

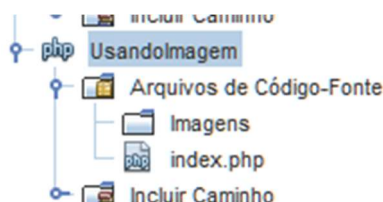
Preparando o projeto em NetBeans para armazenar Imagens

Neste projeto, utilizaremos a linguagem PHP para enviar imagens para o site.

Posteriormente, as imagens serão associadas a algum tipo de informação, como por exemplo, a foto de um aluno, as imagens de um produto em uma loja virtual ou qualquer outra situação na qual uma imagem pode ser utilizada.

O projeto no Netbeans precisa ter uma área para armazenar as imagens, esta área também existirá no servidor Web, portanto é muito importante que o servidor tenha espaço suficiente para o armazenamento das imagens.

Neste Projeto NetBeans, criei uma pasta chamada "Imagens", ela foi criada dentro da pasta "Arquivos de Código-fonte"



Obs. Para inserir uma pasta no projeto você deve escolher o local onde a pasta será criada, depois com um clique no botão direito do mouse escolha "Novo" e de pois "Pasta".

Pronto, o projeto está ajustado para podermos separar as imagens do restante do código do site.

Formulário para carregar um arquivo

A página que utilizaremos para inserir uma imagem também pode ser utilizada para inserir outros tipos de arquivos. O HTML não faz distinção se o arquivo é um Jpeg, Pdf, Icon, etc.

No entanto, o formulário que será utilizado para o envio de arquivos tem uma configuração diferente.

A tag form possui um atributo chamado "enctype", o qual não utilizamos até o momento, na verdade sempre utilizamos o valor padrão para ele, por isto, não o especificamos explicitamente.

Os valores para o atributo enctype são:

Valor	Descrição
application/x-www-form-urlencoded	Padrão (<i>Default</i>). Todos os caracteres são codificados antes do envio (Espaços são convertidos para o símbolo "+", caracteres especiais são convertidos para valores ASCII HEX).
multipart/form-data	Os caracteres não são codificados. Este valor é necessário quando você está usando formulários com tem controle para envio de arquivo.
text/plain	Espaços são convertidos para o símbolo "+", mas os caracteres especiais não são convertidos.

Ex. `<form action="nome_da_página_que_irá_processar_os_dados_do_formulário" enctype="multipart/form-data" method="post">`
`</form>`

OBS. O atributo enctype somente é utilizado com o método post, trecho destacado no exemplo acima.

A tag html utilizada para o envio de arquivo é a input com o tipo definindo com o valor file.

Ex. `<input type="file" name="nome_desta_entrada">`.

Configurando o PHP para aceitar arquivos.

É importante que o interpretador PHP tenha a configuração correta para aceitar os arquivos que serão enviados. Esta configuração deve considerar o local de armazenamento temporário do arquivo enviado, bem como o tamanho máximo que será aceito para cada arquivo.

Consulte o arquivo PHP.ini e localize dentro de dele a seção “File Upload”.

Esta seção apresenta 4 parâmetros que determinam como o PHP vai tratar o arquivos enviados.

- file_uploads=On
Onde On habilita o recebimento de arquivos, Off desabilita o recebimento de arquivos
- upload_tmp_dir="C:\xampp\tmp"
Local onde o arquivo enviado será armazenado temporariamente pelo PHP
- upload_max_filesize=40M
Tamanho máximo aceito para o upload, neste exemplo 40 Megabytes
- max_file_uploads=20
Quantidade máxima de arquivos que poderão ser enviados em cada solicitação

Os parâmetros acima também podem ser modificados durante a execução do programa PHP utilizando a função ini_set().

Esta função solicita o nome do parâmetro e o novo valor que será atribuído.

[Esta é a página de definição da função ini_set\(\)](#)

Recebendo um arquivo e gravando no servidor

A [documentação para upload de arquivo com o método POST](#) do manual do PHP.

Aqui explicaremos os códigos utilizados em projetos.

Para realizar o recebimento da imagem é necessário acessar a variável Global \$_FILES.

[\\$_FILES](#) é uma matriz e por este motivo é importante identificar o nome do índice que nos interessa.

Curso: Análise e Desenvolvimento de Sistemas
Disciplina: Linguagem de Programação IV (PHP)

No projeto, crie o arquivo "Index.php" que possuirá um formulário e dentro deste formulário existirá a tag input chamada "arquivo". Então para acessar o arquivo enviado, será necessário acessar `$_FILES["arquivo"]`, se no formulário o nome da input fosse "umalimagem", então seria usado `$_FILES["umalimagem"]`.

No arquivo "carregaArquivo.php" será realizado um teste com a função [isset\(\)](#) para determinar se a matriz `$_FILES` possui a informação enviada pelo formulário.

Uma vez que a informação é localizada, extraímos as propriedades do arquivo enviado, para isto serão criadas variáveis locais.

\$nomeTemporario	Recebe de <code>\$_FILES["arquivo"]["tmp_name"]</code> , onde "tmp_name" é o nome temporário gerado pelo servidor para armazenar o arquivo em um diretório de arquivos temporários.
\$nomeOriginal	Recebe de <code>\$_FILES["arquivo"]["name"]</code> , onde "name" é o nome original do arquivo que foi enviado para o servidor. O Servidor não utiliza o nome original para o armazenamento temporário para não sobrescrever outro arquivo que tenha o mesmo nome, porém com conteúdo diferente.
\$tamanho	Recebe de <code>\$_FILES["arquivo"]["size"]</code> , onde "size" indica o tamanho do arquivo que foi recebido. Neste exemplo, esta propriedade é utilizada para restringir o envio de arquivos muito grandes.
\$tipo	Recebe de <code>\$_FILES["arquivo"]["type"]</code> , onde "type" determina o tipo de arquivo que foi enviado. Se for necessário, pode-se verificar o tipo de arquivo e impedir a utilização de um tipo que não será utilizado.
\$erro	Recebe de <code>\$_FILES["arquivo"]["error"]</code> , onde "error" determinase outro erro no recebimento do arquivo. O código do arquivo "Receber_arquivo.php" foram usados todos os código de erros disponíveis.

A função [move_uploaded_file\(\)](#) é utilizada para mover o arquivo que está armazenado na pasta temporária do servidor para a pasta imagens do site.

Pode ser utilizado no código a concatenação do diretório de imagens mais a data e hora atuais acrescido do nome originalmente do arquivo fornecido para determinar o local de destino do arquivo. Assim é possível fazer o upload do mesmo arquivo diversas vezes e cada vez que o arquivo for enviado ganhará uma nova data/hora. Se deseja retirar esta possibilidade, basta remover a função [time\(\)](#) usada na concatenação do destino.

Linha de código do programa que permite enviar o mesmo arquivo diversas vezes.

`move_uploaded_file($nomeTemporario, $diretorioImagem.time().$nomeOriginal);`

Linha que sobrescreve arquivos com o mesmo nome, sem a função time().

`move_uploaded_file($nomeTemporario, $diretorioImagem.$nomeOriginal);`

Exibindo imagens em uma página a partir de um diretório

No arquivo "Index.php" acrescente o código abaixo depois da tag `</form>`.

```
<div>
    <form action="excluirArquivo.php">
```

Curso: Análise e Desenvolvimento de Sistemas
Disciplina: Linguagem de Programação IV (PHP)

```
<?php
const CAMINHO = "Imagens/";
$diretorio = opendir(CAMINHO);
while (($arquivo = readdir($diretorio)) !== false ){
    if (filetype(CAMINHO.$arquivo) === "file"){
        echo "<img src='" . CAMINHO . "$arquivo' height='300'/>";
        echo "<input type='checkbox' name='imagem[]' value = '$arquivo'/>";
    }
}
closedir($diretorio);
?>
<input type="submit" value="Excluir arquivos selecionados"/>
</form>
</div>
```

O código acima utiliza a função [opendir\(\)](#) para ter acesso ao diretório que armazena os arquivos. Neste caso a constante CAMINHO foi definida com o valor "Imagens/".

A variável \$diretorio recebe um manipulador de diretório obtido por opendir().

Esta variável \$diretorio é utilizada pela função [readdir\(\)](#) para acessar o conteúdo do diretório, retornando um arquivo que será armazenado na variável \$arquivo, encontrado no diretório e ficando pronto para acessar o próximo. Caso não encontre nenhum arquivo ou o final dos arquivos, a variável \$arquivo ficará com o valor false. Por isto que existe o teste com o valor FALSE no comando while.

Em particular, nesta linha do comando while.

Inicialmente é feita uma atribuição =>

\$arquivo = readdir(\$diretorio)

Após a atribuição é testado o valor de \$arquivo

(\$arquivo = readdir(\$diretorio)) !== false

Enquanto a variável \$arquivo contiver um valor diferente de *false* o processo de repetição do while permanecerá.

A função [filetype\(\)](#) é utilizada para saber se a informação lida por readdir() é um arquivo.

Consulte a documentação de [filetype\(\)](#) para verificar os retornos possíveis.

Neste caso, quando o type do arquivo for "file" o arquivo será carregado como um imagem.

OBS.: Neste exemplo, estou considerando que todos os arquivos enviados para a pasta Imagem são do tipo imagem, mas em uma aplicação final, é importante determinar o tipo do arquivo e fazer a carga utilizando o método adequado ao tipo de arquivo enviado.

Junto com a imagem, foi inserida a tag input para selecionar o arquivo, e neste caso, a seleção será usada para eliminar o arquivo do servidor.

Como está sendo utilizado a tag input, é claro que deve existir a tag form, caso contrário a operação não será processada.

OBS.: Note que a tag input possui a propriedade name com um conteúdo (nome) seguido de colchetes [], isto é feito para informar que todos os checkbox possuem o mesmo nome, e que devem ser tratados pelo php como um vetor.

Assim se duas ou mais imagens forem selecionadas, o php entenderá que o vetor `imagem[]` possuirá diversas posições, iniciando em zero.

Após exibir todas as imagens e percorrer todo o diretório, a função [`closedir\(\)`](#) é executada para liberar o manipulador de diretório.

Excluir arquivos de uma pasta

Para excluir um arquivo ou conjunto de arquivos do servidor é necessário indicar o local onde o arquivo está, para isto é necessário informar o nome do diretório (pasta) e o nome do arquivo.

A função [`unlink\(\)`](#) realiza a exclusão do arquivo.

No arquivo "excluirArquivo.php" realiza a exclusão de imagens que estão exclusivamente na pasta "Imagens".

```
const DIRETORIO = "Imagens/";  
if (isset($_REQUEST["imagem"])){  
    $imagens = $_REQUEST["imagem"];  
    foreach ($imagens as $key => $arquivo) {  
        unlink(DIRETORIO.$arquivo);  
    }  
}
```

```
header("Location:index.php");
```

Neste código-fonte, a constante `DIRETORIO` determina o local (pasta) onde o processo será executado.

A variável `$imagens` recebe por meio da variável global `$_REQUEST` a relação de imagens selecionadas.

O `foreach` percorre todas as posições do vetor `$imagens` e utiliza a função `unlink()` para remover o arquivo indicado.

Lembrando que a variável `$arquivo` é preenchida com o conteúdo do vetor `$imagens` a cada iteração do `foreach`.