

CakePHP

Elton Luís Minetto



Agenda

- **O que é o CakePHP**
- **Conceitos básicos**
- **Requisitos**
- **Instalação**
- **Configuração**
- **Aplicação de exemplo**



O que é o CakePHP

Cake é um framework para PHP que usa padrões de desenvolvimento conhecidos como ActiveRecord e MVC. A idéia principal é ser um framework estruturado que permita a usuários PHP de todos os níveis desenvolver aplicações web robustas sem perda da flexibilidade.



Vantagens

- **Licença flexível (MIT)**
- **Compatibilidade com PHP4 e PHP5**
- **Geração de CRUD para interação com BD**
- **Arquitetura MVC**
- **Validações**
- **Templates**
- **Ferramentas que auxiliam gerar Javascript, AJAX, forms HTML, etc**
- **Comunidade ativa**
- **Funciona em qualquer subdiretório web, com pouca configuração do Apache.**

Conceitos - MVC

Separação da estrutura da aplicação em três partes distintas: Modelo, Visão e Controle:

- **Modelo:** gerencia os dados da aplicação
- **Visão:** gerencia a saída gráfica e textual da parte da aplicação visível ao usuário
- **Controle:** interpreta as entradas de mouse e teclado do usuário, comandando a Visão e o Modelo para se alterarem de forma apropriada.

Conceitos - Diretórios

/app - aplicação

/config - arquivos de configuração, DB, etc

/controllers - controladores da aplicação

/index.php - página inicial da aplicação

/models - modelos da aplicação

/plugins

/tmp - usado para cache e logs

/vendors - bibliotecas de terceiros

/views - visões da aplicação

/elements - elementos, pedaços da visão

/errors - páginas de erros customizadas

/helpers - ajudantes para gerar código

/layouts - arquivos de layout

/pages - visões estáticas

/webroot - DocumentRoot para aplicação

/css - arquivos css

/files - arquivos comuns

/img - imagens

/js - Javascripts

/cake - código fonte do cake



Requisitos

- **Servidor web com as funcionalidades: sessions, mod_rewrite (não obrigatório mas aconselhável)**
- **PHP 4.3.2 ou superior**
- **Uma base dados. Atualmente é suportado MySQL, PostgreSQL e AdoDB. Futuramente Oracle - se meu patch for aceito :-)**



Instalação

- **Download do arquivo em <http://cakephp.org/downloads>**
- **Descompactar e copiar para o diretório root do servidor Web. No Ubuntu:**

```
tar xfvj cake_1.1.10.3825.tar.bz2  
mv mv cake_1.1.10.3825 /var/www/cake
```

- **Testar no navegador no endereço: <http://localhost/cake>**

Configuração Apache

É necessário duas configurações: o `mod_rewrite` e o `AllowOverride` precisam estar ativos.

Alterar o `httpd.conf` (`apache2.conf` no Ubuntu) e descomentar ou adicionar as linhas :

```
LoadModule rewrite_module modules/mod_rewrite.so  
AddModule mod_rewrite.c
```

Adicionar a linha abaixo no `DocumentRoot` e reiniciar o Apache:

```
AllowOverride all
```

talk is cheap...

...show me the code!



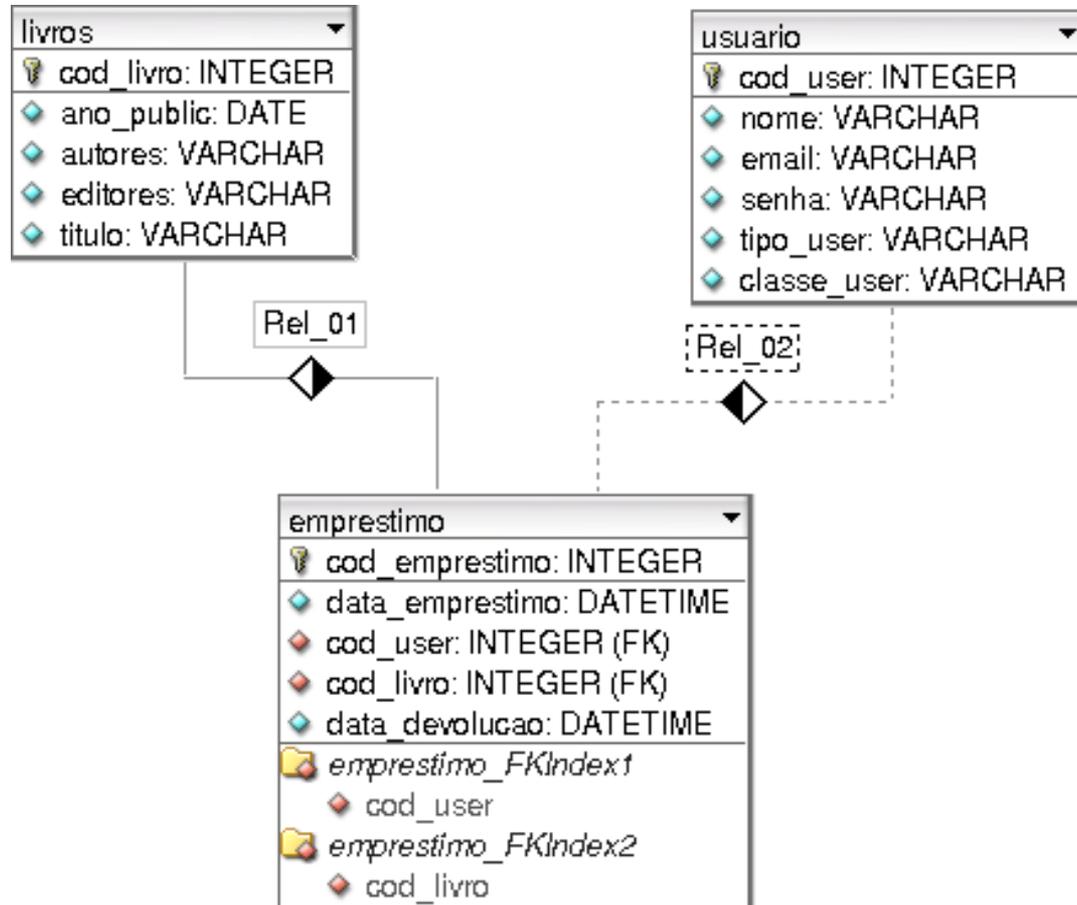
Desenvolvendo

A aplicação

Um professor gostaria de uma aplicação web simples para manter-se informado sobre os livros que empresta a seus alunos. Quando ele entre no sistema pode visualizar e alterar os alunos cadastrados e também os empréstimos.

Desenvolvendo

A modelagem



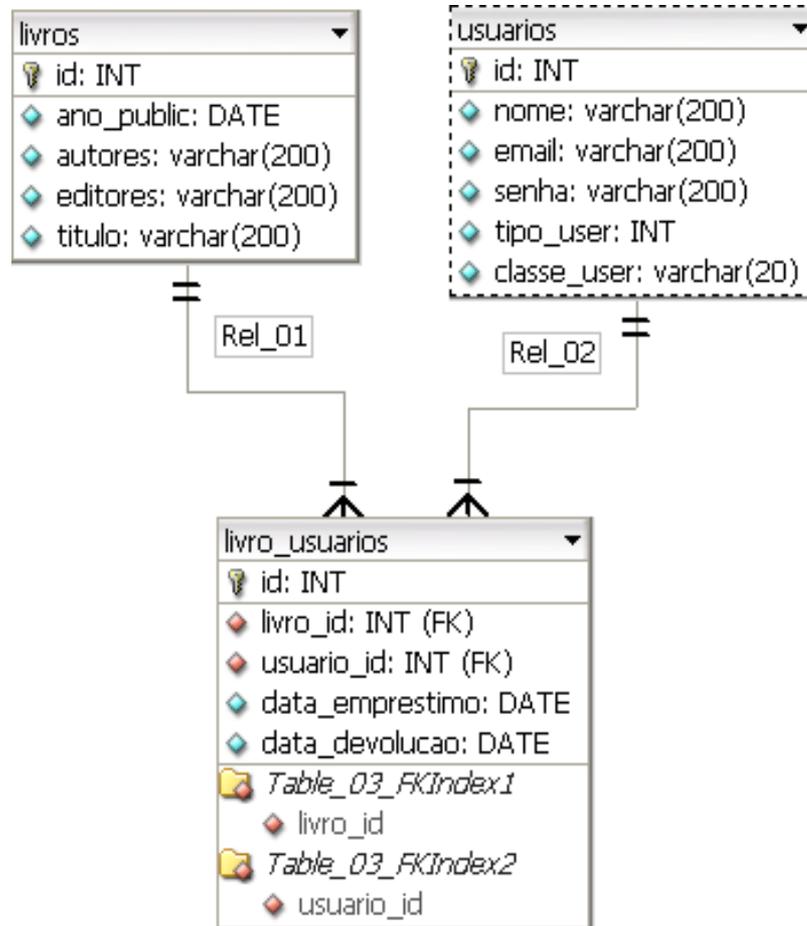
Desenvolvendo

O Cake segue algumas regras de padronização para a base de dados:

- Todas as tabelas devem estar no plural
- A chave primaria para todas as tabelas deve ser o campo "id"
- Para utilizar chave estrangeira, deve possuir nome da tabela no singular acrescido do campo id. Ex: "usuario_id".
- O relacionamento "n - n", deve ser organizado em ordem alfabética, e o primeiro nome no singular acrescido de um "_" nome segunda tabela plural. Ex: livro_usuarios;

Desenvolvendo

Modelagem adaptada:



Desenvolvendo

Iniciando a aplicação

O cake possui uma excelente ferramenta para geração da aplicação e de várias partes de código. A ferramenta chama-se bake.

Desenvolvendo

Criando um projeto:

```
cd /var/www/cake/
```

```
php cake/scripts/bake.php -project /var/www/cake/conference
```

Após responder y para as perguntas, toda a estrutura de diretórios é criada.

Depois de criado o projeto é necessário criar a aplicação.

Desenvolvendo

Criando uma aplicação:

```
php cake/scripts/bake.php -app conference
```

Várias perguntas serão feitas neste momento:

- What database driver would you like to use? (mysql/mysqli/mssql/sqlite/postgres/odbc) – qual database. Neste caso mysql
- What is the hostname for the database server? - localhost
- What is the database username? - elton (ou outro usuario)
- What is the database password? - senha
- What is the name of the database you will be using? - qual é a base de dado usada. Neste exemplo cake. Esta base de dados e as tabelas devem ser previamente criadas.



Desenvolvendo

- Enter a table prefix? - caso as tabelas tenham algum prefixo no seu nome. Neste caso – n
- Look okay? (y/n) – confirmar com y caso tudo esteja correto

Neste momento foi criado as configurações de conexão com a base de dados da aplicação. Estas configurações foram salvas em `/var/www/cake/conference/config/database.php` e podem ser alteradas posteriormente.

Desenvolvendo

Agora é solicitado a opção de gerar o Modelo, a Visão ou o Controle da aplicação.

Gerando o Modelo: O bake conecta na base de dados e encontra as tabelas existentes. Você pode escolher para qual das tabelas irá criar o modelo. Inicialmente vamos escolher a tabela Livro digitando seu número da lista mostrada.

- **Would you like to supply validation criteria for the fields in your model? (y/n)- se deseja fornecer critérios de validação para os campos da tabela. Neste caso y. É apresentado cada um dos campos e, de acordo com seu tipo na tabela é apresentada algumas validações. Por exemplo o campo id pode ser validado para VALID_NOT_EMPTY ou VALID_NUMBER.**

Desenvolvendo

- **Would you like to define model associations (hasMany, hasOne, belongsTo, etc.)? (y/n)** – se é necessário definir associações entre os modelos. Existem quatro associações:
 - **hasOne:** possui um
 - **hasMany:** possui muitos
 - **belongsTo:** pertence a
 - **hasAndBelongsToMany:** tem e pertence a muitosNeste modelo não é preciso associação.
- **Cake test suite not installed. Do you want to bake unit test files anyway? (y/n)** - se é preciso gerar os arquivos para o PHPUnit. Neste caso n

Desenvolvendo

Agora é preciso criar o Controle. Executando novamente o comando:

```
php cake/scripts/bake.php -app conference
```

- **Possible Controllers based on your current database – escolher o Livros**
- **Would you like bake to build your controller interactively? - y**
- **Would you like this controller to use other models besides 'Livro'? (y/n) – se é necessário usar mais de um modelo. Neste caso n**
- **Would you like this controller to use other helpers besides HtmlHelper and FormHelper? (y/n) – se são necessários outros ajudantes além do gerador de Html e Forms. Pode ser usado o de Ajax. Neste momento n**



Desenvolvendo

- **Would you like this controller to use any components? (y/n) – se é preciso usar algum componente especial. n**
- **Would you like to use Sessions? (y/n) – usar sessões sempre é interessante, então y**
- **Would you like to include some basic class methods (index(), add(), view(), edit())? (y/n) - aqui está toda a graça de usar o bake. Ele pede se deseja criar métodos básicos como o de adição, exclusão, etc. Responder y reduz um monte de trabalho depois.**
- **Would you like to create the methods for admin routing? (y/n) – neste caso n**
- **y para confirmar e n para a geração dos arquivos para o PHPUnit**



Desenvolvendo

E, finalmente, para gerar a Visão, executar novamente o comando anterior, escolher a opção V e o número do Livros.

- Would you like to create some scaffolded views (index, add, view, edit) for this controller? - ele vai criar a visão para os métodos básicos de inserção, exclusão, etc.
Responder y
- Would you like to create the views for admin routing?
(y/n) - n

Pronto. Está criado a primeira parte da aplicação. O mesmo pode ser feito para a tabela usuarios.

Desenvolvendo



CakePHP Rapid Development

List Livros

Id	Ano Public	Autores	Editores	Titulo	Actions
1	1952-01-06	elton		teste	View Edit Delete

- [New Livro](#)



Desenvolvendo

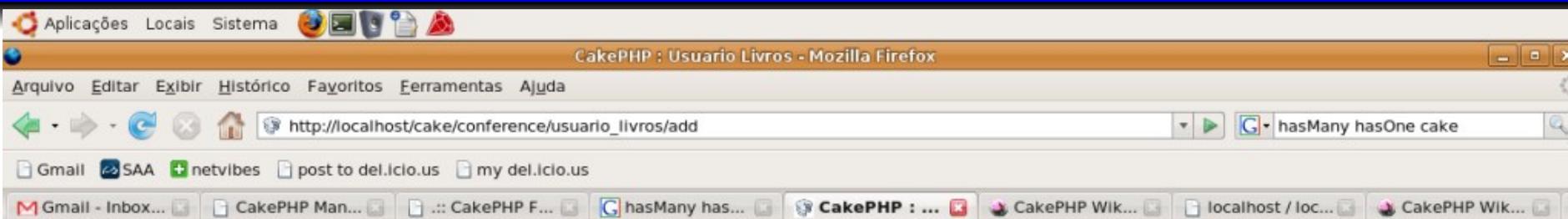
A tabela `livro_usuarios` é uma tabela associativa. Assim, no momento da geração do modelo deve ser respondido `y` para a pergunta: **Would you like to define model associations (hasMany, hasOne, belongsTo, etc.)? (y/n)**

O `bake` vai automaticamente identificar a existência do relacionamento:

`LivroUsuario belongsTo Usuario? (y/n) - y`

`LivroUsuario belongsTo Livro? (y/n) - y`

Desenvolvendo



CakePHP Rapid Development

New Usuario Livro

Usuarios

Livros

Data Emprestimo

Data Devolucao

Add

- [List Usuario Livros](#)
- [View Usuarios](#)
- [Add Usuarios](#)
- [View Livros](#)
- [Add Livros](#)

CAKEPHP POWER



Customizando

Na visão de adição de empréstimos aparece o código do aluno e o código do livro. Vamos alterar para aparecer o nome. No arquivo `controllers/livro_usuarios_controller.php` no método `add()` existe uma linha:

```
$this->set('usuarios', $this->LivroUsuario->Usuario->generateList());
```

O `generateList()` é usado para gerar um array com os dados da tabela. Quando não são passados parâmetros o array é gerado apenas com os ids da tabela.

Customizando

Alterando para :

```
$this->set('usuarios', $this->LivroUsuario->Usuario->generateList(null, 'nome', null, '{n}.Usuario.id', '{n}.Usuario.nome'));
```

Desta forma o array é gerado com id como chave e o nome do usuário como valor. Assim é gerado na tela um select mais interessante. O mesmo pode ser feito com o livro:

```
$this->set('livros', $this->LivroUsuario->Livro->generateList(null, 'titulo', null, '{n}.Livro.id', '{n}.Livro.titulo'));
```

Customizando



CakePHP Rapid Development

New Usuario Livro

Usuarios	<input type="text" value="elton"/>
Livros	<input type="text" value="Anansi Boys"/>
Data Emprestimo	<input type="text" value="November"/> <input type="text" value="12"/> <input type="text" value="2006"/>
Data Devolucao	<input type="text"/> <input type="text"/> <input type="text"/>

- [List Usuario Livros](#)

Customizando - Ajax

O Cake possui uma série de Helpers para auxiliar no desenvolvimento. Um destes Helpers é o de Ajax. Ele usa as bibliotecas JavaScript Prototype e script.aculo.us que devem ser copiadas para o diretório `webroot/js/` da aplicação.

Vamos fazer um pequeno exemplo. Uma pequena busca de livros usando Ajax. Primeiro é preciso criar um novo controlador para a aplicação. No diretório `controllers` foi criado um novo arquivo, chamado `buscas_controller.php`:

Customizando - Ajax

```
<?php
class BuscasController extends AppController
{

    var $name = 'Buscas';
    var $helpers = array('Html', 'Ajax' ); //uso dos helpers
    var $uses = array("Livro"); //indica qual modelo usar

    //funcao da pagina inicial
    function index()
    {
    }

    //funcao que sera executada por AJAX para retornar os livros
    function update ()
    {
        $consulta = $this->params["form"]["titulo"]."%%";
        $this->set('livros',$this->Livro->findAll("titulo LIKE '$consulta'"));
        $this->layout = "ajax";
    }
}
?>
```

Customizando - Ajax

O próximo passo é criar as visões. No diretório views deve-se criar um diretório para as views da aplicação. Neste caso chamamos de buscas. Os arquivos correspondentes aos métodos do controlador foram criados:

`index.thtml`

`update.thtml`

Customizando - Ajax

index.shtml

```
<?php echo $html->charsetTag('UTF-8')?>  
<?php echo $javascript->link('prototype')?>  
<?php echo $javascript->link('scriptaculous.js?load=effects')?>  
  
<?php echo $ajax->form(array('action' => '/buscas/update'), 'livro',  
array('update' => 'resultados')); ?>  
<input type="text" name="titulo">  
<input type="submit" value="Pesquisar">  
<div id="resultados"> </div>
```

As primeiras linhas carregam os arquivos .js necessários. Com o `$ajax->form` é criado um formulário cujo action é a função `update`. Os outros parâmetros são o modelo (livro) e as opções (neste caso está definindo que a função `update` vai alterar a div `resultados`).

Customizando - Ajax

update.shtml

```
<?php echo $ajax->div('resultados'); ?>  
<?php foreach($livros as $livro) {  
    echo $livro['Livro']['titulo']."<br>";  
}  
?>  
<?php echo $ajax->divEnd('resultados'); ?>
```

Este é o código que irá mostrar os resultados.

Conclusões

- **Desenvolver usando frameworks facilita a padronização e trabalho em equipe.**
- **Acelera o ciclo de desenvolvimento.**
- **Cake é fácil de aprender.**
- **Cake usa uma estrutura de arquivos prática e simples**
- **Tem um nome legal :-)**



Referências

- <http://www.cakephp.org>
- <http://www.cakephp.com.br>



Contato

Elton Luís Minetto

eminetto@gmail.com

<http://www.eltonminetto.net>

