

# Curso de Iniciação ao Framework PHP Yii

Igor Rafael  
igor at dcc.ufmg.br

12 de agosto de 2011

# Conteúdo

Visão Geral

Configuração

Fundamentos

Estudo de caso

# Visão Geral

O que é um framework?

- Coleção de bibliotecas
- API - Interface de Programação de Aplicativos
- Funcionalidades genéricas customizáveis

# Visão Geral

O que é um framework?

Frameworks são bibliotecas? **Não!**

# Visão Geral

O que é um framework?

- Inversão de controle
- Comportamento padrão
- Extensibilidade
- Núcleo do código não pode ser modificado <sup>1</sup>

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework) 

# Visão Geral

Quando eu preciso de um framework?

- Aplicações semelhantes: sistemas de gestão, e-commerce, etc
- Aumentar o paralelismo do desenvolvimento o quanto antes
- Reduzir custos de manutenção: fatoração de funcionalidades

# Visão Geral

## Apresentando o framework Yii

- Framework PHP
- Alto-desempenho
- Baseado em componentes
- Aplicações Web: portais, forums, CMS, e-commerce, etc
- Desenvolvimento em larga escala

# Visão Geral

## Benefícios ao utilizar o Yii

- Separação entre interface de usuário, regras de negócio e modelo de dados: MVC
- Segurança: autenticação, autorização e prevenção contra ataques XSS, CSRF e cookies
- Cache: componentes para múltiplos níveis de cache
- Modularização: organização de componentes relacionados
- Geração automática de código: Gii
- Tratamento de erros: facilita associação de erros ao tratador correto
- Internacionalização: tradução de mensagens, formatos de data, hora e números
- e muito mais... Acredite, a lista é longa!

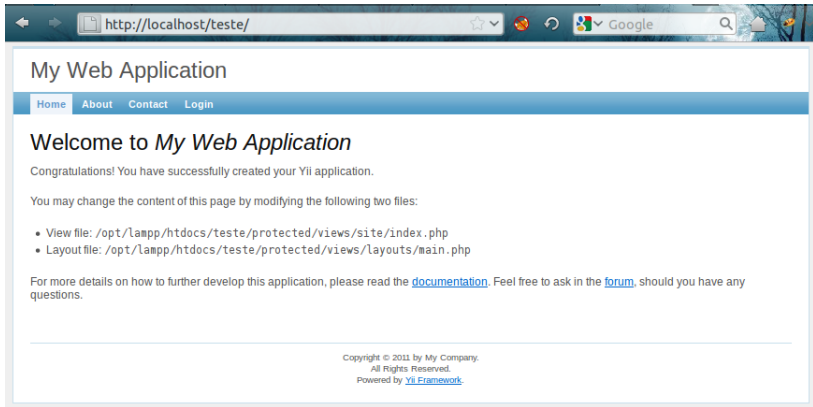
# Configuração do Ambiente Yii

- Pré-requisitos:
  - Yii Framework: versão 1.1.8<sup>2</sup>
  - XAMPP: versão 1.7.4
  - ou WAMPP: versão 2.1e
- Passos:
  - Descompactar o Yii no diretório do apache (htdocs ou www)
  - Checar os requisitos: <http://localhost/yii-1.1.8/requirements/>
  - Criar aplicação de teste:
    - Linux: `yii/framework/yiic webapp teste`
    - Windows: `..\bin\php\php5.3.5\php.exe yii\framework\yiic.php webapp teste`

---

<sup>2</sup><http://www.yiiframework.com/>

# Configuração do Ambiente Yii



The screenshot shows a web browser window with the address bar containing `http://localhost/teste/`. The page title is "My Web Application". The navigation menu includes "Home", "About", "Contact", and "Login". The main content area displays a welcome message and instructions for modifying the application files.

My Web Application

Home About Contact Login

## Welcome to *My Web Application*

Congratulations! You have successfully created your Yii application.

You may change the content of this page by modifying the following two files:

- View file: `/opt/lampp/htdocs/teste/protected/views/site/index.php`
- Layout file: `/opt/lampp/htdocs/teste/protected/views/layouts/main.php`

For more details on how to further develop this application, please read the [documentation](#). Feel free to ask in the [forum](#), should you have any questions.

---

Copyright © 2011 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

# Fundamentos

## Como o Yii funciona

- assets: recursos publicados
- css
- images
- themes
- protected: arquivos protegidos da aplicação
  - components: componentes adicionais
  - config: configuração da aplicação
  - controllers: regras de negócio
  - data: sql de criação/configuração do banco de dados
  - extensions: extensões para o framework
  - messages: tradução de mensagens para UI
  - models: modelo de dados
  - runtime: arquivos temporários
  - tests: scripts de teste da aplicação
  - views: interface gráfica da aplicação

# Fundamentos

## Como o Yii funciona

- Conexão com o banco de dados
  - Criar database “teste” - phpMyAdmin
  - Executar sql `protected/data/schema.mysql.sql`
  - Configurar conexão `protected/config/main.php` - application components

# Fundamentos

## Como o Yii funciona

- Gerando código com o Gii
  - Habilitar o Gii: `protected/config/main.php` - modules
  - Definir uma senha - Não pode ser vazia
  - Acessar `http://localhost/teste/index.php?r=gii`
  - Model Generator: `tbl_user`
  - CRUD Generator: `User`
- `http://localhost/teste/index.php?r=user`

# Fundamentos

## Como o Yii funciona

- Implementa o design pattern<sup>3</sup> MVC - Model View Controller<sup>4</sup>
  - Amplamente utilizado em frameworks Web
  - Separa lógica de negócio da interface de usuário
  - Model (modelo): representa a informação e regras de negócio
  - View (visão): contem os elementos de UI como texto e campos de formulário
  - Controller (controlador): gerencia comunicação entre Model e View, cuida de validação e filtragem de dados

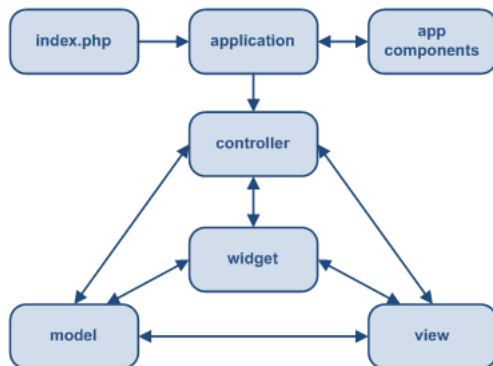
---

<sup>3</sup>[http://en.wikipedia.org/wiki/Design\\_Patterns](http://en.wikipedia.org/wiki/Design_Patterns)

<sup>4</sup><http://en.wikipedia.org/wiki/Model-view-controller>

# Fundamentos

## Como o Yii funciona



# Fundamentos

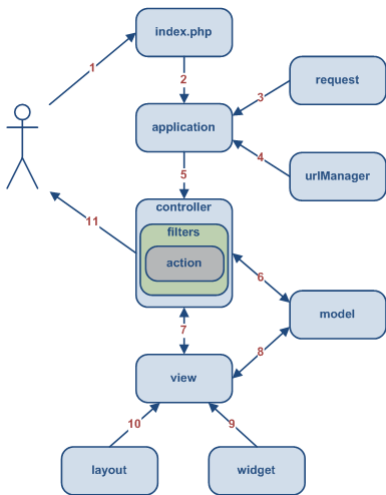
## Como o Yii funciona

Acesso a um blog desenvolvido em Yii

<http://www.example.com/index.php?r=post/showid=1>

# Fundamentos

## Como o Yii funciona



1. requisição à URL aciona `index.php`.
2. criação da aplicação
3. informações da requisição
4. controlador (`PostController`) e ação
5. instancia e aciona ação (`actionShow`)
6. carrega post (`ID=1`) em modelo (`Post`)
7. ação carrega view (`show`) com o modelo
8. view exibe atributos do modelo
9. view carrega widgets necessários
10. view é inserida em um layout
11. ação envia renderização ao usuário

# Fundamentos

## MVC

1. Controladores - CController
2. Modelos - CModel
3. Visões - código html + php

# Estudo de caso: Aplicativo de gestão de horas

## Casos de Uso <sup>5</sup>

- Cadastrar colaboradores da empresa
- Cadastrar projetos, cujo responsável deve ser um colaborador cadastrado
- Cadastrar atividades gerais, que são usualmente utilizadas nos projetos
- Apropriar esforço realizado em uma atividade por um colaborador alocado em um projeto
- Alocar um colaborador à um projeto durante um período de tempo com uma dedicação diária

---

<sup>5</sup>[http://en.wikipedia.org/wiki/Use\\_case](http://en.wikipedia.org/wiki/Use_case)

# Estudo de caso: Aplicativo de gestão de horas

## Histórias de Usuário <sup>6</sup>

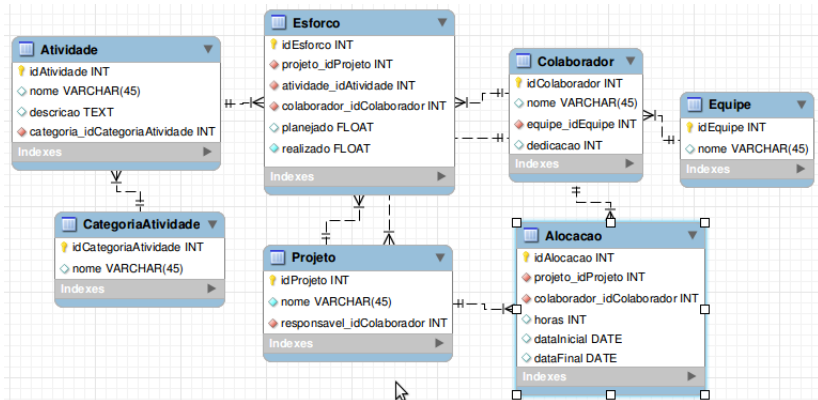
- Como gerente, preciso gerenciar os colaboradores da minha empresa para que eu tenha os registros de todos os funcionários que já passaram pela mesma
- Como gerente, preciso administrar os projetos que realizo para que a diretoria acompanhe o andamento do trabalho
- Como gerente, preciso cadastrar atividades gerais para que possamos acompanhar o esforço investido na mesma em diversos projetos
- Como colaborador, preciso apropriar as horas que trabalhei para facilitar o acompanhamento das minhas atividades
- Como gerente, preciso alocar um colaborador à um projeto durante um período de tempo com uma dedicação diária para dimensionar o tamanho da minha equipe ao longo do projeto

---

<sup>6</sup>[http://en.wikipedia.org/wiki/User\\_story](http://en.wikipedia.org/wiki/User_story)

# Estudo de caso: Aplicativo de gestão de horas

## Modelo de Dados



# Estudo de caso: Aplicativo de gestão de horas

## Configuração

- Importação do banco de dados<sup>7</sup>: phpMyAdmin
- Criação da aplicação: yiic
- Alterar nome da aplicação: config/main.php
- Configuração do banco de dados: config/main.php
- Habilitação do Gii: config/main.php
- Gerar modelos e CRUDs: Gii
- Alterar menu: views/layouts/main.php
- Proteger acesso via login
  - menu: propriedade 'visible'
  - controladores: accessRules

---

<sup>7</sup>[http://homepages.dcc.ufmg.br/~igor/classes/yiiframework/gestao\\_horas.sql](http://homepages.dcc.ufmg.br/~igor/classes/yiiframework/gestao_horas.sql)

# Estudo de caso: Aplicativo de gestão de horas

Alternativa para a proteção de acesso via login

- Alterar o template utilizado pelo Gii
  - Editar o arquivo  
`yii/framework/gii/generators/crud/templates/default/controller.php`
- Outras possibilidades:
  - Extender o comportamento do gerador de código
  - Copiar o template atual para `protected/gii/templates/safe/` e fazer as alterações

# Estudo de caso: Aplicativo de gestão de horas

## Apresentação visual

- Corrigir rótulos: `attributeLabels` nos modelos
- Corrigir textos
  - O Yii gera texto padrão para os formulários (e.g. Create Colaboradors)
  - Tal texto pode ser alterado nas páginas (`admin.php`, `_form.php`, `create.php`, etc) de cada modelo:  
`views/<nomeDoModelo>/<nomeDaVisão>.php`

# Estudo de caso: Aplicativo de gestão de horas

## Utilizando dropDownList em chaves estrangeiras

- Editar `_form.php`: `views/projeto/_form.php`
- Encontrar campo que será alterado:  
`responsavel_idColaborador`
- Substituir `$form->textField(...)` para  
`$form->dropdownlist($model,'responsavel_idColaborador',  
CHtml::listData(Colaborador::model()->findAll(), 'id',  
'nome'))`;
- Alternativa
  - Criar método para montar opções: `getColaboradoresList` no modelo `Projeto`
  - Editar `_form.php` para utilizar o método criado

# Estudo de caso: Aplicativo de gestão de horas

Utilizando janelas modais para formulários<sup>8</sup>

- Incluir no `_form.php` o link para a janela
- Instanciar componente `CJuiDialog`: `_form.php`
- Criar JS para tratar click no link: `_form.php`
- Alterar controlador para tratar chamada Ajax:  
`NomeDoControladorController.php`
- Exercício para próxima aula: Atualizar o `drowpDownList` ao fechar a janela

---

<sup>8</sup><http://www.yiiframework.com/wiki/145/cjuidialog-for-create-new-model>

# Estudo de caso: Aplicativo de gestão de horas

## Internacionalização

- A linguagem padrão do Yii é o Inglês
- Para mudar para o português devemos
  - Editar as configurações da aplicação: `config/main.php`
  - Incluir atributo `'language'=>'pt'` no primeiro nível do arranjo (e.g. logo após a propriedade `name`)
- Nem todas as mensagens serão traduzidas, verifique se as mesmas estão sendo exibidas através da função **Yii::t** (e.g. `<h1><?=Yii::t('app','Contact Us')?></h1>` ao invés de `<h1>Contact Us</h1>` na view `site/contact.php`)
  - Crie a pasta **pt** dentro de `protected/messages`
  - Crie nessa pasta o arquivo `app.php`

# Estudo de caso: Aplicativo de gestão de horas

## Internacionalização

- Conteúdo inicial do arquivo `protected/messages/pt/app.php`

```
<?php  
return array(  
    'Contact Us' => 'Entre em contato'  
);  
?>
```

# Estudo de caso: Aplicativo de gestão de horas

## Atividades de fixação

- Utilizar dropDownList em todas as chaves estrangeiras de todos os modelos
- Traduzir as visões para português
- Utilizar o CJuiDialog para o campo Equipe no modelo Colaborador (semelhante ao que foi feito com CategoriaAtividade e Atividade)
- Esconder chaves primárias de todos os modelos
  - Alterar as tabelas no banco de dados para que as chaves primárias sejam 'auto increment'
  - Retirar o campo ID dos formulários

# Estudo de caso: Aplicativo de gestão de horas

## Melhorando visualização de relacionamentos

- as visões `index/view/admin` por enquanto exibem IDs de chaves estrangeiras
- para substituir pelo item correspondente:
  - alterar nome do relacionamento: `models` (e.g. `Alocacao.php`: mudar **`projetoIdProjeto`** para **`projeto`**)
  - incluir atributo nos labels (e.g. `Alocacao.php`: **`'projeto.nome'=>'Projeto'`**)
  - alterar atributo utilizado nas visões (e.g. `view/alocacao/admin.php`: **`projeto_idProjeto`** para **`projeto.nome`** no `CGridView`)

# Estudo de caso: Aplicativo de gestão de horas

## Melhorando visualização de relacionamentos

- adaptando o CGridView:
  - incluir atributo no modelo (e.g. Alocao.php: **public \$projetoNome;**)
  - alterar regra da busca (método rules) para incluir o atributo (e.g. trocar projeto\_IdProjeto para **projetoNome**)
  - alterar método search para incluir relacionamento

```
criteria -> with = array('projeto');
```

```
criteria -> together = true; .
```

```
.
```

```
.
```

```
criteria -> compare('projeto.nome',this->projetoNome, true);
```

# Estudo de caso: Aplicativo de gestão de horas

## Melhorando visualização de relacionamentos

- adaptando o CGridView:
  - alterar parâmetros do widget CGridView

```
// 'projeto.nome',  
array(  
    'name' => 'projetoNome',  
    'value' => 'data - > projeto - > nome',  
    'type' => 'text',  
    'header' => 'Projeto',  
)
```

# Estudo de caso: Aplicativo de gestão de horas

## Formatação de datas

- admin.php: parâmetros do CGridView

```
//'dataInicial',  
array(  
    'name'=>'dataInicial',  
    'value'=>'Yii::app()->dateFormatter->  
        formatDateTime($data->dataInicial,  
            'medium',null)',  
)
```

- \_view.php:

```
<b><?php echo CHtml::encode($data->getAttributeLabel  
    ('dataInicial')); ?></b>  
<?php echo CHtml::encode(Yii::app()->dateFormatter->  
    formatDateTime($data->dataInicial,'medium',null));  
?><br />
```

# Estudo de caso: Aplicativo de gestão de horas

## Formatação de datas

- view.php: parâmetros do CDetailView

```
array(  
    'name'=>'dataInicial',  
    'value'=>Yii::app()->dateFormatter->  
        formatDate($model->dataInicial,  
            'medium', null),  
),
```

# Estudo de caso: Aplicativo de gestão de horas

## Formatação de datas

- `_form.php`: utilizar o `CJuiDatePicker`

```
<div class="row">
<?php echo $form->labelEx($model,'dataInicial'); ?>
<?php $this->widget('zii.widgets.jui.CJuiDatePicker',
    array('name'=>'dataInicial',
        'model'=>$model,
        'attribute'=>'dataInicial',
        'value'=>$model->dataInicial,
        'options'=>array('dateFormat'=>'dd/mm/yy'),
    ));?>
<?php echo $form->error($model,'dataInicial'); ? >
</div>
```

# Estudo de caso: Aplicativo de gestão de horas

## Formatação de datas

- Alternativa utilizando behaviors: beforeSave

```
public function beforeSave() {  
    if ($this->dataInicial == '')  
        $this->setAttribute('dataInicial', null);  
    else  
        $this->dataInicial=implode('-',array_reverse(  
            explode('/', $this->dataInicial)));  
    if ($this->dataFinal == '')  
        $this->setAttribute('dataFinal', null);  
    else  
        $this->dataFinal=implode('-',array_reverse(  
            explode('/', $this->dataFinal)));  
    return parent::beforeSave();  
}
```

# Estudo de caso: Aplicativo de gestão de horas

## Formatação de datas

- Alternativa utilizando behaviors: afterFind

```
public function afterFind() {  
    if(!is_null($this->dataInicial))  
        $this->dataInicial=Yii::app()->dateFormatter->  
            ($this->dataInicial,'medium',null);  
    if(!is_null($this->dataFinal))  
        $this->dataFinal=Yii::app()->dateFormatter->  
            ($this->dataFinal,'medium',null);  
    return parent::afterFind();  
}
```

# Estudo de caso: Aplicativo de gestão de horas

## Passos no desenvolvimento de uma aplicação Yii

- Criar esqueleto da aplicação: yiic
- Configurar a aplicação: config/main.php
- Criar um modelo para cada entidade do banco de dados
- Criar um controlador para cada tipo de requisição do usuário
- Implementar ações e suas respectivas visões
- Configurar filtros para as ações dos controladores
- Configurar internacionalização e mapear mensagens
- Preparar a aplicação para o ambiente de produção:  
<http://www.yiiframework.com/doc/guide/1.1/en/topics.performance>

# Referências I



Yii Framework Guide

*<http://www.yiiframework.com/doc/guide/>*



CJuiDialog example

*<http://www.yiiframework.com/wiki/145/cjuidialog-for-create-new-model>*