquivo;

//Movendo o arquivo
$retorno = move_uploaded_file($arquivo_temporario, $arquivo_destino);
echo "Arquivo modivo com sucesso?";
var_dump($retorno);

echo "O arquivo existe no diretório de destino?";
var_dump(file_exists($arquivo_destino));
```

É um path válido do arquivo enviado?

boolean true

O diretório de destino é válido?

boolean true

Arquivo modivo com sucesso?

boolean true

O arquivo existe no diretório de destino?

boolean true

Enviando múltiplos arquivos

- Para enviar múltiplos arquivos com PHP, você pode criar vários campos com nomes diferentes, ou criar somente um campo em forma de array no formulário HTML;
- A diferença será que, no primeiro caso, você manipulará o resultado utilizando os nomes dos campos criados, e no segundo caso você manipulará os campos pelos seus índices numéricos;
- A super global é um array com as mesmas propriedades de um array comum, e podemos manipulá-la de várias maneiras;
- Por exemplo, podemos percorrer o array $_FILES com um loop, para manipular vários arquivos em sequência;

Enviando múltiplos arquivos

- Exemplo 1:
  - Criando vários campos com nomes diferentes:

```html
<form enctype="multipart/form-data" action="enviar.php" method="post">
    Arquivos:<br/>
    <input name="arquivo1" type="file" /><br/>
    <input name="arquivo2" type="file" /><br/>
    <input name="arquivo3" type="file" /><br/>
    <input type="submit" value="Enviar" />
</form>
```

```
array (size=3)
  'arquivo1' =>
    array (size=5)
      'name' => string 'Apresentação1.pptx' (length=18)
      'type' => string 'application/vnd.openxmlformats-officedocument.presentationml' (length=60)
      'tmp_name' => string 'C:\wamp\tmp\phpFE67.tmp' (length=23)
      'error' => int 0
      'size' => int 670816
  'arquivo2' =>
    array (size=5)
      'name' => string 'fluxograma_marcela.psd' (length=22)
      'type' => string 'application/octet-stream' (length=24)
      'tmp_name' => string 'C:\wamp\tmp\phpFE77.tmp' (length=23)
      'error' => int 0
      'size' => int 1367191
  'arquivo3' =>
    array (size=5)
      'name' => string '35920.png' (length=9)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\phpFE66.tmp' (length=23)
      'error' => int 0
      'size' => int 2194
```

Enviando múltiplos arquivos

- Exemplo 2:
  - Criando vários campos com mesmo nome, como um array:

```
array (size=1)
  'arquivos' =>
    array (size=5)
      'name' =>
        array (size=3)
          0 => string '35920.png' (length=9)
          1 => string 'Apresentação1.pptx' (length=18)
          2 => string 'fluxograma_marcela.psd' (length=22)
      'type' =>
        array (size=3)
          0 => string 'image/png' (length=9)
          1 => string 'application/vnd.openxmlformats-officedocument.presentationml' (length=60)
          2 => string 'application/octet-stream' (length=24)
      'tmp_name' =>
        array (size=3)
          0 => string 'C:\wamp\tmp\php311D.tmp' (length=23)
          1 => string 'C:\wamp\tmp\php311E.tmp' (length=23)
          2 => string 'C:\wamp\tmp\php312F.tmp' (length=23)
      'error' =>
        array (size=3)
          0 => int 0
          1 => int 0
          2 => int 0
      'size' =>
        array (size=3)
          0 => int 2194
          1 => int 670816
          2 => int 1367191
```

```
1 <form enctype="multipart/form-data" action="enviar.php" method="post">
2     Arquivos:<br/>
3     <input name="arquivos[]" type="file" /><br/>
4     <input name="arquivos[]" type="file" /><br/>
5     <input name="arquivos[]" type="file" /><br/>
6     <input type="submit" value="Enviar" />
7 </form>
```

Enviando múltiplos arquivos

- Exemplo 3:
  - Tratando os arquivos recebidos com um loop:

```html
<form enctype="multipart/form-data" action="enviar.php" method="post">
    Arquivos:<br/>
    <input name="arquivo" type="file" /><br/>
    <input name="arquivo2" type="file" /><br/>
    <input name="arquivo3" type="file" /><br/>
    <input type="submit" value="Enviar" />
</form>
```

```php
<?php
var_dump($_FILES);
foreach($_FILES as $key => $arquivos){
    $arquivo_temporario = $arquivos['tmp_name'];
    $diretorio_destino = "imagens/";
    $nome_arquivo= $arquivos['name'];
    $arquivo_destino = $diretorio_destino.$nome_arquivo;

    echo "Movendo o arquivo: ".$arquivos['name'];
    $retorno = move_uploaded_file($arquivo_temporario, $arquivo_destino);

    echo "<br/>Arquivo modivo com sucesso?";
    var_dump($retorno);
}
```

Enviando múltiplos arquivos

- Exemplo 3:
  - Tratando os arquivos recebidos com um loop:

```
array (size=3)
  'arquivo' =>
    array (size=5)
      'name' => string '35920.png' (length=9)
      'type' => string 'image/png' (length=9)
      'tmp_name' => string 'C:\wamp\tmp\php78D3.tmp' (length=23)
      'error' => int 0
      'size' => int 2194
  'arquivo2' =>
    array (size=5)
      'name' => string 'Apresentação1.pptx' (length=18)
      'type' => string 'application/vnd.openxmlformats-officedocument.presentationml' (length=60)
      'tmp_name' => string 'C:\wamp\tmp\php78D4.tmp' (length=23)
      'error' => int 0
      'size' => int 670816
  'arquivo3' =>
    array (size=5)
      'name' => string 'fluxograma_marcela.psd' (length=22)
      'type' => string 'application/octet-stream' (length=24)
      'tmp_name' => string 'C:\wamp\tmp\php78E4.tmp' (length=23)
      'error' => int 0
      'size' => int 1367191
```

Movendo o arquivo: 35920.png
Arquivo modivo com sucesso?

boolean true

Movendo o arquivo: Apresentação1.pptx
Arquivo modivo com sucesso?

boolean true

Movendo o arquivo: fluxograma_marcela.psd
Arquivo modivo com sucesso?

boolean true

Problemas comuns

- Se o arquivo já existe no diretório de destino, com o mesmo mime type e nome, será sobrescrito pelo mais recente;
- Um problema que pode ocorrer é o seu browser ou servidor de destino possuírem algum tipo de tratativa para não permitir o recebimento ou envio de determinados tipos de arquivo (executáveis, por exemplo). Isso não é comum, mas pode acontecer;
- Outro problema comum é o PHP não possuir permissão para escrita no diretório de destino (a função move_uploaded_file() retornará false);

Problemas comuns

- Para evitar problemas de permissão, tenha certeza de que o PHP possui permissão de escrita na pasta de destino;
- No linux, por exemplo, você pode modificar as permissões de arquivo utilizando chmod;
- No windows, isso varia. Você pode, por exemplo, executar seu pacote de serviços (como o Wamp) em modo administrador, ou definir manualmente as propridades da pasta para o diretório onde seus arquivos web estão;
- O PHP possui uma função chmod(), que recebe um diretório ou arquivo e um número inteiro que representa as permissões de acesso. Entretanto, sua efetividade dependerá do nível de permissão que o usuário do próprio PHP tem na sua máquina.

Exercícios

- 1) Adicione um campo para envio de 2 arquivos no seu formulário de contato:
  - Os usuários somente podem enviar arquivos PDF e imagens;
  - Você deve verificar isso no seu arquivo de destino, e retornar uma mensagem de erro se qualquer tipo não permitido for recebido;
  - Você deve mover o arquivo PDF para um diretório em seu projeto chamado arquivos/pdf e as imagens para um diretório chamado arquivos/imagens;
  - O tamanho máximo dos arquivos enviados deve ser de 5MB;

Exercícios

- 2) [DESAFIO] Utilizando a classe PHP Mailer:
  - Envie pelo menos um dos arquivos recebidos por e-mail, para os mesmos destinatários da mensagem que você já estava enviado antes;
  - No corpo do e-mail, informe:
    - Nome do arquivo;
    - Tamanho;
    - Data de recebimento;
    - Tipo de arquivo;

Referências

- PHP – Upload de arquivos
  - http://php.net/manual/pt_BR/features.file-upload.php
  - http://php.net/manual/pt_BR/features.file-upload.post-method.php
- Códigos de erro e problemas comuns
  - http://php.net/manual/pt_BR/features.file-upload.errors.php
  - http://php.net/manual/pt_BR/features.file-upload.common-pitfalls.php
- Enviando vários arquivos
  - http://php.net/manual/pt_BR/features.file-upload.multiple.php
- is_uploaded_file() e move_upload_file()
  - http://php.net/manual/pt_BR/function.is-uploaded-file.php
  - http://php.net/manual/pt_BR/function.move-uploaded-file.php
- PHP chmod
  - http://php.net/manual/pt_BR/function.chmod.php
- Upload error codes
  - http://php.net/manual/pt_BR/features.file-upload.errors.php
- Mime type list
  - http://www.iana.org/assignments/media-types/media-types.xhtml
- Comando CHMOD e seus valores
  - http://www.vivaolinux.com.br/artigo/Entendento-o-comando-chmod

PHP Intermediário – Aula 1

Eder Martins Franco

efranco23@gmail.com

http://moodle.franco.eti.br

http://fb.me/edermartinsfranco