

# Mapas temáticos gerados com PHP

Por Raquel Dezidério Souto

O artigo mostra como programar em PHP para geração de mapas temáticos através de um exemplo prático. Diversos sistemas de informações geográficas já foram desenvolvidos e oferecem o recurso de geração de mapas temáticos. Entretanto, o tempo gasto no aprendizado do funcionamento de tais sistemas pode ser economizado se forem seguidos os passos simples disponíveis nesse breve tutorial.

A linguagem PHP tem uma biblioteca de funções muito útil, mas ainda pouca usada – a GDLib, biblioteca de exibição gráfica (em inglês: *Graphics Display Library*). A GDLib tem sido utilizada principalmente em aplicações de geração de gráficos, mas seu potencial para desenvolvimento de mapas digitais é grande e não deve ser negligenciado. Leia a apresentação da biblioteca GDLib contida no site oficial do PHP:

“O PHP não está limitado a criar apenas saída em HTML. Ele também pode ser usado para criar e manipular arquivos de imagem em uma diversidade de formatos, incluindo gif, png, jpg, wbmp, e xpm. E ainda mais convenientemente, o PHP pode enviar streams de saída de imagem diretamente para o browser. Você precisará compilar o PHP com a biblioteca GD de imagens de funções para isto funcionar. O PHP e GD também podem requerer outras bibliotecas, dependendo dos formatos de imagens com os quais você queira trabalhar.” [1]

Todas as funções gráficas da biblioteca GDLib possuem descrição detalhada no site oficial do PHP [2] e o leitor pode fazer uso do mesmo para consultas eventuais durante o desenvolvimento do projeto. Abaixo são listadas as funções utilizadas nesse tutorial e a respectiva descrição da sintaxe.

1) **ImageCreate** – Cria uma nova imagem;

ImageCreate ( int \$width , int \$height )

Onde: \$width = largura em pixels; \$height = altura em pixels.

2) **ImageColorAllocate** - Aloca uma cor em uma variável;

imagecolorallocate ( resource \$image , int \$red , int \$green , int \$blue )

Onde: \$image = imagem criada com imagecreate; \$red, \$green e \$blue = códigos RGB para vermelho (R), verde (G) e azul (B).

3) **ImageRectangle** – Desenha um retângulo sem preenchimento;

ImageRectangle ( resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color )

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenada superior esquerda; \$x2,\$y2 = coordenada inferior direita; \$color = identificador da cor da linha.

4) **ImageFilledRectangle** – Desenha um retângulo preenchido;

imagefilledrectangle ( resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color )

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenada superior esquerda; \$x2,\$y2 = coordenada inferior direita; \$color = identificador da cor do preenchimento.

5) **ImagePolygon** – Desenha um polígono sem preenchimento;

ImagePolygon ( resource \$image , array \$points , int \$num\_points , int \$color )

Onde: \$image = imagem criada com imagecreate; \$points = matriz de vértices do polígono; \$num\_points = número de vértices do polígono; \$color = identificador da cor de preenchimento.

Exemplo de matriz: points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1, ...

6) **ImageFilledPolygon** – Desenha um polígono preenchido;

ImageFilledPolygon ( resource \$image , array \$points , int \$num\_points , int \$color )

Onde: \$image = imagem criada com imagecreate; \$points = matriz de vértices do polígono; \$num\_points = número de vértices do polígono; \$color = identificador da cor da linha.

7) **Imagestring** – Desenha uma string horizontalmente;

Imagestring ( resource \$image , int \$font , int \$x , int \$y , string \$string , int \$color )

Onde: \$image = imagem criada com imagecreate; \$font = valores que podem variar de 1 a 5, dependendo do tamanho da fonte (por *default*, a codificação é latin2), onde números maiores correspondem a fontes mais largas; \$x = coordenada x superior à esquerda; \$y = coordenada y superior à esquerda; \$string = string a ser desenhada; \$color = identificador da cor da letra.

8) **Imageline** – Desenha uma linha;

Imageline ( resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color )

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenadas de início da linha; \$x2,\$y2 = coordenadas de fim da linha; \$color = identificador da cor da linha.

9) **ImagePNG** - Envia uma imagem PNG para o browser ou para um arquivo;

ImagePNG ( resource \$image [, string \$filename ] )

Onde: \$image = imagem criada com imagecreate; \$filename = se indicado o caminho, será enviada a imagem para um arquivo.

10) **ImageDestroy** – Destroi uma imagem, liberando a memória associada a ela.

ImageDestroy ( resource \$image )

Onde: \$image = imagem criada com imagecreate.

O tutorial de programação de scripts em PHP para geração de mapas temáticos segue os passos básicos listados a seguir. A execução demonstrada nesse artigo fundamenta-se no trabalho de classificação de municípios do Rio de Janeiro, onde foram gerados mapas temáticos que são visualizados diretamente no navegador [3].

## 1 . Preparação do PHP para execução das funções gráficas GD

Para começar a desenvolver projetos com GD, o programador deve estar atento à compilação do PHP com a GDLib para que as funções possam ser interpretadas corretamente. Além disso, se precisar expandir os recursos da GD, o programador deve estar atento à instalação de outras bibliotecas. A lista básica de bibliotecas a serem instaladas para que tenha mínimo de independência na programação segue adiante:

- **GDLib** - <http://www.libgd.org/releases/>

- **Lib JPEG** – Para gerar imagens jpeg a partir de scripts PHP com GD - <ftp://ftp.uu.net/graphics/jpeg>.

- **FreeType 2** - <http://www.freetype.org>.

No tutorial, a imagem a ser criada possui extensão PNG, que é suportada em versões de GD superiores à gd-1.6. Com relação a outras extensões de imagens suportadas pela GD, observe que:

- GIF - Suportado apenas em versões do GD anteriores a gd-1.6. Suporte apenas para leitura está disponível com o PHP 4.3.0 e a biblioteca GD empacotada. Suporte para escrita está disponível desde o PHP 4.3.9 e PHP 5.0.1.

- JPG - Quando compilando a biblioteca jpeg-v6b (antes de compilar o PHP) você precisa usar a opção --enable-shared no passo de configuração. Se você não fizer, irá receber um erro dizendo *libjpeg.(also) not found* quando você estiver no passo de configuração para compilar o PHP.

- XPM - É provável que a biblioteca já esteja disponível se o sistema tiver um ambiente X instalado.

Descomprima e instale os pacotes no diretório de bibliotecas do seu sistema. Todos os trechos que estão entre colchetes devem ser substituídos pelas informações relativas à sua instalação.

Instale o GD:

```
# cd [diretorio do pacote gd]
# ./configure --with-jpeg-dir=[caminho do diretorio da biblioteca Libjpeg] --
with-png-dir=[caminho do diretorio da biblioteca Libjpeg]
-- with-freetype-dir=[caminho do diretório da biblioteca freetype]
# make
# make install WITH_XPM=yes WITHOUT_X11=yes
```

Pare o servidor Apache se ele ainda estiver sendo executado:

```
# /usr/local/apache2/bin/apachectl stop
```

Recompile o PHP:

```
# ./configure - --with-apxs2 = [caminho do diretório apxs] --enable-track-vars --
with-mysql --enable-ftp --with-zlib --with-gd --with-jpeg-dir=[caminho do
diretório da biblioteca jpeg] --with-png-dir=[caminho do diretório da biblioteca
png] --with-freetype-dir=[caminho do diretório da biblioteca Freetype]
# make
# make install
```

Reinicie o Apache:

```
# /usr/local/apache2/bin/apachectl start
```

Execute a função `phpinfo()` para verificar se as bibliotecas foram instaladas corretamente.

OBS: Muitos documentos na web versam sobre como compilar o PHP com GD. Outras configurações de compilação são possíveis dependendo do servidor e/ou da versão das bibliotecas adquiridas e do PHP em uso. Aconselha-se que os pacotes de bibliotecas sejam baixados e que procure um tutorial específico na rede para a versão do Apache e do PHP que estiver utilizando, antes de começar a instalação.

## 2 . Planejamento do projeto

Muitas aplicações podem ser desenvolvidas com uso de mapas temáticos. Atualmente a informação geográfica tem lugar cada vez maior nos projetos de Internet. Esse artigo apresenta o exemplo de uma classificação temática realizada em 34 municípios do Rio de Janeiro, envolvendo 22 temas. Para planejar o projeto, o programador deve ter em mente as necessidades de cada proposta. No exemplo, as principais características levadas em conta foram as seguintes:

- Temas – Como são muitos temas, foi criada uma codificação para facilitar a busca em banco de dados;
- Unidade geográfica de análise utilizada – Municipal.
- Meta-informações – Também armazenadas em banco de dados para exibição automática – Títulos de mapas, descrição dos temas, nomes dos municípios, nomes de fontes de dados etc.

## 3 . Planejamento do banco de dados

O tamanho e complexidade do banco de dados dependerão das características do projeto. Aqui, é apresentada uma estrutura de banco de dados simplificada apenas para entendimento básico do processamento de informações na geração de um mapa temático, com unidade geográfica municipal. Daqui, poderá ser personalizado e expandido, de acordo com cada projeto.

Tabelas:

- **mun** – tabela com meta-informações dos municípios. Campos: `cod_mun` (código do município); `nome` (nome do município);
- **temas** – tabela com meta-informações dos temas adotados. Campos: `cod_tema` (código de identificação do tema); `desc_tema` (descrição do tema);
- **fator** – tabela com os limites de intervalos de classes utilizados e com os valores mínimo e máximo de cada série de dados temáticos. No exemplo, foram usadas três classes, denominadas como “alto impacto”, “médio impacto” e “baixo impacto”. Assim, a tabela contém os campos: `cod_tema` (código de identificação do tema), `L1` (valor numérico do limite entre as classes “baixo impacto” e “médio impacto”), `L2` (valor numérico do limite entre as classes “médio impacto” e “alto impacto”), `min` (valor numérico mínimo encontrado na série de dados de determinado tema), `max` (valor numérico máximo encontrado na série de dados de determinado tema).

Para armazenar os dados referentes a cada tema, deve ser criada uma tabela para cada qual, com os seguintes campos: `cod_mun` (código de identificação do município) e `valor` (valor numérico do tema referente a determinado município). Essa medida torna a consulta de dados pelo programa de geração do mapa digital mais ágil.

## 4 . Programando o script de geração dos mapas

O script de geração do mapa deve estar puro, destinado exclusivamente a esse fim. Se o programador inserir o script em uma página que já tem um cabeçalho HTML, incorrerá em erro. O uso da função `include()` do PHP também não é permitido.

A estrutura do programa PHP para geração de mapas digitais consiste das partes: 1) Cabeçalho do arquivo; 2) Conexão com o banco de dados; 3) Definição das cores; 4) Desenho da moldura do mapa; 5) Desenho do Título e do rodapé; 6) Desenho da

legenda; 7) Desenho dos polígonos e 8) Geração, exibição e destruição da imagem. A seguir, os trechos do código que correspondem a cada item e a explicação. Todos os trechos que estão em negrito devem ser substituídos com informações referentes ao seu projeto.

#### 1) Cabeçalho do arquivo

```
header("Content-type:image/png");
$cod = código do tema que serve de base para esse mapa;
$im = ImageCreate(890,650); //Criação da imagem.
```

#### 2) Conexão com o banco de dados

```
$con = mysql_connect ("endereço do host do banco de dados", "usuário", "senha");
$db = nome do banco de dados;
mysql_select_db ($db);
$tab = $cod; //seleciona a tabela correspondente ao tema em questão.
```

#### 3) Definição das cores

```
//cores da legenda
//cor amarela
$cor_02['R']=249;$cor_02['G']=248;$cor_02['B']=191;
//cor verde
$cor_01['R']=165;$cor_01['G']=200;$cor_01['B']=165;
//cor vermelha
$cor_03['R']=244;$cor_03['G']=150;$cor_03['B']=150;
//cor azul
$cor_04['R']=79;$cor_04['G']=191;$cor_04['B']=232;
//cor bege do fundo
$cor_05['R']=248;$cor_05['G']=244;$cor_05['B']=228;
//aloca as cores
$cor01 = ImageColorAllocate($im, $cor_01['R'],$cor_01['G'],$cor_01['B']);
$cor02 = ImageColorAllocate($im, $cor_02['R'],$cor_02['G'],$cor_02['B']);
$cor03 = ImageColorAllocate($im, $cor_03['R'],$cor_03['G'],$cor_03['B']);
$cor04 = ImageColorAllocate($im, $cor_04['R'],$cor_04['G'],$cor_04['B']);
$cor05 = ImageColorAllocate($im, $cor_05['R'],$cor_05['G'],$cor_05['B']);
$white = ImageColorAllocate($im, 255, 255, 255);
$black = ImageColorAllocate($im, 0, 0, 0);
```

#### 4) Desenho da moldura do mapa

```
imagerectangle($im,5,5,880,640,$black);
```

#### 5) Desenho do título e do rodapé

```
//Escreve o titulo
//Pesquisa o nome do indicador no banco de dados

$result0 = mysql_query("SELECT desc_tema FROM temas WHERE cod_tema = '$cod'")or
die ("Nao pude pesquisar ". mysql_error());
while ($row0 = mysql_fetch_array($result0)){
$desc_tema = $row0[0];
}
Imagestring($im,6,20,40,$desc_tema,$black);
//Escreve o rodapé
$string_fonte = "Fonte: Atlas de Indicadores de Sustentabilidade para os
Municípios Costeiros do Estado do Rio de Janeiro, Brasil.";
imagestring($im,3,20,570,$string_fonte,$black);
$string_end = "http://www.vides.org/atlas/mapas_ind.php, acessado em "
.date("d") . "/" . date("n") . "/" . date("Y");
imagestring($im,3,20,590,$string_end,$black);
$string_mail = "Contacte a autora: Raquel Dezidério
(raquel.deziderio@gmail.com)";
imagestring($im,3,20,610,$string_mail,$black);
```

#### 6) Desenho da legenda

```
//Cria o box da legenda
Imagefilledrectangle($im,712,340,872,560,$white);
//Cria a moldura do box da legenda
Imagerectangle($im,712,340,872,560,$black);
//Escreve a palavra "LEGENDA"
```

```

$string = "LEGENDA";
imagestring($im,5,765,350,$string,$black);
//Cria a linha que sublinha a legenda
ImageLine($im, 750, 370, 840, 370, $black);
//Pesquisa os intervalos das classes
$result0 = mysql_query("SELECT max, min, L1, L2 FROM fator WHERE cod_tema =
'cod'")or die("Nao pude pesquisar ". mysql_error());
while ($row0 = mysql_fetch_array($result0)){
$max = $row0[0];
$max_aux = $max;
$max = $max + 0.0100;
$min = $row0[1];
$min_exibe = $min;
$min = $min - 0.0100;
$L1 = $row0[2];
$L1_aux = $L1 + 0.0001;
$L2 = $row0[3];
$L2_aux = $L2 + 0.0001;
}
//Desenha os retangulos dos intervalos
//Escreve o primeiro intervalo
$string1 = "Impacto Baixo";
$string1c = "(" . $min_exibe . " - " . $L1 . ")";
imagestring($im,5,748,396,$string1,$black);
imagestring($im,3,748,411,$string1c,$black);
//Escreve o segundo intervalo
$string2 = "Impacto Médio";
$string2c = "(" . $L1_aux . " - " . $L2 . ")";
imagestring($im,5,748,456,$string2,$black);
imagestring($im,3,748,471,$string2c,$black);
//Escreve o terceiro intervalo
$string3 = "Impacto Alto";
$string3c = "(" . $L2_aux . " - " . $max_aux . ")";
imagestring($im,5,748,516,$string3,$black);
imagestring($im,3,748,531,$string3c,$black);
//Desenha os quadradinhos
imagefilledrectangle($im,720,400,740,420,$cor01);
imagerectangle($im,720,400,740,420,$black);
imagefilledrectangle($im,720,460,740,480,$cor02);
imagerectangle($im,720,460,740,480,$black);
imagefilledrectangle($im,720,520,740,540,$cor03);
imagerectangle($im,720,520,740,540,$black);

```

## 7) Desenho dos polígonos

```

//Cria o poligono do mar
$values_mar = array(
0 => 5,
1 => 544,
2 => 5,
3 => 640,
4 => 880,
5 => 640,
6 => 880,
7 => 84,
8 => 736,
9 => 133,
10 => 739,
11 => 232,
12 => 500,
13 => 413,
14 => 430,
15 => 443,
16 => 380,

```

```

17 => 382,
18 => 340,
19 => 390,
20 => 286,
21 => 453,
22 => 215,
23 => 430,
24 => 151,
25 => 452,
26 => 120,
27 => 437,
28 => 49,
29 => 459,
30 => 5,
31 => 544,
);
$n_mar = 16;
//Desenha o poligono do mar
imagefilledpolygon($im,$values_mar,$n_mar,$cor04);
imagepolygon($im,$values_mar,$n_mar,$black);
//Desenha o poligono do município 31 e atribui a cor
$cod_mun = 31;
//Pesquisa o valor da medida para o municipio ...
$result31 = mysql_query("SELECT valor FROM $tab WHERE cod_mun = '$cod_mun'")or
die ("Nao pude pesquisar ..." . mysql_error());
while ($row31 = mysql_fetch_array($result31)){
$valor31 = $row31[0];
$valor31_aux = $valor31;
$valor31 = number_format($valor31,2,".",".");
}
//Define o poligono para o municipio 31
$values_31 = array(
0 => 294,
1 => 423,
2 => 296,
3 => 423,
4 => 297,
5 => 422,
6 => 298,
7 => 422,
8 => 299,
9 => 421,
10 => 306,
11 => 421,
12 => 305,
13 => 412,
14 => 301,
15 => 412,
16 => 300,
17 => 413,
18 => 297,
19 => 413,
20 => 295,
21 => 414,
22 => 295,
23 => 416,
24 => 293,
25 => 416,
26 => 292,
27 => 417,
28 => 291,
29 => 418,
30 => 291,
31 => 419,

```

```

32 => 294,
33 => 422,
34 => 294,
35 => 423,
);
$n_31 = 18;
//Testa em que intervalo está o valor recuperado para o município
if (($valor31 > $min)&&($valor31 < $L1)){
imagefilledpolygon($im,$values_31,$n_31,$cor01);
}
if (($valor31 >= $L1)&&($valor31 <= $L2)){
imagefilledpolygon($im,$values_31,$n_31,$cor02);
}
if (($valor31 <= $max)&&($valor31 > $L2)){
imagefilledpolygon($im,$values_31,$n_31,$cor03);
}
//Desenha a borda do poligono 31
imagepolygon($im,$values_31,$n_31,$black);

```

#### 8) Geração, exibição e destruição da imagem

```

ImagePNG($im);
Imagedestroy($im);

```

### Considerações finais

Aqui só está descrito o trecho de código de um dos municípios avaliados para fim de exemplificação. Para ver o código PHP utilizado na aplicação real que serviu de base para esse tutorial, visite: [http://www.ivides.org/atlas/mapas\\_ind.php](http://www.ivides.org/atlas/mapas_ind.php) e clique no link “código fonte do mapa”.

### Referências e links sugeridos

- [1] [http://br2.php.net/manual/pt\\_BR/intro.image.php](http://br2.php.net/manual/pt_BR/intro.image.php) - Apresentação da GDLib no php.net.
- [2] [http://br2.php.net/manual/pt\\_BR/ref.image.php](http://br2.php.net/manual/pt_BR/ref.image.php) – Lista das funções da GDLib no php.net.
- [3] [http://www.ivides.org/atlas/mapas\\_ind.php](http://www.ivides.org/atlas/mapas_ind.php) – Atlas de Indicadores de Sustentabilidade para os Municípios Costeiros do Estado do Rio de Janeiro.

### Sobre a autora

Raquel Dezidério Souto ([raquel.deziderio@gmail.com](mailto:raquel.deziderio@gmail.com)) é Bacharel em Oceanografia pela Universidade do Estado do Rio de Janeiro (2005) e programadora PHP/MySQL, atuando desde 2004 em desenvolvimento de aplicações administrativo-financeiras e geocientíficas. Atualmente seus principais projetos são: portal do “Atlas de Indicadores de Sustentabilidade para os Municípios Costeiros do Estado do Rio de Janeiro” - <http://www.ivides.org/atlas> e portal do “Instituto Virtual para o Desenvolvimento Sustentável - IVIDES.org” - <http://www.ivides.org>. Procura por parceiros de atividades e coloca-se à disposição para resolver dúvidas geradas no uso desse tutorial.