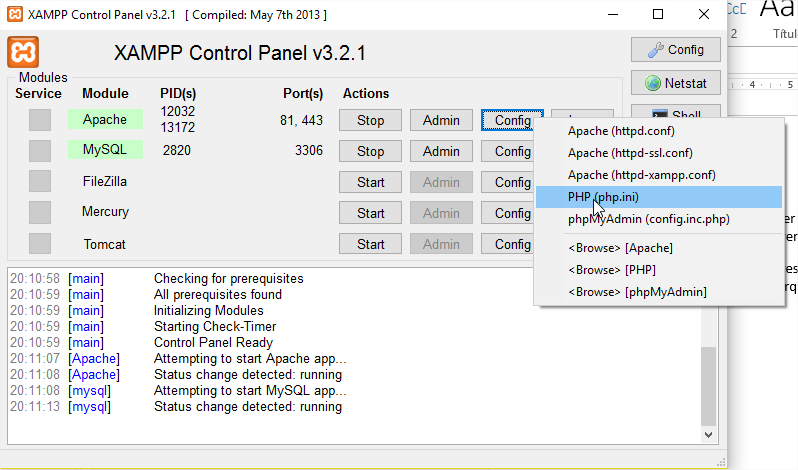
|  |  |
| --- | --- |
|  | **INST. FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS**  **Campus Ouro Branco**  **Disciplina:** Programação Web **Professor:** Márcio Assis Miranda  **Curso:** Tec. em Informática / Sistemas de Informação / Lic. em Computação |

**UPLOAD MÚLTIPLOS DE ARQUIVOS**

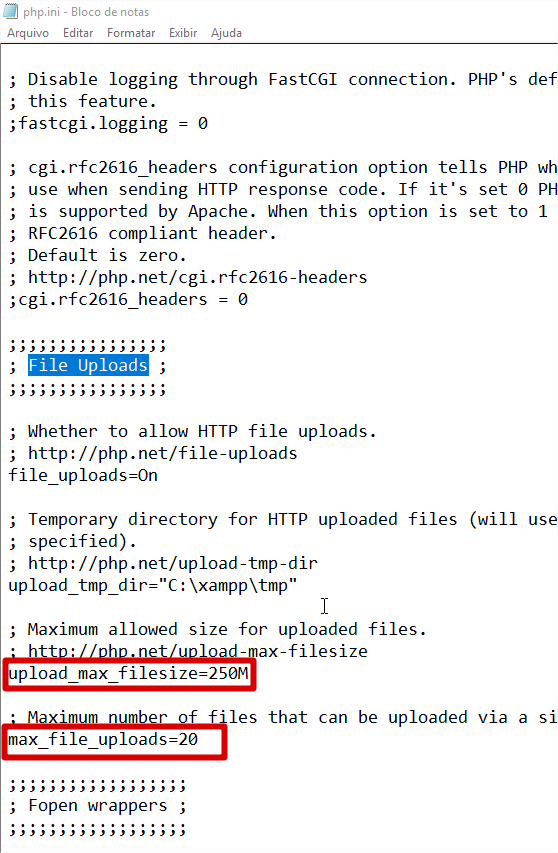
Esse tutorial apresenta a implementação de uma aplicação web com upload de múltiplos arquivos (vamos trabalhar com imagens).

Se você nunca implementou upload de arquivos, haverá a necessidade de conferir alguns parâmetros de comunicação no arquivo **php.ini**, conforme orientações a seguir.

1. Abra o **XamppServer** e posteriormente o arquivo **php.ini**.



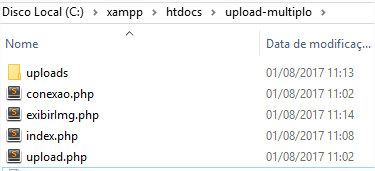
1. Procure (CTRL + F) pelo termo “file uploads” e altere os dois parâmetros em destaque, conforme dados abaixo na imagem. Os valores dos parâmetros vai depender dos arquivos em que você vai trabalhar nos uploads.



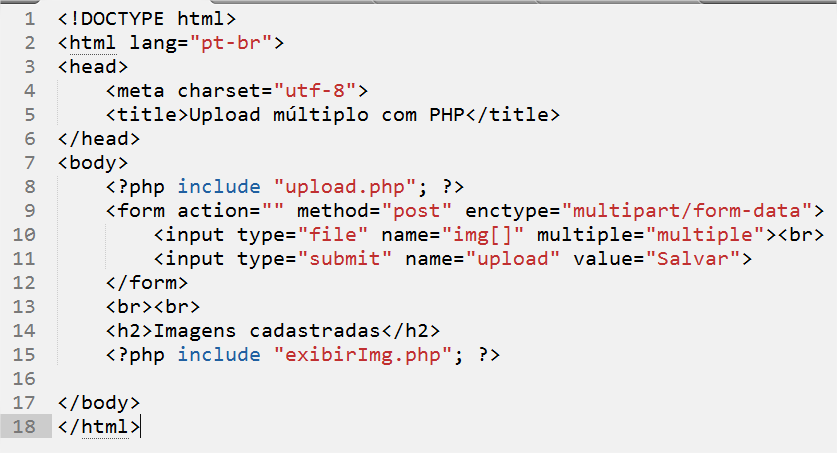
1. Salve o arquivo php.ini.
2. Reinicie o XamppServer.

**Agora crie a aplicação web para a realização dos uploads.**

1. Crie um banco de dados com o nome **upload-multiplo**. Crie uma tabela com o nome **tbFile** com os campos **idImg** (int, chave primária, auto incremento) e **nameImg** (varchar, tamanho 100, não nulo, índice único). Observe que no banco de dados vamos gravar apenas o nome e não o arquivo. O arquivo será gravado na pasta **uploads** que será criada conforme orientações a seguir.
2. Dentro da pasta **htdocs**, crie uma subpasta chamada **upload-multiplo** e dentro dessa, crie outra subpasta chamada **uploads**.
3. Dentro da pasta **upload-multiplo**, crie os arquivos **index.php**, **upload.php. exibirImg.php e conexão.php**, conforme a imagem abaixo.



1. Implemente o código abaixo no arquivo **index.php**.



**Linha 7:** inclui o arquivo **upload.php**, para carregar e executar aqui o script desse arquivo.

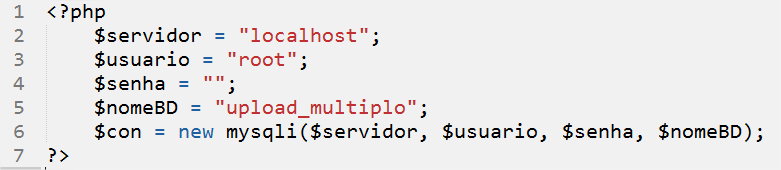
**Linha 8:** O atributo **enctype** específica como os dados do formulário devem ser codificados para enviá-lo ao servidor. O valor **multipart/form-data** é usado para quando for fazer upload de arquivos.

**Linha 9: <input type="file" name="img[]" multiple="multiple">**

- img[] – determina que seja enviada várias imagens, sendo que a variável de envio é um array.

- o atributo **multiple** permite escolher vários arquivos, ou seja, permite ao usuário selecionar várias imagens para o upload.

1. Implemente o código abaixo no arquivo **conexao.php**.



1. Implemente o código abaixo no arquivo **upload.php**.

**<?php**

//verifica se foi enviado algum valor

**if (isset($\_POST["upload"])) {**

//recebe os valores enviados pelo formulário

**$file = $\_FILES["img"];**

//permite debugar e ver o que foi enviado

//var\_dump($file);

//conta quantas imagens foram enviadas

**$numFile = count(array\_filter($file['name']));**

//define qual pasta a imagem será salva

**$folder = "uploads";**

//define os tipos suportados de arquivos enviados

**$permite = array("image/tif", "image/jpeg","image/png");**

**$maxSize = 1024 \* 1024 \* 5;**

//Mensagens

**$msg = array(); //cria um array vazio**

//cria um array e já atribui valores das mensagens a ele.

**$erro = array(**

**1 =>"O arquivo é maior que o limite definido no max\_filesize",**

**2 => "O arquivo ultrapassa o limite de tamanho permitido no MAX\_FILE\_SIZE",**

**3 => "O upload do arquivo foi feito parcialmente",**

**4 => "Não foi feito o upload do arquivo"**

**);**

**if ($numFile<=0) {**

**echo "Selecione pelo menos uma imagem.";**

**} else {**

**for ($i=0; $i < $numFile; $i++) {**

**$name = $file["name"][$i];** //pega o nome

**$type = $file["type"][$i];** //pega o tipo **$size = $file["size"][$i];** //pega o tamanho

**$error = $file["error"][$i];** //pega os erros

**$tmp = $file["tmp\_name"][$i];** //pega o nome temporário do arquivo quando ele está sendo passado do cliente para o servidor

**$extensao = @end(explode(".", $name));** //pega extensão de cada arquivo

//var\_dump($extensao); //para debugar e ver se está pegando a extensão dos arquivos

**$novoNome = rand().".$extensao";** //gera um novo nome único para os arquivos. Poderia também usar a função md5 do php

//abaixo estamos tratando as mensagens a serem exibidas para o usuário, em casos de erro ou sucesso no upload

**if ($error != 0) {** //se não houver erro ao carregar a imagem

**$msg[] = "<b>$name: </b>".$errorMsg[$error];**

**} else if(!in\_array($type, $permite)) {**

**$msg[] = "<b>$name: </b> Erro - Tipo de arquivo não suportado. Escolha arquivo do tipo .jpg e .png.";**

**} else if ($size > $maxSize) {**

**$msg[] = "<b>$name: </b>Erro - Tamanho do arquivo é maior que o permitido.";**

**} else {**

//move o arquivo para a pasta definida

**if (move\_uploaded\_file($tmp, $folder."/".$novoNome)) {**

//trecho para gravar a imagem na pasta e para gravar os dados na pasta da imagem no banco de dados

**include "conexao.php"; //inclui o arquivo de conexão com o banco de dados**

//define o comando sql para inserção do nome da imagem no banco de dados

**$SQL = "INSERT INTO tbimagem (nameImg) VALUES ('".$novoNome."')";**

**if ($con->query($SQL) === TRUE){** //verifica se o comando foi executado com sucesso

**$msg[] = "<b>$name: </b> Upload realizado com sucesso.";**

**}**

**else{**

//mensagem exibida caso ocorra algum erro na execução do comando sql

**$msg[] = "<b>$name: </b> Erro - ".$con->error;**

**}**

**} else {**

**$msg[] = "<b>$name: </b> Desculpe! Ocorreu um erro ao fazer upload da imagem.";**

**}**

**}**

**}**

//var\_dump($msg); //para debugar e ver o conteúdo da variável $msg

**foreach ($msg as $mensagem) {**

**echo $mensagem."<br>";**

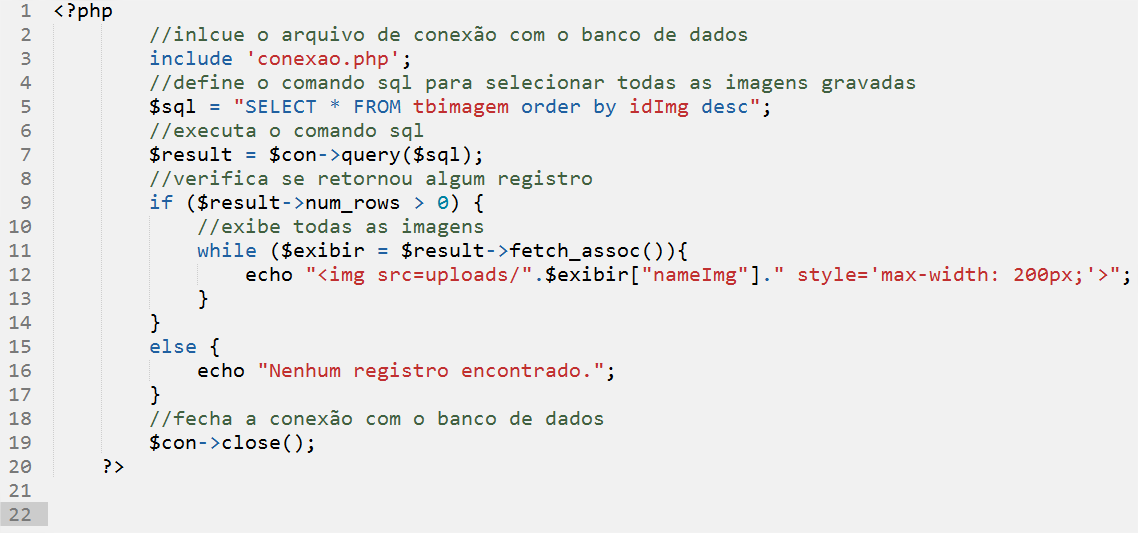
**}**

**}**

**}**

**?>**

1. Implemente o código abaixo no arquivo **exibirImg.php**.



1. Execute a aplicação e veja o resultado.
2. Realize os testes, inclusive com arquivos não suportados para ver se realmente as mensagens de erros estão sendo exibidas corretamente.
3. Entenda todo o código e adapte para a sua necessidade.

Bom trabalho!