

# COMO FAZER

## Desenhando gráficos de equações de primeiro grau com uma variável usando php

Albert Josuá

albertjosua@yahoo.com.br

Rio Branco – Acre

06 de fevereiro de 2011

## Equações de primeiro grau com uma variável

Equações de primeiro grau com uma variável são expressas em sua forma geral como

$$ax + b = 0$$

Onde:

a e b são números conhecidos e  $a > 0$ .

Exemplos de equação de primeiro grau com uma variável.

- $2x + 8 = 0$ ;
- $1x + 6 = 9$ ;
- $5x - 4 = 6x + 8$ .

Podemos resolver este tipo de equação subtraindo b de ambos os membros da equação e obtendo:

$$ax + b = 0 \quad (1)$$

$$ax + b - b = 0 - b \quad (2)$$

$$ax = -b \quad (3)$$

$$x = \frac{-b}{a} \quad (4)$$

Toda equação de primeiro grau com uma variável pode ser reduzida a forma descrita em (4).

## PHP

Neste tutorial nós construiremos dois arquivos .php, o primeiro denominado equacao.php consiste simplesmente na exibição de um formulário HTML para que o usuário possa entrar com os dados ( este arquivo não necessitaria obrigatoriamente de ser na extensão .php por conter somente tags HTML), o segundo, eqc\_1\_gr.php é o script que construirá o gráfico no navegador do usuário.

Para uma melhor visualização os códigos estarão disponíveis em arquivos pdf sepados.

### Equação.php

Este arquivo contém somente um formulário HTML para que o usuário possa inserir os dados necessários para o processamento e construção do gráfico. O código apresentado neste arquivo não contém nenhuma formatação HTML ou CSS que o torne apresentável, ficando como sugestão a melhoria deste código para sua melhor apresentação.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />

<title>Equação</title>

</head>

<body>

<form action="eqc_1_gr.php" method="post">

<label>Insira a valores da equação de primeiro grau na forma  $ax + b = c$ </label>

<br/>

<label>a</label>

<input type="text" name="a" size="10" />

<br/>

<label>b</label>

<input type="text" name="b" size="10" />

<br/>

<label>c</label>

<input type="text" name="c" size="10"/>
```

```
<br/>
<input type="submit" value="Desenhar" />
</form>
</body>
</html>
```

## **eqc 1 gr.php**

Este script após receber os valores da equação realiza o processamento e a geração do gráfico. Explicaremos todas as funções usadas para construção e toda a codificação realizada.

### **Funções PHP**

#### **ImageCreate(\$largura, \$altura)**

Esta função cria uma nova imagem, sendo que recebe dois parâmetros com a definição da largura e da altura da imagem, o retorno é um identificador de imagem.

```
$sid_imagem = ImageCreate(200, 200);
```

#### **ImageColorAllocate(\$Id\_imagem, \$r, \$g, \$b)**

Esta função define uma cor usando o sistema de cores RGB, os parâmetros desta função são \$id\_imagem que consiste em um identificador de imagem, e três valores inteiros (\$r, \$g, \$b) com os números da cor desejada no padrão RGB, o retorno é um identificador de cor

```
$preta = ($sid_imagem, 0, 0, 0);
```

#### **ImageFill(\$id\_imagem, \$coord\_x, \$coord\_y, \$preta)**

Esta função desenha um fundo para nossa aplicação, os parâmetros desta função são um identificador de imagem (\$id\_imagem), as coordenadas iniciais da área do fundo (o canto superior esquerdo possui coordenador 0, 0) e o último parâmetro é um identificador da cor usada para pintar a área de fundo(\$preta).

```
ImageFill($sid_imagem, 0, 0, $preta);
```

#### **ImageLine(\$id\_imagem, \$coord\_ini\_x, \$coord\_ini\_y, \$coord\_fim\_x, \$coord\_fim\_y, \$preta)**

Esta função desenha uma linha entre dois pontos definidos nos seus parâmetros, os parâmetros aceitos por ImageLine são um identificador de imagem(\$id\_imagem), um par de coordenadas para o fixar o começo da linha(\$coord\_ini\_x e \$coor\_ini\_y), um par de coordenadas para fixar o fim da

linha(\$coord\_fim\_x e \$coord\_fim\_y) e o último parâmetro é um identificador de cor(\$preta).

```
ImageLine($id_image, 0,0, 400, 400, $preta);
```

**ImageString(\$id\_image, \$fonte, \$coord\_x, \$coord\_y, \$texto, \$preta)**

Esta função insere um texto, os parâmetros desta função são um identificador de imagem(\$id\_image), uma fonte que é um valor inteiro de 1 a 5(\$fonte), um par de coordenadas para demarcar o local do início do texto (\$coord\_x e \$coord\_y), o seu quinto parâmetro é uma string com o texto a ser inserido(\$texto) e o último é um identificador de cor para o texto.

```
ImageString($id_image, 5, 40, 40, "Olá", $preta);
```

**Header(\$tipo\_de\_arquivo)**

Esta função informa ao navegador do usuário que tipo de arquivo que o script envia, o parâmetro desta função é uma string com a definição do arquivo.

```
Header('Content-type: image/png');
```

**ImagePng(\$id\_imagem)**

Esta função envia a saída para o navegador, o parâmetro desta função é um identificador de imagem(\$id\_imagem).

```
ImagePng($id_imagem);
```

**ImageDestroy(\$id\_imagem)**

Esta função, como o nome sugere, destrói uma imagem gerada, o seu parâmetro é um identificador de imagem.

```
ImageDestroy($id_imagem);
```

Após mostrarmos as funções envolvidas no desenvolvimento de nosso script passaremos para nosso código:

```
<?php
```

```
//Recebe variáveis enviadas pelo formulário através do método POST.
```

```
$a = $_POST['a'];
```

```
$b = $_POST['b'];
```

```
$c = $_POST['c'];
```

/\*Configura a área de desenho.

É importante que ao definir o tamanho da largura e do comprimento seja levado em consideração o fato de que a definição desses tamanhos está diretamente ligada ao valor definido para a unidade de comprimento (logo abaixo), porque se notarmos cada dimensão (horizontal e vertical) é dividida em duas partes, uma para o lado negativo e a outra para o lado positivo. Cada uma dessas duas parte precisa ter um número inteiro de unidades de comprimentos para a correta centralização do gráfico.

Exemplo:

horizontal = 2000

unidade de comprimento = 40

este valor é definido por tentativas e erros quando do desenvolvimento deste script notou-se que este valor é viável para uma melhor visualização.

$2000/2 = 1000$

cada lado do eixo horizontal terá tamanho 1000 e como cada unidade de comprimento possui tamanho 40, ao dividirmos os valores obtemos um número inteiros de unidades de comprimento em cada parte do eixo.

$1000/40 = 25$ .

Para o eixo vertical a ideia é a mesma.

\*/

\$horizontal = 2000; //Largura

\$vertical = 1200; //Comprimento

/\*Nossa área de trabalho tem 2000 na horizontal e 1000 na vertical.

Vamos considerar que cada unidade de comprimento tenha 40. Desta forma podemos representar na horizontal 50 unidades de comprimento e na vertical 30 unidades de comprimento.

\*/

\$uc = 40; //Unidade de comprimento.

//ImageCreate gera a área de desenho com as dimensões defindas acima.

\$area = ImageCreate(\$horizontal, \$vertical);

```

//Define cores que são utilizadas.

$preta = ImageColorAllocate($area, 0, 0, 0);

$azul = ImageColorAllocate($area, 0, 0, 64);

$branca = ImageColorAllocate($area, 255, 255, 255);

//Desenha área.

ImageFill($area, 0, 0, $preta);

//Desenha a linha dos eixos do gráfico.

ImageLine($area, 0, 600, 2000, 600, $branca); //x;

ImageLine($area, 1000, 0, 1000, 1200, $branca); //y;

//Desenha as unidades de área na horizontal(negativo).

for($uc;$uc<1000;$uc+=40){

    ImageString($area, 5,$uc,588,"|", $branca); // Na horizontal.

    ImageString($area, 5,$uc,600,"|", $branca); // Na horizontal.

}

//Desenha as unidades de área na horizontal(positivo).

for($uc=1040;$uc<2000;$uc+=40){

    ImageString($area, 5,$uc,588,"|", $branca); // Na horizontal.

    ImageString($area, 5,$uc,600,"|", $branca); // Na horizontal.

}

//Desenha as unidades de área na vertical(positivo).

for($uc=40;$uc<600;$uc+=40){

    ImageString($area, 5,988,$uc,"----", $branca); // Na vertical.

}

//Desenha as unidades de área na vertical(negativo).

for($uc=640;$uc<1200;$uc+=40){

    ImageString($area, 5,988,$uc,"----", $branca); // Na vertical.

```

```

}

//Insere textos no desenho.

ImageString($area, 5, 1980, 640, "X", $branca);

ImageString($area, 5, 1040, 20, "Y", $branca);

//Calcula resultado da equação.

$х = ($c - $b)/$a;

//Reseta valor da unidade de comprimento.

$uc=40;

/*

Verifica em qual lado do gráfico a reta passará.

Se o resultado da equação for um número positivo a reta passará do lado positivo do eixo horizontal,
senão passará do lado negativo.

$pos_resultado = Posicionamento do resultado no gráfico.

Como foi determinado acima cada lado do eixo horizontal possui o tamanho 1000 se o resultado da
equação for positivo a reta passará no lado positivo do eixo ( da metade para o fim).

Em $pos_resultado = 1000 + ($x*$uc);

1000 representa o centro do gráfico(0,0)

($x*$uc) representa o resultado da equação $x multiplicado pelo tamanho da unidade de comprimento
definida acima $uc. O resultado desta multiplicação é quantas unidades de comprimento o resultado da
equação representa. Ao somarmos ao número 1000 temos quantas unidades de comprimento o
resultado deve ser posicionado.

Para o caso de o resultado da equação for um número negativo, a idéia é a mesma, basta para isto que
se use a função abs() do php para usar o modulo de um número.

*/

if($x>0){

    $pos_resultado = 1000 + ($x*$uc);

}

else{

    $pos_resultado = 1000 - abs($x*$uc);

}

```

```
//Insero o resultado da equação no gráfico.
```

```
ImageString($area, 5,$pos_resultado ,640,$x, $branca);
```

```
/*Define orientação da reta.
```

Se a for positivo a reta terá uma orientação crescente, sendo desta forma a reta traçada em duas partes a primeiro para o lado crescente da reta e a outra para o lado decrescente da reta.

$pos\_resultado+(0.40*pos\_resultado)$  é um valor aproximado para a inclinação da reta.

Se a for negativo a ideia é a mesma.

```
*/
```

```
if($a>0){
```

```
    //Seta a reta no gráfico
```

```
    //Parte positiva.
```

```
    ImageLine($area,$pos_resultado,600,$pos_resultado+(0.40*$pos_resultado),100,$branca);
```

```
    //Parte negativa.
```

```
    ImageLine($area,$pos_resultado,600,$pos_resultado-(0.40*$pos_resultado),1100,$branca);
```

```
}
```

```
//Envia gráfico para saída.
```

```
Header('Content-type: image/png');
```

```
ImagePng($area);
```

```
ImageDestroy($area);
```

```
?>
```