

MÓDULO

PROGRAMAÇÃO PHP

UNIDADE

PROCESSAMENTO DE IMAGENS COM GD

ÍNDICE

OBJETIVOS.....	3
INTRODUÇÃO.....	4
1. INTRODUÇÃO.....	5
2. CARREGAR UMA IMAGEM PARA O SERVIDOR.....	6
3. FUNÇÕES PADRÃO EM GD	9
CONCLUSÃO.....	17
AUTOAVALIAÇÃO	19
SOLUÇÕES.....	23
PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO.....	24
BIBLIOGRAFIA	25

OBJETIVOS

Com esta unidade didática, pretende-se que desenvolva os seguintes objetivos de aprendizagem:

- Fazer upload de imagens para o servidor usando um formulário.
- Reconhecer as principais constantes e funções da biblioteca de processamento de imagens GD.
- Tratar de uma imagem enviada para o servidor.

INTRODUÇÃO

O PHP é frequentemente utilizado, não apenas para a criação de páginas web, mas também para a gestão das mesmas. Para isso, é possível aceder a dados SQL, modificá-los, excluí-los, criar sistemas de segurança de conexão, entre outros aspetos.

Nesta unidade aprenderá a fazer o upload e a processar imagens para o servidor, uma vez que muitas páginas contêm galerias de fotografias que devem ser processadas para terem um tamanho específico ou um formato definido.

1. INTRODUÇÃO

A capacidade de fazer upload de ficheiros para o servidor é uma funcionalidade que todas as linguagens de programação da web (PHP, ASP, JSP) possuem. A vantagem do PHP sobre as demais é que possui uma série de bibliotecas, como a GD (que é a que vai ser abordada nesta unidade), que permitem criar e modificar ficheiros do tipo imagem. Permitem modificar imagens nos formatos GIF, PNG e JPG e também permitem saber informações sobre o tamanho e medidas das imagens.

A biblioteca de processamento de imagem GD está incluída na instalação do PHP a partir da versão 4.3.

2. CARREGAR UMA IMAGEM PARA O SERVIDOR

Em PHP, como noutras linguagens de programação, é possível recolher um ficheiro de qualquer tipo, enviado de um formulário HTML preparado para enviar ficheiros. Para isso, deve ser criado com o valor do atributo **enctype**, que indica o tipo de dados enviados com o valor **multipart/form-data** e se é possível enviar ficheiros completos a partir do formulário. A sintaxe do formulário é, assim, a seguinte.

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

Neste formulário, além dos campos normais que pode incluir num formulário, também terá a opção de usar o tipo **file**, que permite incluir um campo com um botão “Escolher um ficheiro”, para percorrer unidades do computador e seleccionar um ficheiro para anexar ao formulário.

```
<input type="file" name="imagemoriginal" id="imagemoriginal"
size="30" />
```

Este campo vai conter o ficheiro que terá o mesmo nome que possui no sistema de directórios do computador.

O formulário irá chamar a página desejada, que receberá na variável global **\$_FILES** o **array** com o ficheiro e as suas características:

- **\$_FILES** [‘ficheiro’] [‘size’]: indica o tamanho em *bytes* do ficheiro recebido.

- \$_FILES ['ficheiro'] ['type']: indica o tipo MIME do ficheiro (a sua extensão).
- \$_FILES ['ficheiro'] ['name']: indica o nome original do ficheiro.
- \$_FILES ['ficheiro'] ['tmp_name']: indica o nome dado ao ficheiro temporariamente armazenado no servidor até que seja processado.

Com os valores recebidos no array e através da função copy do sistema de ficheiros PHP é possível completar um website que carregue ficheiros. A esta função será passado, como parâmetro, o valor temporário do nome do ficheiro e o valor final do nome que pretenda que tenha.

Exemplo

Este exemplo simples permite fazer upload de imagens diretamente para o servidor:

```
<?php
$ok = '';
if ($_POST["acc"] == "submeter") {
    // se vier do formulário, obter os dados do ficheiro
    $tamanho = $_FILES["ficheiro"]["size"];
    $tipo = $_FILES["ficheiro"]["type"];
    $ficheiro = $_FILES["ficheiro"]["name"];
    if ($ficheiro != "") {
        // salvar o ficheiro na raiz, desde que não esteja vazio, chamando
        // de 'novo_' e o nome que tinha no computador
        $destino = "/novo_".$ficheiro;
        if (copy($_FILES["ficheiro"]["tmp_name"],$destino)) {
            $ok = 'Ficheiro submetido: sim';
        } else {
            $ok = 'Ficheiro submetido: não';
        }
    } else {
        $ok = 'Nenhum ficheiro selecionado.';
    }
}
```

```
}  
?>  
<html>  
<body><p>  
<?php echo $ok ?>  
<form action="" method="post" enctype="multipart/form-data">  
<input name="ficheiro" type="file" size="35" />  
<input name="enviar" type="submit" value="Submeter ficheiro" />  
<input name="acc" type="hidden" value="submeter" />  
</form>  
</p></body>  
</html>
```

3. FUNÇÕES PADRÃO EM GD

A biblioteca GD fornece várias funções que permitem trabalhar com as imagens carregadas no servidor. Seguem-se as mais importantes:

- **Getimagesize (nome do ficheiro):** devolve um array com valores do tamanho da imagem:
 - O elemento 0 é a largura da imagem.
 - O elemento 1 é a altura da imagem.
 - O elemento 2 é uma constante IMAGETYPE_XXX e indica o tipo de imagem.
 - O elemento 3 é uma string com o valor height = "value" width = "value", muito útil para indicar nos elementos HTML img os tamanhos da imagem a ser carregada.

```
<?php
$tam = getimagesize('novo_DSCF0157.jpg');
echo ' largura: '.$tam[0].'<br/>';
echo 'altura: '.$tam[1].'<br/>';
echo 'tipo: '.$tam[2].'<br/>';
echo 'string: '.$tam[3].'<br/>';
?>
```

- **image_type_to_extension (constante IMAGETYPE):** devolve a extensão válida para o tipo de imagem passada como parâmetro.

```
<?php
echo image_type_to_extension(IMAGETYPE_PNG). '<br/>';
echo image_type_to_extension(IMAGETYPE_GIF). '<br/>';
echo image_type_to_extension(IMAGETYPE_JPEG). '<br/>';
?>
```

- **imagecreatefromjpeg (ficheiro):** cria uma nova imagem a partir de um ficheiro ou caminho URL de uma imagem em formato .jpg, devolvendo um identificador para poder trabalhar com a imagem selecionada.

```
$fonte = imagecreatefromjpeg('castelo.jpg');
```

- **imagecreatefromgif (ficheiro):** cria uma nova imagem a partir de um ficheiro ou caminho URL de uma imagem em formato .gif, devolvendo um identificador para poder trabalhar com a imagem selecionada.

```
$fonte = imagecreatefromjpeg('castelo.gif');
```

- **imagecreatefrompng (ficheiro):** cria uma nova imagem a partir de um ficheiro ou caminho URL de uma imagem em formato .png, devolvendo um identificador para poder trabalhar com a imagem selecionada.

```
$fonte = imagecreatefromjpeg('castelo.png');
```

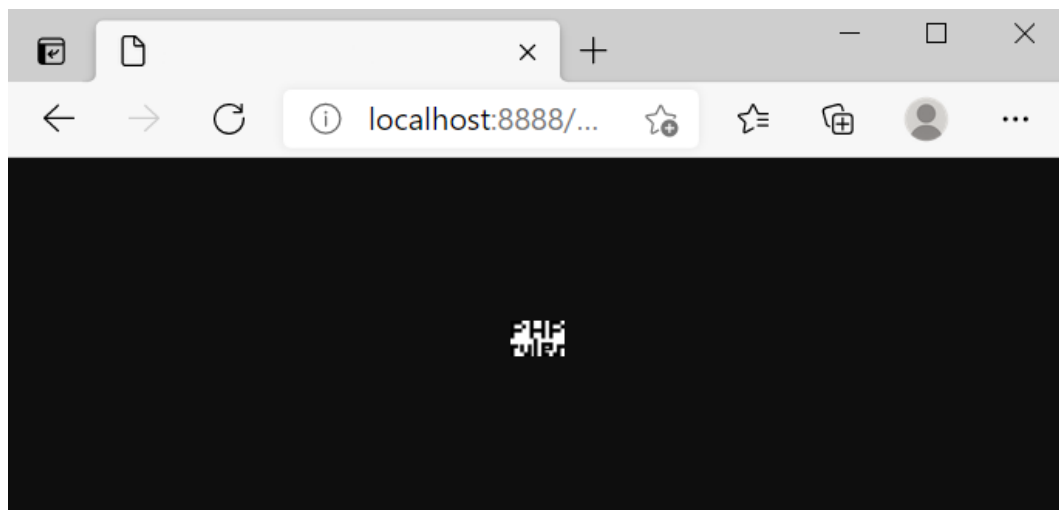
- **imagecreatefromstring (string):** cria uma imagem a partir de uma string de dados passada pelo parâmetro. Esta string deve ser o conteúdo físico no modo de texto de uma imagem:

```
<?php
$dados = 'iVBORw0KGgoAAAANSUhEUgAAABwAAAASCAMAAAB/2U7WAAAAB1' .
'BMVEUAAAD///+l2Z/dAAAASU1EQVR4XqWQUQoAIAxC2/0vXZDr' .
'EX4IJTRkb7lobNUStXsB0jIXIAMSsQnWlsV+wULF4Avk9fLq2r' .
'8a5HSE35Q3e02XP1A1wQkZSgETvDtKdQAAAABJRu5ErkJggg==';

$dados = base64_decode($dados);
$im = imagecreatefromstring($dados);
```

```
if ($im !== false) {  
    header('Content-Type: image/png');  
    imagepng($im);  
    imagedestroy($im);  
}  
else {  
    echo 'Ocorreu um erro.';  
}  
?>
```

□ Resultado:



- **imagedestroy (identificador da imagem):** liberta a memória associada à imagem passada por parâmetro.
- **imagejpeg (identificador da imagem):** envia uma imagem do formato .jpeg para a janela do navegador ou para um ficheiro; por defeito, envia-a para o navegador como se abrisse uma imagem com ela. Se pretender criar uma nova imagem, é passado o caminho completo, incluindo o nome da nova imagem, como parâmetro.
- **imagegif (identificador de imagem):** envia uma imagem do formato .gif como a função anterior.

- **imagepng (identificador da imagem):** envia uma imagem do formato .png como a função anterior.
- **imagecreate (altura, largura):** cria uma imagem com as medidas enviadas pelos parâmetros e a paleta padrão.

```
$destino = imagecreate (180, 140);
```

- **imagecreatetruecolor (altura, largura):** cria uma imagem com as medidas enviadas por parâmetros e a paleta de cores reais (preto).

```
$destino = imagecreatetruecolor (180, 140);
```

- **imagegammaconvert (identificador de imagem, entrada gama, saída gama):** permite aplicar uma correção gama à imagem passada por parâmetro.

```
<?php
// Criar instâncias de imagem
$fonte = imagecreatefromjpeg('castelo.jpg');
// Gama, saída correta = 1.937
imagegammaconvert($fonte, 1.0, 1.937);
imagejpeg($fonte, 'castelo2.jpg');
?>
```

□ Resultado:



À esquerda está a imagem original, à direita está a imagem com a gama corrigida.

- **imagecopy** (**destino, origem, destino_x_coordenar, destino_y_coordenar, origem_x_coordenar, origem_y_coordenar, largura_de_origem, altura_de_origem**): copia parte de uma imagem dependendo dos dados passados por parâmetro onde é indicada a imagem de origem, a imagem de destino, as coordenadas a copiar e o tamanho que terá.

```
<?php
// Criar instâncias da imagem
$fonte = imagecreatefromjpeg('castelo.jpg');
$destino = imagecreatetruecolor(250, 250);
// Copiar
imagecopy($destino, $fonte, 0, 0, 1750, 1200, 250, 250);
// Imprimir e libertar
header('Content-Type: image/jpeg');
imagejpeg($destino);
?>
```

□ Resultado:



À esquerda está a imagem original, à direita a imagem cortada.

- **imagefilter** (**identificador de imagem, filtro, opção1, opção2, opção3, opção4**): esta função permite aplicar um filtro à imagem passada e várias opções não obrigatórias. Os valores possíveis para cada parâmetro são:

□ Para filtro:

IMG_FILTER_NEGATE	Inverte as cores da imagem.
IMG_FILTER_GRAYSCALE	Converte para escala de cinza.
IMG_FILTER_BRIGHTNESS	Altera o brilho da imagem (usa a opção1 para indicar o nível).
IMG_FILTER_CONTRAST	Muda o contraste da imagem (usa a opção1 para indicar o nível).
IMG_FILTER_COLORIZE	O mesmo que passar para tons de cinza, mas definindo uma cor; opção1, opção2 e opção3 são usadas para dar a cor RGB Red (vermelho), Green (verde), Blue (azul) e opção4 para o alfa (os intervalos de cada cor vão de 0 a 255).
IMG_FILTER_EDGEDETECT	Destaca as bordas da imagem.
IMG_FILTER_EMBROSS	Coloca a imagem em relevo.
IMG_FILTER_GAUSSIAN_BLUR	Desfoca a imagem.
IMG_FILTER_SELECTIVE_BLUR	Desfoca a imagem.
MG_FILTER_MEAN_REMOVAL	Adiciona um efeito superficial à imagem.
IMG_FILTER_SMOOTH	Suaviza a imagem (usa a opção1 para indicar o nível de suavização).
IMG_FILTER_PIZELATE	Aplica um filtro de pixelização (usa a opção1 para dizer o tamanho do pixel e a opção2 para dizer o modo de pixelização).

□ Para a opção1:

IMG_FILTER_BRIGHTNESS	Nível de brilho.
IMG_FILTER_CONTRAST	Nível de contraste.
IMG_FILTER_COLORIZE	Valor de RED (vermelho).
IMG_FILTER_SMOOTH	Nível da suavidade.
IMG_FILTERPIXELATE	Tamanho do pixel.

□ Para opção2:

IMG_FILTER_COLORIZE	Valor de GREEN (verde).
IMG_FILTER_PIXELATE	Usa ou não o efeito de pixelização avançado (false, por defeito).

□ Para opção3:

IMG_FILTER_COLORIZE	Valor de BLUE (azul).
---------------------	-----------------------

□ Para opção4:

IMG_FILTER_COLORIZE	Valor alfa entre 0 e 127, sendo 0 totalmente opaco e 127 transparente.
---------------------	--

```
<?php
$fonte = imagecreatefromjpeg('castelo.jpg');
imagefilter($fonte, IMG_FILTER_GRAYSCALE);
imagejpeg($fonte, 'castelo3.jpg');
?>
```



À esquerda está a imagem original, à direita a imagem depois do filtro.

- **imagecopyresized** (imagem alvo, imagem fonte, alvo x coordenada, alvo e coordenada, fonte x coordenada, fonte e coordenada, largura alvo, altura alvo, largura fonte, altura fonte): esta função terá uma área retangular indicada por um parâmetro de largura e altura e irá colocá-lo numa nova imagem com as suas coordenadas e tamanhos enviados. Se as coordenadas ou tamanhos de origem e destino não coincidirem, será efetuado um aumento ou redução da peça da imagem selecionada.
- **imagerotate** (imagem, ângulo, cor): esta função permite girar uma imagem num determinado ângulo e preencher as lacunas restantes com a cor especificada.

```
<?php
$fonte = imagecreatefromjpeg('castelo.jpg');
$rodar = imagerotate($fonte, 74, 255);
imagejpeg($rodar, 'castelo4.jpg');
?>
```

□ Resultado:



À esquerda está a imagem original, à direita a imagem com rotação.

CONCLUSÃO

A biblioteca PHP GD permite controlar as imagens carregadas no servidor. Graças às suas funções predefinidas, é possível alterar tamanhos, cores e até aplicar filtros às imagens.

Agora já será capaz de criar galerias de imagens interativas nas quais as imagens podem ser carregadas, os seus tamanhos alterados e até mesmo aplicados efeitos.

AUTOAVALIAÇÃO

1. **A biblioteca de processamento de imagem GD está incluída na instalação do PHP?**
 - a) Apenas a partir da versão 5.
 - b) Não, é instalada separadamente.
 - c) Sim.
 - d) Apenas a partir da versão 4.3.

2. **Qual das seguintes linguagens não possui um método de upload de ficheiros para o servidor?**
 - a) ASP.
 - b) PHP.
 - c) JSP.
 - d) Nenhuma, todas as linguagens possuem um método de upload.

3. **Qual é a principal vantagem da biblioteca PHP GD sobre as outras linguagens de programação?**
 - a) Vem logo instalada com o PHP.
 - b) Permite modificar tamanhos e tratar as imagens com filtros, etc.
 - c) Permite enviar várias imagens ao mesmo tempo.
 - d) Não usa o JavaScript para fazer upload de imagens como ASP ou JSP.

4. Qual é o valor do atributo enctype que deve ser dado ao formulário para poder fazer o upload de imagens?
 - a) multipart/form-data.
 - b) multipart-file.
 - c) multipart/file-upload.
 - d) multipart/data.

5. Os ficheiros enviados do formulário são recolhidos em PHP com a função global:
 - a) \$_FILES.
 - b) \$_FILE.
 - c) \$FILES.
 - d) \$_ARCHIVES.

6. Qual das seguintes opções não é um elemento do array PHP que recolhe dados de ficheiros enviados por um formulário?
 - a) type.
 - b) name.
 - c) size.
 - d) tmp.

7. Qual é a função do sistema de ficheiros PHP que permite copiar um ficheiro de origem para outro destino?
 - a) copyfile.
 - b) copy.
 - c) copy_file.
 - d) copy-file.

- 8. O que é que devolve a função GD `getimagesize`?**
- a) A largura e a altura da imagem.
 - b) A altura da imagem.
 - c) O tipo de imagem em valor constante.
 - d) A altura, a largura e o tipo de imagem em valor constante.
- 9. Qual é a função GD que devolve uma extensão válida para o tipo de imagem enviada como um valor constante?**
- a) `image_type`.
 - b) `const_to_image_type`.
 - c) `image_type_to_extension`.
 - d) `image_type_to_mime`.
- 10. Com que função define um identificador numa imagem física em formato .jpeg?**
- a) `imagejpeg`.
 - b) `imagefromjpeg`.
 - c) `imagecreatefromjpeg`.
 - d) `Imagecreatejpeg`.

SOLUÇÕES

1.	d	2.	d	3.	b	4.	a	5.	a
6.	d	7.	b	8.	d	9.	c	10.	c

PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

Para conhecer todas as funções da biblioteca GD, veja o seguinte link:

- http://www.php.net/manual/pt_BR/ref.image.php

BIBLIOGRAFIA

- Cabezas, L. M. (2010), *Php 5*. Madrid: Anaya Multimedia.
- The PHP Group (2001), "Processamento de Imagem(GD)". Disponível em:
https://www.php.net/manual/pt_BR/book.image.php
Consultado a 3 de abril de 2021.

