

Tratamento de imagens

# Carregamento de imagens

- Usar um formulário HTML com o atributo `enctype="multipart/form-data"` para permitir o envio de arquivos de imagem.
- Acessar o arquivo enviado usando a variável superglobal `$_FILES`.
- Validar o arquivo para garantir que seja uma imagem válida (tipo MIME, tamanho, etc.).
- Mover o arquivo temporário para um local permanente no servidor usando `move_uploaded_file()`.

# Exibição de imagens:

- Usar a tag HTML <img> com o atributo src apontando para o caminho da imagem no servidor.
- Configurar corretamente as permissões de arquivo para que a imagem possa ser lida pelo servidor web

# Redimensionamento de imagens:

- Usar as funções GD ou Imagick para redimensionar, recortar ou manipular imagens.
- Utilizar funções como `imagecreatefromjpeg()`, `imagecreatefrompng()`, `imagecreatefromgif()` para criar um recurso de imagem a partir de um arquivo existente.
- Utilizar funções como `imagecopyresized()`, `imagecrop()`, `imagefilter()` para aplicar as transformações desejadas na imagem.
- Salvar a imagem resultante em um novo arquivo ou substituir o original.

# Redimensionamento de imagens:

- As funções GD(Graphics Library) e Imagick são bibliotecas populares em PHP usadas para manipular e processar imagens.
- Aqui estão alguns exemplos de como redimensionar, recortar e manipular imagens usando essas bibliotecas:
  - Exemplo\_gd
  - Exemplo\_imagick

- GD:

- A biblioteca GD é uma biblioteca gráfica de código aberto que fornece funções para criar, manipular e renderizar imagens em PHP.
- Ela suporta uma variedade de formatos de imagem, como JPEG, PNG, GIF, entre outros.
- Com o GD, você pode redimensionar, recortar, rotacionar, aplicar filtros, desenhar formas e textos, e muito mais.

- Imagick:
  - A biblioteca Imagick é uma extensão do PHP que fornece uma interface orientada a objetos para manipulação de imagens usando a biblioteca de software ImageMagick.
  - Ela oferece uma ampla gama de recursos avançados, como redimensionamento de alta qualidade, recorte preciso, ajustes de cor e contraste, aplicação de efeitos especiais, composição de várias imagens, entre outros.
  - A Imagick suporta uma variedade de formatos de imagem e oferece controle refinado sobre as operações de processamento de imagens.

# Funções de imagens

- Utilizar funções como `imagecreatefromjpeg()`, `imagecreatefrompng()`, `imagecreatefromgif()` para criar um recurso de imagem a partir de um arquivo existente nos formatos JPEG, PNG e GIF, respectivamente. Aqui está um exemplo de como usar essas funções:
- Criar um recurso de imagem a partir de um arquivo JPEG:
- `$imagem = imagecreatefromjpeg('caminho/para/imagem.jpg');`
- Criar um recurso de imagem a partir de um arquivo PNG:
- `$imagem = imagecreatefrompng('caminho/para/imagem.png');`



- Criar um recurso de imagem a partir de um arquivo GIF:
- `$imagem = imagecreatefromgif('caminho/para/imagem.gif');`
- Essas funções retornam um recurso de imagem que pode ser usado posteriormente para realizar manipulações e processamentos na imagem, como redimensionamento, recorte, aplicação de filtros, desenho de formas, entre outros.
- É importante ressaltar que, ao utilizar essas funções, é necessário ter a extensão GD habilitada no servidor PHP. Além disso, verifique se o caminho fornecido para o arquivo está correto e que o arquivo tenha permissão de leitura. Caso contrário, ocorrerá um erro durante a criação do recurso de imagem. Certifique-se também de liberar a memória após a conclusão do processamento da imagem usando a função **`imagedestroy()`**.

- As funções **imagecopyresized()**, **imagecrop()** e **imagefilter()** são utilizadas para aplicar diferentes transformações em uma imagem manipulada com a biblioteca GD. Aqui está um exemplo de como utilizar essas funções:
- Redimensionar uma imagem:
- `// Criar uma nova imagem redimensionada`
- `$novaLargura = 500;`
- `$novaAltura = 300;`
- `$novalmagem = imagecreatetruecolor($novaLargura, $novaAltura);`
- `// Copiar e redimensionar a imagem original para a nova imagem`
- `imagecopyresized($novalmagem, $imagemOriginal, 0, 0, 0, 0, $novaLargura, $novaAltura, $larguraOriginal, $alturaOriginal);`

- Recortar uma imagem:
- // Criar uma nova imagem recortada
- \$x = 100;
- \$y = 100;
- \$larguraRecorte = 200;
- \$alturaRecorte = 150;
- \$novaImagem = imagecrop(\$imagemOriginal, ['x' => \$x, 'y' => \$y, 'width' => \$larguraRecorte, 'height' => \$alturaRecorte]);

- Aplicar um filtro na imagem:
- `// Aplicar filtro de desfoque`
- `imagefilter($imagemOriginal, IMG_FILTER_GAUSSIAN_BLUR);`
- ssas são apenas algumas das transformações que podem ser realizadas na imagem usando as funções da biblioteca GD. Existem várias outras funções disponíveis para desenhar formas, aplicar efeitos, alterar cores, entre outros. Você pode explorar a documentação oficial do PHP para obter mais detalhes sobre as funções disponíveis e seus parâmetros.
- Lembre-se de liberar a memória após o uso do recurso de imagem utilizando a função **`imagedestroy($imagem)`**.

# Salvar Imagem

- Para salvar a imagem resultante em um novo arquivo ou substituir o arquivo original, você pode usar as funções **imagejpeg()**, **imagepng()** e **imagegif()** da biblioteca GD, dependendo do formato de imagem que deseja salvar. Aqui está um exemplo de como fazer isso:
- Salvar a imagem resultante em um novo arquivo JPEG:
- `$novoCaminho = 'caminho/para/nova_imagem.jpg';`
- `imagejpeg($novalmagem, $novoCaminho);`

# Como definir cabeçalho

- Para definir o cabeçalho apropriado e garantir que o navegador reconheça o arquivo como uma imagem, você pode utilizar a função **header()** do PHP. A função **header()** é usada para enviar um cabeçalho HTTP para o navegador. No caso de imagens, é necessário definir o tipo de conteúdo corretamente. Veja um exemplo:
  - Header.php

- Nesse exemplo, você deve substituir **'caminho/para/imagem.jpg'** pelo caminho e nome do arquivo de imagem que deseja exibir.
- Ao enviar o cabeçalho com o tipo de conteúdo adequado, o navegador irá reconhecer o arquivo como uma imagem e exibi-lo corretamente. Certifique-se de chamar a função **header()** antes de qualquer saída de conteúdo no PHP, caso contrário, pode ocorrer um erro de "headers already sent" (cabeçalhos já enviados).

- Lembre-se de que, ao usar essa abordagem, o arquivo de imagem será exibido diretamente no navegador e não será baixado como um arquivo. Se você deseja forçar o download da imagem, será necessário adicionar cabeçalhos adicionais, como **Content-Disposition** e **Content-Length**.



- Exportar para SVG: O formato SVG (Scalable Vector Graphics) é um formato de imagem baseado em XML que descreve imagens vetoriais. Para exportar uma imagem para SVG, você pode usar a biblioteca de manipulação de imagem Imagick, que suporta a exportação para esse formato. Você pode usar a função **writeImage()** para salvar a imagem no formato SVG. Aqui está um exemplo básico:

# Exportar imagens

- Para exportar uma imagem para outros formatos além dos formatos de imagem comuns (JPEG, PNG, GIF), como PDF ou SVG, é necessário utilizar bibliotecas ou extensões adicionais em PHP. Vou fornecer algumas informações sobre como realizar essa exportação para cada formato:
- Exportar para PDF:
  - Para exportar uma imagem para PDF, você pode usar bibliotecas como TCPDF ou FPDF.
  - Essas bibliotecas permitem criar documentos PDF e adicionar elementos gráficos, como imagens, texto e outros elementos.
  - Você precisará seguir as instruções e documentação específicas da biblioteca escolhida para realizar a exportação da imagem.

- Lembre-se de ter a biblioteca Imagick instalada e habilitada em seu servidor para poder utilizar essa funcionalidade.
- É importante mencionar que exportar imagens para formatos específicos, como PDF ou SVG, requer conhecimentos mais avançados de bibliotecas ou extensões específicas. Recomenda-se consultar a documentação oficial das bibliotecas ou extensões mencionadas para obter mais detalhes sobre as funcionalidades e possíveis configurações.

- Salvar a imagem resultante em um novo arquivo PNG:
- `$novoCaminho = 'caminho/para/nova_imagem.png';`
- `imagepng($novalmagem, $novoCaminho);`

- Salvar a imagem resultante em um novo arquivo GIF:
- `$novoCaminho = 'caminho/para/nova_imagem.gif';`
- `imagegif($novalmage, $novoCaminho);`

# Processamento de imagens:

- Aplicar filtros, efeitos ou ajustes às imagens, como brilho, contraste, saturação, etc.
- Usar as funções de manipulação de imagem para desenhar formas, adicionar texto ou sobrepor elementos gráficos.
- Salvamento e exportação de imagens:

# Filtros

- Para aplicar filtros, efeitos ou ajustes às imagens, como brilho, contraste, saturação, entre outros, você pode usar a função **imagefilter()** da biblioteca GD. A função **imagefilter()** permite aplicar uma variedade de filtros pré-definidos a uma imagem. Aqui está um exemplo de como utilizar essa função para ajustar o brilho, contraste e saturação de uma imagem:
- Exemplo\_gd/filtro.php
- Neste exemplo, a função **imagefilter()** é usada para aplicar os filtros de brilho, contraste e saturação à imagem. Os parâmetros **IMG\_FILTER\_BRIGHTNESS**, **IMG\_FILTER\_CONTRAST** e **IMG\_FILTER\_SATURATION** especificam o tipo de filtro a ser aplicado. Os valores passados após o tipo de filtro são os ajustes desejados para cada filtro.

# Processamento de imagens:

- Salvar a imagem resultante em um novo arquivo usando funções como `imagejpeg()`, `imagepng()`, `imagegif()`.
- Definir o cabeçalho apropriado para que o navegador reconheça o arquivo como uma imagem.
- Exportar a imagem em outros formatos, se necessário, como PDF ou SVG.



- Lembre-se de substituir '`caminho/para/nova_imagem.jpg`', '`caminho/para/nova_imagem.png`' e '`caminho/para/nova_imagem.gif`' pelos caminhos e nomes de arquivos desejados.

- Após aplicar os filtros, você pode salvar a imagem resultante em um novo arquivo usando a função **imagejpeg()** (ou **imagepng()** ou **imagegif()** para outros formatos). Não se esqueça de liberar a memória usando a função **imagedestroy()**.
- Você pode explorar outros filtros disponíveis na função **imagefilter()** para aplicar diferentes efeitos e ajustes às imagens, como desfoque, nitidez, sepia, entre outros.
- **Consulte a documentação oficial do PHP para obter mais detalhes sobre os filtros disponíveis e seus parâmetros.**

- Adicionar texto à imagem:
- // Definir a cor do texto
- `$textColor = imagecolorallocate($image, 0, 0, 255); // Azul`
  
- // Adicionar texto à imagem
- `$text = 'Exemplo de texto';`
- `imagettftext($image, 20, 0, 100, 150, $textColor, 'caminho/para/fonte.ttf', $text);`

- Com a biblioteca GD, você pode usar várias funções para manipular imagens, incluindo a capacidade de desenhar formas, adicionar texto e sobrepor elementos gráficos. Aqui estão alguns exemplos:
- Desenhar uma forma geométrica:
- `// Criar uma nova imagem`
- `$image = imagecreatetruecolor(500, 500);`
- `// Definir a cor de preenchimento`
- `$color = imagecolorallocate($image, 255, 0, 0); // Vermelho`
- `// Desenhar um retângulo preenchido`
- `imagefilledrectangle($image, 50, 50, 200, 200, $color);`

- Sobrepor uma imagem em outra:

- php

- // Carregar a imagem de fundo
- `$background = imagecreatefromjpeg('caminho/para/imagem_fundo.jpg');`
- // Carregar a imagem a ser sobreposta
- `$overlay = imagecreatefrompng('caminho/para/imagem_sobreposta.png');`
- // Sobrepor a imagem sobreposta na imagem de fundo
- `imagecopy($background, $overlay, 50, 50, 0, 0, imagesx($overlay), imagesy($overlay));`

# Bancos de imagens

- <https://pixabay.com/pt/>
- <https://br.freepik.com/fotos-vetores-gratis/banco-de>
- <https://br.depositphotos.com/>