

# **Desenvolva rapidamente utilizando o framework CakePHP**

**Elton Luís Minetto**

# Agenda

- **Ambiente Web**
- **PHP**
- **Problemas**
- **Frameworks**
- **CakePHP**
- **Demonstração**

# Ambiente Web

**É o ambiente formado por algumas tecnologias:**

- **Servidor Web**
- **Protocolo HTTP e HTTPS**
- **Navegadores**
- **HTML, PHP, JSP, CSS, AJAX**
- **Web sites e sistemas**



# PHP

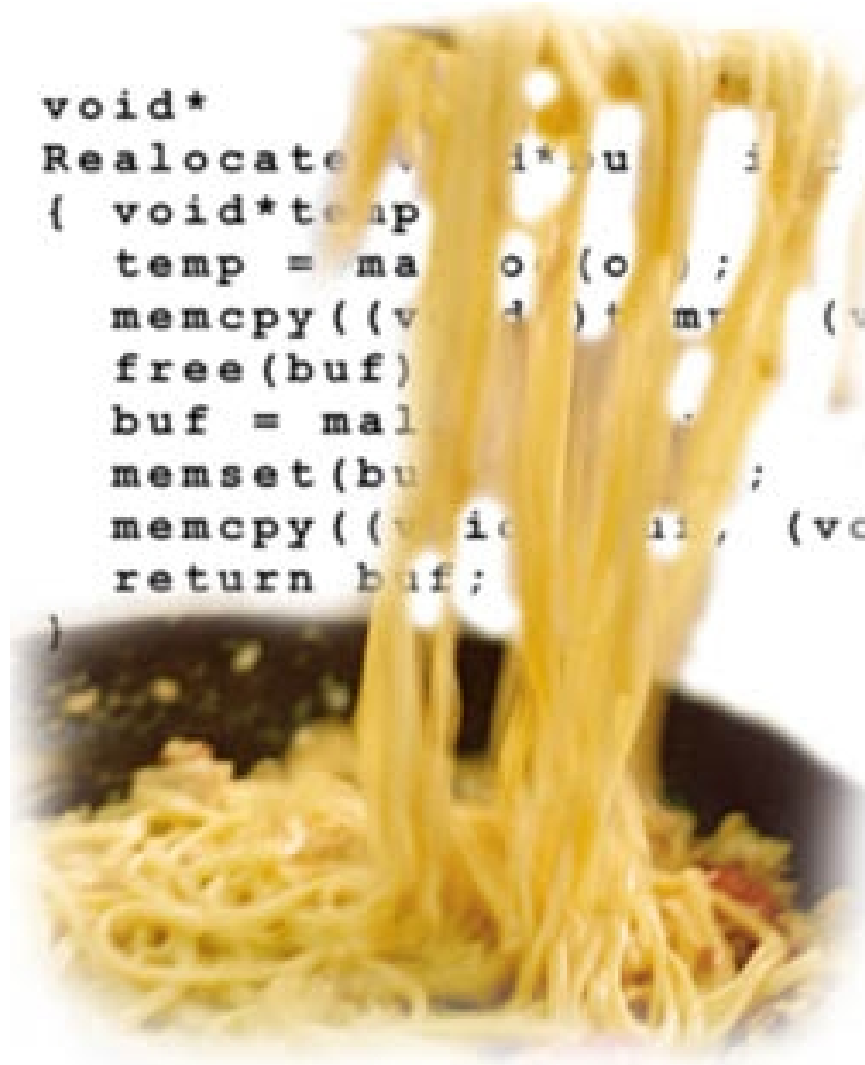
- **Livre**
- **Fácil de usar**
- **Grande documentação**
- **Grande biblioteca de funções**
- **Orientação a Objeto**

# Problemas Ambiente Web/PHP

- **Repetição de tarefas**
- **Dificuldade de debug**
- **Várias formas de fazer a mesma coisa**
- **Existem poucos padrões e os que existem não são cumpridos (W3C x Microsoft)**
- **Os design patterns foram criados para aplicações cliente-servidor, somente agora estão sendo portados para web**
- **Clientes querem alterar tudo muito rápido**
- **Desenvolvimento lento**
- **Pouco reaproveitamento de código**
- **Não existem ferramentas RAD como Delphi**
- **Dificuldade de fazer relatórios**
- **Navegadores não renderizam corretamente algumas coisas**
- **Novas tecnologias surgindo a todo minuto**
- **Mistura de layout e lógica**

# Problemas Ambiente Web/PHP

```
void*
Reallocate (void* buf, int size)
{ void* temp;
  temp = malloc (size);
  memcpy ((void*) temp, (void*) buf, size);
  free (buf);
  buf = temp;
  memset (buf, 0, size);
  memcpy ((void*) buf, (void*) temp, size);
  return buf;
}
```



# Frameworks

- É uma “base” de onde pode-se desenvolver algo maior ou mais específico. É uma coleção de códigos, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos softwares.
- **Vantagens**
  - Todos programam parecido graças a padronizações. Fácil manutenção
  - Automatiza tarefas repetitivas. DRY
  - Separação de layout e lógica. *no more spaghetti!*
  - *Reaproveitamento de códigos*

# Padrões

**Os frameworks são baseados em uma série de padrões de desenvolvimento já consolidadas:**

## **MVC**

**MVC é um acrônimo para Model, View, Controller (Modelo, Visão e Controlador). A idéia é separar todo o desenvolvimento de uma aplicação nestas três partes, ou camadas:**

- Model – gerencia o comportamento dos dados da aplicação.**
- View – gerencia a saída gráfica e textual da parte da aplicação visível ao usuário**
- Controller – interpreta as entradas de mouse e teclado do usuário, comandando a Visão e o Modelo para se alterarem de forma apropriada.**



# Padrões

## ActiveRecord

**Este padrão de design facilita a manipulação de dados contidos em uma base de dados por aplicações desenvolvidas com o paradigma orientados a objetos. Uma tabela de uma base de dados é vista na forma de uma classe, enquanto que cada linha da tabela é considerada um objeto desta classe. Quando um objeto é criado, alterado ou excluído esta ação é automaticamente refletida na base de dados. Desta forma não é necessário que o desenvolvedor conheça uma linguagem de manipulação de dados como SQL, além de manter toda a aplicação desenvolvida no paradigma orientado a objetos.**

# 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.**



# CakePHP - 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 - 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, SQLite, ODBC e AdoDB. Futuramente Oracle**

# Instalação

**Download do arquivo em  
<http://cakephp.org/downloads>**

**Descompactar e copiar para o diretório root do servidor Web. No Ubuntu:**

```
tar xfvj cake_1.1.15.5144.tar.bz2  
mv cake_1.1.15.5144 /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

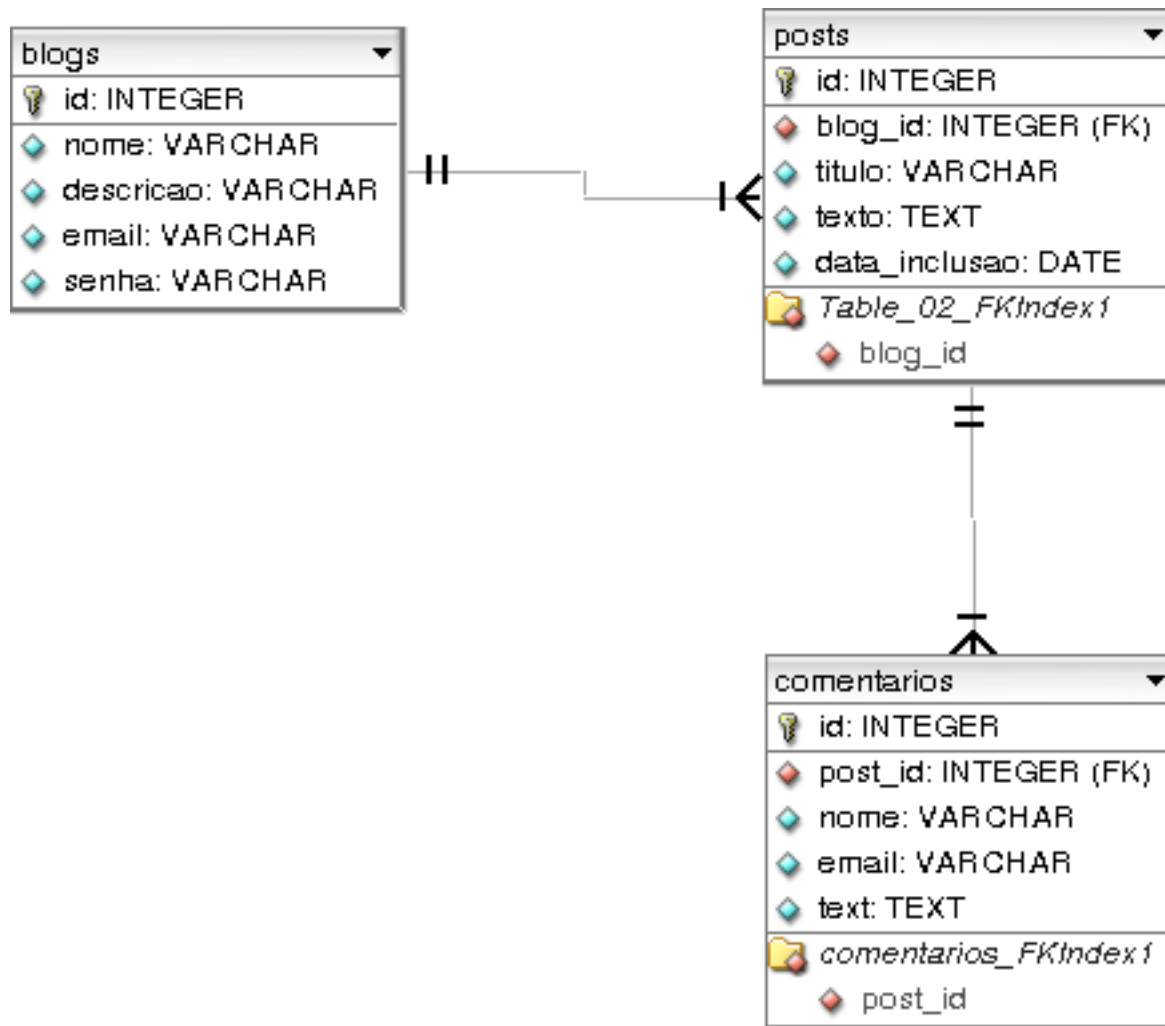
**A aplicação consiste de um sistema de blogs. Um usuário postar suas notícias e visitantes podem fazer comentários nas notícias**

# Desenvolvendo

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

- **Todas as tabelas devem estar no plural.**
- **A chave primária 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: blog\_id.**
- **O relacionamento "n-n", deve ser organizado em ordem alfabética e os nomes devem estar no plural. Exemplo: livros\_usuarios**
- **Modelos são no singular e controladores no plural**

# Modelagem



# Usando o CakePHP

## *Iniciando a aplicação*

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



# Usando o CakePHP

## ***Demonstração do bake***

- ***Criando o projeto***
- ***Configurando a base de dados***
- ***Criando as aplicações***
  - ***Modelos***
  - ***Controladores***
  - ***Visões***

# Usando o CakePHP

## ***Associações entre Tabelas***

***hasOne: possui um***

***hasMany: possui muitos***

***belongsTo: pertence a***

***hasAndBelongsToMany: tem e pertence a muitos***

# Customizando

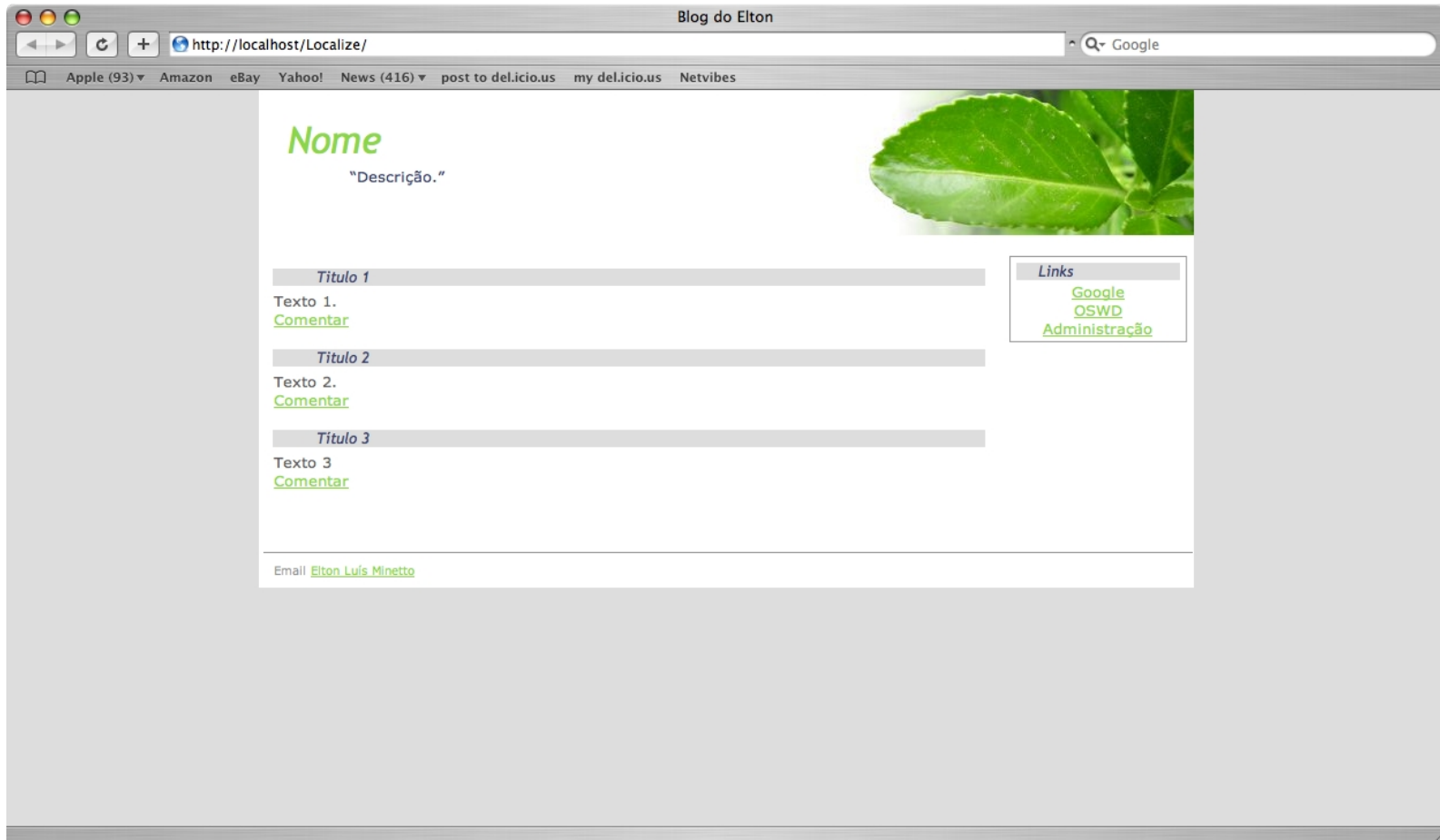
*Após a utilização do bake temos um "esqueleto" de aplicação.*

*Agora podemos customizar os códigos gerados para criarmos a aplicação final.*



# Layout

***Nosso designer nos mandou o layout do site. Agora precisamos colocar este visual nas aplicações geradas pelo CakePHP.***



# Layout

***O layout principal das aplicações encontra-se no arquivo `views/layouts/default.thtml`. Mudando este arquivo podemos transformar automaticamente todas as aplicações geradas.***

***Os arquivos `.css` devem ser salvos no diretório `webroot/css/`***

***Os arquivos `.js` (Javascript) caso existam devem ser salvos no diretório `webroot/js/`***



**Code Monkey Like You.**

# Página inicial

*Quando acessamos o link*

*<http://localhost/cake/blog/>*

*iremos redirecionar para a página que lista os posts.  
Para isso deve-se alterar o arquivo*

*[views/pages/home.thtml](#)*

*Seu novo conteúdo é:*

```
<?php
```

```
header('location:posts');
```

```
?>
```

# Posts

***O controlador `posts_controller.php` gerado pelo `bake` possui diversos métodos. Vamos alterá-lo e deixar somente o método `index()` e os métodos `admin`.***

***O método `index()` é o responsável por mostrar todos os posts.***



# Posts

***Além disso é preciso alterar o arquivo  
views/posts/index.shtml***

***para mostrar os dados vindos da tabela.***



# Comentários

***O visitante possui duas opções na página de posts: visualizar os comentários e adicionar um novo comentário.***

***Para isso iremos utilizar as páginas geradas pelo bake. Quando clicar em "Ver comentários" ele será direcionado para a página de visualização de comentários e clicando em "Comentar" será direcionado para o formulário de adição.***

***Foram necessárias algumas alterações no `comentarios_controller.php`***



Code Monkey Like You.

# Controle de acesso

**Quando o usuário clicar na opção Administração na página inicial será direcionado a um formulário para fazer o login na aplicação.**

**Para realizar esta tarefa é preciso os seguintes passos:**

- 1) Criar o controlador em `controllers/auth_controller.php`. Neste arquivo deve-se criar os métodos `login`(para mostrar o formulário e fazer a validação) e `logout`(sair do sistema)**
- 2) Criar o diretório `views/auth/` com o arquivo `login.thtml`, que é o formulário de login**

# Controle de acesso

**3) *Verificar se foi realizado o login no momento do acesso da interface de administração. Alterar o arquivo `app_controller.php`.***





# 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>***

***<http://bakery.cakephp.org/>***

***<http://cakeforge.org/>***

**Atenção!**

***O próximo slide é  
uma prova da  
cara-de-pau do  
palestrante!***

# Propaganda

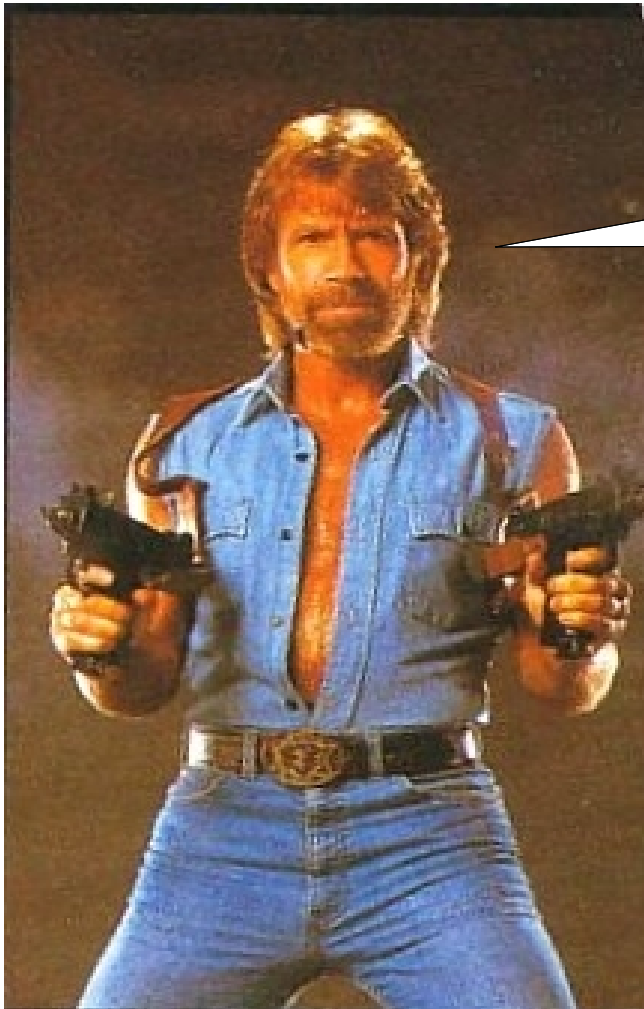


## Frameworks para Desenvolvimento em PHP

Elton Luís Minetto

novatec

# Depoimentos



Melhor que um  
roundhouse kick!

# Depoimentos



A força sinto nesse livro!

# Contato

**Elton Luís Minetto**

**eminetto@gmail.com**

**<http://www.eltonminetto.net>**