

Roni Márcio Fais

PHP-GENERATOR

CRIAÇÃO DE APLICATIVOS COM PHP E MySQL



Gerador de aplicativos PHP criado no projeto TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE FREEWARE do VIVA A ESCOLA 2009, um programa de complementação curricular do Estado do Paraná, sob orientação do Professor Roni Marcio Fais, mantido e distribuído pelo site www.rmfaais.com.

1ª Edição



Nota: muito zelo foi empregado na revisão e edição desta obra. No entanto, algumas falhas podem ter passado despercebidas. Se de alguma forma você puder contribuir para melhorias na próxima edição, envie sua dúvida, sugestão ou crítica ao endereço de e-mail rmfais@yahoo.com.br

Material de apoio relacionado a esta obra disponível em:
www.rmfais.com

Dados de Catalogação na Publicação

Criação de aplicativos com PHP e MySQL / Roni Márcio Fais / 1ª edição

Clube de Autores, 2015 – Joinville/SC

1. Educação Profissionalizante – Brasil;
2. Ensino Médio – Brasil;
3. Ensino Pós Médio – Brasil;
4. Ensino Superior – Brasil;
5. Programação HTML, JAVASCRIPT e PHP;
6. Banco de Dados MySQL, Lógica de Programação e Algoritmos.

Roni Márcio Fais

PHP-GENERATOR

CRIAÇÃO DE APLICATIVOS COM PHP E MySQL

Gerador de aplicativos PHP criado no projeto TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE FREEWARE do VIVA A ESCOLA 2009, um programa de complementação curricular do Estado do Paraná, sob orientação do Professor Roni Marcio Fais, mantido e distribuído pelo site www.rmfaais.com.

1ª edição



PHP-GENERATOR
CRIAÇÃO DE APLICATIVOS COM PHP E MySQL
Roni Márcio Fais

Nenhuma parte desta obra pode ser reproduzida ou duplicada sem autorização expressa do autor.

Direitos para esta edição
PROF. RONI MÁRCIO FAIS
www.rmfaiss.com

Impresso no Brasil.

DEDICATÓRIA

Aos meus avós **Benedito Fernandes** e **Angelina da Costa Fernandes** e ao meu tio **José Mário Fernandes** que, juntos, foram a base das minhas conquistas.

AGRADECIMENTOS

A Deus, pelo Dom da Vida, pelas incontáveis Bênçãos recebidas e pela Graça de poder concluir este trabalho.

A todos os meus alunos e alunas que, com seus questionamentos e anseios, foram a grande fonte de inspiração para criação desta obra.

SUMÁRIO

1. Introdução.....	09
2. Processo de Instalação.....	09
3. Módulos do PHP-GENERATOR.....	10
4. Arquivos do PHP-GENERATOR.....	18
5. Dicas para Criação de Projetos.....	20
6. Programação SQL para MySQL.....	26
7. Considerações Finais.....	33
8. Dados do Autor.....	35
9. Referências.....	35

1. INTRODUÇÃO

O PHP-GENERATOR, versão 2015, é um gerador de aplicativos que, por padrão, armazena dados em arquivo do tipo texto, mas que pode receber implementações para armazenamento no banco de dados no MySQL. Valendo lembrar que ele foi projetado para ser multiusuário, tanto no armazenamento em arquivo texto, quanto em banco de dados.

Com ele é possível automatizar tarefas como:

- Cadastro de clientes, produtos, vendas, agenda de amigos, etc;
- Sistema para downloads e uploads de arquivos;
- Fórum de discussões;
- Formulário para envio de e-mails;
- Vitrines virtuais;
- Além da criação de Sites, entre muitos outros aplicativos que atendam necessidades específicas.

2 PROCESSO DE INSTALAÇÃO

- Aquisição do servidor apache (distribuição xampp) + PHP-GENERATOR pelo site www.rmfaeis.com ou pelo e-mail rmfaeis@yahoo.com.br;
- Para ambiente windows, copiar a pasta xampp na raiz da unidade c:, acessar o painel de controle, dentro da pasta xampp e clicar em start para as opções apache e MySQL;
- Para ambiente linux, copiar a pasta lampp em /opt como super usuário e digitar no terminal `/opt/lampp/lampp start` para ativar o apache e mysql;

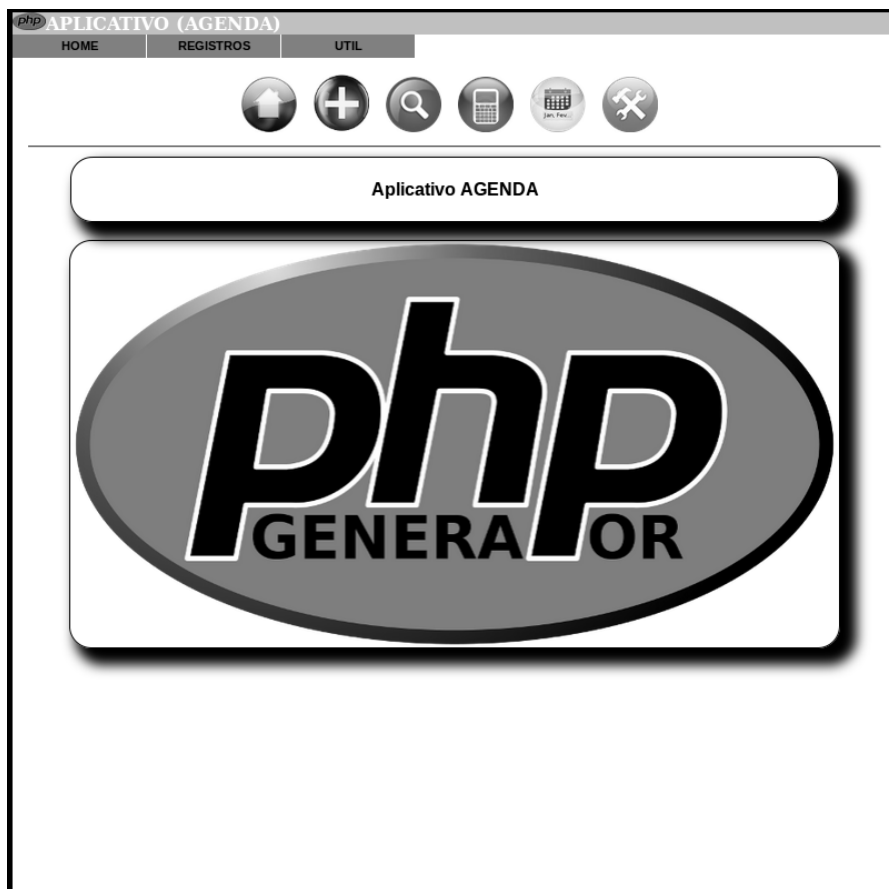
- Para executar o PHP-GENERATOR basta digitar na barra do navegador de Internet `http://localhost/phpgenerator`;
- Tudo que for desenvolvido no PHP-GENERATOR poderá ser compartilhado em outros computadores da rede local, informando-se neles, no lugar de localhost , o IP do computador onde foi instalado o servidor apache com o PHP-GENERATOR;
- O acesso poderá se dar também pelo Web, informando-se, no lugar de localhost, o IP de Internet.

3. MÓDULOS DO PHP-GENERATOR

O endereço **`http://localhost/phpgenerator`** abre no navegador o módulo para geração de novos projetos de aplicativos.

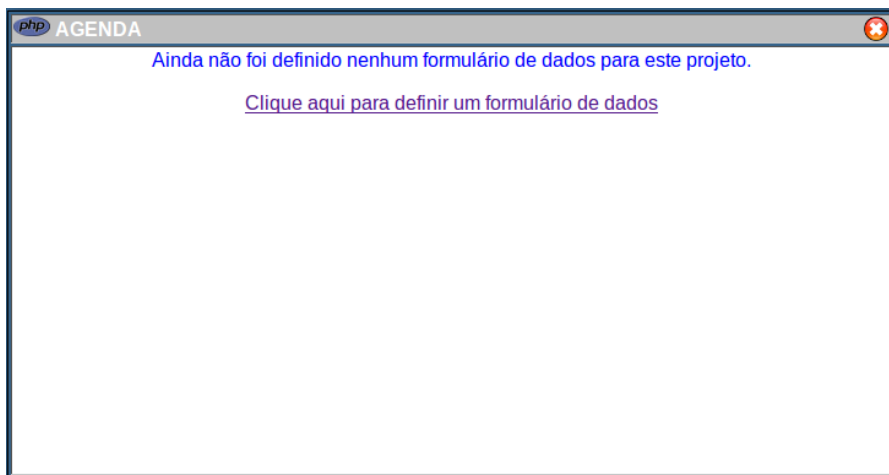


Ao dar um nome para o projeto e clicar em “Create Project”, uma subpasta com a estrutura básica de um novo aplicativo será criada.



Essa subpasta ficará armazenada dentro da pasta **htdocs** da distribuição Apache **xampp**, ou **lampp**, de acordo com o sistema operacional que está sendo utilizado. Clique no sinal de mais(+), que representa o ícone cadastro, para abrir a tela de cadastro, que, na primeira vez que for aberta, solicitará a definição do formulário de dados.

Tela de cadastro



Clique no link “definir um formulário de dados”, informe a senha “1234” e clique em “Enter”.



Especifique os campos que farão parte do formulário, informando nome, tipo e tamanho, sem seguida clique em salvar

AGENDA

MINI EDITOR DE ARQUIVO - edit.php

Gerador de campos para o formulário de dados

email caracter Gerar Código

```
//*****  
echo"<tr><td>nome</td><td><input name='nome' value='$campo[$c]' size='50' maxlength='50'  
onBlur='value=value.toUpperCase();'></td></tr>";  
$c++;  
//*****  
//*****  
echo"<tr><td>fone</td><td><input name='fone' value='$campo[$c]' size='15' maxlength='15'  
onBlur='value=valida(value);'></td></tr>";  
$c++;  
//*****
```

salvar voltar Arquivo: campos abrir

Os dados serão salvos no arquivo “campos.php”, que pode ser editado em um editor simples como bloco de notas.

Ao fechar o editor de formulários e clicar no ícone de cadastro, que tem o sinal de mais (+), a seguinte tela de cadastro será exibida, com opções básicas para manutenção de dados.



nome JOSE DA SILVA

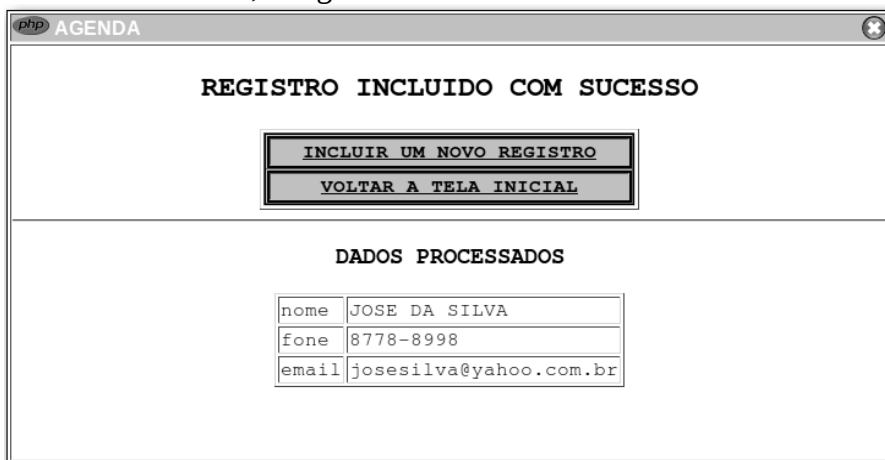
fone 8778-8998

email josesilva@yahoo.com.br

Salvar Consultar Sair

O arquivo “manifile.php”, que salva esses os dados, requer que todos os campos sejam preenchidos, ou seja, nenhum campo do formulário devesse ficar em branco.

Ao clicar em salvar, a seguinte tela será exibida:



REGISTRO INCLUIDO COM SUCESSO

INCLUIR UM NOVO REGISTRO

VOLTAR A TELA INICIAL

DADOS PROCESSADOS

nome	JOSE DA SILVA
fone	8778-8998
email	josesilva@yahoo.com.br

Depois de salvos, os dados poderão ser consultados a qualquer momento, clicando-se no ícone lupa, na tela principal, que abre o módulo de consulta, com várias opções de busca, conforme tela abaixo:

php

AGENDA

MODULO DE CONSULTA

Filtro de Busca:

JOSE

Qtde de Campos.:

4

Paginação

30

☐ relacionar dados

Cabeçalho.....:

doc,nome,fone,email,

☒ Listar em ordem numérica crescente

☐ Listar em ordem numérica decrescente

☐ Listar em ordem alfabetica pelo campo

2

☐ Listar registros excluidos

Gerar relatorio

O resultado, ao clicar em “Gerar relatório”, será um relatório com a seguinte estrutura:

php

AGENDA

APLICATIVO AGENDA - RELATORIO EMITIDO EM 08/07/2015 AS 20:27

doc	nome	fone	email
1	JOSE DA SILVA	8778-8998	josesilva@yahoo.com.br

Voltar

Imprimir

Total de registros encontrados= 1

Enviar e-mail a todos os registros listados

O aplicativo dispõe também de um módulo de configuração básica, onde podem ser alteradas cores, efeitos, senhas de proteção, tipo de armazenamento de dados (MySQL ou arquivo texto), ativação ou desativação de módulos de segurança, menus, ícones e edição de alguns arquivos. Para isso, basta clicar no ícone “martelo/chave” na tela principal.


AGENDA



MODULO QUE ALTERA ALGUMAS CONFIGURAÇÕES DO APLICATIVO

CONFIGURAÇÃO DE CORES	OUTRAS CONFIGURAÇÕES
Plano de fundo....: <input type="text" value="white"/> Barra de título...: <input type="text" value="silver"/> Barra de rodapé...: <input type="text" value="#BEBEBE"/> Tela de aplicativo: <input type="text" value="white"/> Fonte do fundo....: <input type="text" value="black"/> Fonte do título...: <input type="text" value="#FFFFFF"/> Fonte da tela.....: <input type="text" value="#000000"/> Barra de menu.....: <input type="text" value="gray"/> Fonte/Barra/Menu..: <input type="text" value="black"/> Tarja de menu.....: <input type="text" value="#49A3FF"/> Fonte/Tarja/Menu..: <input type="text" value="#FFFFFF"/> Tela de Menu.....: <input type="text" value="#EAEBD8"/> Fonte/Tela/Menu...: <input type="text" value="#2875DE"/> GradienteFundo....: <input type="text"/> GradienteTelas....: <input type="text"/> GradienteTitulo...: <input type="text"/> GradienteExtra....: <input type="text"/>	<input type="checkbox"/> Ativar módulo de segurança <input type="checkbox"/> Desativar menu <input type="checkbox"/> Desativar ícones <input type="checkbox"/> Ativar armazenamento no MySQL Nova senha de administrador <input type="text"/> Nova senha de cadastro <input type="text"/> Editar: <div> <input type="button" value="menu"/> <input type="button" value="ícones"/> <input type="button" value="campos"/> <input type="button" value="relacoes"/> </div> <div> <input type="button" value="creditos"/> <input type="button" value="css"/> </div> Arquivo PNG altera LogoMarca e JPG fundo <div> <input type="button" value="Selecionar arquivo..."/> Nenhum arquivo selecionado. </div> <div> <input type="button" value="Enviar Arquivo PNG"/> <input type="button" value="Enviar Arquivo JPG"/> </div>

Ao ativar o módulo “armazenamento no MySQL” e clicar em “SALVAR ALTERAÇÕES”, outras opções irão surgir, possibilitando definir usuário e senha do banco de dados MySQL, ativar gerador de relatórios, editor de comandos SQL para o MySQL, além de opções de “backup” e “restore” para criação e restauração de cópias de segurança.

MODULO QUE ALTERA ALGUMAS CONFIGURAÇÕES DO APLICATIVO

CONFIGURAÇÃO DE CORES

Plano de fundo....:

Barra de título....:

Barra de rodapé....:

Tela de aplicativo:

Fonte do fundo....:

Fonte do título....:

Fonte da tela.....:

Barra de menu.....:

Fonte/Barra/Menu...:

Tarja de menu.....:

Fonte/Tarja/Menu...:

Tela de Menu.....:

Fonte/Tela/Menu...:

GradienteFundo....:

GradienteTelas....:

GradienteTitulo....:

GradienteExtra....:

OUTRAS CONFIGURAÇÕES

☐ Ativar módulo de segurança

☐ Desativar menu

☐ Desativar icones

☐ Desativar armazenamento no MySQL

Nova senha de administrador

Nova senha de cadastro

Configuração básica do MySQL

usuário	Senha	Banco
root	<input type="text"/>	test

Edição de códigos SQL

Editar:

Arquivo PNG altera LogoMarca e JPG fundo

Nenhum arquivo selecionado.

Obs: Para que o módulo “armazenamento no MySQL” funcione, é necessário desenvolver dentro do banco de dados “test”, do MySQL, uma tabela de dados com o mesmo nome do projeto da pasta de dados gerado pelo PHP-GENERATOR, contendo os mesmos campos especificados no formulário do arquivo “campos.php”, com o detalhe de que o primeiro campo da tabela deverá se chamar código do tipo INT(6 ou 7 ou 8 ou 9...), marcado como chave primária de incremento automático;

4. ARQUIVOS DO PHP-GENERATOR

A programação no PHP-GENERATOR é baseada em package (pacote), pois cada novo aplicativo é moldado em cima de scripts PHP predefinidos em uma pasta chamada protótipo com o seguinte conteúdo:

Tabela de Arquivos

Arquivos e pastas	Descrição
_protec.php	Arquivo que contém senhas de proteção
cadastro.php	Aplicativo que contém formulário de cadastro html
calculadora.php	Aplicativo contendo calculadora simples em javascript
calendario.php	Aplicativo contendo calendário simples em javascript
campos.php	Arquivo onde serão definidos os campos do formulário html
creditos.php	Arquivo que contém dados exibidos na tela principal (arquivo desktop.php)
css.php	Arquivo que contém código de programação CSS
close.png	Arquivo de imagem
config.php	Aplicativo de configuração
consulta.php	Aplicativo para consulta de dados armazenados
contato.txt	Arquivo que contém mensagem para envio de e-mail
cores.txt	Arquivo que contém códigos de cores utilizadas
desktop.php	Aplicativo que monta tela principal
edit.php	Aplicativo para edição de arquivos PHP
email.php	Aplicativo que faz envio de e-mails
email.txt	Arquivo que contém e-mail do administrador de

	sistema
erro.php	Aplicativo que exibe mensagem de erro
files.php	Arquivo que contem extensões válidas para uploads/downloads
funcoes.php	Arquivo que contém funções php
funcoes.txt	Arquivo que contem funções javascript
icones	Pasta que contém ícones utilizados no desktop
icones.php	Arquivo que contém lista dos ícones mostrados no desktop
index.php	Aplicativo que exibe pagina principal sem menus
index2.php	Aplicativo que exibe página principal com menus
labels.txt	Arquivo que contém labels de relatorio
logomarca.png	Arquivo de imagem a ser exibido no desktop
manifile.php	Aplicativo que grava dados do formulário html no arquivo regs.php
Livro.pdf	Manual de instrução do PHP-GENERATOR-2015
menu.php	Arquivo que contém itens do menu
mysql.php	Arquivo onde devera ser informado usuário e senha do MySQL
mysqlexecuta	Script que executa uma instrução sql no MySQL
mysqlretorna.php	Script retorna dados de uma instrução sql feita ao MySQL
pesquisa.php	Aplicativo que pesquisa dados em ordem de código
pesquisa_sql.php	Aplicativo que realiza consulta no banco de dados MySQL
php.png	Arquivo de imagem
phpgenerator.php	Arquivo com breve descrição sobre o PHP-

	GENERATOR-2015-TXT
phpgenerator.png	Arquivo de imagem
regs.php	Arquivo que armazena os dados do formulário html
relation.php	Script que relaciona campos de diferentes tabelas de dados
relacoes.php	Script que relaciona diferentes arquivos de dados ou tabela de dados.
relatorio_sql.php	Aplicativo que gera relatórios dos dados armazenados no MySQL
relatorio.php	Aplicativo que exida dados armazenados no arquivo regs.php
relatorios.php	Arquivo que contem comandos SQL para ser usados no aplicativo consulta.php
senha.php	Aplicativo que solicita senha de acesso
table.php	Aplicativo que cria telas
webmail.php	Aplicativo que cria formulário de e-mail html

5. DICAS PARA CRIAÇÃO DE PROJETOS

5.1. O formulário de dados deve ser criado no arquivo “campos.php”, conforme o seguinte exemplo:

```
//*****
echo"<tr><td>nome</td><td><input name='nome' value='$campo[$c]'
size='30' maxlength='50' onBlur='value=value.toUpperCase();'></td></tr>";
$c++;
//*****
```

No exemplo dado será criado um campo de formulário para

digitação do nome de uma pessoa com largura de 30 pixels, podendo receber 50 caracteres, convertidos todos para maiúsculos por meio do evento javascript `onBlur`, que faz chamada ao método `toUpperCase`. Para converter para minúsculas poderia ter sido utilizado o método `toLowerCase`.

O evento `onBlur` pode ser utilizado para fazer chamada a outras funções javascript pré-definidas pelo PHP-GENERATOR no arquivo **funcoes.txt**, utilizando-se as seguintes sintaxes:

- `onBlur='value = validadata(value);'` - valida um campo data no formato dd/mm/aaaa;
- `onBlur='value = validanum(value);'` - valida um campo só para números;
- `onBlur='value=validaint(value);'` - valida um número inteiro;
- `onBlur='value = validacpf(value);'` - valida digitação de um CPF;
- `onBlur='value = validacnpj(value);'` - valida digitação de um CNPJ;
- `onBlur='value = validames(value);'` - valida digitação de mês com 2 dígitos;
- `onBlur='value = validaano(value);'` - valida digitação de ano com 4 dígitos.
- `onBlur='value = validasenha(value);'` - valida confirmação da digitação de uma senha.

5.2. Relacionamentos de dados em diferentes tabelas ou arquivos podem ocorrer da seguinte forma:

- Para de dentro de um formulário pesquisar dados de outro, basta usar a função

pesquisar("nome_do_outro_cadastro"); e acrescentar no input do referido campo a ser pesquisado a cláusula `html id='nome_do_outro_cadastro'`, se houver necessidade de mais de um campo ser pesquisado neste outro cadastro, pode se usar `id='nome_do_outro_cadastro02'`, etc.
exemplo: `pesquisar("cidades");`

- Para de dentro de um formulário criar uma caixa de seleção com dados de outro, basta usar a função `selecionar("nome_do_campo","value_do_campo","nome_do_outro_cadastro",0);`
exemplo:
`selecionar("CodCidade","$campo[5]","cidades",0);` onde 0(zero) determina que será retornado o código do outro registro;
- Os dados de um cadastro criado com o PHP-GENERATOR serão armazenados, por padrão, em um arquivo texto chamado `regs.php`.

5.3. Uploads de arquivos e senhas de segurança podem ser implementados da seguinte maneira:

- Para enviar um arquivo para o servidor e fazer com que este seja exibido junto com os dados do cadastro, no caso desse arquivo ser uma imagem com extensão “jpg”, crie um campo `type=file name=_file` no formulário; Se o arquivo enviado for uma imagem “jpg”, ele também poderá ser visualizado no aplicativo de

relatório, se não for uma imagem, ficará disponível para download no aplicativo de relatório.

- Para incluir uma senha de acesso individualizado em cada registro, crie um campo com name=senha com type=password; Nesse caso é importante também criar um campo `<input name=_senha onBlur='value = validasenha(value);'>` para validar a senha;
- A senha informada na segunda linha do arquivo _protec.php serve como senha de administrador ou programador, dando acesso irrestrito a todos os registros;
- Para criar uma chave ou senha de confirmação para gravação de registro, basta criar um campo type=password com name=_chave. Este campo será usado para verificar a senha informada na terceira linha do arquivo chave.php. Se o arquivo _protec.php for renomeado para protec.php é desnecessário criar o campo name=_chave, pois será acionado um módulo de segurança que, automaticamente, protegerá os módulos cadastro, consultas e relatórios.
- A senha padrão para acesso ou consulta de registros é **1234**. A senha padrão para proteção do aplicativo cadastro é **4321** e podem ser alteradas no arquivo _protec.php;
- Executar o módulo **config.php** no navegador, informando a senha padrão **1234**, caso ela não tenha sido alterada no arquivo _protec.php, facilita configurar

algumas dessas restrições de segurança.

5.4. Para criar ou alterar ícones edite o arquivo `ícones.php` e siga as seguintes instruções:

- informe, na segunda linha do arquivo, separados por vírgula, nomes dos ícones a serem criados. Para cada nome deverá haver um arquivo png, na pasta “ícones”, e um arquivo php com o mesmo nome do ícone;
- A segunda linha não deve ser mudada de lugar. Caso não existir arquivos png e php com um dos nomes acima, serão executados os arquivos “icone.png” e “erro.php”;
- Para que não seja criado nenhum ícone no aplicativo, basta excluir ou renomear o arquivo `ícones.php`. Aconselha-se renomear, colocando-se o símbolo underline (`_`) no início do nome do arquivo.;
- Pode-se, também, apagar a linha que contém os nomes de ícones, deixando uma linha em branco no lugar, com isso nenhum ícone será criado no aplicativo.

5.6. Para criar ou alterar menus edite o arquivo `menu.php` e siga as seguintes instruções

- Cada novo menu pode ser criado a partir da segunda linha do arquivo, sendo que cada menu com as suas respectivas opções deve ter uma linha própria;
- A primeira palavra informada será o título do menu e as demais as opções, devendo, todas as palavras, serem separadas por vírgula;
- Para cada opção deverá ser criado um arquivo PHP com o mesmo nome da opção informada no menu.

5.5. Implementações para armazenamento no banco de dados MySQL pode se dar da seguinte forma:

- Renomear o arquivo `_mysql.php` para `mysql.php`;
- Desenvolver dentro do banco de dados **test** do MySQL uma tabela de dados com o mesmo nome de projeto da pasta de dados gerada pelo PHP-GENERATOR, contendo os mesmos campos especificados no formulário do arquivo `campos.php`, com o detalhe de que o primeiro campo da tabela deverá se chamar **codigo do tipo INT(6 ou 7 ou 8 ou 9...), marcado como chave primária de incremento automático**;
- Informar dentro do arquivo `mysql.php` nome, senha de usuário e banco de dados do MySQL a ser usado.
Com isso o aplicativo estará pronto para gravar, consultar e alterar dados diretamente no MySQL.
- Executar o módulo **config.php** no navegador, informando a senha padrão **1234**, caso ela não tenha sido alterada no arquivo `_protec.php`, facilita configurar algumas dessas configurações relacionadas ao MySQL.

6. PROGRAMAÇÃO SQL PARA MySQL

A linguagem de programação SQL (Structured Query Language), linguagem de consulta estruturada, surgiu na década de 70, em um dos laboratórios da empresa estadunidense IBM, tornando-se a linguagem padrão para bancos de dados relacionais.

Nesta parte do livro, serão mostrados alguns **exemplos básicos** de programação SQL, testados em um banco de dados MySQL.

Nesses exemplos, o nomes de bancos, tabelas e campos não estão entre os sinais de crase. Caso isso seja exigido em sua versão do MySQL, acrescente esse sinal em suas sintaxes.

6.1. Exemplos de alguns tipos de campos mais utilizados pelo banco de dados MySQL.

Tipo	Descrição	Mínimo	Máximo
INT ou INTEGER	Número inteiro	-2147483648	+2147483647
BIGINT	Inteiro longo	-9223372036854775808	+9223372036854775807
FLOAT	Ponto flutuante simples	-1.175494351E-38	+3.402823466E+38
DOUBLE	Ponto flutuante normal	-2.2250738585072014 E-308	+1.7976931348623157E+308
VARCHAR	Caracteres diversos		
DATE	Data	'1000-01-01'	'9999-12-31'

6.2. Principais operadores relacionais, lógicos e matemáticos

Operador Relacional	Descrição
=	igual
<>	diferente
>	maior
<	menor
>=	Maior ou igual
<=	Menor ou igual
Operador Lógico	Descrição
AND	e
OR	ou
Operadores Matemáticos	Descrição
+	adição
-	subtração
*	multiplicação
/	divisão

6.3. Exibindo bancos de dados existentes.

```
SHOW DATABASES;
```

6.4. Criando um novo banco de dados chamado “sistema”.

```
CREATE DATABASE sistema;
```

6.5. Colocando um banco de dados já existente em uso.

```
USE test;
```

6.6. Criando uma tabela chamada “**idades**” em um banco já existente chamado “**test**”. Se o banco de dados “**test**” já estiver em uso. por meio do comando **USE “test”**, é desnecessário informá-lo antes do nome da tabela “**idades**”.

```
CREATE TABLE test.idades (codigo INT( 6 ) UNSIGNED  
NOT NULL AUTO_INCREMENT,
```

```
nome VARCHAR(50),  
uf VARCHAR(2),
```

```
PRIMARY KEY ( codigo ) ) ;
```

UNSIGNED significa que código é um numero positivo, ou seja, maior que zero.

6.7. Exibindo tabelas do banco de dados “**test**”.

```
SHOW tables;
```

6.8. Exibindo informações sobre a tabela **idades**, do banco de dados “**test**”.

```
DESCRIBE test.idades;
```

6.9. Inserindo dados na tabela “**idades**” do banco de dados “**test**”.

```
INSERT INTO test.cidades (codigo,nome,uf) VALUES  
('','LOANDA','PR');  
INSERT INTO test.cidades (codigo,nome,uf) VALUES  
('','ROSANA','SP');  
INSERT INTO test.cidades (codigo,nome,uf) VALUES  
('','RIO DE JANEIRO','RJ');  
INSERT INTO test.cidades (codigo,nome,uf) VALUES  
('','NOVO HORIZONTE','SP');  
INSERT INTO test.cidades (codigo,nome,uf) VALUES  
('','NOVO HORIZONTE','BA');
```

Observe que, em VALUES, código não é informado porque na criação da tabela ele foi definido como número inteiro, não nulo, de auto-incremento.

6.10. Selecionando registros inseridos na tabela “**idades**” do banco de dados “**test**” para consulta de dados.

```
Seleciona todos os registros da tabela cidades  
SELECT * FROM test.cidades;  
Seleciona apenas os campos especificados de todos os registros  
SELECT codigo,nome FROM test.cidades;  
  
Seleciona apenas os registros em que o campo uf='SP'  
SELECT * FROM test.cidades WHERE uf='SP';  
  
Seleciona apenas os registros em que o campo uf='SP' ou 'PR'  
SELECT * FROM test.cidades WHERE uf='SP' or uf='PR';  
  
Seleciona apenas os registros em que o campo nome='NOVO  
HORIZONTE' e uf='SP'
```

```
SELECT * FROM test.cidades WHERE nome='NOVO  
HORIZONTE' and uf='SP';
```

Seleciona todos os campos de todos os registros em ordem crescente pelo campo nome

```
SELECT * FROM test.cidades ORDER BY nome;
```

Seleciona todos os campos de todos os registros em ordem decrescente pelo campo nome

```
SELECT * FROM test.cidades ORDER BY nome DESC;
```

Seleciona todos os registros em que o campo nome contém o texto 'HORIZONTE'

```
SELECT * FROM test.cidades WHERE nome LIKE  
'%HORIZONTE%'
```

6.11. Alterando registros inseridos na tabela “**cidades**” do banco de dados “**test**”

Altera o conteúdo atual pelo conteúdo especificado em SET para o registro em que codigo = 1

```
UPDATE test.cidades SET  
codigo="",nome="CAMPINAS",uf="SP" WHERE codigo=1;
```

6.12. Deletando registros inseridos na tabela “**cidades**” do banco de dados “**test**”

Exclui o registro em que codigo = 12

```
DELETE FROM test.cidades WHERE codigo = 12;
```

Exclui todos os registros em que uf='PR'

```
DELETE FROM test.cidades WHERE uf = 'PR';
```

Limpa todos os dados da tabela zerando o auto-incremento do campo codigo

```
TRUNCATE TABLE test.cidades;
```

6.13. Alterando a estrutura de uma tabela

Altera o campo nome para 40 caracteres

```
ALTER TABLE test.cidades MODIFY nome VARCHAR(40)
```

Altera inserindo uma nova coluna chamada tamanho no inicio

```
ALTER TABLE test.cidades ADD tamanho INT(15) FIRST;
```

Altera inserindo uma nova coluna chamada cep no final

```
ALTER TABLE test.cidades ADD cep VARCHAR(15);
```

Altera inserindo uma nova coluna chamada regioao depois da coluna uf

```
ALTER TABLE test.cidades ADD regioao VARCHAR(15)  
AFTER uf;
```

Altera excluindo o campo cep e a coluna cep

```
ALTER TABLE test.cidades DROP cep
```

Altera renomeando-a para localidades

```
ALTER TABLE cidades RENAME localidades
```

6.14. Copiando ou renomeando uma tabela de dados

Copia a tabela cidades e toda sua estrutura para a tabela localidades

```
CREATE TABLE localidades SELECT * FROM cidades;
```

Renomeia tabela cidades para municipios

```
RENAME TABLE test.cidades TO test.municipios;
```

6.15. Exportando e importando tabela de dados (muito útil na realização de backups de tabelas)

Exporta uma tabela de dados para um arquivo txt

```
SELECT * FROM test.cidades INTO OUTFILE  
'/tmp/cidades.txt' FIELDS TERMINATED BY '#' LINES  
TERMINATED BY '\n';
```

Importa uma tabela de dados

```
LOAD DATA LOCAL INFILE '/tmp/cidades.txt' INTO TABLE  
cidades FIELDS TERMINATED BY '#' LINES TERMINATED  
BY '\n';
```

Onde: o símbolo suspenso(#) separa um campo de outro e (\n) é um quebra de linha, fazendo com que cada registro seja exportado para uma nova linha no arquivo texto.

6.16. Removendo uma tabela ou um banco de dados

Remove permanentemente a tabela cidades

```
DROP TABLE test.cidades;
```

Remove permanentemente o banco de dados test

```
DROP DATABASE test;
```

6.17. Manutenção de tabelas de dados

Remove permanentemente a tabela cidades

```
DROP TABLE test.cidades;
```

Remove permanentemente o banco de dados test

```
DROP DATABASE test;
```


7. CONSIDERAÇÕES FINAIS

Ter tido a oportunidade de ser professor orientador do Projeto Tecnologia em Desenvolvimento de Software Freeware, do Programa Viva a Escola, em 2009, no Colégio Paraná, em Loanda, foi uma oportunidade ímpar de atingir grandes objetivos, com base nos anseios de alunos que buscavam ir um pouco além da simples utilização de frameworks, no sentido de prover novas tecnologias e não apenas utilizá-las.

Mostrar o logística que existe por trás de um gerador de software e responder a inúmeras indagações, foi o ponto de partida para uma primeira versão do PHP-GENERATOR, que, nos anos seguintes, iria se apresentar como excelente ferramenta didática no ensino de programação para novas turmas que estariam iniciando o curso técnico em informática.

Em pouco tempo o modesto gerador de softwares em PHP foi usado na elaboração de um controle de biblioteca e aplicativos online para o site do mesmo colégio onde ele fora desenvolvido.

O projeto mostrou-se tão útil e eficaz que não poderia parar por aí. O mesmo inspirava novas versões com mais recursos e melhoramentos. Para tanto seria preciso novas críticas e sugestões. Era necessário disponibilizá-lo para o maior número de pessoas possíveis. Uma tarefa que se concretizou devido a divulgação do download pela Agência Estadual de Notícias do Paraná, rádios de Curitiba e outros veículos de comunicação.

Por cinco anos o projeto ficou disponível para download e utilização online em meu site pessoal: www.rmfaeis.com, podendo receber as tão necessárias críticas e sugestões, que resultou em uma versão mais refinada da original, que é alvo de estudo nesse pequeno tutorial, em forma de livro, eternamente inconclusivo! Mas, incontestavelmente útil, podendo sempre ser melhorado por incentivadores de software livre.

Este pequeno tutorial sequer mostra todos os recursos do PHP-GENERATOR. Muitos recursos podem ser explorados de forma intuitiva. O objetivo aqui não é ir a fundo em todas as possibilidades, pois elas são infinitas, considerando que o projeto é de código aberto.

A maioria dos procedimento de programação encontram-se

documentados no próprio código fonte e podem ser alterados, melhorados, ou detonados de acordo com a habilidade do aventureiro. Para isso, basta estudar e analisar cada um dos arquivos descritos na **tabela de arquivos**.

Os aplicativos gerados pelo PHP-GENERATOR são responsivos, ou seja, projetados para se adaptarem automaticamente a diferentes tipos de dispositivos, desde computadores desktops até smartphones.

É sempre bom lembrar que um gerador de aplicativos facilita em muito o trabalho de desenvolvimento, mas não é a resposta para todos os problemas, razão pelo qual é imprescindível que o programador conheça pelo menos a estrutura básica das linguagens utilizadas no desenvolvimento de seus aplicativos, ou mesmo de um sistema mais complexo.

Nesse quesito, o PHP-GENERATOR é um forte aliado, pois tem como princípio e base a filosofia de software livre com código open source, que dá total liberdade de ação aos desenvolvedores.

8. DADOS DO AUTOR

Roni Márcio Fais

Nascido em Novo Horizonte/SP, aos 28/09/1974. Graduado em Ciência da Computação em 16/02/2001. Pós Graduado em Administração, Supervisão e Orientação Educacional em 17/09/2004. Professor Acadêmico nos anos de 2002 a 2005. Atuou como docente na educação infantil e ensino fundamental no ano de 2004. Concluiu Formação Pedagógica pela UEM (Universidade Estadual de Maringá/PR) em 2011. Atualmente é professor da rede pública de ensino no Estado do Paraná em cursos profissionalizantes.

E-mail do autor: rmfais@yahoo.com.br

Site do autor: www.rmfais.com

9. REFERÊNCIAS

Fais. Roni Márcio. Programação para Internet. Firstinbooks, São Paulo, 2011, 1ª edição.

<http://php.net/docs.php>. Manual oficial de programação da linguagem PHP, acessado em julho de 2015.

MySQL 5.6 Reference Manual. Manual de referência do MySQL fornecido junto com a distribuição xampp do servidor de Internet Apache.