

## Uma viagem pelo PHP-GTK2

Por Pablo Dall'Oglio

### TAGS

PHP, GTK, Widgets, temas, diálogos.

### Resumo

Neste artigo veremos algumas das novas características presentes no PHP-GTK2, através de exemplos práticos de utilização de diálogos, imagens, botões e estilos. Antes de tudo, teremos uma breve introdução e contextualização histórica do projeto PHP-GTK.

### 1. Introdução

Em 2001, foi lançada a primeira versão do PHP-GTK. Andrei Zmievski, seu criador, iniciou o projeto como um experimento para provar que era possível utilizar os recursos da biblioteca gráfica GTK no PHP, mostrando que o PHP tinha muito potencial também fora do ambiente web, como uma linguagem de propósitos gerais. Sua tentativa foi um sucesso, e o que muitas pessoas torciam o nariz no início se tornou real e dia após dia mais pessoas utilizam PHP para produzir aplicações cliente utilizando esta biblioteca multi-plataforma (GTK).



foi originalmente desenvolvido para o GIMP (GNU

O GTK é um conjunto de bibliotecas, desenvolvido originalmente por Peter Mattis, Spencer Kimball e Josh MacDonald, cujo propósito é servir ao desenvolvedor como base para criar aplicações gráficas. O GTK (GIMP ToolKit)

Image Manipulation Program), o software para artes gráficas mais conhecido para Linux. Ele tem crescido muito desde o início do projeto e hoje é utilizado como parte central do Gnome, uma das interfaces gráficas e plataformas de desenvolvimento mais utilizadas para Linux. O GTK+ também tem sido portado para o BeOS e Win32, fazendo da linguagem a escolha perfeita para o desenvolvimento de aplicações gráficas livres ou comerciais, uma vez que é licenciado sob a GPL (General Public License).

O PHP-GTK é uma extensão do PHP. Ao habilitarmos esta extensão, temos à disponibilidade um conjunto de classes e métodos a serem utilizados para construir o visual de nossa aplicação, com janelas, botões, painéis, etc. A primeira “geração” do PHP-GTK consistia na utilização do PHP4 com a biblioteca GTK1.2. Esta versão ainda é utilizada por muitos projetos. Assim que o PHP5 foi lançado, Andrei iniciou a árdua tarefa de reescrever a base do PHP-GTK para utilizar as novas facilidades do PHP5 e do GTK2. Grande parte do trabalho foi o de tirar vantagem do novo modelo de objetos existente no PHP5 e das novas características do GTK2, que traz novos componentes, um visual mais atraente e um desenvolvimento mais flexível.

É importante destacar que o PHP-GTK permite desenvolver aplicações cliente, stand-alone, com um ambiente de janelas. O PHP-GTK não está aí para substituir as aplicações web. Existem nichos de aplicações que simplesmente é mais interessante desenvolvermos uma aplicação cliente do que uma aplicação web e outros no qual ter uma interface gráfica é um diferencial positivo. Em muitos projetos é necessário termos um ambiente híbrido com

alguns módulos na web e outros módulos cliente. Imagine um sistema de supermercado, por exemplo. Podemos desenvolver pela web toda parte de cadastros, administração, portal de compras, etc. Mas para alguns usos como o ponto de vendas, onde se usa impressora fiscal e leitor de códigos de barras é interessante termos uma aplicação cliente, devido à sua velocidade, agilidade e interatividade com dispositivos periféricos. Para quem já conhece o PHP e sabe os conceitos de orientação a objetos verá que é muito simples aprender o PHP-GTK, visto que para cada componente existe uma classe correspondente.

Utilizando PHP-GTK, você criará uma aplicação que possui conectividade com o servidor (banco de dados, acesso à arquivos etc.), como todos os outros programas escritos em PHP, mas que, pelo fato de rodar na máquina cliente, também tem total acesso aos recursos desta (executar aplicações, escrever arquivos e acessar dispositivos locais). Para tanto, o PHP-GTK precisa ser instalado em cada máquina-cliente que executará uma aplicação PHP-GTK.

Neste artigo veremos apenas algumas funcionalidades introduzidas pela biblioteca GTK2 em relação ao GTK1. Você poderá conferir mais exemplos reais no site da comunidade PHP-GTK Brasil ([www.php-gtk.com.br](http://www.php-gtk.com.br)).

## 2. Rótulos com Estilo

No PHP-GTK1 era um tanto quanto complicado exibir rótulos de texto com diferentes formatações em tela. Tarefas simples como colocar o texto em negrito, itálico, sublinhado ou em cores necessitavam um maior nível de conhecimento. Agora o PHP-GTK2 usa o Pango, um framework open source que lida com todas tarefas relativas à layout e renderização. Com o Pango é possível utilizar uma linguagem de marcação derivada da SGML para formatar textos para exibição, uma forma simples de definir estilo e cores. Apesar de não demonstrado aqui, é possível escrever textos verticalmente ou usando ângulos (em graus). Este recurso é bastante útil em diversas situações que precisamos destacar algum item na tela.



Figura 1 – Rótulos de texto com estilo

```
<?php
$window = new GtkWindow;
$window->set_position(GTK::WIN_POS_CENTER);
$window->set_border_width(4);
$window->set_default_size(200,200);

$vbox = new GtkVBox;
$label = new GtkLabel();
$label->set_markup('sample: <b>Bold
text</b>');
$vbox->pack_start($label);

$label = new GtkLabel();
$label->set_markup('sample:
<i>Italic</i>');
$vbox->pack_start($label);

$label = new GtkLabel();
$label->set_markup('sample:
<u>Underline</u>');
$vbox->pack_start($label);

$label = new GtkLabel();
$label->set_markup('sample:
<b><i><u>Bold, Italic
and Underline</u></i></b>');
$vbox->pack_start($label);

$label = new GtkLabel();
$label->set_markup('<span foreground="blue"
background="gray"> <b>blue text and gray
background</b></span>');
$vbox->pack_start($label);

$label = new GtkLabel();
$label->set_markup('<span foreground="red"
background="orange"> <b>Red Text and orange
background</b></span>');
$vbox->pack_start($label);

$window->add($vbox);
$window->show_all();
gtk::main();
?>
```

## 3. Ícones de Estoque

Quando desenvolvemos aplicações é muito comum necessitarmos criar ícones com imagens como: salvar, abrir, fechar, deletar, adicionar, limpar, sair, sim, não, dentre outras. Quase todo tempo necessitamos isto. Para padronizar o visual de aplicações, o GTK2 oferece-nos itens de estoque, um conjunto de imagens comuns para se utilizar em aplicações para botões, menus, listagens, dentre outros. Usando imagens de estoque, não precisamos nos preocupar em procurar imagens na internet, pois a biblioteca já traz ícones para a maioria das tarefas comuns em aplicações.

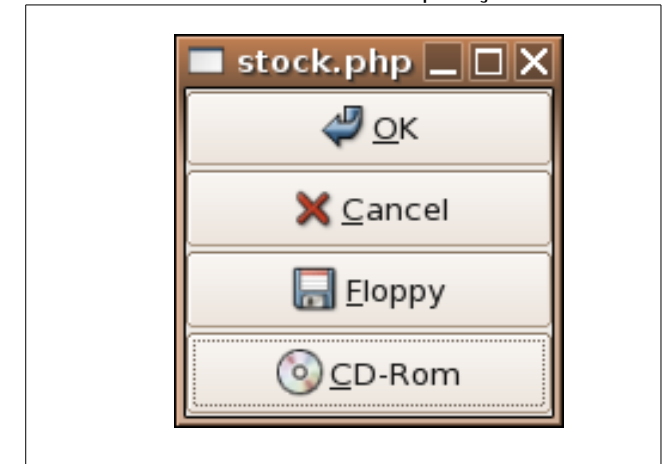


Figura 2 – Botões de Estoque

```
<?php
$window = new GtkWindow;
$window->set_default_size(120,100);
$vbox = new GtkVBox;

# Cria botões de estoque
$button =
GtkButton::new_from_stock(Gtk::STOCK_OK);
$vbox->pack_start($button);

$button =
GtkButton::new_from_stock(Gtk::STOCK_CANCEL
);
$vbox->pack_start($button);

$button =
GtkButton::new_from_stock(Gtk::STOCK_FLOPPY
);
$vbox->pack_start($button);

$button =
GtkButton::new_from_stock(Gtk::STOCK_CDRUM)
;
$vbox->pack_start($button);

$window->add($vbox);
$window->show_all();
gtk::main();
?>
```

## 4. Exibição de Imagens

O formato padrão de imagens utilizado pelo PHP-GTK1 era Xpixmap (XPM), que consiste em um formato de representação de imagens no formato ASCII. Apesar de fácil de se utilizar, arquivos XPM não são tão populares e não são indicados para imagens de alta resolução, por que seu tamanho cresce rapidamente.

Uma alternativa para o PHP-GTK1 era habilitar a biblioteca externa GdkPixbuf, que suporta os

formatos de imagens mais populares como PNG, JPEG e GIF. A partir do GTK2, a manipulação de imagens pode ser realizada diretamente pela classe GtkImage, que suporta os formatos mais comuns de imagens como PNG, JPEG, GIF, dentre outros. O código abaixo mostra como exibir uma imagem PNG no PHP-GTK2.



Figura 3 – Exibição de Imagens

```
<?php
# cria a janela
$window = new GtkWindow;
$window->set_default_size(200,200);
# lê arquivo de imagem
$imagem =
GtkImage::new_from_file('gnome.png');
# adiciona à janela
$window->add($imagem);
# exhibe
$window->show_all();
gtk::main();
?>
```

## 5. Seleção de Arquivos

O widget que o PHP-GTK1 oferecia para lidar-se com arquivos (abrir, salvar) era o GtkFileSelection. Este widget era relativamente fácil de se utilizar e oferecia uma interface simples. Mas não era muito amigável para aqueles acostumados com diálogos de arquivos de outros sistemas operacionais. Para o GTK2, Federico Mena Quintero, um dos mais ativos hackers por trás do GTK, escreveu o widget GtkFileChooser, um novo diálogo para arquivos, com um visual atraente e também mais extensível. Agora o GtkFileChooser é usado praticamente por todas aplicações Gnome, como Evolution, Gnumeric, Gaim, Gimp, dentre outros. Através deste simples exemplo abaixo, percebemos como rodar o diálogo para abrir um arquivo.

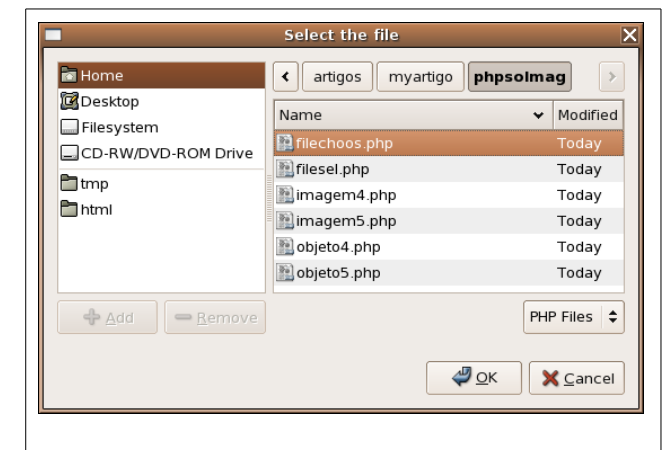


Figura 4 – Seleção de Arquivos

```
<?php
// Instancia classe GtkFileChooser
$dialog = new
GtkFileChooserDialog('Selecione o arquivo',
NULL, Gtk::FILE_CHOOSER_ACTION_OPEN,
array(Gtk::STOCK_OK, Gtk::RESPONSE_OK,
Gtk::STOCK_CANCEL, Gtk::RESPONSE_CANCEL));

$dialog->set_select_multiple(true);
$response = $dialog->run();

// Verifica a resposta
if ($response == Gtk::RESPONSE_OK)
{
    echo 'Você escolheu : ' .
        ($dialog->get_filename());
}
$dialog->destroy();

Gtk::main();
?>
```

## 6. Visual e Temas

O toolkit GTK é construído no topo da biblioteca GDK (Gimp Drawing Kit), que é uma biblioteca construída sobre um conjunto de funções de baixo nível para acessar as funções do sistema de janelas. O mecanismo de renderização de temas no GTK1 era um pouco limitado e programas PHP-GTK1 não tinham uma aparência de aplicação nativa quando rodavam em sistemas como o Windows. Com o PHP-GTK2, temos mais opções de temas para utilizar. Para usuários windows, por exemplo, há o tema "Wimp", que deixa a aplicação exatamente com o mesmo visual que as aplicações nativas.

Para obter este visual nativo sobre Windows, aplicamos um tema. Há muitos temas para GTK disponíveis em <http://gtk.themes.org>. O tema Wimp (utilizado aqui) está disponível em [wimp.sourceforge.net. No site PHP-GTK Brasil \(\[www.php-gtk.com.br\]\(http://www.php-gtk.com.br\)\), temos um instalador do PHP-GTK2 para windows que já instala o Wimp automaticamente. Para utilizar um tema, precisamos somente adicionar uma linha no início do código:](http://gtk-</a></p>
</div>
<div data-bbox=)

```
<?
Gtk::rc_parse('Wimp/gtk-2.0/gtkrc');
$window = new GtkWindow;
$window->set_default_size(200,200);
/*
 * Código da aplicação
 */
$window->show_all();
Gtk::main();
?>
```

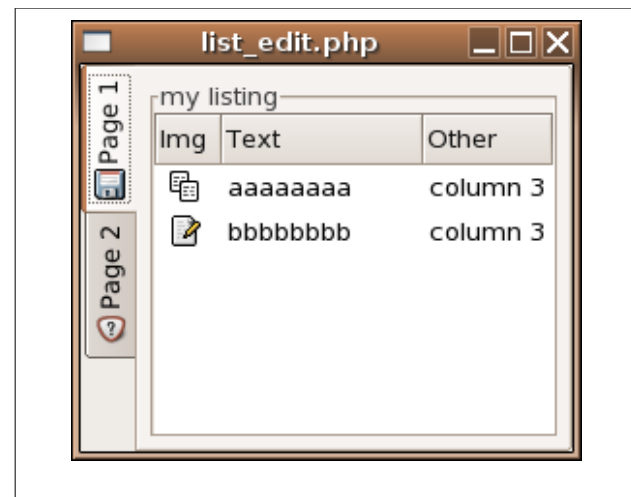


Figura 5 – Visual de uma janela no Ubuntu

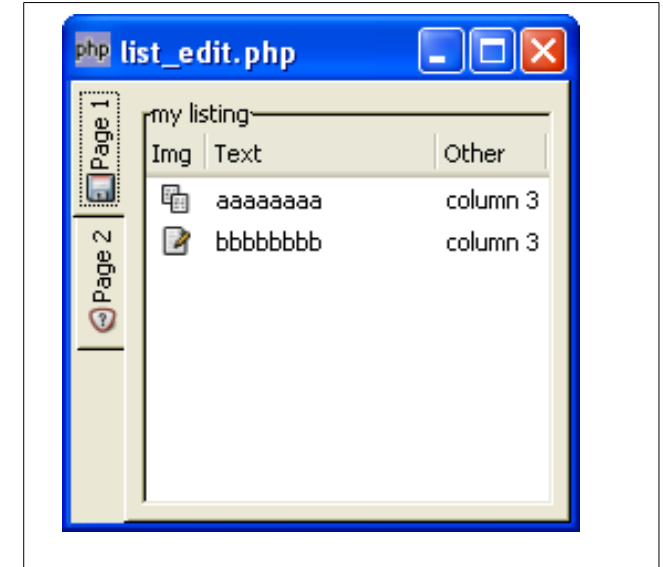


Figura 6 – Visual de uma janela no Windows

## 7. Diálogos

PHP-GTK2 oferece-nos um widget flexível para caixas de diálogo (GtkMessageDialog), que pode ser utilizado em muitas situações (mensagens de warning, erro, diálogos de confirmação, caixas de entrada, etc). Podemos também utilizar um conjunto de ícones de estoque com suas respectivas respostas. Aqui, demonstraremos dois exemplos usando a mesma classe GtkMessageDialog. A primeira exibindo uma mensagem de erro e a segunda perguntando por uma confirmação.



Figura 7 – Mensagem de Erro

```
<?php
// Cria um MessageDialog do tipo ERROR
$diallog = new GtkMessageDialog(null,
Gtk::DIALOG_MODAL, Gtk::MESSAGE_ERROR,
Gtk::BUTTONS_OK);
// Define o rótulo utilizando marcação
$diallog->set_markup('<b><i> Illegal
operation </i></b>');
$response = $diallog->run();

// Verifica a resposta
if ($response == Gtk::RESPONSE_OK)
{
    echo "Você clicou no botão OK\n";
}
$diallog->destroy();
?>
```

A seguir, temos um diálogo de confirmação para perguntar ao usuário final se o mesmo deseja continuar a operação. Veja que o terceiro parâmetro identifica o tipo de mensagem (Gtk::MESSAGE\_QUESTION) e o parâmetro seguinte identifica os botões que estarão presentes no diálogo (Gtk::BUTTONS\_YES\_NO). A seguir temos a mensagem a ser exibida. O diálogo será exibido em tela pelo método run(). Após isto, temos

um conjunto de IF's para testar as respostas do usuário.

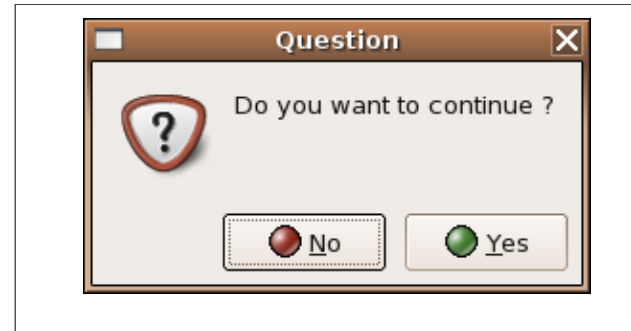


Figura 8 – Mensagem de Confirmação

```
<?php
// Cria um MessageDialog do tipo QUESTION
$diallog = new GtkMessageDialog(null,
Gtk::DIALOG_MODAL, Gtk::MESSAGE_QUESTION,
Gtk::BUTTONS_YES_NO, 'Do you want to
continue ?');

$response = $diallog->run();
// Verifica a resposta
if ($response == Gtk::RESPONSE_YES)
{
    echo "Você escolheu por continuar\n";
}
else if ($response == Gtk::RESPONSE_NO)
{
    echo "Você escolheu não continuar\n";
}
else if ($response ==
Gtk::RESPONSE_DELETE_EVENT)
{
    echo "Você fechou a janela\n";
}
$diallog->destroy();
?>
```

## Considerações finais

Neste artigo procuramos dar um overview sobre algumas das características presentes no PHP-GTK2 de forma simples e didática. Nos próximos artigos iremos explorar mais o potencial de uso do PHP-GTK2 em relação a orientação a objetos e também em exemplos mais práticos que podem ser utilizados no cotidiano.

## Referências e links sugeridos

[PHP-GTK Brasil] – <http://www.php-gtk.com.br>

[PHP-GTK Home] – <http://gtk.php.net>

[Site do Autor] – <http://www.pablo.blog.br>

Sobre o autor	
<b>Nome</b>	Pablo Dall'Oglio
<b>E-mail</b>	pablo@dalloglio.net
Pablo Dall'Oglio é formado em Análise de Sistemas pela UNISINOS. É autor de projetos consagrados em PHP-GTK, como o Agata Report e o Tulip. Entusiasta de PHP-GTK desde sua criação em 2001, se tornou amigo de seu criador, Andrei Zmievski. É membro do time de documentação e criador da comunidade brasileira de PHP-GTK ( <a href="http://www.php-gtk.com.br">www.php-gtk.com.br</a> ). Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions ( <a href="http://www.adianti.com.br">www.adianti.com.br</a> ) onde atua como consultor de tecnologia e provedor de soluções. Pode ser contatado pelo e-mail <a href="mailto:pablo@php.net">pablo@php.net</a> .	