

PHP-GTK

construa um gerenciador de downloads

FTP

faça upload seguro de arquivos

Banco de Dados

mapeamento objeto-relacional

Imagens

geração de mapas

Integração de Tecnologias

Javascript, MySQL e PHP



hello world





PHP-GTK

Criando Aplicações Gráficas com PHP

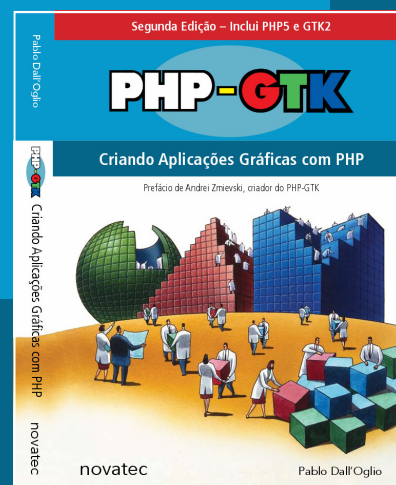
Todo programador já deve ter ouvido falar sobre PHP, uma das linguagens mais utilizadas no mundo. A popularidade do PHP deve-se à sua facilidade em criar aplicações dinâmicas para a internet, com suporte à maioria dos gerenciadores de bancos de dados existentes, além de um conjunto de funções que permitem desde a criação de simples portais até complexas aplicações de comércio eletrônico.

O que poucos entenderiam até alguns anos atrás é a possibilidade de criar aplicações-cliente, com janelas, botões, listas e painéis em PHP. Pois este é justamente o foco de PHP-GTK Criando Aplicações Gráficas com PHP. O PHP tem crescido muito nos últimos anos e se tornou uma linguagem de propósitos gerais. Este livro é prova disto. Por intermédio do PHP-GTK, você pode utilizar o PHP para desenvolver aplicações com visual atraente e nativo para várias plataformas, como Linux, Windows e Mac, utilizando a mesma linguagem que utiliza para criar aplicações para a Web.

Principais tópicos abordados no livro:

- Introdução ao PHP, arrays, strings, bancos de dados
- Tutorial completo sobre orientação a objetos
- Exemplos de utilização de janelas, árvores, listas, botões
- Exemplos de utilização de menus, diálogos, formulários
- Crie suas próprias classes, estendendo o GTK
- Exemplos com bancos de dados (inserção, listagem, navegação)
- Utilização da ferramenta Glade (geração de notas fiscais)

www.novatec.com.br



Um gerenciador de Downloads em PHP-GTK

Neste artigo iremos desenvolver um pequeno utilitário em PHP-GTK para baixar arquivos remotos. Assim como nos artigos anteriores, nosso objetivo não é desenvolver nenhuma ferramenta profissional, mas mostrar de forma didática como utilizar os recursos do PHP em conjunto com o GTK.

1. O Programa

A interface do programa será bastante simples. Como pode ser visto na Figura 1, a janela é composta de um campo onde o usuário irá digitar o endereço remoto do arquivo a ser baixado (objeto **\$file**), um botão "OK" para iniciar o download (objeto **\$inicia_btn**) e um botão "Cancelar" (objeto **\$cancel_btn**) para interromper o download. Logo a seguir, tem-se uma barra de progresso (objeto **\$progress**) que indica o percentual que já foi baixado do arquivo até o momento.

Para criar a interface do programa, a classe `GtkWindow`, que representa uma janela, será estendida. Desta forma, em seu método construtor, definiremos quais objetos estarão contidos na janela principal e também as ações do botão "OK", que executará o método **download()** e do botão "Cancelar" que executará o método **abort()**.

```
<?php
/**
 * classe para baixar arquivos
 * @author Pablo Dall'Oglio
 */
class Downloader extends GtkWindow
{
    private $file; //local do arquivo
    private $inicia_btn; //botão ok
    private $cancel_btn; //botão cancelar
    private $progress; //barra de progresso
    private $aborted; //controle de abortar

    /**
     * cria a interface
     */
    function __construct()
    {
        parent::__construct();
        parent::set_position(Gtk::WIN_POS_CENTER);
        parent::set_size_request(430,-1);
        parent::set_title('pDownloader!');

        // cria uma caixa vertical
        $vbox = new GtkVBox;
        $this->aborted = FALSE;
```

```
// cria uma caixa para o arquivo
$this->file = new GtkEntry('http://file.tgz');
// cria botões de OK e Cancelar
$this->inicia_btn = GtkButton::new_from_stock...
$this->cancel_btn = GtkButton::new_from_stock...
// define as ações dos botões
$this->inicia_btn->connect_simple('clicked',
    array($this, 'download'));
$this->cancel_btn->connect_simple('clicked',
    array($this, 'abort'));

// cria uma barra de progresso
$this->progress = new GtkProgressBar();
$this->progress->set_fraction(0);
$this->progress->set_text('Aguardando...');

// cria uma caixa horizontal
$hbox = new GtkHBox;
$hbox->pack_start($this->inicia_btn, FALSE,...
$hbox->pack_start($this->cancel_btn, FALSE,...

// organiza a interface
$vbox->pack_start($this->file, TRUE, TRUE);
$vbox->pack_start($hbox, FALSE, FALSE);
$vbox->pack_start($this->progress, FALSE,...

// adiciona a caixa vertical na janela
parent::add($vbox);
parent::show_all();
}
```

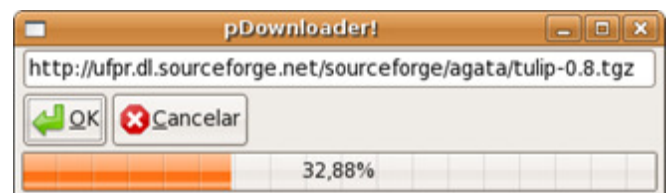


Figura 1 – Aplicação rodando

O próximo método que iremos escrever é o método **abort()**. Este método será executado sempre que o usuário clicar no botão cancelar. Seu funcionamento será bastante simples. Ele irá definir o valor da propriedade **\$aborted** como verdadeiro (TRUE) e também irá exibir uma mensagem ao usuário por meio de um diálogo de

Warning, alertando que o download foi cancelado.

```
/**
 * aborta o processamento
 */
function abort()
{
    // define o valor desta propriedade
    $this->aborted = TRUE;
    // exibe mensagem de warning
    $dialog = new GtkMessageDialog(null,
    Gtk::DIALOG_MODAL, Gtk::MESSAGE_WARNING,
    Gtk::BUTTONS_OK, 'Download Abortado !!');
    $response = $dialog->run();
    $dialog->destroy();
}
```

2. Baixando o arquivo

O método **download()** será responsável por baixar o arquivo. Para tal, ele irá abrir ambos arquivos remoto e local por meio do método **fopen()**. Em um laço de repetições, utilizaremos o método **fgets()** para ler os dados remotos e o método **fwrite()** para gravar no arquivo local. A função **events_pending()** e **main_iteration()** são utilizadas para que o GTK atualize a interface principal com o percentual de andamento do download e para que a mesma não fique congelada até o final do processo.

Neste método utilizamos a função **get_headers()** para obter a informação do tamanho do arquivo, antes de iniciar o download, para poder calcular o percentual concluído, e a função **pathinfo()**, para extrair somente o nome do arquivo da URL. Assim, o arquivo remoto será salvo na pasta **/tmp**, mantendo o mesmo nome.

O índice de conclusão do download (variável **\$indice**) será obtido acumulando a quantidade de bytes já baixados, por meio da variável **\$downloaded** e fazendo a sua proporção em relação ao tamanho total do arquivo (variável **\$filesize**).

```
/**
 * baixa um arquivo e salva na pasta /tmp
 */
function download()
{
    // obtém o nome do arquivo
    $url = $this->file->get_text();

    // obtém o tamanho do arquivo
    $header = get_headers($url, 1);
    $filesize = $header['Content-Length'];
    $remote = fopen($url, "r"); // abre o arquivo
    if (!$remote) // se arquivo não existe
    {
        return;
    }
    // obtém somente o nome do arquivo
    $pathinfo = pathinfo($url);
    $basename = $pathinfo['basename'];

    // cria um arquivo localmente
    $local = fopen ("/tmp/{$basename}", 'w');
    $downloaded = 0;

    // laço de repetições para baixar o arquivo
```

```
while(!feof($remote) and !$this->aborted)
{
    // atualiza a interface
    while (gtk::events_pending())
    {
        gtk::main_iteration();
    }
    // baixa até 1kb
    $var = fgets($remote, 1024);
    // grava o conteúdo localmente
    fwrite($local, $var);
    // total de bytes baixados
    $downloaded += strlen($var);
    // calcula índice para barra de progresso
    $indice = $downloaded/$filesize;
    $percent = ((int) ($indice * 10000))/100;
    // atualiza barra de progresso
    $this->progress->set_fraction($indice);
    $this->progress->set_text ("{$percent}%");
}

$this->aborted=FALSE;
// fecha arquivos
fclose($remote);
fclose($local);
}
```

3. Instanciando a classe principal

Para finalizar o programa, basta instanciarmos a classe principal para que a janela seja exibida na tela.

```
// instancia classe
new Downloader;
Gtk::main();
?>
```

Considerações finais

Neste artigo, demos continuidade à nossa série de artigos sobre PHP-GTK, sempre demonstrando uma funcionalidade específica da linguagem PHP útil no dia-a-dia. Devido ao espaço restrito aqui, algumas linhas foram truncadas. Logo a seguir, na seção referências, encontra-se o [Site do Programa], onde o mesmo poderá ser baixado integralmente.

Referências e links sugeridos

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

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

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

[Site do Programa] – <http://downloader.php-gtk.com.br>

Pablo Dall'Oglio - pablo@dalloaglio.net

Pablo Dall'Oglio é bacharel em Informática pela UNISINOS. Autor dos livros "PHP Programando com Orientação a Objetos" e "PHP-GTK - Criando Aplicações Gráficas com PHP", pela editora Novatec. Também foi o criador do PHP-GTK Brasil. Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions (www.adianti.com.br).