

MÓDULO

PROGRAMAÇÃO PHP

PRÁTICA 32

EDIÇÃO DE IMAGENS COM PHP ATRAVÉS DE BIBLIOTECA GD

EDIÇÃO DE IMAGENS COM PHP ATRAVÉS DE BIBLIOTECA GD

1.1. DESCRIÇÃO DA PRÁTICA

Tempo estimado: 1 h.

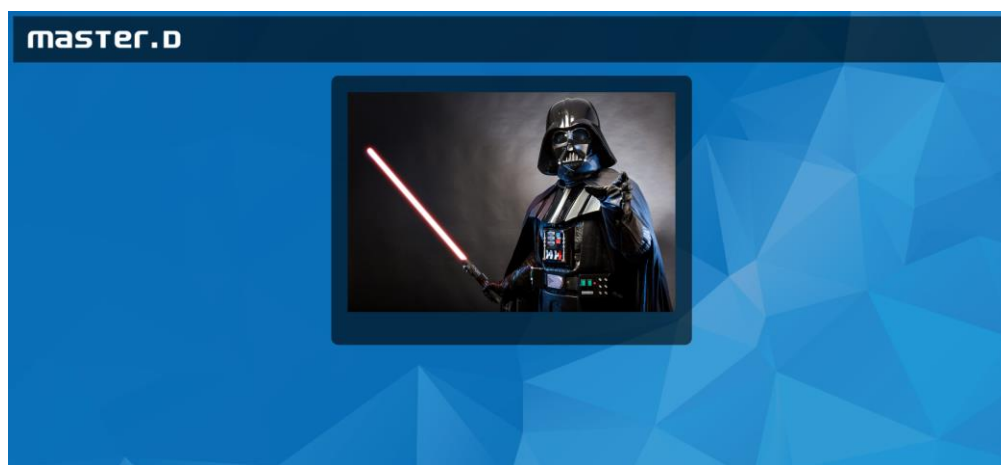
REQUISITOS
■ Ter concluído a unidade didática “Processamento de imagens com GD”.
OBJETIVOS
■ Aprender a aplicar dinamicamente marcas de água em imagens através do PHP e da biblioteca GD.
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
■ Editor de código. ■ Servidor web.

1.2. DESENVOLVIMENTO DA PRÁTICA

Nesta prática, vai ser abordada uma forma mais específica de editar as imagens com o PHP através da biblioteca GD, ou seja, não se limitará a cortar, redimensionar ou aplicar filtros, mas aprenderá a adicionar marcas de água à imagem.

Assim, criar-se-ão duas marcas de água diferentes: a primeira criando uma marca de água translúcida usando PHP com `imagecopymerge()` e a segunda criando uma marca de água com uma imagem.

Para a realização desta prática deverá descarregar a pasta de ficheiros de apoio disponibilizada no seu Campus Virtual e utilizar ficheiro PHP e o respetivo CSS, bem como uma imagem à sua escolha. Deve obter inicialmente um aspeto semelhante ao seguinte:



Foi colocada uma imagem de teste apenas para representar onde ficará a imagem com a marca de água.

Todo o processo de acrescentar a marca de água é realizado dentro de uma tag PHP. Por isso, dentro da mesma, irá começar por definir a imagem original onde será desenhada a marca de água com a função `imagecreatefromjpeg()`. A extensão da imagem estará de acordo com a imagem por si escolhida. No caso de ser uma imagem com extensão jpeg, deverá usar a função `imagecreatefromjpeg()`.

```
$image = imagecreatefrompng('darthvader.png');
```

O próximo passo será criar a marca de água, que, neste caso, será um retângulo preto com um pouco de transparência. Dentro desse retângulo, vai estar escrito a branco MASTER D.

```
$stamp = imagecreatetruecolor(300, 100);  
imagefilledrectangle($stamp, 0, 0, 200, 50, 0x000000);  
imagestring($stamp, 5, 100, 30, 'MASTER D', 0xffffffff);
```

Depois, é necessário definir onde é que a marca de água estará, por isso, vão ser definidas uma margem da direita para a esquerda e uma margem de baixo para cima. Também será necessário definir o tamanho da marca de água, mas, como já foi definido anteriormente o tamanho do retângulo, basta colocar a variável \$stamp nas funções imagesx() e imagesy().

```
$right = 350;  
$bottom = 50;  
$sx = imagesx($stamp);  
$sy = imagesy($stamp);
```

Para terminar, será criada uma cópia da imagem original que terá a marca de água já “carimbada”. Para isso, será utilizada a função imagecopymerge(), que junta a segunda variável à primeira variável passada como parâmetro, seguindo as coordenadas e as proporções também indicadas. O último valor (70) define a combinação das imagens de 0 a 10, ou seja, no caso de ser zero, não haverá nenhuma modificação e, no caso de ser dez, a variável \$stamp seria colada com 100% de opacidade (é aqui que a opacidade do retângulo preto criado anteriormente é definida).

```
imagecopymerge($image, $stamp, imagesx($image) - $sx -  
$right, imagesy($image) - $sy -  
$bottom, 0, 0, imagesx($stamp), imagesy($stamp), 70);
```

Por fim, basta guardar a imagem, neste caso como `darthvader_stamp.png`, e depois libertar a imagem criada na variável `$image`.

```
imagepng($image, 'darthvader_stamp.png');  
  
imagedestroy($image);*/
```

Na tag referente à imagem, deve então incluir o nome da nova imagem criada.

```

```

Obtemos então:



No entanto, podemos querer adicionar uma imagem como marca de água.

Para a marca de água ser uma imagem, por exemplo o logótipo (fornecido na pasta de apoio a esta prática e de extensão .png), basta indicar à variável \$stamp o novo ficheiro correspondente ao logótipo (deixando de ter os dados da string). Não será utilizada a função imagecopymerge(), mas sim a função imagecopy(), que será utilizada da mesma forma, mas sem o último valor (50).

```
$image = imagecreatefrompng('darthvader.png');  
$stamp = imagecreatefrompng('logo.png');  
  
$right = 350;  
$bottom = 50;  
$sx = imagesx($stamp);  
$sy = imagesy($stamp);  
  
imagecopy($image, $stamp, imagesx($image) - $sx -  
    $right, imagesy($image) - $sy -  
    $bottom, 0, 0, imagesx($stamp), imagesy($stamp));  
  
imagepng($image, 'darthvader_stamp.png');  
imagedestroy($image);
```

O resultado final deve ser semelhante a este (com alguns ajustes nas margens):



Os códigos finais compilados serão, respetivamente, para o primeiro e segundo caso:

```
<html>

<head>

    <meta charset="UTF-8">

    <title>Prática 31</title>

    <link href="estilos.css" rel="stylesheet">

</head>

<body>

    <div class="caixa0">

        <span id="logo"></span>

    </div>

    <div class="caixa1" >

    </div>

</body>

</html>
```

```
<?php

$image = imagecreatefrompng('darthvader.png');

$stamp = imagecreatetruecolor(300, 100);
imagefilledrectangle($stamp, 0, 0, 200, 50, 0x000000);
imagestring($stamp, 5, 100, 30, 'MASTER D', 0xffffffff);

$right = 350;
$bottom = 50;

$sx = imagesx($stamp);
$sy = imagesy($stamp);

imagecopymerge($image, $stamp, imagesx($image) - $sx -
    $right, imagesy($image) - $sy -
    $bottom, 0, 0, imagesx($stamp), imagesy($stamp), 70);

imagepng($image, 'darthvader_stamp.png');
imagedestroy($image);

?>

</div>

</body>

</html>
```



```
<html>

<head>

    <meta charset="UTF-8">

    <title>Prática 31</title>

    <link href="estilos.css" rel="stylesheet">

</head>

<body>

    <div class="caixa0">

        <span id="logo"></span>

    </div>

    <div class="caixa1" >

        <?php

$image = imagecreatefrompng('darthvader.png');

$stamp = imagecreatefrompng('logo.png');

$right = 350;

$bottom = 50;

$sx = imagesx($stamp);

$sy = imagesy($stamp);
```

```
imagecopy($image, $stamp, imagesx($image) - $sx -  
    $right, imagesy($image) - $sy -  
    $bottom, 0, 0, imagesx($stamp), imagesy($stamp));  
  
imagepng($image, 'darthvader_stamp.png');  
  
imagedestroy($image);  
  
?>  
  
</div>  
  
</body>  
  
</html>
```