



## \* Frameworks PHP

- \* Edgar Rodrigues Sandi
- \* @EdgarSandi
- \* [edgar@season.com.br](mailto:edgar@season.com.br)
- \* [edgarsandi.com.br](http://edgarsandi.com.br) – under construction

# Minibio

- ▶ ZCE – Zend Certified Enginner
- ▶ Gerente de projetos
- ▶ Desenvolvedor PHP e Java
- ▶ Ministra os treinamentos:
  - Linguagens de Programação
    - PHP I – Fundamentos ( Oficial **Zend** )
    - PHP II – Estruturas Superiores ( Oficial **Zend** )
    - Academia do Programador ( Oficial **Globalcode** )
  - Bancos de Dados (MySQL / PostgreSQL e Oracle)
  - MS Project
  - WebDesign (Suíte Adobe)
- ▶ Instrutor homologado **Globalcode**

# Season

## CONSULTORIA E TREINAMENTOS AVANÇADOS EM INFORMÁTICA

- \* Quem é a **Season Treinamentos**?
- \* A **Season Treinamentos** é o único centro autorizado a realizar treinamentos oficiais das tecnologias **Zend** no Brasil.
- \* Outras parcerias de treinamentos oficial:



# Cursos Oficiais da Zend no Brasil



The PHP Company

- \* Treinamentos oficiais:
  - \* **PHP I – Fundamentos**
  - \* **PHP II – Estruturas Superiores**
- \* Próximos treinamentos oficiais:
  - \* **Zend Framework**
  - \* **Zend Server**
  - \* **Zend Studio**
  - \* Preparatório para as certificações **ZCE** e **ZFC**

# Cursos Oficiais da Zend no Brasil



The PHP Company

## ► Treinamentos oficiais em São Paulo:



### PHP I - Fundamentos

Treinamento Oficial Zend



**Carga Horária**

24 horas

### Próximas Turmas



**São Paulo/SP**

**06.06.2011 Seg. Quar. e Sexta - Noite**



### PHP II – Estruturas Superiores

Treinamento Oficial Zend



**Carga Horária**

40 horas

### Próximas Turmas



**São Paulo/SP**

**20.06.2011 Seg. Quar. e Sexta - Noite**

# Cursos Oficiais da Zend no Brasil



The PHP Company

## ▶ Próximo minicurso em São Paulo:



**A certificação ZCE**  
**22.06.2011 Quarta feira – 19h**



**Carga Horária dos minicursos**  
**3 horas**

# Quem é Zend Technologies?



The PHP Company

- \* Quem é **Zend Technologies**?
  - \* **Zend** é uma empresa norte-americana fabricante de software.
  - \* Seus produtos são orientados para a plataforma PHP com ênfase no gerenciamento e melhoria do desempenho de aplicações web utilizando esta tecnologia.

# Objetivo

- \* Objetivo deste minicurso:
  - \* Discutir frameworks e
  - \* Apresentar de forma superficial 5 dos frameworks PHP existentes



# Prefácio

- 🕒 Falando de frameworks
  - \* Sua definição
  - \* Seu uso
  - \* Vantagens e Desvantagens
- \* E um pouco sobre
  - \* CakePHP
  - \* CodeIgniter
  - \* Symfony
  - \* YiiFramework
  - \* Zend Framework

# Falando de Frameworks

- \* Definição

Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação.

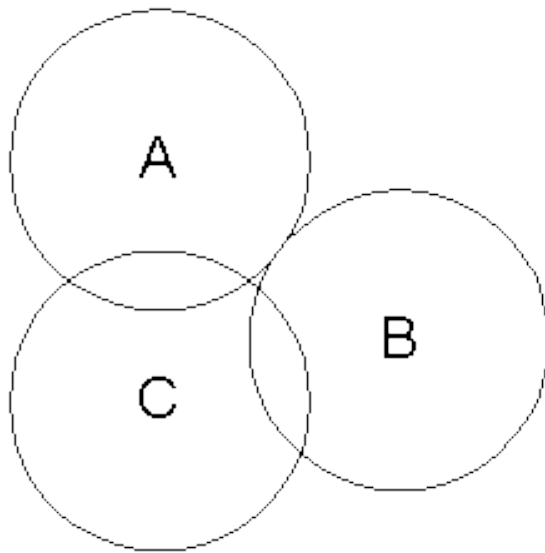
-- Fayad e Schmidt

- \* Podem atingir uma determinada funcionalidade por uma configuração
- \* Dita o fluxo da aplicação

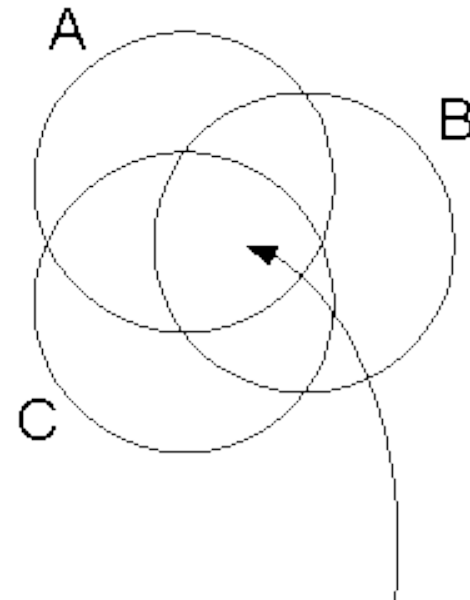
# Falando de Frameworks

- \* Definição

- \* Framework conceitual
- \* Framework de software



Impossível criar  
Framework

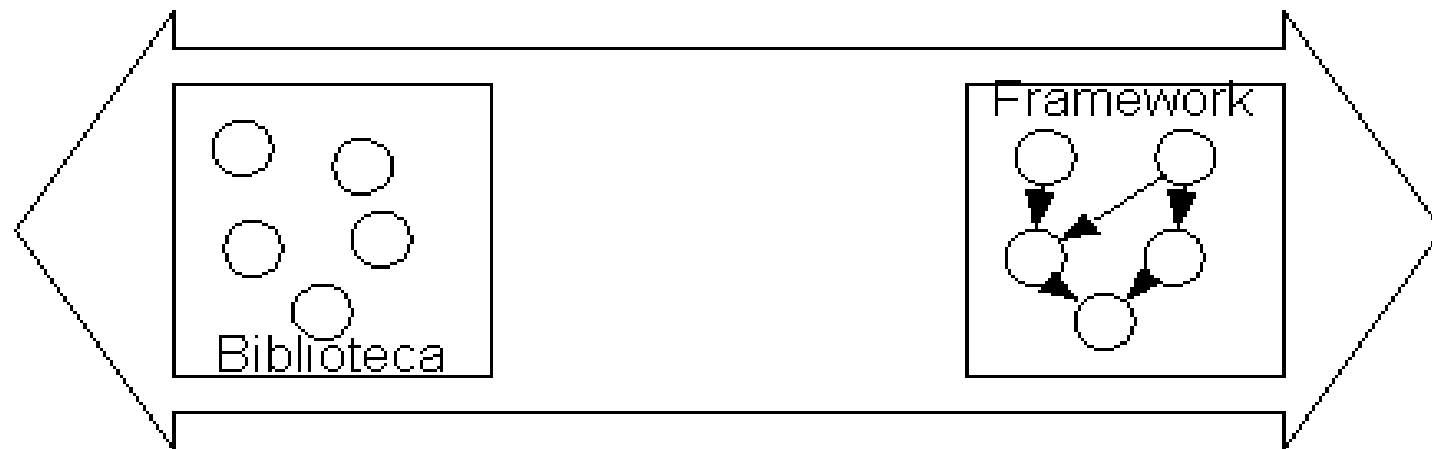


Interseção grande  
Possível criar Framework

# Falando de Frameworks

## \* Definição

### \* Framework versus Bibliotecas



- Classes instanciadas pelo cliente
- Cliente chama funções
- Não tem fluxo de controle predefinido
- Não tem interação predefinida
- Não tem comportamento default

- Customização com subclasse ou composição
- Chama funções da "aplicação"
- Controla o fluxo de execução
- Define interação entre objetos
- Provê comportamento default

# Falando de Frameworks

- \* Definição

- \* Framework *versus* Padrões de Projeto

- \* Design patterns são mais abstratos do que frameworks
    - \* Design patterns são elementos arquiteturais menores do que frameworks
    - \* Design patterns são menos especializados do que frameworks

# Falando de Frameworks

- \* Vantagens e Desvantagens

- \* Vantagens de criar um Framework

- \* Redução de custos
    - \* Redução de time-to-market
    - \* Maximização de re-uso (análise, design, código, testes)
    - \* Desenvolvedores se concentram em adicionar valor em vez de reinventar a roda
    - \* Menos manutenção
    - \* Estabilização melhor do código (menos defeitos) devido ao uso em várias aplicações
    - \* Fatoração de aspectos comuns a várias aplicações
    - \* Melhor consistência e compatibilidade entre aplicações
    - \* Alavancagem do conhecimento de especialistas

# Falando de Frameworks

- \* Vantagens e Desvantagens

- \* Desvantagens de criar um Framework

- \* Construir um framework é complexo
    - \* Re-uso não vem sozinho: deve ser planejado
    - \* É mais complexo e demora mais fazer uma aplicação tendo que construir um framework
    - \* Benefícios são realizados em longo prazo

# Falando de Frameworks

- \* Definição

- \* Características básicas de um Framework

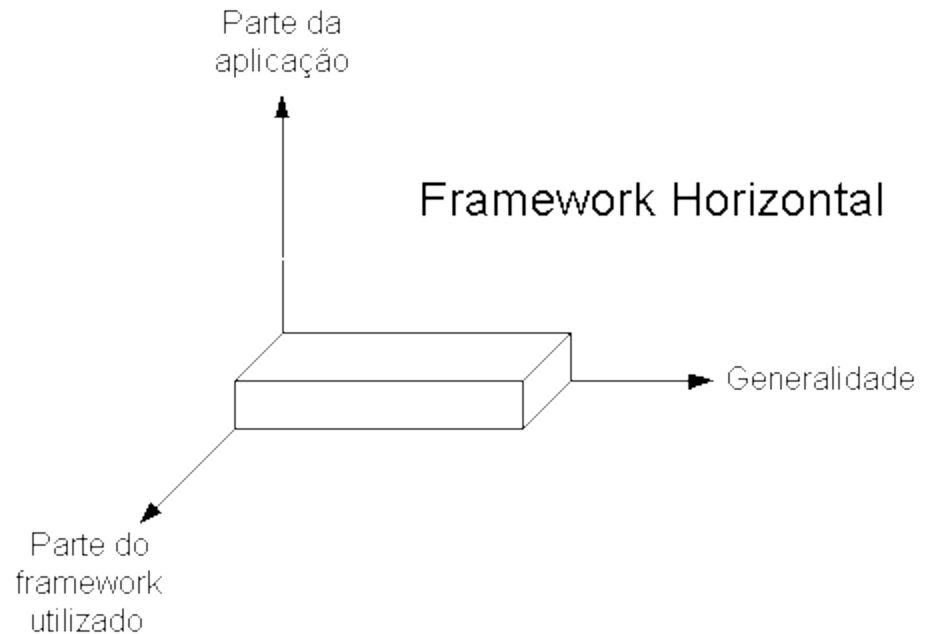
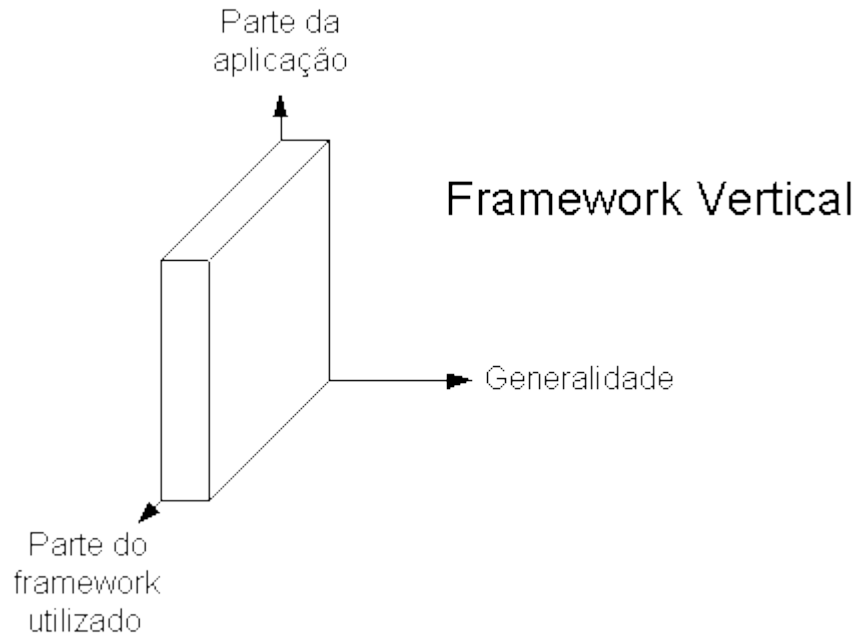
- \* Um framework deve ser reusável
    - \* Deve ser extensível
    - \* Deve ser de uso seguro
    - \* Deve ser eficiente
    - \* Deve ser completo



# Falando de Frameworks

- \* Definição

- \* Framework de suporte
- \* Frameworks vertical
- \* Frameworks horizontal



# Falando de Frameworks

- \* Definição
  - \* Partes de um Framework
    - \* *Frozenspots*
      - \* *Partes fixas*
    - \* *Hotspots*
      - \* *Partes extensíveis*

# Falando de Frameworks

- \* Seu uso

- \* Há 3 formas de uso para um Framework:
  - \* Inheritance-focused – Foco na herança
  - \* Composition-focused – Foco na composição
  - \* Híbridos

# Falando de Frameworks

- \* Seu uso
  - \* Inheritance-focused
    - \* Também chamado de white-box ou architecture-driven
    - \* Estende ou modifica funcionalidade pela definição de sub-classes com override de métodos

# Falando de Frameworks


- \* Seu uso
  - \* Composition-focused
    - \* Também chamado de black-box ou data-driven
    - \* Usa a funcionalidade já presente no framework
      - \* Os códigos internos do framework não podem ser vistos ou alteradas
      - \* Deve-se usar as interfaces fornecidas
  - \* As instanciações e composições feitas determinam as particularidades da aplicação

# Falando de Frameworks

- \* Seu uso
  - \* Híbridos
    - \* A maioria dos frameworks é inheritance-focused com alguma funcionalidade pronta (composition-focused)

# E um pouco sobre

## \* Comparativo entre frameworks PHP

PHP Framework	PHP4	PHP5	MVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules
<a href="#">CakePHP</a> 	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓
<a href="#">CodeIgniter</a> 	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	-
<a href="#">Symfony</a> 	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓
<a href="#">Yii</a> 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">Zend</a> 	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓

# E um pouco sobre



## \* Características

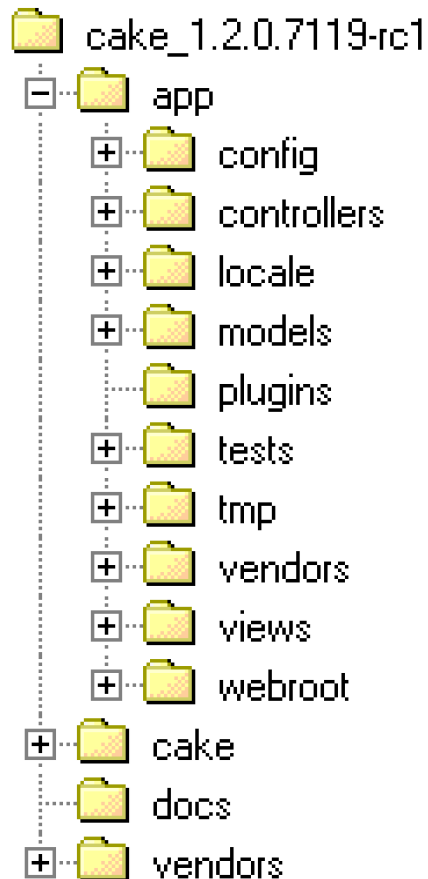
- \* Configuração zero (padronização sobre configuração)
- \* ✓ ORM transparente
- \* ✓ Simplicidade
- \* ✓ Melhores práticas
- \* ✓ Orientação a objetos real
- \* ✓ Geração de código
- \* ✓ Internacionalização
- \* ✗ Curva de aprendizado



# E um pouco sobre



## \* Estrutura de diretórios



# E um pouco sobre



## \* Padrões de nomenclatura

**DB**

*bolos*  
(no plural)

**Model**

*bolo.php*  
*Bolo*  
(no singular)

**Controller**

*bolos\_controller.php*  
*BolosController*  
(no plural)

**View**

*bolos/assar.ctp*  
*bolos/confeitar.ctp*  
(templates)

# E um pouco sobre



- \* Padrões de nomenclatura

**URLs:**

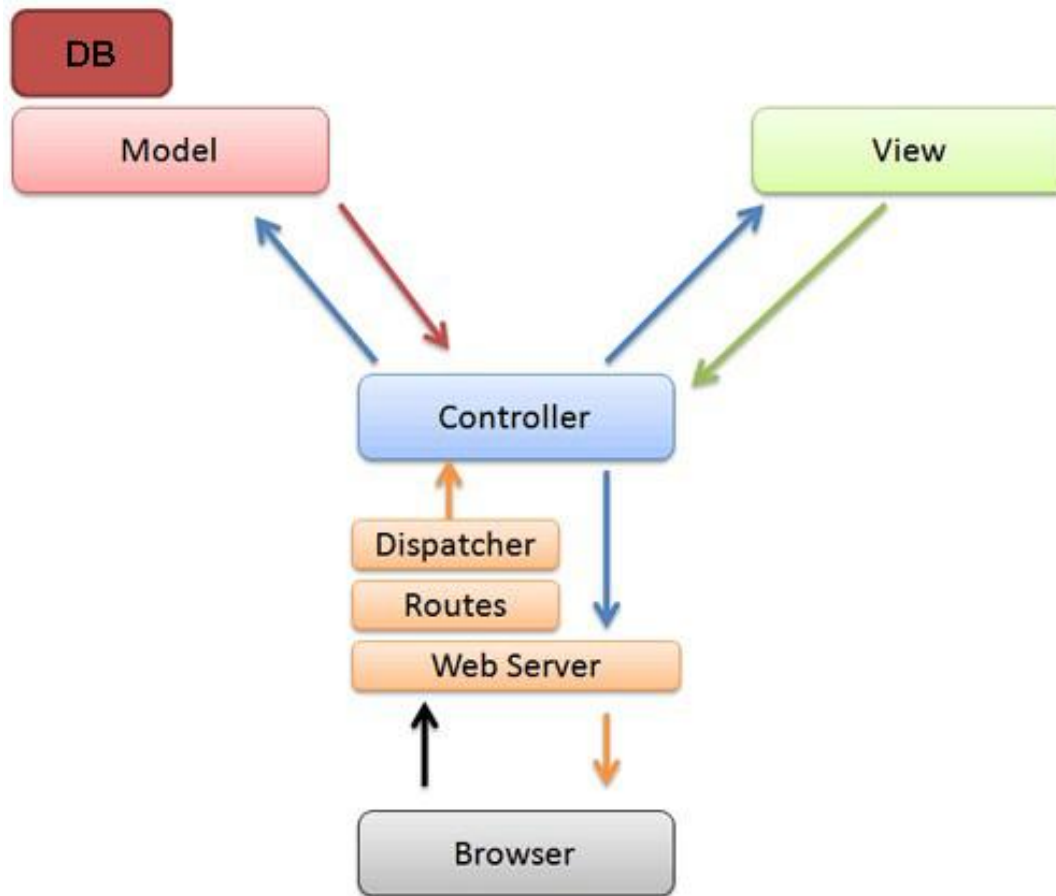
<domínio> / <controller> / <método> / <param1> ...

ex: *www.dominio.com/bolo/assar/1*

# E um pouco sobre



## \* Raiox do funcionamento



# E um pouco sobre



- \* Características

- \* Versatilidade
- \* Compatibilidade
- \* Facilidade de instalação
- \* Flexibilidade
- \* Leveza
- \* Documentação tutorializada
- \* Instanciamento dinâmico
- \* Junção de componentes
- \* Singularidade dos componentes
- \* Classes, bibliotecas e “helpers” nativos para diversas funcionalidades

# E um pouco sobre

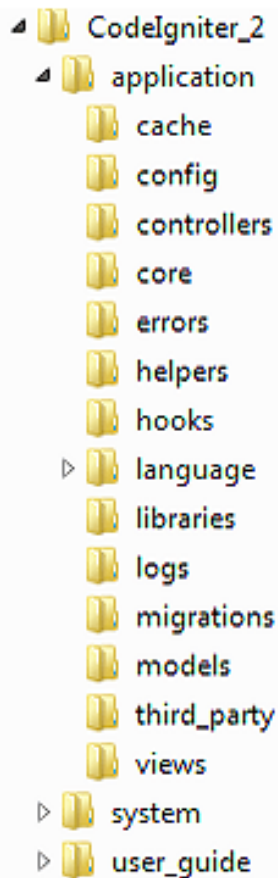


- \* Características
  - \* Algumas das funcionalidades nativas
    - \* Calendários
    - \* Bancos de dados
    - \* E-mails
    - \* Criptografia
    - \* Formulários
    - \* Manipulação de imagens
    - \* FTP
    - \* Compactação de arquivos
    - \* Controle de cookies e sessões PHP

# E um pouco sobre



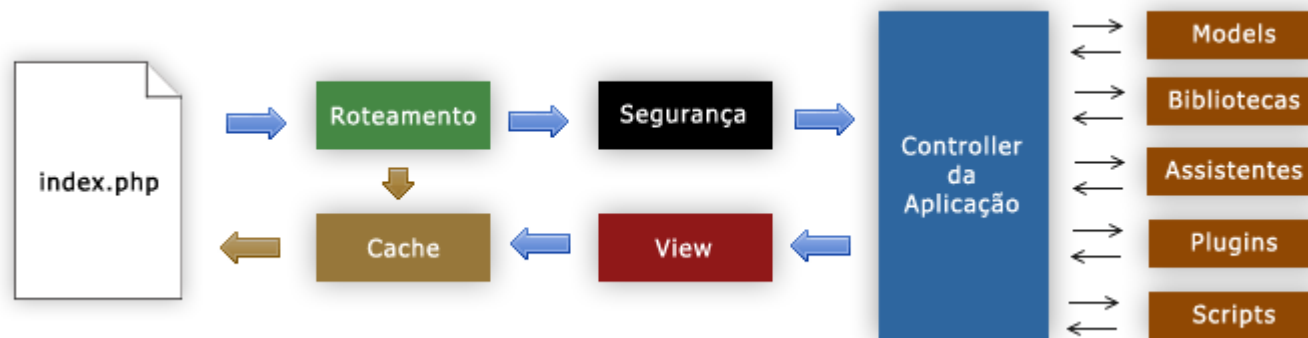
## \* Estrutura de diretórios



# E um pouco sobre



## \* Raiox do funcionamento





# E um pouco sobre

The Symfony logo, which consists of the word "symfony" in a white, lowercase, sans-serif font, set against a dark blue rectangular background.

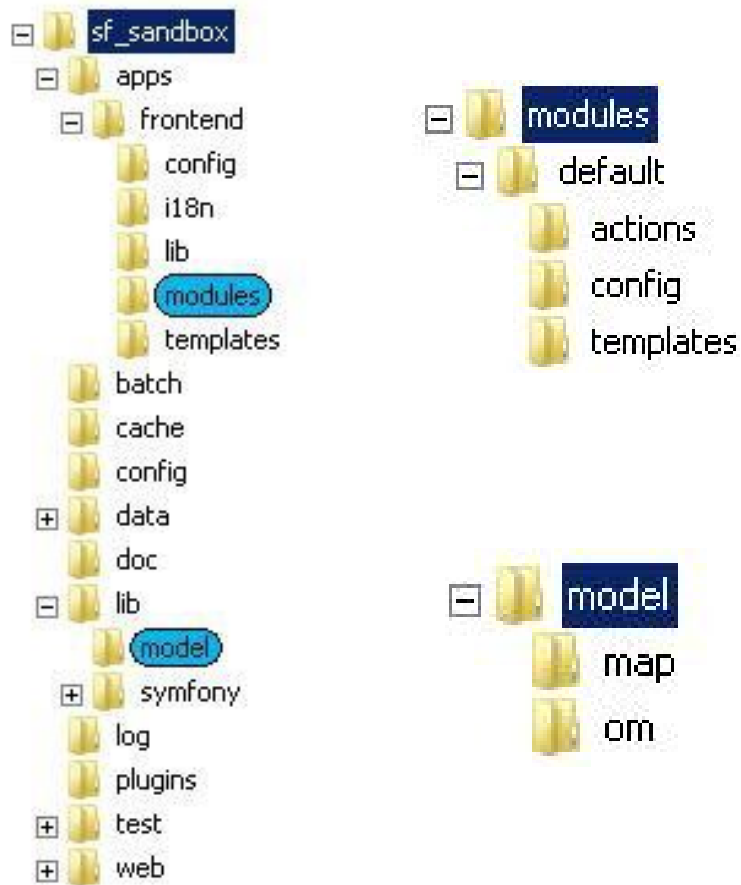
- \* Características

- \* Documentação
- \* Configurabilidade
- \* Proteção XSS
- \* Ferramentas de Depuração
- \* Testes Funcionais
- \* Extensibilidade
- \* Admin Generator
- \* ORM
- \* i18n / l10n

# E um pouco sobre

**symfony**

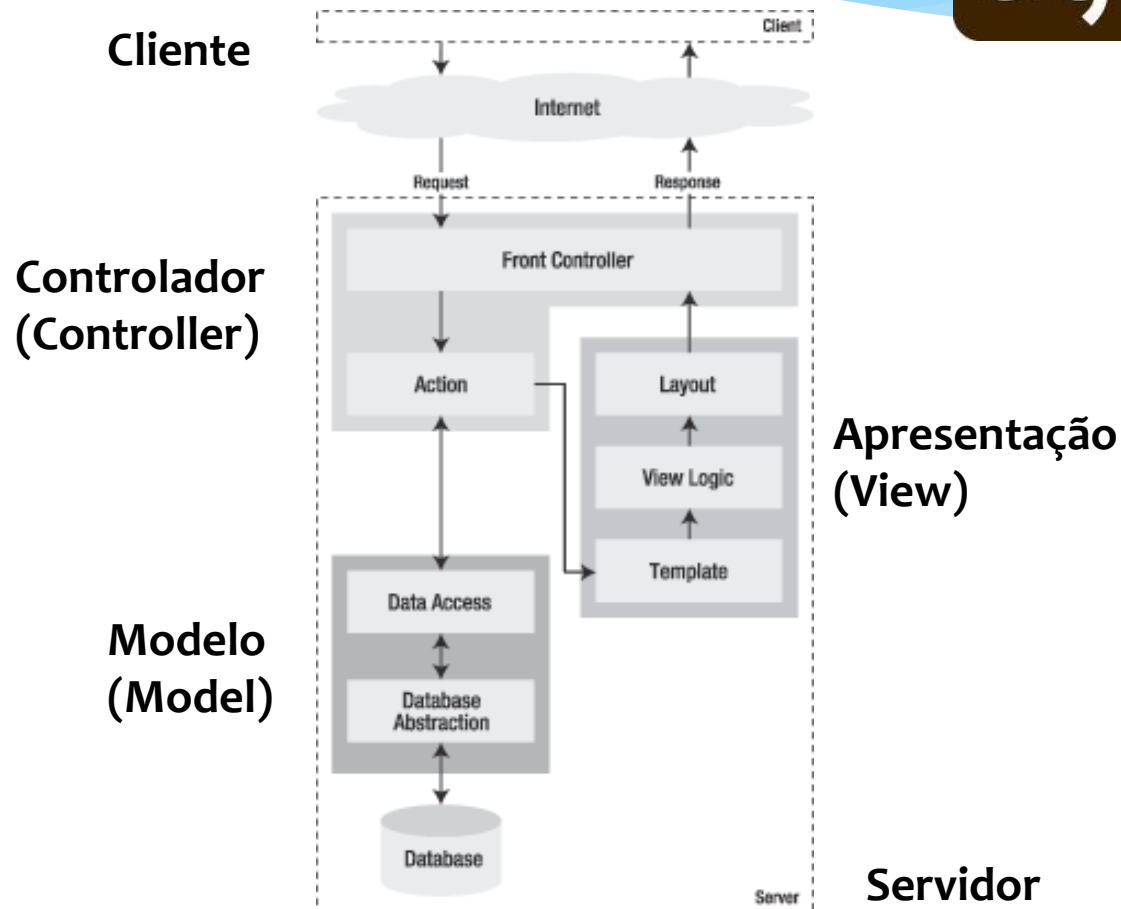
## \* Estrutura de diretórios



# E um pouco sobre

**symfony**

## \* Raios do funcionamento



**Servidor**

@EdgarSandi

# E um pouco sobre



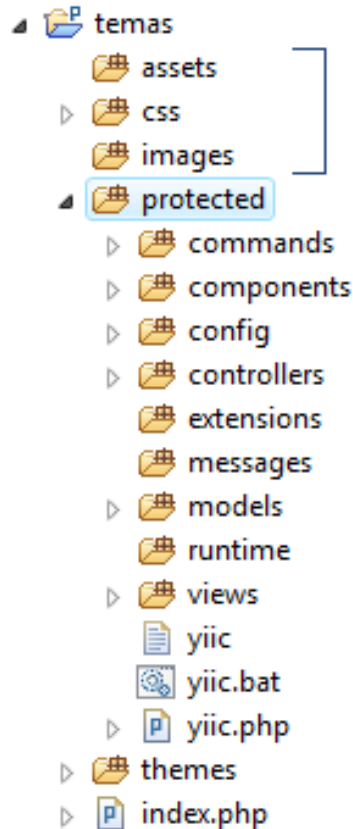
- \* Características

- \* Mapeamento do Banco
- \* Widgets integrados com JQuery
- \* Integrado com plugins do JQuery
- \* Internacionalização e localização
- \* Segurança
  - \* (prevenção cross-site scripting (XSS) ,
  - \* prevenção cross-site request forgery (CSRF) ,
  - \* prevenção de adulteração cookie, etc)
- \* Autenticação e Controle de Acesso
- \* Componentes e helpers voltados para web 2.0
- \* Temas e tradução

# E um pouco sobre



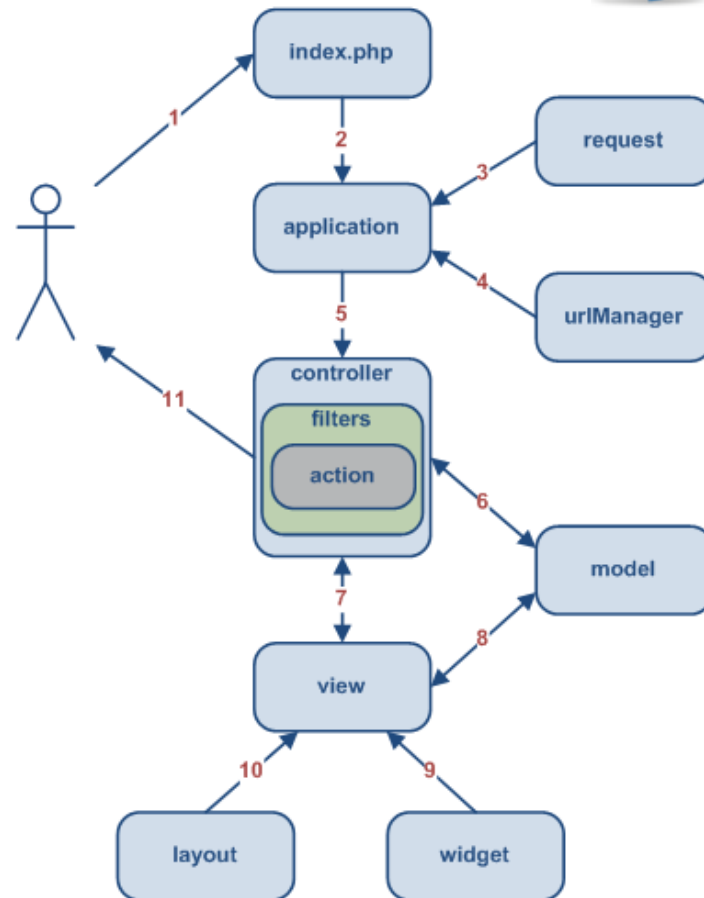
## \* Estrutura de diretórios



- 1 Arquivos como JavaScripts, CSS, imagens e outras pastas personalizáveis, por exemplo, crie uma pasta para os swfs.
- 2 Arquivos de configuração, bibliotecas, arquivos de internacionalização e visões. Visões para páginas estáticas também são mantidas aqui.
- 3 Arquivo de entrada. Porta principal da sua aplicação, responsável por instanciar sua aplicação em um objeto que reflete todas as configurações e componentes em atributos.

# E um pouco sobre

## \* Raiox do funcionamento



# E um pouco sobre



- \* Características

- \* Fácil integração com outros frameworks
- \* Fracamente acoplado
- \* PHP 5
- \* Implementa tendências da web atual (web services, ajax, etc)
- \* Segurança
- \* Modular
- \* Arquitetura *self-service*
- \* Suporte a múltiplos BD
- \* Internacionalização e Localização
- \* Gerenciamento de sessão
- \* Autenticação e autorização baseado em ACL

# E um pouco sobre



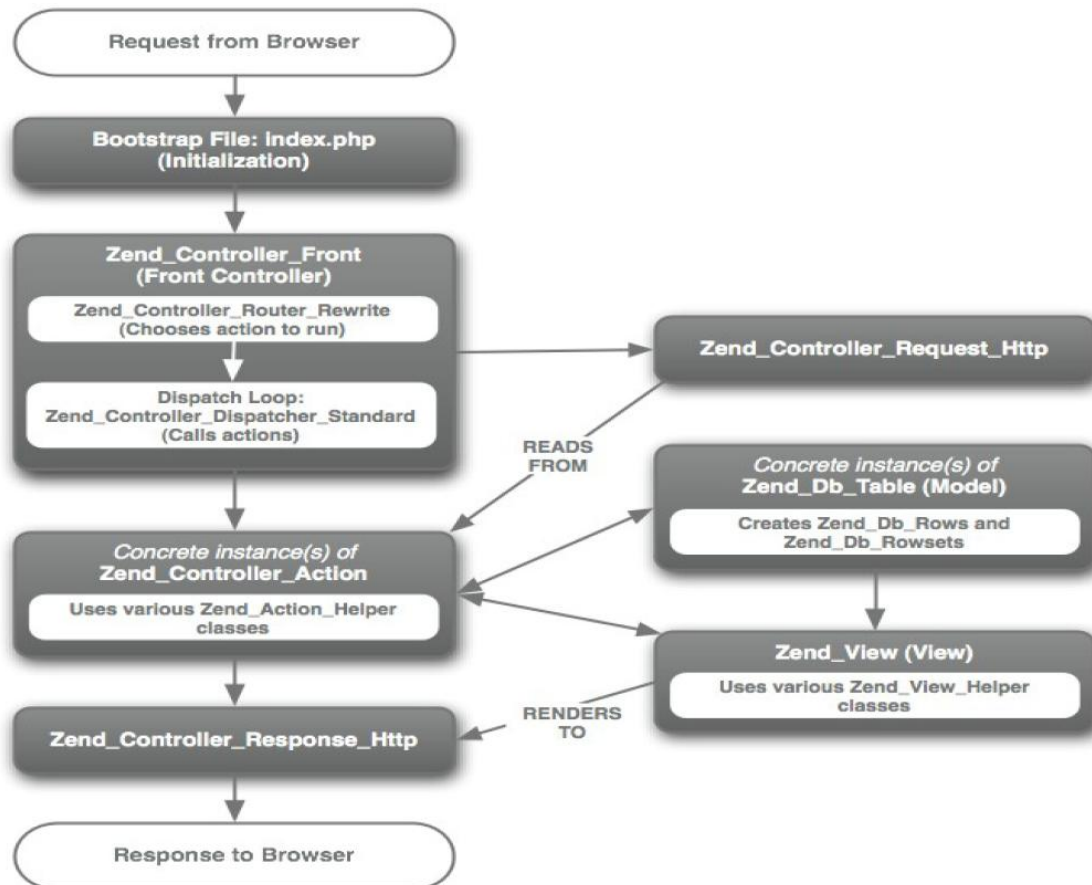
## \* Estrutura de diretórios





# E um pouco sobre

## \* Raiox do funcionamento



# Perguntas e respostas

