

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Alguns termos úteis

# Alguns termos úteis

## ▶ Sistema Modelo

- ▶ O nome do sistema que estamos usando para desenvolver os vídeos e demais materiais relacionados ao curso.
- ▶ Espera que, tudo que seja feito para construir o Sistema Modelo, você possa replicar no seu contexto e desenvolver o seu próprio sistema.

## ▶ Projeto

- ▶ É a pasta do seu sistema dentro do ambiente de desenvolvimento.
- ▶ Exemplo: Dentro do ambiente Eclipse, criamos um projeto PHP chamado sistemamodelo.

## ▶ CRUD é um acrônimo para ‘create, read, update and delete’, que são operações comuns em sistemas com gerenciamento de informações. Significam: criar, ler, atualizar e excluir.

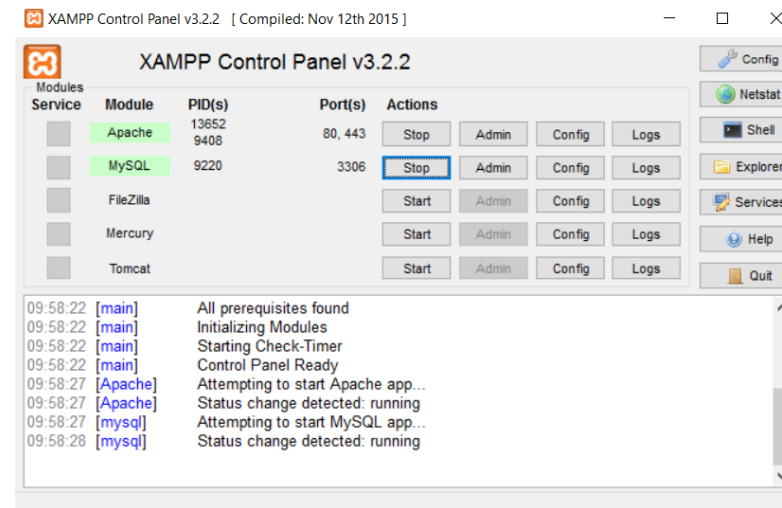
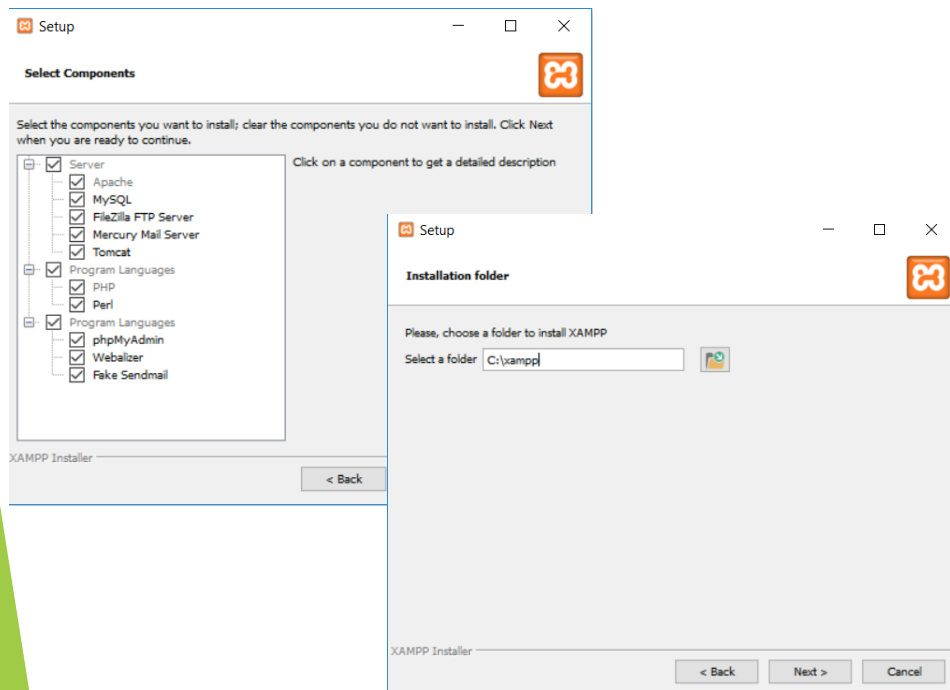
## ▶ IDE

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Preparação do Ambiente de Trabalho

# Preparação do Ambiente de Trabalho

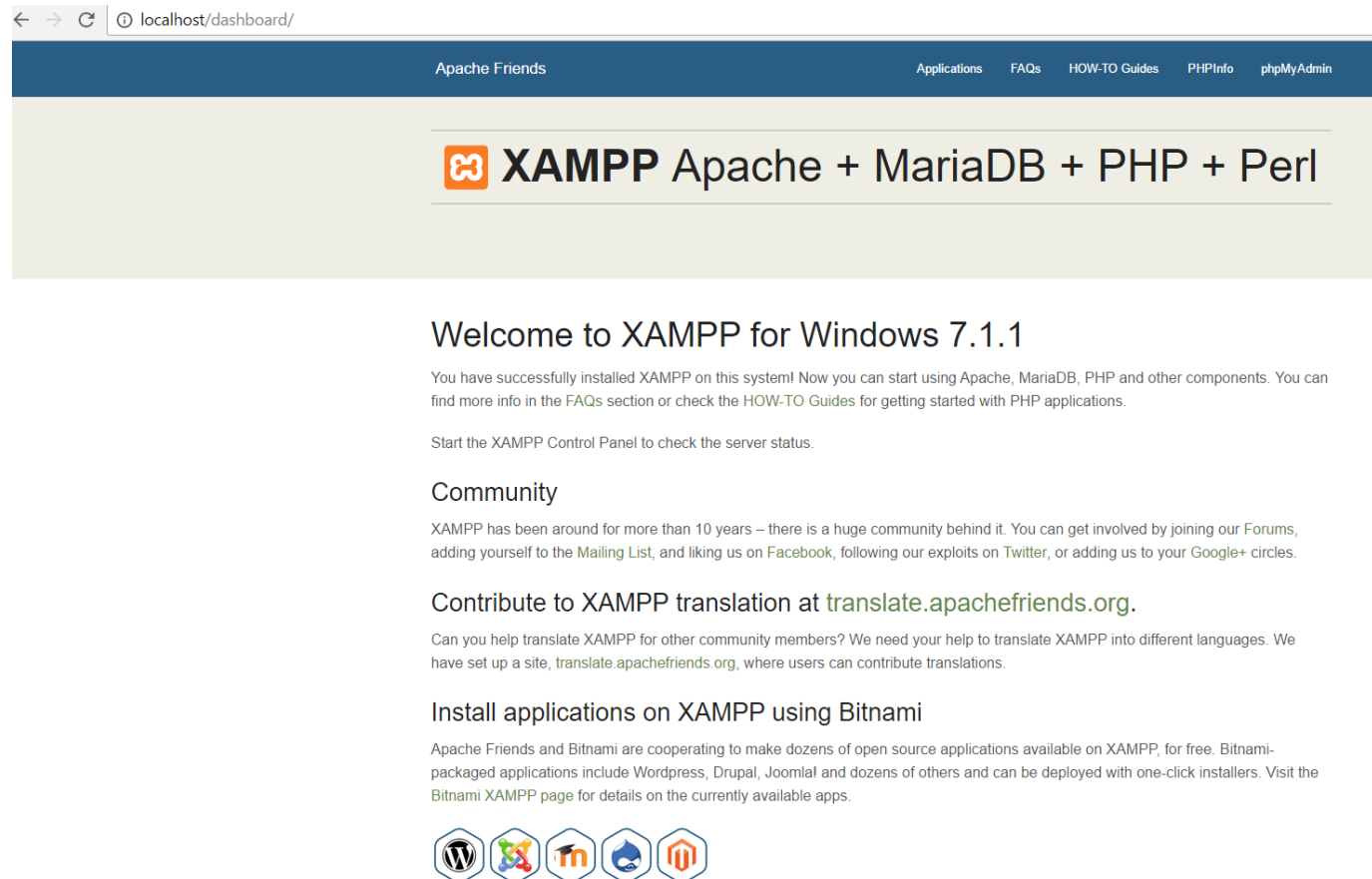
- ▶ Servidor web local, suporte à linguagem PHP, acesso ao banco de dados.
- ▶ Agregadores: Xampp? Wamp? Outros?
- ▶ Xampp
  - ▶ Download: [https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html)
  - ▶ Instalação Padrão:



Sempre verifique  
se os serviços  
estão ativos!

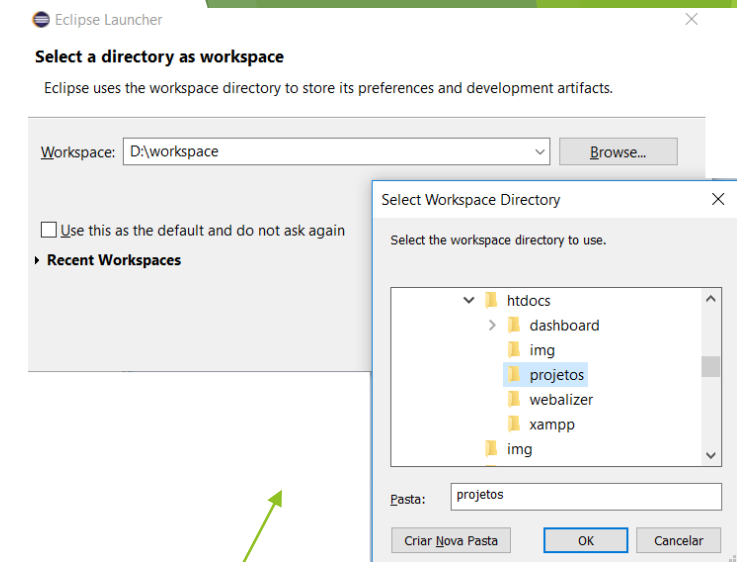
# Preparação do Ambiente de Trabalho

- ▶ Para testar se o servidor web local está funcionando:
- ▶ Navegador: localhost



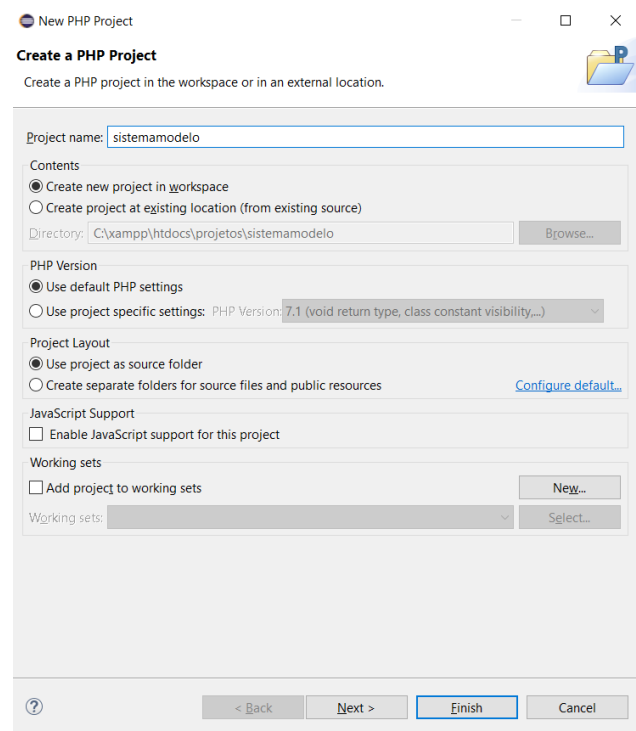
# Preparação do Ambiente de Trabalho

- ▶ Qual ferramenta (IDE, Editor, etc) utilizar? Front? Back?
  - ▶ Netbeans? Eclipse? Aptana?
  - ▶ Bloco de notas? Wordpad? Sublime?
- ▶ Eclipse for PHP Developers
  - ▶ <http://www.eclipse.org/downloads/packages/eclipse-php-developers/neon3>
  - ▶ Descompactar o arquivo, renomear para eclipse e colocar no diretório C: (por exemplo)
  - ▶ Executar o eclipse.exe
  - ▶ Erro? Pode ser que esteja faltando o JDK (Java Development Kit )
  - ▶ Setar workspace (ex: crie uma pasta “projetos” dentro de htdocs do Xampp)
- ▶ Download do JDK
  - ▶ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
  - ▶ Execute o arquivo, assim que ele finalizar.
  - ▶ C:\Program Files\Java\jre1.8.0\_121 (pasta default)



# Preparação do Ambiente de Trabalho

- ▶ Dentro do Eclipse
  - ▶ Crie um projeto PHP novo (File - New PHP Project)
  - ▶ Dê um nome para o seu projeto.
    - ▶ No nosso exemplo, vamos utilizar sistemamodelo (letras minúsculas).
  - ▶ Habilite JavaScript Suport.



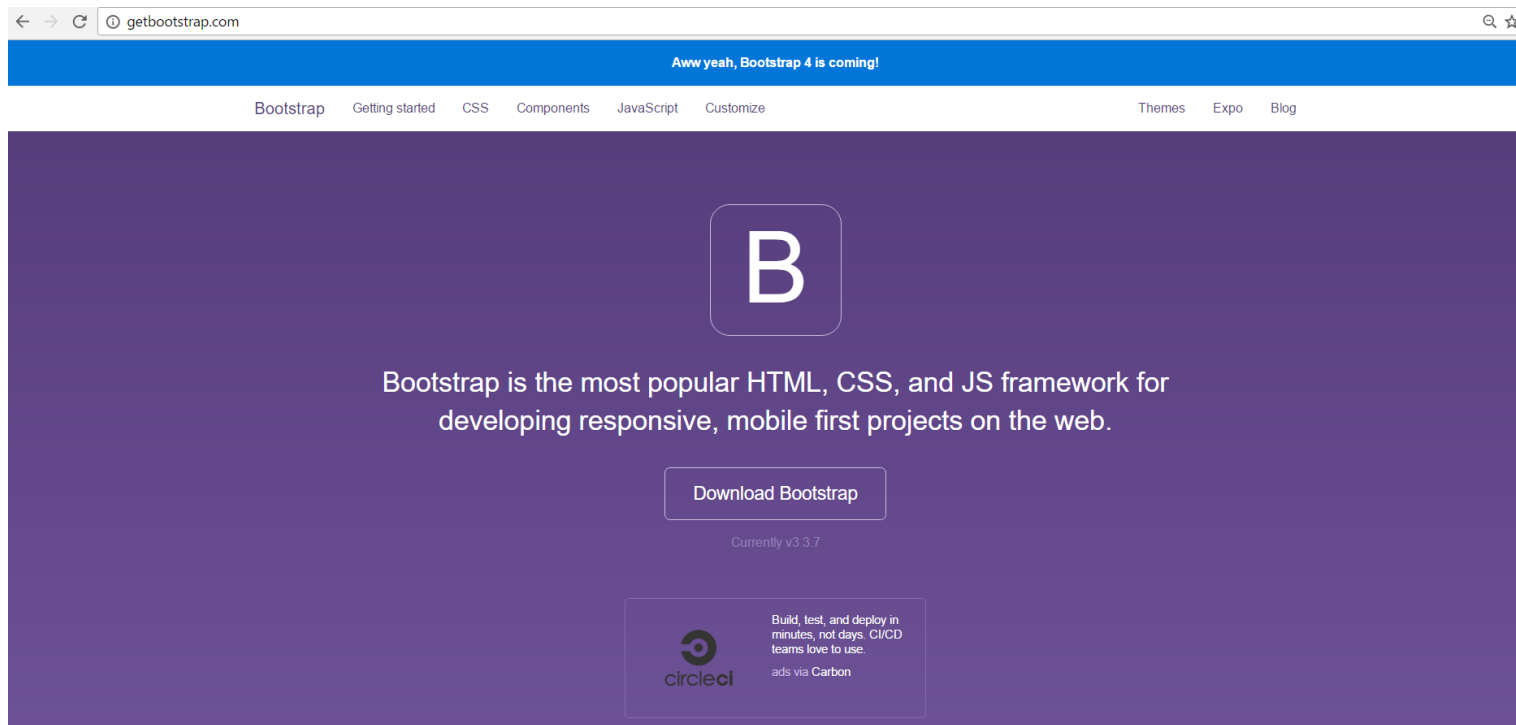
# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Download e Configuração Inicial do Bootstrap



# Download e Configuração Inicial do Bootstrap

- ▶ Para fazer o download do Bootstrap, basta acessar o site <http://getbootstrap.com> e clicar no botão “Download Bootstrap”.
- ▶ Versão em português: <http://getbootstrap.com.br>
  - ▶ No momento da elaboração deste vídeo, o Bootstrap estava na versão 3.3.7.



# Download e Configuração Inicial do Bootstrap

- Existem 3 opções de download:



getbootstrap.com/getting-started/#download

## Download

Bootstrap (currently v3.3.7) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

### Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

Download Bootstrap

### Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

Download source

### Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

Download Sass

## Bootstrap CDN

The folks over at [MaxCDN](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN](#) links.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css"
integrity="sha384-rhYon11rSVX4nD0JutInGAs1CJuc7uWjduW9SVrLVRyOoPp2bWYgmgJQIXw1/Sp" crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc51Qib027qyqjSMFHJ0MakfufuVx2XUPnCA712mCWNlIp69mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>
```

Copy

## Install with Bower

You can also install and manage Bootstrap's Less, CSS, JavaScript, and fonts using [Bower](#).

- “Download Bootstrap”
  - contém somente arquivos necessários para o início do desenvolvimento do projeto (CSS, JS e FONT).
- “Download source”
  - contém todos os arquivos para o desenvolvimento do projeto e documentação do bootstrap.
- “Download SASS”
  - contém os arquivos SASS e LESS para desenvolvedores que utilizam SASS ou LESS ao invés de utilizar CSS diretamente. SASS e LESS são linguagens que adicionam funcionalidades ao CSS.

# Download e Configuração Inicial do Bootstrap

- ▶ Neste vídeo, nós usamos a opção “Download Source”.
- ▶ Descompacte o arquivo “bootstrap-3.3.7.zip” (Escolha a opção extrair aqui).
  - ▶ Por enquanto, vamos usar apenas a pasta “dist”.
  - ▶ Ela contém “css, js e font”. E pode conter imagens e outros arquivos de apoio.
- ▶ Copie a pasta “dist” e arraste para dentro do seu projeto (no Eclipse).
- ▶ Adicionalmente, coloque na pasta dist/css o arquivo “main.css”.
  - ▶ Este arquivo está disponível na plataforma do curso.
  - ▶ Ele contém estilos de formatações adicionais, criados por nós e não pelo Bootstrap.

# Download e Configuração Inicial do Bootstrap

- ▶ Criar uma pasta “img” dentro de “dist”.
- ▶ Colar as figuras que serão usadas no seu projeto.
- ▶ Por enquanto, faça o download destes arquivos disponíveis na plataforma do curso.

- ▶ Avatar.png

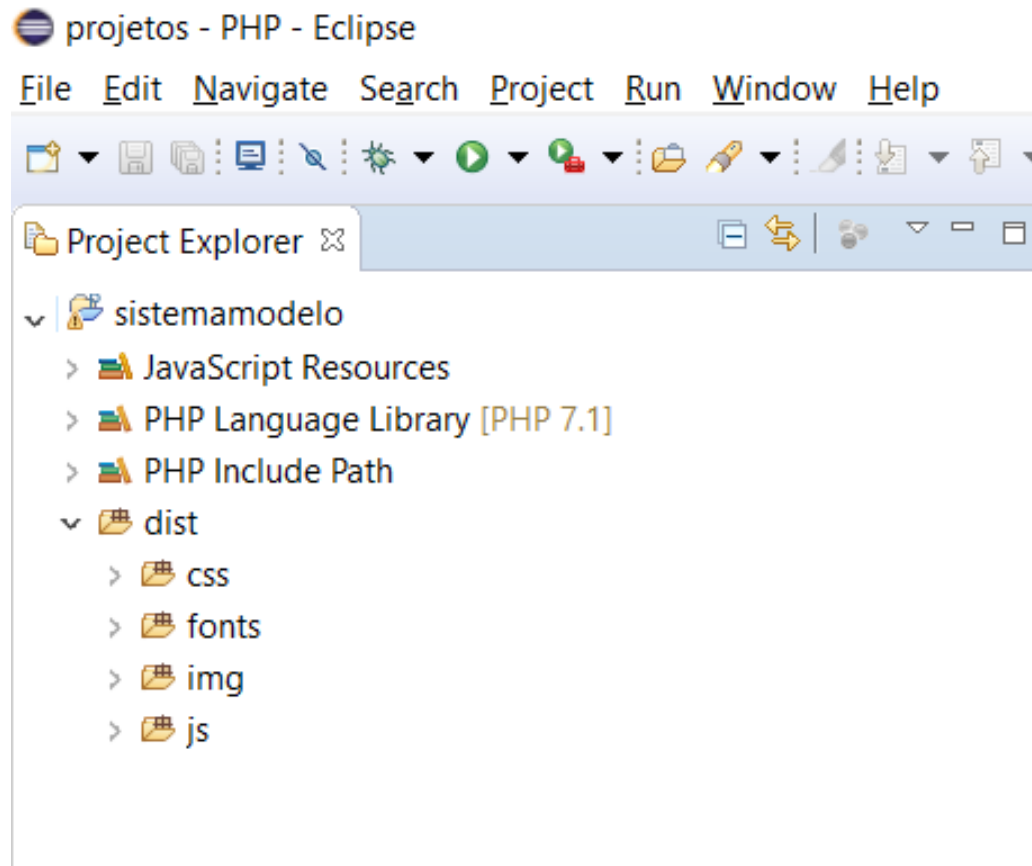
- ▶ Logo.png

- ▶ News.jpg

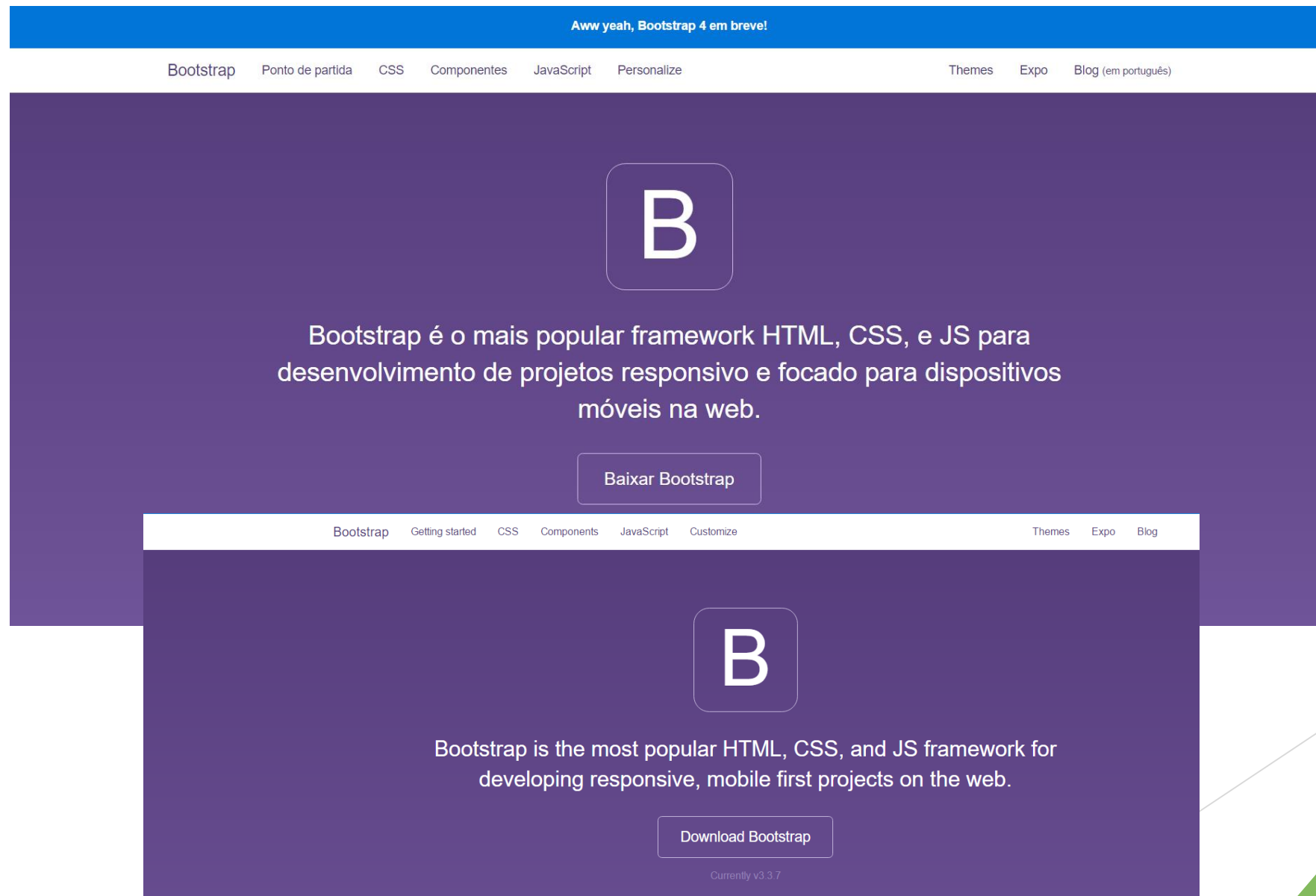


# Download e Configuração Inicial do Bootstrap

- ▶ Ao final deste vídeo, esta é a estrutura de pastas e arquivos necessária para continuarmos:



Aproveite para explorar outras informações disponíveis no site do Bootstrap, seja na versão em inglês ou em português!



# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Olá Mundo! No Bootstrap

# Olá Mundo! No Bootstrap

- ▶ Vamos iniciar o desenvolvimento de telas estáticas no Bootstrap.
- ▶ Algumas opções:
  - ▶ iniciar de um template básico ou modificar exemplos de templates prontos para Bootstrap.
- ▶ Primeiro Passo: template básico
- ▶ Acessar o portal do Bootstrap.
  - ▶ Menu: Getting started.
  - ▶ Basic Template.
  - ▶ Copie o código HTML disponível (estude ele antes!!!).
  - ▶ Na pasta do projeto(ex: sistemamodelo), crie um arquivo HTML chamado “ola.html” e cole o código.
- ▶ Para testar o funcionamento de “ola.html”:
  - ▶ No Eclipse: clique com o botão direito no arquivo e escolher a opção Open with (abrir com) - Web Browser (navegador web).
  - ▶ Servidor web local: acesse o navegador e digite localhost/projeto/<nome da pasta do projeto>



← → ↻ ⓘ localhost/projetos/sistemamodelo/ola.html

**Hello, world!**



# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

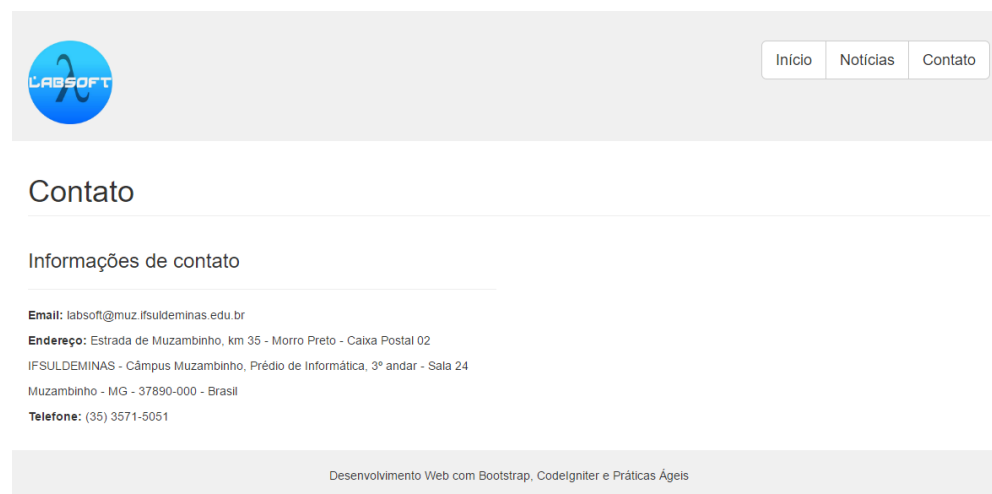
Telas do Sistema Modelo

# Telas do Sistema Modelo

- ▶ O nosso SistemaModelo é separado em Ambiente de Usuário e Ambiente Administrativo.
- ▶ O Ambiente de usuário é composto, em sua primeira versão, por 4 telas:



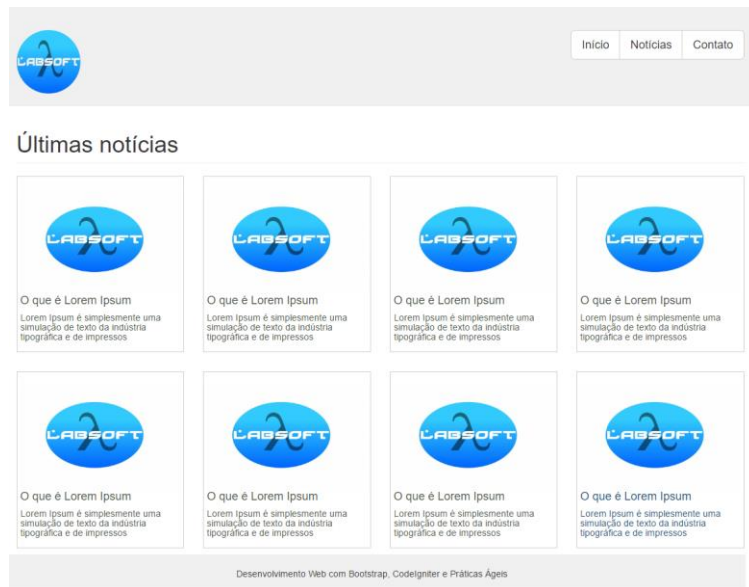
Página Inicial



Contato (dados estáticos)

# Telas do Sistema Modelo

## ► Continuação do Ambiente de Usuário:



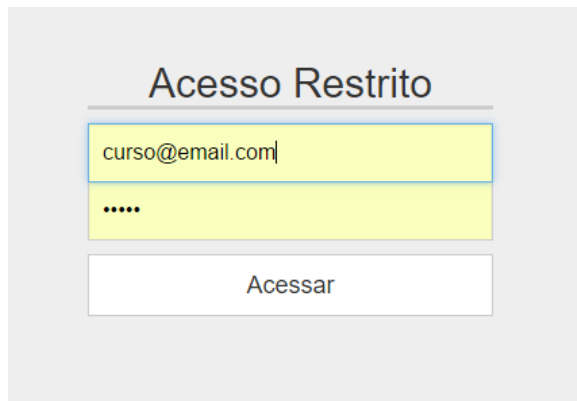
## Menu Notícias - Listagem



## Ver Notícia

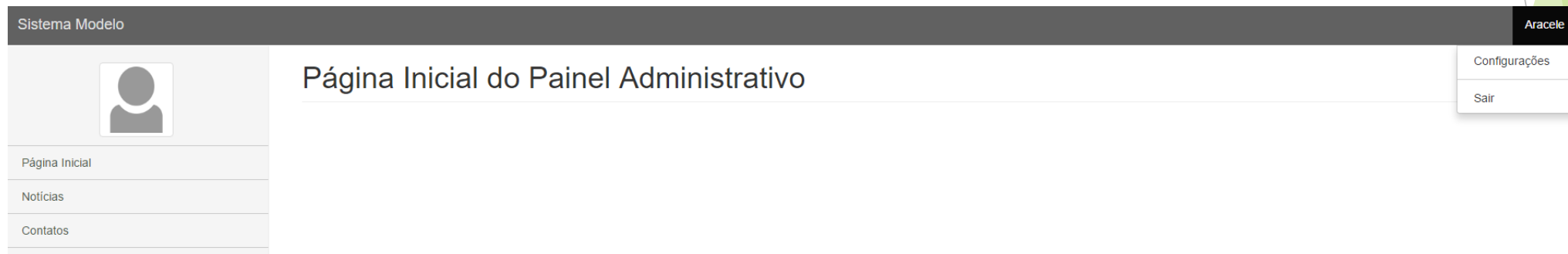
# Telas do Sistema Modelo

- ▶ Já o **Ambiente Administrativo** é composto por 6 telas:



A screenshot of a login form titled "Acesso Restrito". It features two yellow input fields: the first contains the email "curso@email.com" and the second contains masked characters "\*\*\*\*\*". Below these fields is a white button labeled "Acessar".

Formulário de Login




Página Inicial do Ambiente Administrativo

# Telas do Sistema Modelo

## ► Continuação das telas do Ambiente Administrativo:

Projeto Notícias

Curso Codeigniter



Página Inicial

Notícias

Contatos

## Notícias

Show 10 entries

Search:

Nova notícia

#	Título	Subtítulo	Tipo	
2	O que é Lorem Ipsum	Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.	Evento	<div>Ver Notícia</div> <div>Atualizar</div> <div>Deletar</div>
5	O que é Lorem Ipsum	Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.	Curso promovido	<div>Ver Notícia</div> <div>Atualizar</div> <div>Deletar</div>
6	O que é Lorem Ipsum	Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos	Apresentação	<div>Ver Notícia</div> <div>Atualizar</div> <div>Deletar</div>
7	O que é Lorem Ipsum	Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos	Apresentação	<div>Ver Notícia</div>


## Listagem de Notícias

# Telas do Sistema Modelo

- Continuação das telas do Ambiente Administrativo:

Projeto Notícias

Curso Codeigniter



Página Inicial

Notícias

Contatos

## Cadastro de notícias

Título

Subtítulo

Conteúdo

Enviar

Limpar

Slug

Data

Tipo

Apresentação

Imagem da noticia

Choose File

No file chosen

Cadastro de Notícias

# Telas do Sistema Modelo

## ► Continuação das telas do Ambiente Administrativo:

Projeto Notícias

Curso Codeigniter



[Página Inicial](#)

[Notícias](#)

[Contatos](#)

### Atualização de notícias

**Título**

**Subtítulo**

**Conteúdo**

O que é Lorem Ipsum

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker. Porque nós o usamos?

Enviar

Limpar

**Slug**

**Data**

**Tipo**

Evento

**Imagem da noticia**

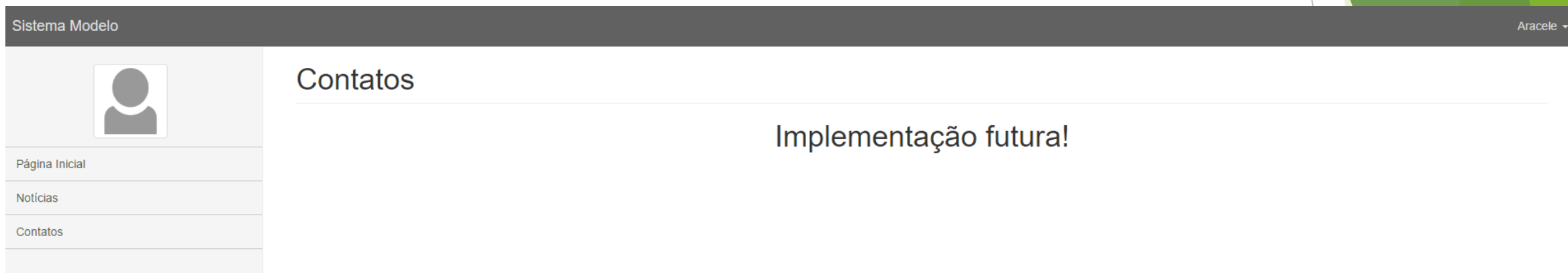
Choose File

 No file chosen

## Atualização de Notícias

# Telas do Sistema Modelo

- Continuação das telas do Ambiente Administrativo:



Contatos



# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Usando templates do Bootstrap para criar as telas de  
acesso e a base do ambiente administrativo

# Telas do Ambiente Administrativo: Preparação

- ▶ Opções para trabalhar com telas no Bootstrap:
  - ▶ 1) fazer todas as telas, testá-las, depois migrá-las para a aplicação (já com o CodeIgniter) e fazer as devidas mudanças.
  - ▶ 2) fazer as telas já dentro da estrutura do CodeIgniter, mas há uma dificuldade em testar interface constantemente. A solução é testar só no final, o que dificulta achar erros.
- ▶ Ideia: criar algumas telas em um projeto paralelo, sem CodeIgniter (opção 1)
- ▶ Neste vídeo: acesso (login) e a base do ambiente administrativo:

Acesso Restrito

curso@email.com|

.....

Acessar



# Telas do Ambiente Administrativo: Preparação

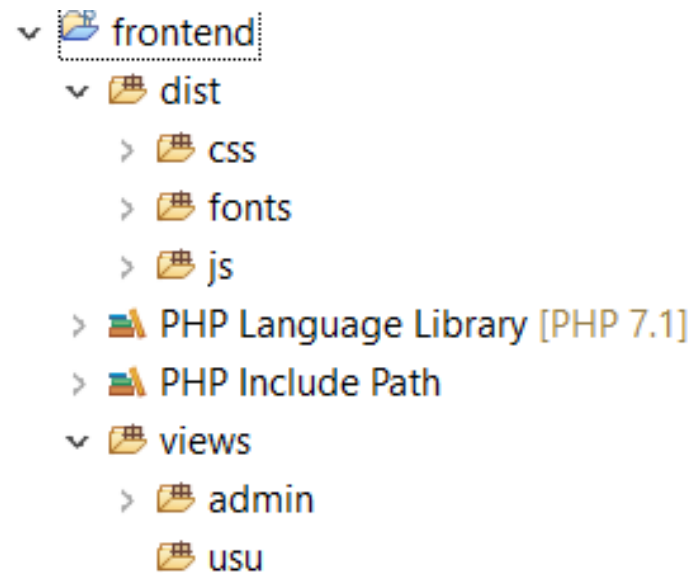
## ► Preparação:

- Crie um novo projeto temporário chamado “frontend”.
- Copie a pasta bootstrap-3.3.7\dist para dentro do projeto.
- Crie as pastas

- views

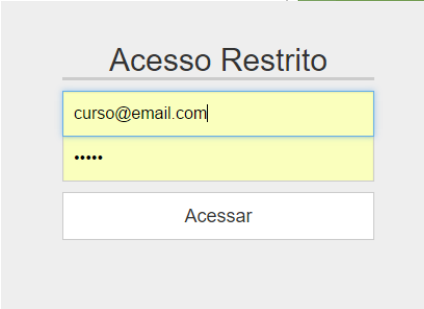
- admin

- usu



# Telas do Ambiente Administrativo: Acesso

- ▶ <http://getbootstrap.com/> - Getting started
- ▶ Examples - Clicar em “Sign-in page”
- ▶ Uma opção para pegar o código:
  - ▶ Botão direito - View Page Source (Ver Código Fonte) - Copiar todo o código
  - ▶ Dentro de “views/admin”, criar um arquivo php chamado “acessoview.php”.
  - ▶ Colar o código do template.
- ▶ Outra forma de fazer:
  - ▶ Copie o arquivo “index.html” que está em bootstrap-3.3.7\docs\examples\signin
  - ▶ Cole no projeto temporário frontend/views/admin
  - ▶ Renomeie para acessoview.php
  - ▶ Verifique o caminho do css “signin.css”: `<link href="../../dist/css/signin.css" rel="stylesheet">`
  - ▶ Arraste o css correspondente “signin.css” e coloque dentro da pasta “dist”.
- ▶ **1º Teste:**
  - ▶ abra o navegador e digite localhost/frontend/views/admin/acessoview.php



Acesso Restrito

curso@email.com|

.....

Acessar

acessoview.php

# Telas do Ambiente Administrativo: Acesso

## ► Configurações iniciais no arquivo views/admin/acessoview.php

- Acrescentar os nomes de email e senha:

```
<div class="container">

  <form class="form-signin">
    <h2 class="form-signin-heading">Please sign in</h2>
    <label for="inputEmail" class="sr-only">Email address</label>
    <input type="email" id="inputEmail" class="form-control" placeholder="Email address" name="email" required autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" class="form-control" placeholder="Password" name="senha" required>
    <div class="checkbox">
      <label>
        <input type="checkbox" value="remember-me"> Remember me
      </label>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
  </form>

</div> <!-- /container -->
```

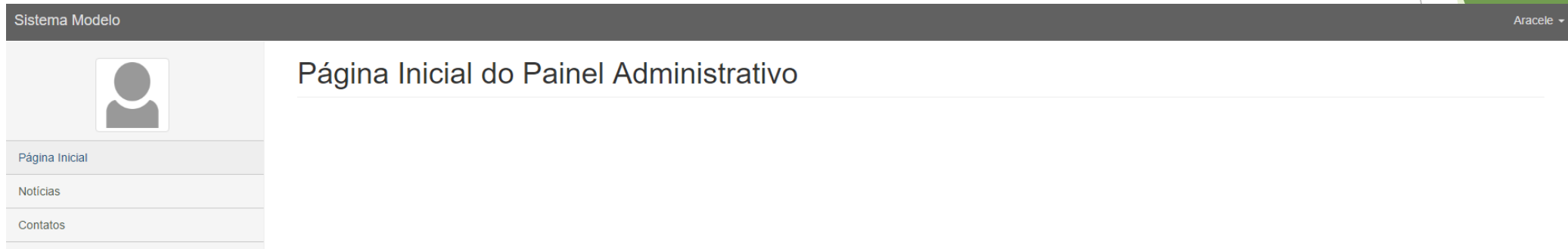


## ► Fazer as demais alterações e salve o arquivo acessoview.php

- <title>Signin Template for Bootstrap</title> → Acesso ao Ambiente Administrativo
- Please Sign in → Faça o login
- Email address → E-mail
- Password → Senha
- Remember me → Lembrar
- Sign in → Acessar

# Telas do Ambiente Administrativo: Painel Inicial

- ▶ Essa é a página Inicial da área administrativa do Sistema Modelo:



- ▶ Vamos criar a base de todo o ambiente administrativo (exceto o acesso) a partir de um template disponível no portal do Bootstrap:
  - ▶ Template: dashboard

# Telas do Ambiente Administrativo: Painel Inicial

## ► Acessar o portal do Bootstrap: <http://getbootstrap.com>

- Menu: Getting started.
- Examples
- Verificar templates disponíveis: dashboard.
- Clicar para acessá-lo.
- Botão direito - ver o código fonte.
- Copiar todo o código (ctrl+a).
- Na pasta do projeto temporário frontend, dentro de ‘/views/admin’, crie um arquivo chamado ‘painelview.php’ e adicione a ele o código copiado.
- Salve o arquivo painelview.php



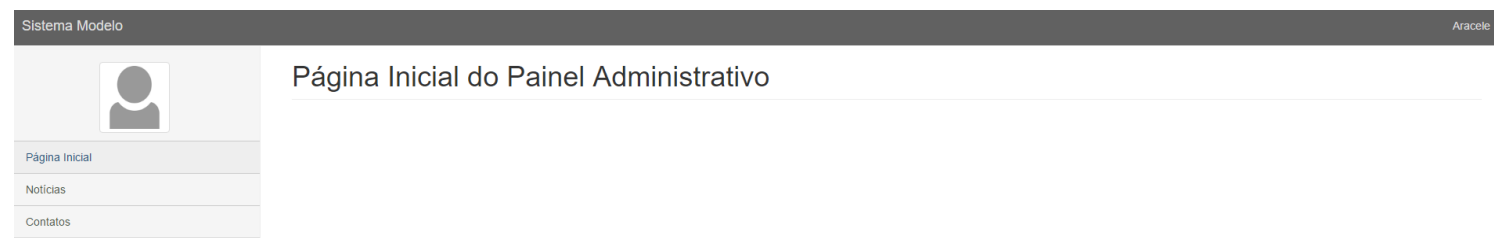
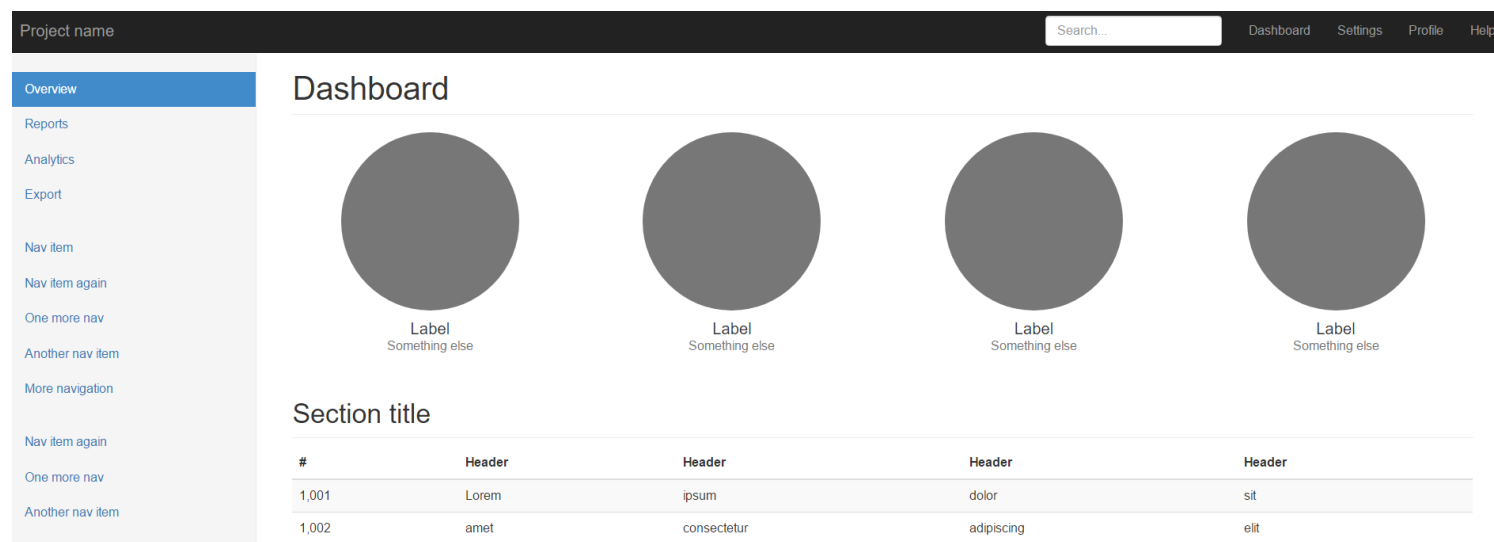
## ► Outra forma de fazer:

- Copie o arquivo index.html que está em bootstrap-3.3.7\docs\examples\dashboard
- Cole-o no projeto temporário frontend/views/admin
- Renomeie para painelview.php
- Verificar o caminho do css “dashboard.css”: `<link href=".../dist/css/dashboard.css" rel="stylesheet">`
- Mover o css correspondente “dashboard.css” para dentro da pasta “dist”.

## ► Teste no navegador: <http://localhost/projetos/frontend/views/admin/painelview.php>

# Telas do Ambiente Administrativo: Painel Inicial

- Agora só falta customizar o template para as nossas necessidades.





# Telas do Ambiente Administrativo: Painel Inicial

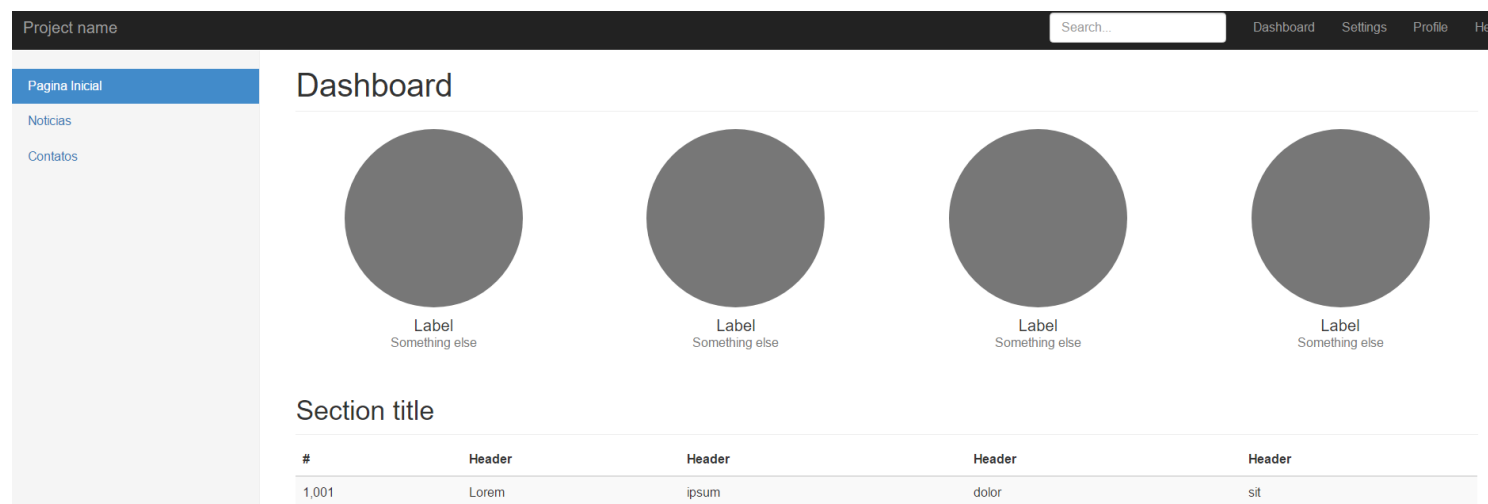
- Customização do template para criar a tela inicial do ambiente administrativo:

- Exemplo:

- Em painelview.php, procure por Project Name e troque para “Sistema Modelo”
- Procure por “Overview” e troque para “Página Inicial”
- Procure por “Repots” e troque para “Notícias”.
- Procure por “Analytics” e troque para “Contatos”.
- Apague o restante dos itens deste menu. →
- Salve o painelview.html
- Verifique o resultado:

```
<ul class="nav nav-sidebar">
  <li><a href="">Nav item</a></li>
  <li><a href="">Nav item again</a></li>
  <li><a href="">One more nav</a></li>
  <li><a href="">Another nav item</a></li>
  <li><a href="">More navigation</a></li>
</ul>
<ul class="nav nav-sidebar">
  <li><a href="">Nav item again</a></li>
  <li><a href="">One more nav</a></li>
  <li><a href="">Another nav item</a></li>
</ul>
```

- Navegador: <http://localhost/projetos/frontend/views/admin/painelview.html>



# Telas do Ambiente Administrativo: Painel Inicial

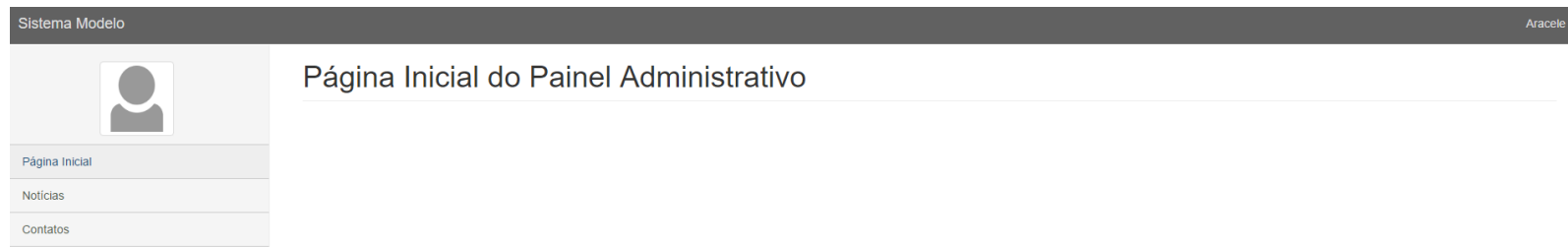
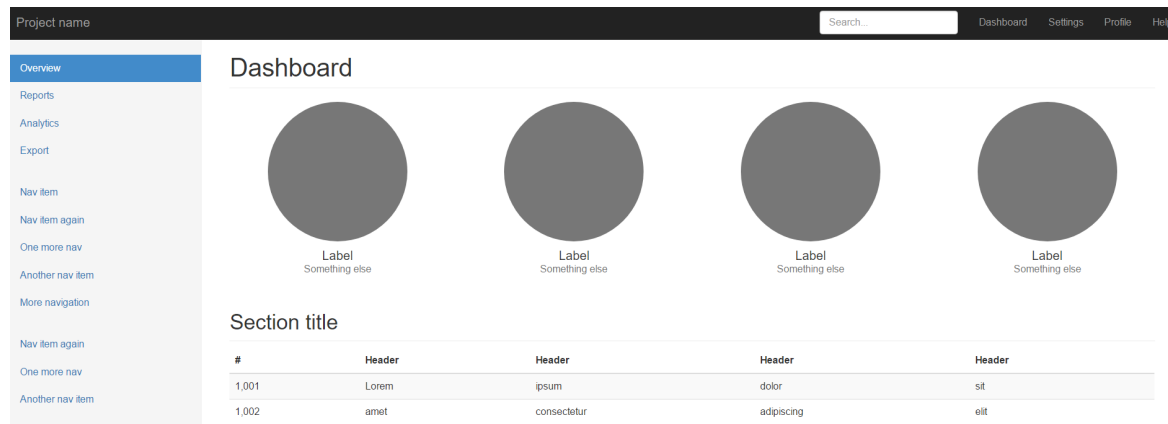
- ▶ Não precisamos da parte que contém os círculos. Você pode excluir:
- ▶ Apague a `<div class="row placeholders">` e todo o trecho relacionado:

```
<div class="row placeholders">
  <div class="col-xs-6 col-sm-3 placeholder">
    
    <h4>Label</h4>
    <span class="text-muted">Something else</span>
  </div>
  <div class="col-xs-6 col-sm-3 placeholder">
    
    <h4>Label</h4>
    <span class="text-muted">Something else</span>
  </div>
  <div class="col-xs-6 col-sm-3 placeholder">
    
    <h4>Label</h4>
    <span class="text-muted">Something else</span>
  </div>
  <div class="col-xs-6 col-sm-3 placeholder">
    
    <h4>Label</h4>
    <span class="text-muted">Something else</span>
  </div>
</div>
```

- ▶ Salve o arquivo e teste o resultado no navegador:  
<http://localhost/projetos/frontend/views/admin/painelview.html>

# Telas do Ambiente Administrativo: Painel Inicial

- ▶ Agora que você já entendeu a ideia principal, faça as demais alterações necessárias.
- ▶ Basta comparar as figuras das respectivas telas e apagar o trecho de código associado.
  - ▶ Ou comentar o código !!!



# Telas do Ambiente Administrativo: Considerações

## ► Resumindo:

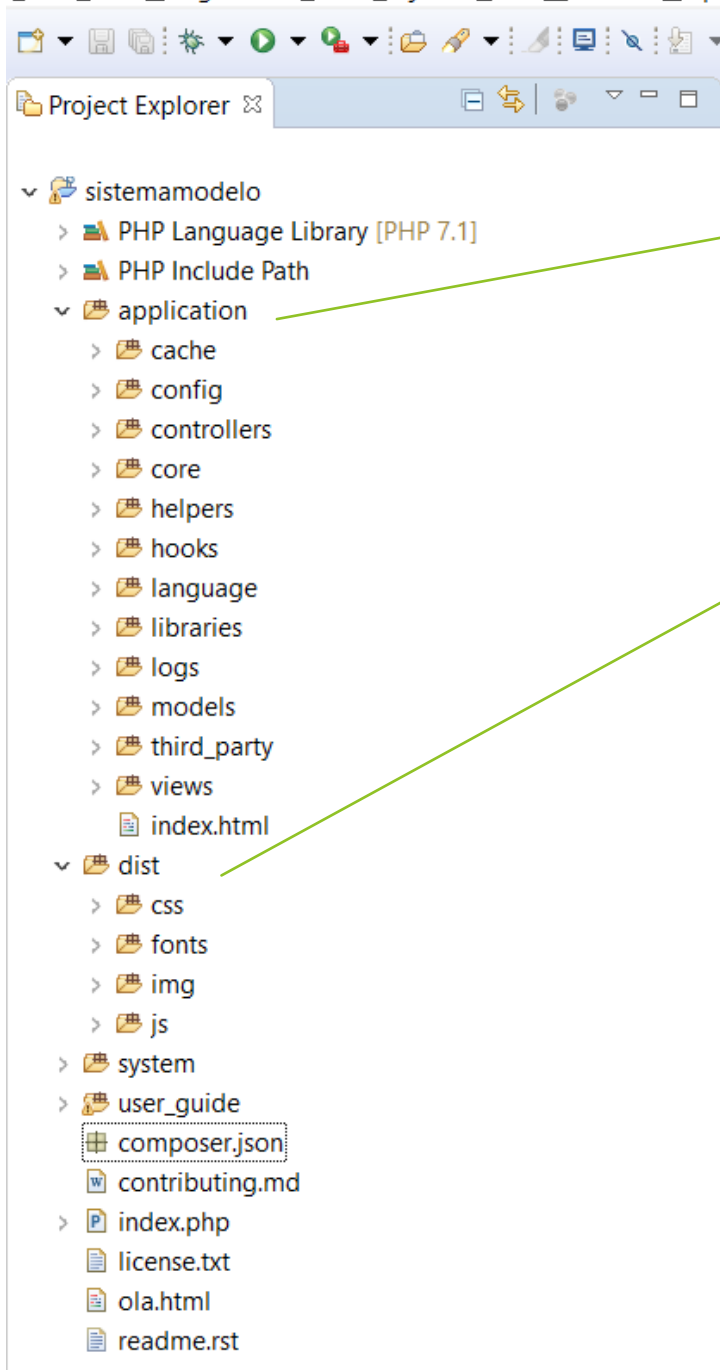
- Este vídeo apresentou uma ideia geral para usar templates prontos a partir do portal do Bootstrap.
- Dica: também é possível encontrar templates gratuitos em outros sites (Google it!)
- Próximos passos:
  - Criar as demais telas.
  - Migrá-las para a estrutura do CodeIgniter.
  - Fazer as devidas alterações.

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Download do CodeIgniter

# Download do CodeIgniter

- ▶ Acesse a página de download do site CodeIgniter: <https://www.codeigniter.com>
- ▶ Clique em Download
  - ▶ Observe que na ocasião desta gravação, a última versão disponível era a 3.1.4
- ▶ Descompacte o arquivo de download (botão direito - Extrair aqui).
- ▶ Mova as pastas nativas dele para dentro do projeto criado na IDE escolhida
  - ▶ Exemplo: Eclipse - sistemamodelo



## Estrutura de Pastas e Arquivos do projeto após a inserção dos arquivos originados do CodeIgniter e do Bootstrap

Primeira forma de integração Bootstrap e CodeIgniter.

Observe as pastas mais importantes:

- Config
- Controllers
- Models
- views

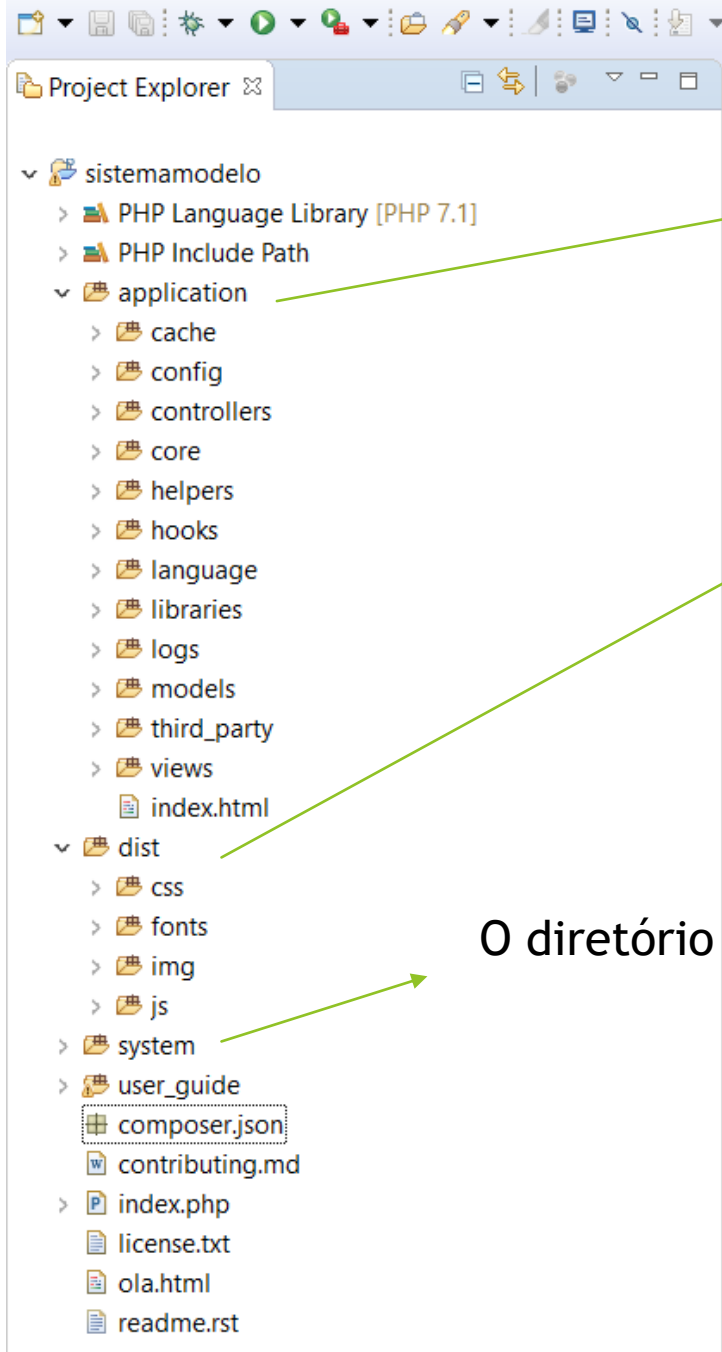
# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Funcionamento Geral do CodeIgniter



# Funcionamento Geral do CodeIgniter

- ▶ O CodeIgniter trabalha com o padrão de arquitetura MVC.
- ▶ Ou seja, ele possui os diretórios principais (Models, Views e Controllers) e também alguns diretórios auxiliares (Cache, Config, Core, Helpers, Hooks, Language, Libraries, Logs, Models, Thirdy\_party).
- ▶ Os diretórios mais utilizados:
  - ▶ Controllers, Models e Views.
  - ▶ Config: Onde serão armazenados os arquivos de configurações (database.php, config.php, routes.php, etc.);
  - ▶ Helpers: Para criar funções auxiliares.
  - ▶ Libraries: Semelhates os helpers, porém para guardar bibliotecas de funções maiores, por exemplo: login por Facebook, Google e Twitter;
  - ▶ Core: extensões de arquivos nativos existentes.



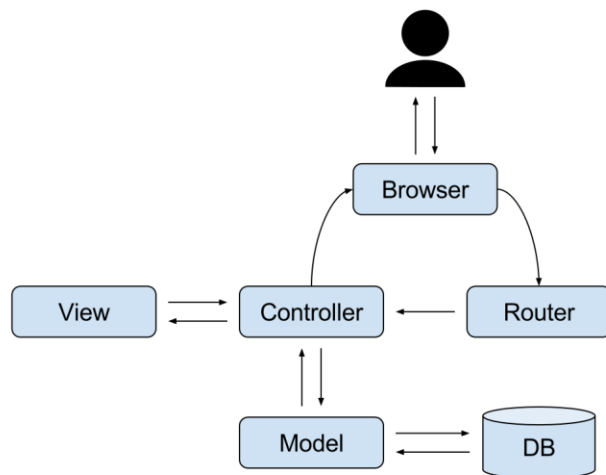
Estrutura de Pastas e Arquivos do projeto após a inserção dos arquivos originados do CodeIgniter e do Bootstrap

O diretório “/system” possui as bibliotecas nativas do CodeIgniter.

# Funcionamento Geral do CodeIgniter MVC

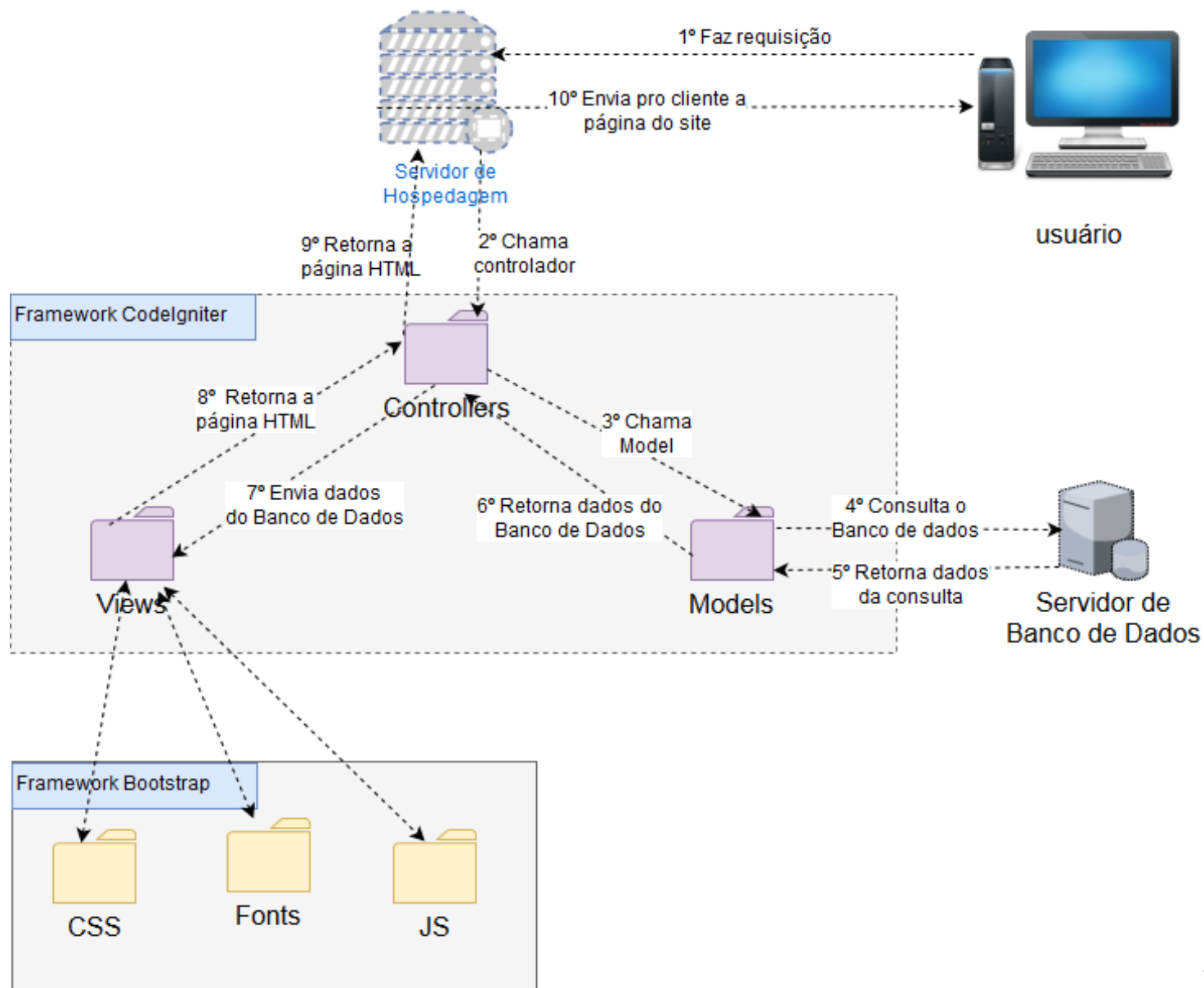
- ▶ O padrão de arquitetura MVC (model-view-controller) ou modelo-visão-controlador ajuda a organizar o projeto, facilita o reuso: desenvolvimento em camadas.

- ▶ Fluxo Básico



- ▶ Implementação do MVC por frameworks de desenvolvimento:
  - ▶ É importante compreender como um framework de desenvolvimento implementa MVC.
  - ▶ Podem existir algumas variações estruturais que impactam o fluxo dos dados e mudanças de nomes/termos.
  - ▶ De forma geral, o CodeIgniter implementa MVC sem muitas mudanças drásticas.

# Funcionamento do framework CodeIgniter, desde o cliente até o servidor de banco de dados:



# Funcionamento Geral do CodeIgniter

## Controladores (controllers)

### ► Estrutura de URL

- Existe uma relação entre a estrutura de pastas e arquivos de um Sistema criado com o CodeIgniter e as URLs de acesso.
- Exemplo:
  - [www.sistemamodelo.com.br/noticias/eventos/evento\\_outubro](http://www.sistemamodelo.com.br/noticias/eventos/evento_outubro).
  - sistemamodelo: pasta geral do Sistema.
  - noticias: classe noticias (controlador notícias).
  - eventos: método (ou função) que existe dentro da classe (controlador) notícias.
  - evento\_outubro: é o parâmetro ou id que o método “eventos” recebe.

### ► Outro Exemplo:

- Ao digitar [www.sistemamodelo.com.br/admin](http://www.sistemamodelo.com.br/admin) uma requisição é enviada ao servidor web, a fim de encontrar uma classe admin (controlador) contida em um arquivo “admin.php”.
- Caso tal controlador seja encontrado, ele entra em ação e fica responsável por controlar o fluxo do programa.
- Os Controladores devem ficar em /application/controllers.
- É importante existir um controller para cada “funcionalidade”.
- Nome de Classes com inicial maiúscula e o restante em letras minúsculas.

# Funcionamento Geral do CodeIgniter

## Visão (View ou views)

### ► Visão (view):

- Responsável pela apresentação, o que será exibido para o usuário.
- Formada por HTML, CSS, JS.
- Devem ser armazenadas em **application/views**.
- Em geral, views são diretamente ligadas a um controller (view noticia, controller noticia).
- Pode-se criar subpastas (módulos) para organizar as views (ex: views/admin views/usu)

### ► Funções:

- “Renderizar” (formatar/ajustar) informações vindas dos controllers;
- Requisitar atualizações ao controller correspondente;
- Enviar ações de usuários ao controller correspondente;

### ► Outras considerações:

- O controller passa dados para uma view dessa forma:
  - `$this->load->view('noticia', $array_noticias);`
- É possível chamar várias views, o controller faz a “montagem” em uma única exibição

```
$this->load->view('admin/includes/topo');  
$this->load->view('admin/includes/menu');  
$this->load->view('admin/painelview');  
$this->load->view('admin/includes/rodape');
```

# Funcionamento Geral do CodeIgniter

## Modelo (model ou models)

### ► Modelo (model)

- São classes PHP e projetadas para trabalhar com informações do banco de dados.
- Contém, usualmente, métodos para inserção, atualização, recuperação/seleção e exclusão de dados no banco de dados.

### ► Visão Geral de Classe Modelo no CodeIgniter:

- Classes Modelo são armazenadas no diretório **application/models/**
- Nomes de classes: primeira letra maiúscula e o restante com letras minúsculas.
  - Ex.: **Acessomodel**
- Uma classe modelo será “carregada” por um controlador (letras minúsculas):
  - `$this->load->model('admin/acessomodel');`
- Uma vez carregada, por padrão, você vai acessar os métodos da classe Modelo usando um **objeto** que tem o mesmo nome da classe Modelo:
  - `$usuario = $this->acessomodel->logar($email, $senha);`
  - Neste caso aqui, o método `logar` recebe 2 parâmetros e retorna um valor que é armazenado na variável `$usuario`.

# Funcionamento Geral do CodeIgniter

## Modelo (model or models)

- Exemplo de um controlador que carrega um modelo e repassa para a visão:

```
class Blog_controller extends CI_Controller {  
  
    public function blog()  
    {  
        $this->load->model('blog');  
  
        $data['query'] = $this->blog->get_last_ten_entries();  
  
        $this->load->view('blog', $data);  
    }  
}
```

Fonte: Documentação do CodeIgniter  
<https://www.codeigniter.com/userguide3/general/models.html>



# Funcionamento Geral do CodeIgniter

## Modelo (model or models)

### ► CodeIgniter possui uma classe chamada Query Builder.

- Ela permite a manipulação de dados (inclusão, exclusão, etc) com o mínimo de código. Em alguns casos, uma ou duas linhas de código são necessárias para realizar uma ação do banco de dados.
- Também permite a independência entre aplicação e banco de dados.
  - Ex.: hoje a sua aplicação usa o MySQL. Amanhã pode usar SqlServer.
- Se deseja criar suas próprias consultas ao banco, basta desabilitar QueryBuilder.

### ► Exemplos de código:

► A) `$query = $this->db->get('mytable');` *// Produces: SELECT \* FROM mytable*

► B) `$query = $this->db->get('mytable', 10, 20);`

*// Executes: SELECT \* FROM mytable LIMIT 20, 10  
// (in MySQL. Other databases have slightly different syntax)*

► C) `$this->db->select('title, content, date');`  
`$query = $this->db->get('mytable');`

*// Executes: SELECT title, content, date FROM mytable*

► D) `$this->db->order_by('title', 'DESC');`  
*// Produces: ORDER BY `title` DESC*

# Funcionamento Geral do CodeIgniter

## URI (Uniform Resource Identifier ) Routing

- ▶ URI é uma forma generalizada para denominar URLs (Uniform Resource Locator) e URNs (Uniform Resource Name).
- ▶ **Contexto geral:**
  - ▶ De forma geral, existe um relacionamento 1-para-1 entre uma URI e uma classe do tipo controlador e seus métodos.
  - ▶ Uma URL costuma ter o seguinte padrão: `example.com/class/function/id/`
  - ▶ Entretanto, em alguns casos, você pode remapear esse relacionamento para uma nova classe e/ou método.
- ▶ **Como fazer?**
  - ▶ Regras de roteamento são definidas no arquivo: `application/config/routes.php`
  - ▶ Exemplo: `$route['journals'] = 'blogs';`
  - ▶ A cabeça é composta por uma URI (journals) e o corpo contém o destino para onde a URI deve ser roteada ou encaminhada (controlador blogs).
  - ▶ Rotas são executadas na ordem em que são definidas.
  - ▶ Rotas definidas primeiro possuem maior precedência ou prioridade.

# Funcionamento Geral do CodeIgniter

## URI (Uniform Resource Identifier ) Routing




































- ▶ Relembrando o padrão de regras de roteamento: `example.com/class/function/id/`
- ▶ **Outro exemplo:** `$route['blog/joe'] = 'blogs/users/34';`
  - ▶ A URI contendo o seguimento “blog/joe” será remapeada para a class Controller “blogs”, o método (function) “users” e o ID receberá o valor “34”.
- ▶ **Expressões regulares:**
  - ▶ Você pode usar expressões regulares para definir regras de roteamento:  
`$route['products/([a-z]+)/(\d+)'] = '$1/id_$2';`
- ▶ **Rotas reservadas:** `www.sistemamodelo.com.br`
  - ▶ Quando o usuário acessa a URL principal: `$route['default_controller'] = 'welcome';`
    - ▶ O corpo desta regra aceita um valor do tipo controller/método
    - ▶ Caso não seja especificado o método, o index() da classe controller será chamado.
    - ▶ Exemplo: controllers/Welcome/index()
  - ▶ Quando uma página não é encontrada: `$route['404_override'] = '';`

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Customização da estrutura de pastas do CodeIgniter  
para atender as necessidades do projeto

# Customização da estrutura de pastas do CodeIgniter

- ▶ O nosso projeto chamado SistemaModelo é separado em **Ambiente de Usuário** e **Ambiente Administrativo**.
- ▶ Por isso, para cada pasta Model-View-Controller, crie as pastas:
  - ▶ admin
  - ▶ usu
- ▶ Veja o resultado:

- ▼  sistemamodelo
  - >  PHP Language Library [PHP 7.1]
  - >  PHP Include Path
  - ▼  application
    - >  cache
    - >  config
    - ▼  controllers
      -  admin ←
      -  usu
      -  index.html
      - >  Welcome.php
    - >  core
    - >  helpers
    - >  hooks
    - >  language
    - >  libraries
    - >  logs
  - ▼  models
    -  admin ←
    -  usu
    -  index.html
    - >  third\_party
  - ▼  views
    -  admin
    - >  errors ←
    -  usu
    -  index.html
    - >  welcome\_message.php
    -  index.html
  - >  dist
  - >  system
  - >  user\_guide
  -  composer.json
  -  contributing.md
  - >  index.php

# Customização da estrutura de pastas do CodeIgniter

- ▶ O nosso Sistema Modelo permite o cadastro de Notícias
- ▶ Cada notícia tem uma figura associada.
- ▶ Ao cadastrar uma notícia e adicionar uma figura, ela precisa ser salva em uma pasta do seu projeto.
- ▶ Então, dentro do projeto sistemamodelo:
  - ▶ Crie uma subpasta “imagens”.
  - ▶ E, dentro de “imagens”, crie uma subpasta “noticias” (sem acento).

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Configuração Inicial do CodeIgniter



# Configuração Inicial do CodeIgniter

## ► Configuração do arquivo config.php:

- Acesse o arquivo /application/config/config.php e especifique qual vai ser a URL base de acesso ao sistema:

Onde está: `$config['base_url'] = '';`

Altere para: `$config['base_url'] = 'http://localhost/projetos/sistemamodelo/';`

- Ainda neste arquivo config.php, altere o valor da página index.

- Dessa forma, quando alguém digitar [www.sistemamodelo.com.br](http://www.sistemamodelo.com.br), não precisará informar a página inicial index.php ([www.sistemamodelo.com.br/index.php](http://www.sistemamodelo.com.br/index.php)) porque automaticamente será redirecionado para a index.php.

Onde está: `$config['index_page'] = 'index.php'`

Altere para: `$config['index_page'] = '';`

- Não esqueça de salvar o arquivo.

- Teste no navegador: <http://localhost/projetos/sistemamodelo/>



# Configuração Inicial do CodeIgniter

## ► Criação do Banco de Dados

- No navegador, digite localhost.



- Acesse o phpMyAdmin
- Clique em “Base de Dados”
  - Criar Base de Dados: sistemamodelo
  - Clique em “Criar”
- Faça o download do script MySQL (estará disponível na plataforma do curso).
- Execute o script MySQL para criar o Banco De Dados.
  - Importar - File to import - Choose File
  - Escolha o script “database.sql” disponibilizado e clique em “Executar”.

# Configuração Inicial do CodeIgniter

## ► Configuração do Banco de Dados no CodeIgniter

- No Eclipse, acesse o arquivo /application/config/database.php e configure as seguintes linhas, considerando a configuração do seu banco de dados local.

`'hostname' => 'localhost',`

`'username' => 'root',`

`'password' => '',`

`'database' => 'sistemamodelo',`

Faça as adaptações  
necessárias, considerando o  
nome do banco de dados e os  
dados de acesso!

Não esqueça de salvar o  
arquivo database.php

# Configuração Inicial do CodeIgniter

## ► Configuração do arquivo autoload.php

- Útil para informar quais recursos serão automaticamente carregados por padrão. Dessa forma, a aplicação fica leve, usando recursos mínimos para funcionar.
- Essa configuração também faz com que o CodeIgniter carregue as bibliotecas nativas “database” e “session” para a manipulação de banco de dados e controle de sessões dos usuários, bem como códigos prontos de apoio (helpers, funções nativas).

- Acesse o arquivo /application/config/autoload.php e configure as linhas:

Onde está: `$autoload['libraries'] = array();`

Altere para: `$autoload['libraries'] = array('database','session');`

Onde está: `$autoload['helper'] = array();`

Altere para: `$autoload['helper'] = array('url','file');`

- Salve o arquivo autoload.php

# Configuração Inicial do CodeIgniter

## ► Configurando o arquivo “.htaccess”

- Permite a reescrita de uma URL.
  - No nosso caso, estamos permitindo a reescrita da URL para remover o termo “index.php” da URL de acesso principal ao site;
  - Ou seja, em vez de usar assim localhost/projetos/sistemamodelo/index.php, a nossa aplicação criada com CodeIgniter usará localhost/projetos/sistemamodelo.
- No Eclipse, fora da pasta “/application” (raiz do projeto), crie (se não existir) um arquivo chamado “.htaccess”.
  - New - other - general - file (novo - outro - geral - arquivo).
  - Nome: .htaccess
  - Coloque o seguinte código e salve

*RewriteEngine on*

*RewriteCond %{REQUEST\_FILENAME} !-f*

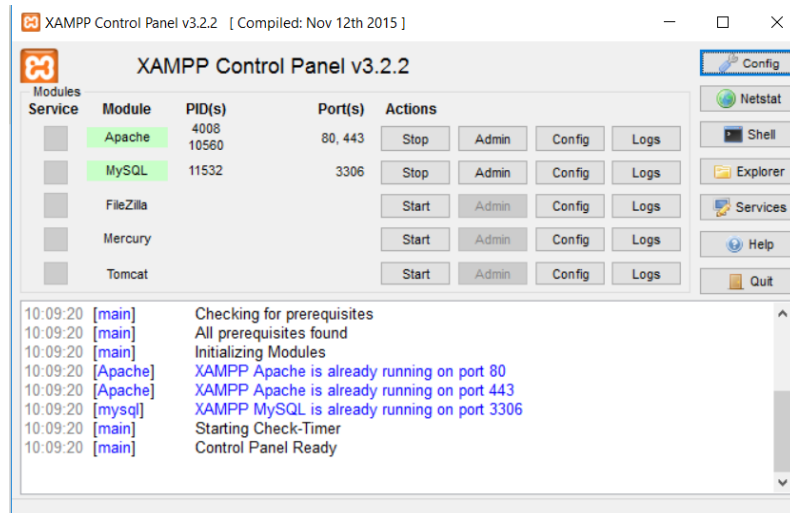
*RewriteCond %{REQUEST\_FILENAME} !-d*

*RewriteRule .\* index.php/\$1 [PT,L]*

# Configuração Inicial do CodeIgniter

## ► Configurando o arquivo “httpd.conf”

- O servidor precisa saber que você está reconfigurando o próprio servidor a partir do arquivo “.htaccess” no seu site.
- Então é necessário (caso você esteja usando o Xampp) acessar o arquivo “httpd.conf”, localizado em C:\xampp\apache\conf e descomentar esta linha:  
`#LoadModule rewrite_module modules/mod_rewrite.so`
- Ficará da seguinte forma:  
*`LoadModule rewrite_module modules/mod_rewrite.so`*
- Depois disto, reinicie novamente o seu servidor Apache.



# Configuração Inicial do CodeIgniter

## ► Configuração do arquivo routes.php

- No CodeIgniter, uma URL e um controlador tem esse relacionamento: pasta\_do\_projeto/classe/metodo
  - Exemplo: sistemamodelo/noticias/insere\_noticias
- Entretanto, às vezes é necessário “remapear” este relacionamento.
- No CodeIgniter, isso é feito através de **routes** (rotas).
- Usa-se o arquivo *routes.php* para definir novas rotas para atender uma requisição.
- Se nenhum controlador é chamado (ex:localhost/projetos/sistemamodelo/), o CodeIgniter chamará o controlador default.
- Por isso, é importante configurar um controlador default para todo o Sistema.
- Quando qualquer página é acessada, o controlador principal do Sistema é chamado, no Sistema Modelo será:
  - “.../controllers/admin/acesso.php”
  - “.../controllers/usu/paginainicial.php”

# Configuração Inicial do CodeIgniter

## ► Configuração da configuração do arquivo routes.php

- Para criar um controlador default, acesse o arquivo /application/config/routes.php

Onde está `$route['default_controller'] = 'welcome';`

Troque para `$route['default_controller'] = 'home';`

- **Nota:** Essa parte será alterada posteriormente, não precisa fazer nada agora, mas saiba da importância de se definir um controlador padrão!



# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Ambiente Administrativo - Acesso

# Ambiente Administrativo - Acesso

- ▶ View “acessoview.php”:
  - ▶ Copiar “acessoview.php” de frontend e colocar em application/views/admin
  - ▶ O arquivo “signin.css” deve estar dentro da pasta “dist/css”
- ▶ Alterações
  - ▶ `<link href="<?= base_url() ?>dist/css/bootstrap.min.css" rel="stylesheet">`
  - ▶ `<link href="<?= base_url() ?>dist/css/signin.css" rel="stylesheet">`

```
<div class="container">
    <form class="form-signin" action="<?= base_url('admin/acesso/Logar') ?>" method="post">
        <?php if (isset($alert)) { ?>
            <div class="alert alert-<?php
                $a = explode('-', isset($alert) ? $alert : '');
                echo $a[0];
            ?>">
                <button type="button" class="close" data-dismiss="alert">Ã-</button>
                <?php
                    $a = explode('-', isset($alert) ? $alert : '');
                    echo $a[1];
                ?>
            </div>
        <?php } ?>
        <h2 class="form-signin-heading">Acesso Restrito</h2>

        <label for="inputEmail" class="sr-only">Email</label>
        <input type="email" class="form-control" name="email" placeholder="Email" required autofocus>
        <label for="inputPassword" class="sr-only">Senha</label>
        <input type="password" class="form-control" placeholder="Senha" name="senha" required>
        <button class="btn btn-lg btn-default btn-block" type="submit">Acessar</button>
    </form>
</div>
```

# Ambiente Administrativo - Acesso

- ▶ Model “Acessomodel.php”:
  - ▶ Criar o arquivo “Acessomodel.php” dentro de application/models/admin
  - ▶ Colocar o código referente ao modelo de acesso.

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class AcessoModel extends CI_Model {

    public function login($email, $senha) {
        $this->db->where('email_usuario', $email);
        $this->db->where('senha_usuario', $senha);
        return $this->db->get('usuarios')->result();
    }

}
```

# Ambiente Administrativo - Acesso

- ▶ Controller “Acesso.php”:
  - ▶ Criar o arquivo “Acesso.php” dentro de application/controllers/admin
  - ▶ Colocar o código referente ao controller de acesso.
    - ▶ (Ver código disponível na plataforma)

# Ambiente Administrativo - Acesso

- ▶ Teste: <http://localhost/projetos/sistemamodelo/admin>
  - ▶ Page not found
- ▶ Não existe um controlador chamado “admin”.
- ▶ O método principal do controlador “Acesso.php” é login (\$alert = null).
- ▶ Será necessário alterar o arquivo “routes.php” que está em application/config/routes.php
  - ▶ `$route['admin'] = 'admin/aceso';`
  - ▶ `$route['admin/aceso'] = 'admin/aceso/login';`
  - ▶ `$route['admin/aceso/(:num)'] = 'admin/aceso/login/$1';`
- ▶ Salve o arquivo routes.php
- ▶ Novo teste: <http://localhost/projetos/sistemamodelo/admin>
  - ▶ Page not found: admin/painelview.php
  - ▶ Falta preparar as demais telas.

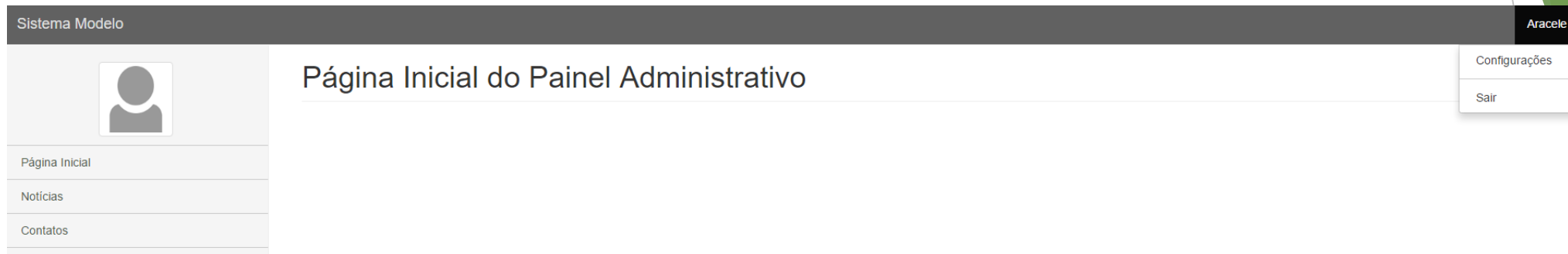
# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Ambiente Administrativo

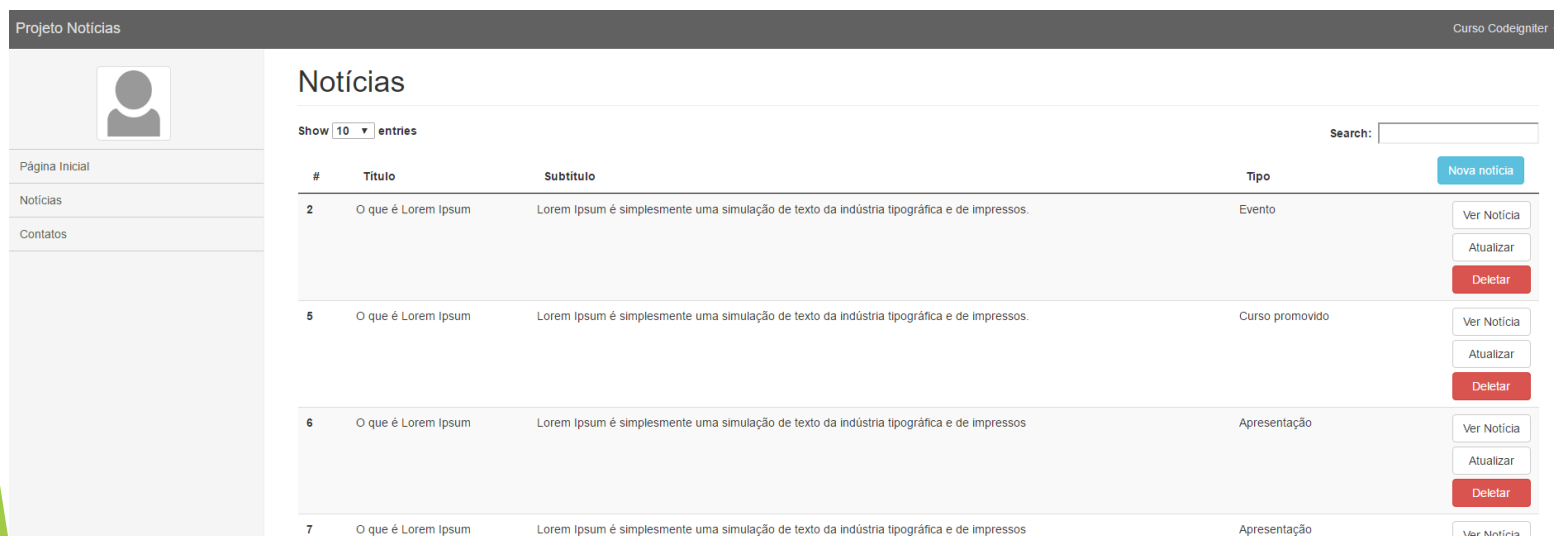
Preparando a base para as demais telas

# Ambiente Administrativo

## Preparando a Base para as demais telas



## Página Inicial do Ambiente Administrativo



## Listagem de Notícias

# Ambiente Administrativo

## Preparando a Base para as demais telas

- ▶ O template dashboard (Bootstrap) será dividido em 3 partes: topo, menu e rodapé.
- ▶ Assim, o “meio” será preenchido com os dados das demais telas:
  - ▶ Cadastro de notícias, listagem de notícias, etc
- ▶ Crie uma pasta chamada “includes” dentro de “application/views/admin”.
  - ▶ Coloque os arquivos (faça download da plataforma).
    - ▶ topo.php
    - ▶ menu.php
    - ▶ rodape.php
- ▶ Crie e/ou faça a adaptação para os seguintes arquivos:
  - atualizanoticiaview.php
  - configuracoesview.php
  - contatosview.php
  - listanoticiasview.php
  - novanoticiaview.php
  - painelview.php



# Ambiente Administrativo

## Preparando a Base para as demais telas

### ► CSS:

- A pasta “dist/css” deve conter os arquivos relacionados ao templates do Bootstrap que estamos utilizando:
  - Dashboard.css (usado no restante das telas do Painel Administrativo)
  - Signin.css (usado no acessoview.php)
- Adicionalmente, podemos criar um arquivo de estilo adicional, específico para uso em determinados pontos do projeto (paginação de dados, pesquisa na página, avatar, etc)
- Também podemos adicionar outros recursos que demandam css específicos. Todos devem ser colocados dentro de “dist/css”
- Neste momento, coloque os seguintes arquivos dentro de “dist/css”.
  - jquery.datatables.min.css
  - main.css

### ► JS (a mesma coisa pode ocorrer para arquivos Javascript)

- Coloque o arquivo “jquery.datatables.min.js” dentro de “dist/js”

# Ambiente Administrativo

## Preparando a Base para as demais telas

- Observe que esses arquivos css e js adicionais estão sendo usados no topo.php, menu.php e rodape.php, para estilizar a tabela de notícias, paginação, avatar e campo busca. São recursos adicionais não incluídos no template padrão do Bootstrap (dashboard).

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Sistema Modelo</title>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="../../favicon.ico">

    <link href="<?=base_url(); ?>dist/css/bootstrap.min.css" rel="stylesheet">
    <link href="<?=base_url(); ?>dist/css/main.css" rel="stylesheet">
    <link href="<?=base_url(); ?>dist/css/jquery.dataTables.min.css" rel="stylesheet">

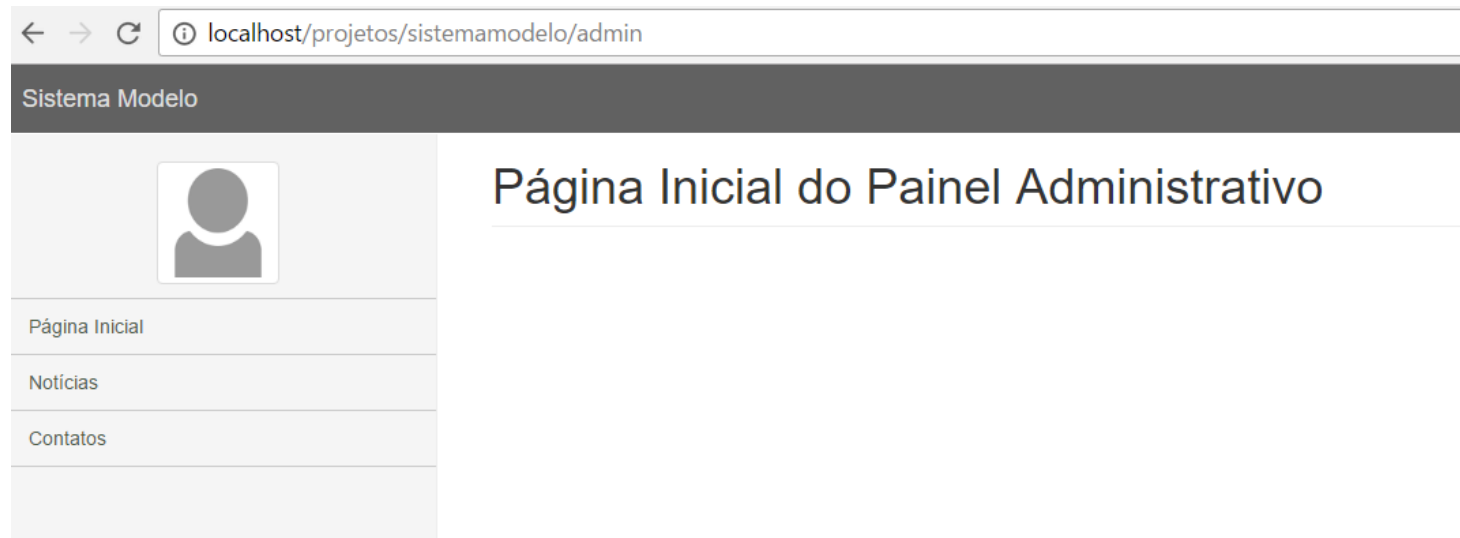
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
      <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body style="padding-top: 50px;">
```

topo.php

# Ambiente Administrativo

## Preparando a Base para as demais telas

- ▶ Se quiser, faça um novo teste agora:
  - ▶ Localhost/projetos/sistemamodelo/admin
- ▶ Entretanto, precisamos finalizar os controllers e os models das demais telas.



# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Ambiente Administrativo  
Criando controllers e models restantes

# Modelos

- ▶ Em .../apliation/models, crie o arquivo “model” correspondente:
  - ▶ adm
    - ▶ noticiasmodel.php
- ▶ O arquivo está disponível no ambiente do curso.

# Controles

- ▶ Em .../application/controllers/admin, crie os arquivos:
  - ▶ Noticias.php
  - ▶ Contato.php

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Ambiente Administrativo  
Testes e Configurações Adicionais

# Ambiente Administrativo - Teste Nova Notícia

- ▶ Tente criar uma nova notícia.
- ▶ Qual foi o resultado final?
  - ▶ <http://localhost/projetos/sistemamodelo/admin/noticias/1>
  - ▶ The page you requested was not found.
- ▶ Inclua no arquivo “routes.php”
  - ▶ `$route['admin/noticias/(:num)'] = 'admin/noticias/index/$1';`



# Ambiente Administrativo - Teste Ver Notícia

- ▶ Tente ver uma notícia:
- ▶ Qual foi o resultado final?
  - ▶ <http://localhost/projetos/sistemamodelo/usu/paginainicial/noticias/o-que-e-lorem.2>
  - ▶ The page you requested was not found.
- ▶ A tela para ver notícias está ligada à área de usuário.
- ▶ Ela ainda precisa ser criada.

# Ambiente Administrativo - Outros testes

- ▶ Faça novos testes:
  - ▶ Atualizar uma notícia.
  - ▶ Excluir uma notícia.
  - ▶ Contatos.
  - ▶ Configuração.
  - ▶ Sair.

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

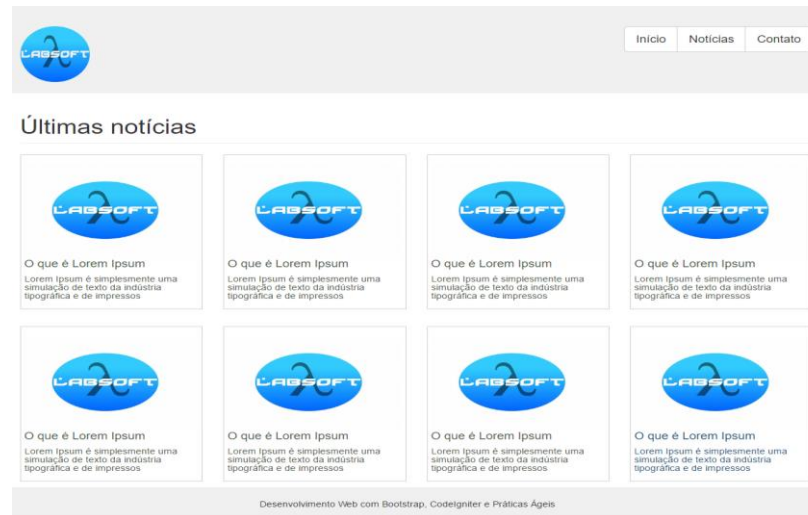
Ambiente de Usuário  
Telas, Controladores e Modelos

# Ambiente de Usuário - Telas

- ▶ O Ambiente de usuário é composto, em sua primeira versão, por 4 telas.
- ▶ Topo, menu e rodapé são comuns para ambas as telas.
- ▶ Exemplo:



Página Inicial



Menu Notícias - Listagem

# Ambiente de Usuário - Telas

- ▶ Criando as telas do Ambiente de Usuário:
  - ▶ Crie uma pasta “includes” dentro de “application/views/usu”.
  - ▶ Insira os 3 arquivos: menu.php, rodape.php e topo.php
    - ▶ Preparados para este curso.
    - ▶ Neste caso, não usamos template pronto, mas você pode usar um, sem problema!
  - ▶ Crie o “body” dos demais arquivos dentro de “application/views/usu”.
    - ▶ contatoview.php
    - ▶ listanoticiasview.php (lista todas as notícias)
    - ▶ listanoticiaview.php (exibe apenas a notícia selecionada)
    - ▶ paginainicialview.php

# Ambiente de Usuário - Controlador

- ▶ Criando o arquivo controlador do Ambiente de Usuário:
  - ▶ Insira o arquivo “Paginainicial.php” dentro de “application/controllers/usu”
- ▶ Observe que este arquivo faz uso deste model, para exibir notícias:
  - ▶ “application/models/admin/noticiasmodel.php”
  - ▶ Sendo assim, neste momento, não existe uma classe modelo associada ao controlador “usu/Paginainicial.php”



# Ambiente de Usuário - Teste Página Inicial

- ▶ Ao acessar <http://localhost/projetos/sistemamodelo/> queremos que o usuário acesse a página inicial:



- ▶ Precisamos informar isso no arquivo routes.php
- ▶ 1ª ideia: Poderíamos fazer `$route['default_controller'] = 'usu/paginainicial';`
- ▶ `$route['default_controller']` não admite subpasta no corpo da regra.
- ▶ Alternativa: estender o arquivo routes.php

## Ambiente de Usuário - Estendendo routes.php

- ▶ Precisamos estender o arquivo “routes.php”, ou seja, acrescentar novas funcionalidades, para que ele permita o direcionamento para uma rota que contenha subpastas.
- ▶ O CodeIgniter permite estender uma classe:
  - ▶ [https://www.codeigniter.com/userguide3/general/core\\_classes.html?highlight=core](https://www.codeigniter.com/userguide3/general/core_classes.html?highlight=core)
- ▶ Para resolver a nossa questão:
  - ▶ coloque o arquivo “MY\_router.php” dentro de “application/core”.
- ▶ Faça um novo teste:
  - ▶ <http://localhost/projetos/sistemamodelo/>



## Ambiente de Usuário - Testes Adicionais

- ▶ Clique em “Mais Notícias”.
- ▶ Veja a URL correspondente: <http://localhost/projetos/sistemamodelo/noticias>
  - ▶ The page you requested was not found.
- ▶ Precisamos informar ao arquivo “routes.php” o que deve acontecer caso chegue uma URL “sistemamodelo/noticias”, já que o controlador do Ambiente Administrativo está dentro de “sistemamodelo/usu/paginainicial”.
- ▶ Acesse “routes.php” e acrescente:
  - ▶ `$route['noticias'] = 'usu/paginainicial/noticias';`
- ▶ Teste novamente.

# Ambiente de Usuário - Testes Adicionais

- ▶ Clique em “Mais Notícias” e selecione uma notícia qualquer.
  - ▶ Veja a URL Correspondente:  
<http://localhost/projetos/sistemamodelo/noticias/slughoje>
  - ▶ The page you requested was not found.
- ▶ Novamente, precisamos alterar o arquivo “routes.php”
  - ▶ Se chegar noticias/alguma coisa → encaminhe para o método correspondente.
  - ▶ `$route['noticias/(:any)'] = 'usu/paginainicial/noticias/$1';`
  - ▶ Salve o arquivo routes.php
- ▶ Faça um novo teste:
  - ▶ <http://localhost/projetos/sistemamodelo>
  - ▶ Mais notícias
  - ▶ Selecione uma notícia.

# Ambiente de Usuário - Testes Adicionais

- ▶ Agora clique no menu “Contato”.



- ▶ URL correspondente: <http://localhost/projetos/sistemamodelo/contato>
  - ▶ Page not found
- ▶ Altere o arquivo routes.php
  - ▶ `$route['contato'] = 'usu/paginainicial/contato';`
  - ▶ Salve o arquivo e repita o teste.

# Desenvolvimento Web com Bootstrap, CodeIgniter e Práticas Ágeis

Ajustes Adicionais - Paginação em Português

# Ajustes adicionais - Paginação em Português

- Observe que a paginação e o campo de busca estão em inglês:



- Datatables é um plugin js para paginação de tabelas. Neste site <https://datatables.net/> pode-se encontrar várias informações relacionadas.
- O campo de busca, quantidade de registro por páginas e ordem de listagem são programados dentro do plugin. No entanto podemos atualizar algumas coisas para adaptar ao nosso projeto.

# Ajustes adicionais - Paginação em Português

- ▶ Abra o arquivo “application/views/admin/includes/rodape.php”.
- ▶ Vá até o código final neste arquivo “rodape.php”
- ▶ Onde está:

```
<script>
    $(document).ready(function () {
        $('#dataTable').DataTable();
    });
</script>
```

- ▶ Coloque o código disponível no arquivo:
  - ▶ script rodape admin paginacao em português.txt
- ▶ Acrescente os arquivos css e js
  - ▶ jquery.datatables.min.css (dist/css)
  - ▶ jquery.datatables.min.js (dist/js)
- ▶ O Código foi reusado de: <https://datatables.net/plug-ins/i18n/Portuguese-Brazil>

# Ajustes adicionais - Paginação em Português

► Veja o resultado:

The screenshot shows a web application interface for managing news items. The browser address bar indicates the URL is `localhost/projetos/sistemamodelo/admin/noticias`. The page title is "Sistema Modelo". The left sidebar contains a user profile icon and a menu with "Página Inicial", "Notícias", and "Contatos". The main content area is titled "Notícias". Below the title, there is a dropdown menu showing "10 resultados por Pagina". To the right of the dropdown is a search bar labeled "Pesquisar" with the number "5" entered. Below the search bar is a button labeled "Nova noticia". The main content area displays a table with the following columns: "#", "Título", "Subtítulo", and "Tipo". The table contains one row with the following data: "# 5", "Título O que é Lorem Ipsum", "Subtítulo Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.", and "Tipo Curso promovido". To the right of the table, there are three buttons: "Ver Noticia", "Atualizar", and "Deletar". Below the table, there is a pagination control showing "Mostrando de 1 ate 1 de 1 registros (Filtrados de 13 registros)". The pagination control includes buttons for "Anterior", "1", and "Proximo".

Notícias

10 resultados por Pagina

Pesquisar 5

Nova noticia

#	Título	Subtítulo	Tipo
5	O que é Lorem Ipsum	Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.	Curso promovido

Ver Noticia

Atualizar

Deletar

Mostrando de 1 ate 1 de 1 registros (Filtrados de 13 registros)

Anterior 1 Proximo