

# Montando interfaces

O PHP-GTK oferece todo o poder da linguagem PHP à criação de interfaces gráficas com a biblioteca GTK+.

por Pablo Dall'Oglio



Cecile Graat - www.sxc.hu

Este artigo será o primeiro de uma série, onde conheceremos o *PHP-GTK*<sup>[1]</sup>, extensão da linguagem de programação *PHP* que nos permite desenvolver aplicações gráficas. Neste primeiro artigo, iremos conhecer melhor o porquê de sua criação e também alguns mitos e verdades a respeito dessa ferramenta que está conquistando cada vez mais desenvolvedores.

O PHP é originalmente uma linguagem de script *server-side*, ou seja,

scripts executados pelo servidor, e não pelos clientes, para efetuar a tarefa de interesse. Batizada primeiramente como *Personal Home Pages*, a linguagem foi concebida no outono de 1994 por Rasmus Lerdorf. Ele a escreveu como um modo de rastrear os visitantes de seu currículo online. A primeira versão data de 1995, época na qual Rasmus achou que se o PHP fosse de Código Aberto, teria suas falhas corrigidas por outras pessoas. O núcleo do PHP seria

reescrito mais tarde por Zeev Suraski e Andi Gutmans, formando o mecanismo que hoje é conhecido como *Zend Engine*. Atualmente, PHP significa *PHP: Hypertext Preprocessor*, e é uma poderosa linguagem de programação mantida por um grupo de desenvolvedores que

conta com uma vasta comunidade de programadores.

O *GTK* é um conjunto de bibliotecas cujo propósito é servir ao desenvolvedor como base para criar aplicações gráficas. Ele (*GIMP Toolkit*) foi originalmente desenvolvido para o software de manipulação de imagens *GIMP* (*GNU Image Manipulation Program*). Tem crescido muito desde o início do projeto, e hoje é utilizado como parte central do ambiente desktop *Gnome*. O *GTK+* também tem sido portado para o *BeOS* e a plataforma *Win32*, fazendo da linguagem a escolha perfeita para o desenvolvimento de aplicações gráficas multi-plataforma.

## PHP-GTK

O *PHP-GTK* não é uma nova linguagem de programação, mas um projeto que visa a disponibilizar toda a biblioteca de componentes gráficos *GTK* como uma extensão



**Figura 1** Um mesmo código em *PHP-GTK* pode ter visual nativo tanto em sistemas Linux quanto em Windows, sem necessidade de alteração.

do PHP. Com isso, permite a criação de aplicações gráficas com a linguagem PHP e tudo que ela oferece, como a sintaxe flexível, a tipagem dinâmica, a quantidade enorme de documentação a respeito, uma multiplicidade de livros e a grande comunidade ativa, além, é claro, da capacidade de se comunicar com praticamente todos os bancos de dados existentes e permitir a geração de gráficos, planilhas, documentos PDF, entre outros, através de suas demais extensões. O PHP-GTK é a primeira extensão da linguagem PHP que permite escrever aplicações *client-side* com interface gráfica. Foi escrita, em parte, para provar que o PHP é uma linguagem completa e de grandes propósitos.

## Multi-Plataforma

Ao escrever um programa em PHP-GTK, o desenvolvedor poderá rodá-lo da mesma forma nas principais plataformas do mercado – Linux, Windows e Mac –, sem exigir a recompilação do código. Isso é possível por que o código do PHP não é compilado, mas interpretado. Para o usuário, é necessário somente ter o interpretador do PHP com suporte à biblioteca GTK instalado no sistema operacional.

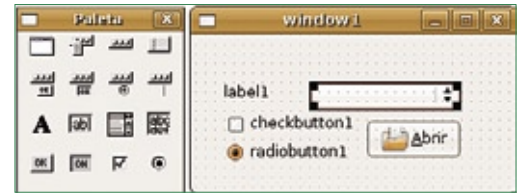
Atualmente na sua segunda versão, a biblioteca GTK possui um visual extremamente atraente. É possível escrever programas para rodarem com visual nativo em sistemas Linux e também em Windows®, como mostra a **figura 1**. Para isso, não será necessário alterar uma linha sequer de seu código-fonte.

## Desempenho

Muitas pessoas têm o receio de que um código interpretado seja lento ao rodar aplicações *standalone*, mas isso é um engano. O mecanismo Zend que existe por trás do PHP otimiza

em muito a execução do código, uma vez que ele é carregado na memória. O Zend realiza uma espécie de pré-compilação do código e, uma vez que as classes estejam carregadas na memória, a aplicação tem desempenho excepcional, não deixando nada a dever para linguagens compiladas.

Por exemplo, o carregamento de um simples grid com dados de mil pessoas de um banco de dados *PostgreSQL* não otimizado tem um tempo de carregamento de cerca de dois segundos em um computador atual. Há vários casos de aplicações em PHP-GTK rodando com folga em antigas máquinas Pentium 200 com 128 MB de memória RAM, com interface gráfica leve (*Fluxbox*) e um terminal de atendimento público, durante vários meses de forma ininterrupta, sem apresentar qualquer tipo



**Figura 2** A ferramenta *Glade* facilita muito a criação de interfaces gráficas.

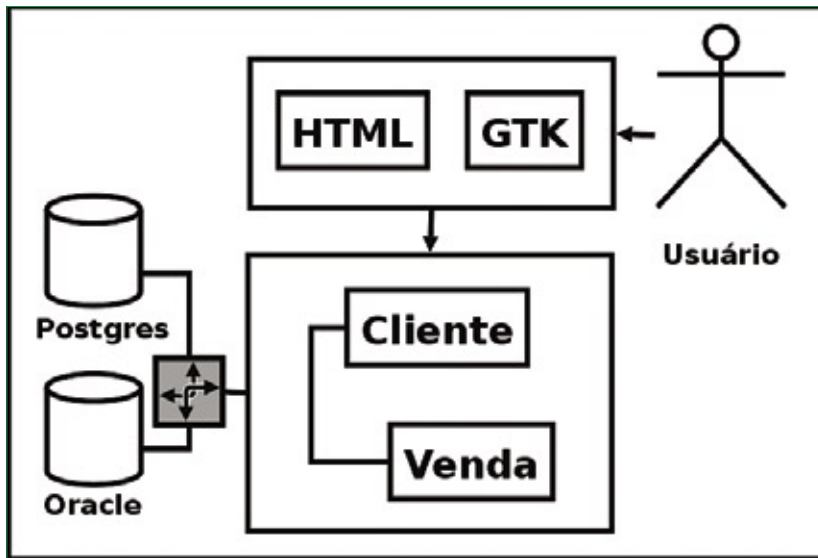
de problema e realizando centenas de transações por dia.

## Produtividade

De que adianta uma ferramenta de desenvolvimento ser rápida e multi-plataforma se não oferecer uma boa produtividade? A versão 5 do PHP, ou *PHP5*, surpreendeu o mundo ao adicionar recursos fabulosos no quesito de orientação a objetos. O PHP-GTK tira proveito de todos esses recursos ao máximo. Nesse sentido, o desenvolvedor pode criar facil-

### Exemplo 1: Programa simples em PHP-GTK

```
01 <?php
02 // cria a janela
03 $janela = new GtkWindow;
04 $janela->set_size_request(300,200);
05 $janela->set_border_width(20);
06 $janela->set_title('Título da Janela');
07
08 // cria duas caixas verticais
09 $pagina1 = new GtkVBox;
10 $pagina2 = new GtkVBox;
11
12 // adiciona alguns objetos
13 $pagina2->add(new GtkLabel('Código'));
14 $pagina2->add(new GtkEntry('1234'));
15
16 // cria um notebook
17 $note = new GtkNotebook;
18 $note->append_page($pagina1, new GtkLabel('Título 1'));
19 $note->append_page($pagina2, new GtkLabel('Título 2'));
20 // adiciona o notebook na janela
21 $janela->add($note);
22
23 // exhibe a janela
24 $janela->show_all();
25
26 Gtk::Main();
27 ?>
```



**Figura 3** Disposição do código do programa em PHP, usando uma interface Web ou GTK.

mente, por meio do mecanismo de herança, novos componentes, como formulários e listagens, baseados nos componentes nativos do GTK.

Além disso, o desenvolvedor pode fazer uso do *Glade* (figura 2), uma ferramenta reconhecida por toda a comunidade Linux por permitir desenhar interfaces para qualquer projeto GTK, independente da linguagem de implementação do aplicativo. O Glade salva toda a interface gráfica da aplicação num arquivo XML que é invocado e interpretado por ela em tempo de execução.

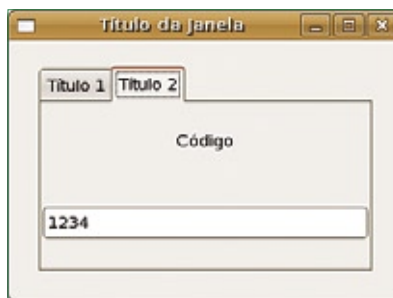
## Reutilização de código

Um dos grandes desafios da engenharia de software é a reutilização de código. A utilização de alguns padrões, como a modularização e a programação em camadas, maximizam a reutilização. Em PHP, não é raro termos de criar sistemas híbridos, com uma interface predominantemente Web e alguns módulos em GTK que rodam na máquina do cliente. Diversas situações tornam isso necessário, principalmente em interfaces que demandam maior agilidade, como terminais de atendimento e pontos de vendas, onde precisamos de te-

clas de atalho e comunicação com dispositivos periféricos.

Nesses casos, podemos reaproveitar toda a lógica de negócio da aplicação já escrita para a Web, agora na aplicação cliente, em GTK. Mas para isso, é praticamente imprescindível que a aplicação esteja orientada a objetos e que a camada com a lógica de negócio não contenha qualquer código visual (marcações HTML, por exemplo). Nesses casos, podemos utilizar as mesmas classes da Web, preocupando-nos apenas com a parte visual da aplicação (GTK).

Na figura 3, procuramos demonstrar uma aplicação em camadas, onde o usuário se comunica com a camada visual (HTML ou GTK), sendo que esta dispara eventos da camada



**Figura 4** Janela simples com duas abas criada pelo código do exemplo 1.

de negócio, que contém relacionamentos entre os objetos pertinentes ao modelo conceitual (*Cliente* e *Venda*, na figura) e trata da persistência destes no banco de dados.

## Prática

Para finalizarmos nosso primeiro artigo com um exemplo prático, vamos criar no exemplo 1 uma janela (objeto *GtkWindow*, linha 3) com duas abas (figura 4). A primeira aba estará vazia, e a segunda aba conterá uma caixa vertical com dois objetos. O primeiro objeto será um rótulo de texto (objeto *GtkLabel*, linha 13), enquanto o segundo será uma caixa de entrada de dados (objeto *GtkEntry*, linha 14).

## Conclusão

Neste primeiro artigo, vimos alguns conceitos básicos. Nos próximos artigos iremos abordar aspectos avançados, como a criação da interface, a ferramenta Glade, conceitos de sinais e *callbacks*, além de conexão com bancos de dados. ■

### Mais informações

- [1] PHP-GTK Brasil: <http://www.php-gtk.com.br>
- [2] Livro PHP-GTK: <http://www.php-gtk.com.br/book>
- [3] Site do Autor: <http://www.pablo.blog.br>

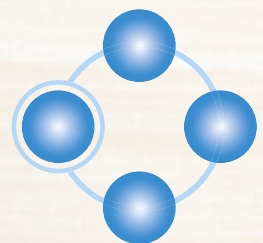
### Sobre o autor

**Pablo Dall'Oglio** é graduado em Análise de Sistemas pela UNISINOS. Autor do livro sobre PHP-GTK pela Novatec Editora, programa em PHP-GTK desde sua criação em 2001. É membro do time de documentação e criador da comunidade brasileira de PHP-GTK ([www.php-gtk.com.br](http://www.php-gtk.com.br)). Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions ([www.adianti.com.br](http://www.adianti.com.br)) onde atua como consultor de tecnologia e engenheiro de software. Pode ser contatado pelo e-mail [pablo@php.net](mailto:pablo@php.net).

A man in a dark suit and black shoes is holding a black briefcase with both hands. The briefcase has a large black bomb attached to it by a chain. The bomb is on the floor next to the man's feet. The background is a light-colored wooden floor.

**O ERP que você  
usa está travando  
o seu negócio?**

**Conheça  
a solução  
flexível  
Kenos  
ADempiere.**



**[www.kenos.com.br](http://www.kenos.com.br)  
(11) 4082-1305**

**Kenos**  
Sistemas de Gestão Integrada