

COM222

DESENVOLVIMENTO DE SISTEMAS WEB

Aula - Framework php

Conteúdo da aula

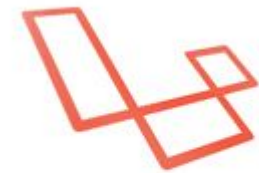
1. O que é um framework?
 - a. PHP FIG e as PSRs
2. CodeIgniter
 - a. Instalação
 - b. Estrutura de Arquivos
 - c. Configurações
 - d. Rotas
 - e. Controle de acesso
 - f. Banco de Dados
3. Exercício

1) O que é um framework?

“União de diversos artefatos de uso comum para atingir algum objetivo.”

No caso para programação web com php podemos citar artefatos como: AutoLoader, direcionamento de rotas, acesso ao banco de dados e validador de formulários.

1) O que é um framework?



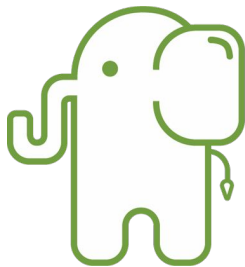
laravel



CodeIgniter



Symfony



Slim Framework

1) PHP FIG e as PSRs

O PHP FIG é um grupo que estabelece padrões para os frameworks de php, com a intenção de padronizar e estabelecer o conteúdo ideal.

<http://www.php-fig.org/>

Isso vai desde como o código deve ser escrito até as ferramentas necessárias.

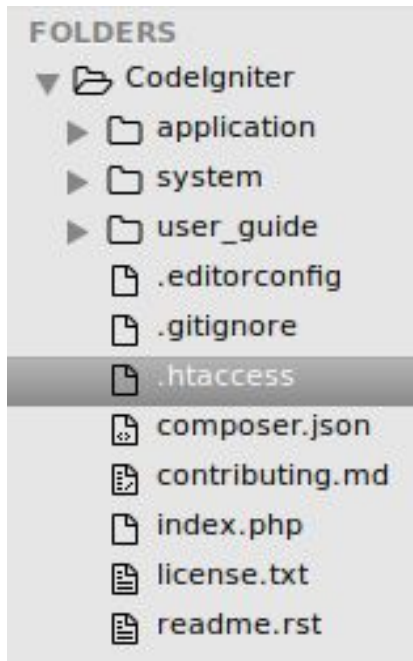
As PSR(PHP Standard Recommendation) são os padrões recomendados.

2) CodeIgniter : Instalação

Faça o download da versão mais nova no site oficial:

<https://www.codeigniter.com/>

extraia os arquivos para a pasta do projeto.



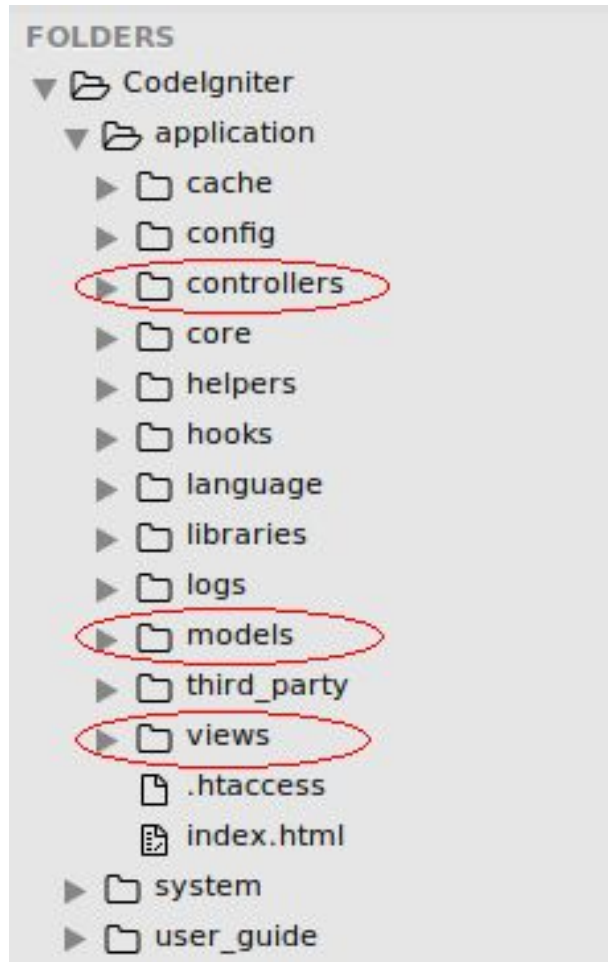
Crie um .htaccess com módulo de reescrita, no diretório raiz da aplicação.

2) CodeIgniter : Estrutura de Arquivos



Nossa aplicação ira ser desenvolvida dentro da pasta **application** ...

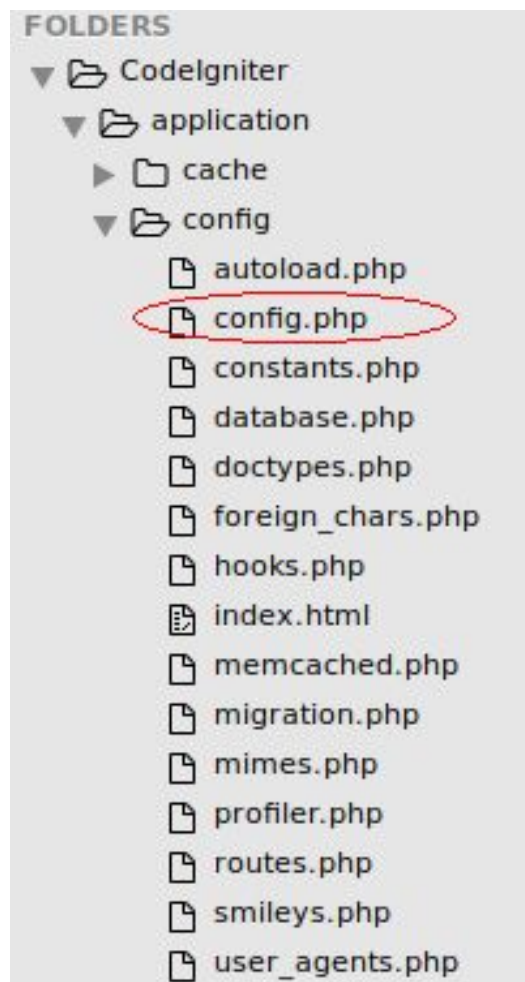
2) CodeIgniter : Estrutura de Arquivos



Nossa aplicação ira ser desenvolvida dentro da pasta **application**.

O CodeIgniter já da suporte nativo para uma aplicação estruturada no padrão MVC. Basta criar os controllers, views e models em suas respectivas pastas

2) CodeIgniter : Configuração

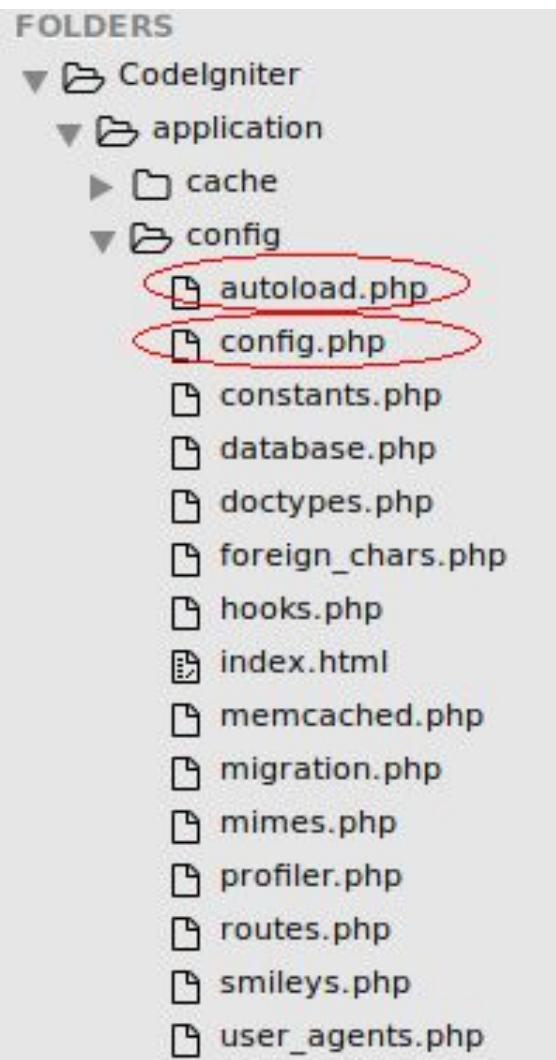


Para configurar algumas coisas do CI abra o arquivo config.php.

```
$config['base_url'] = 'sua url';
```

Essa alteração ira indicar ao framework a url base de nosso site.

2) CodeIgniter : Configuração

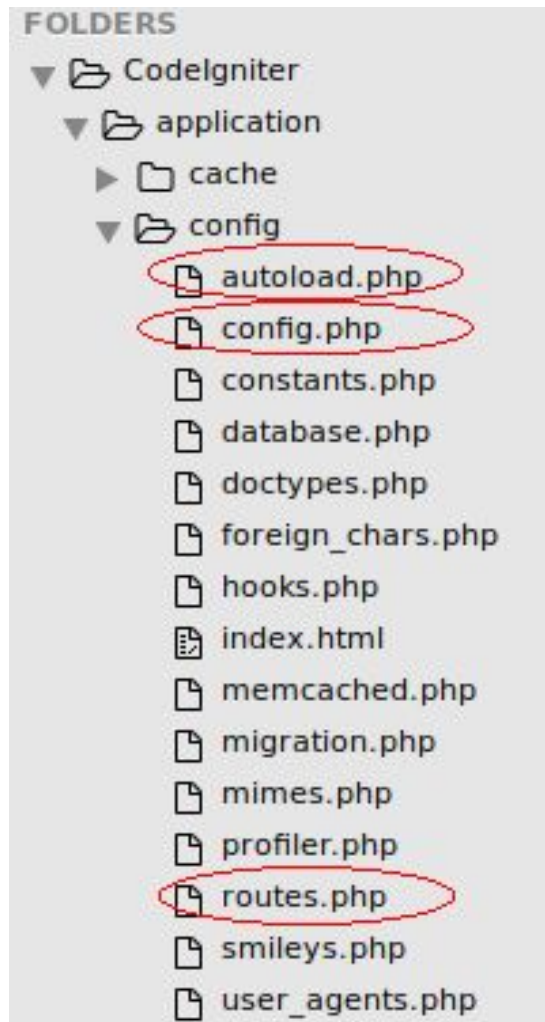


O arquivo autoload carrega automaticamente as bibliotecas e helpers

```
$autoload['helper'] = array('url');
```

Vamos carregar o helper url pois usaremos muito em nossa aplicação.

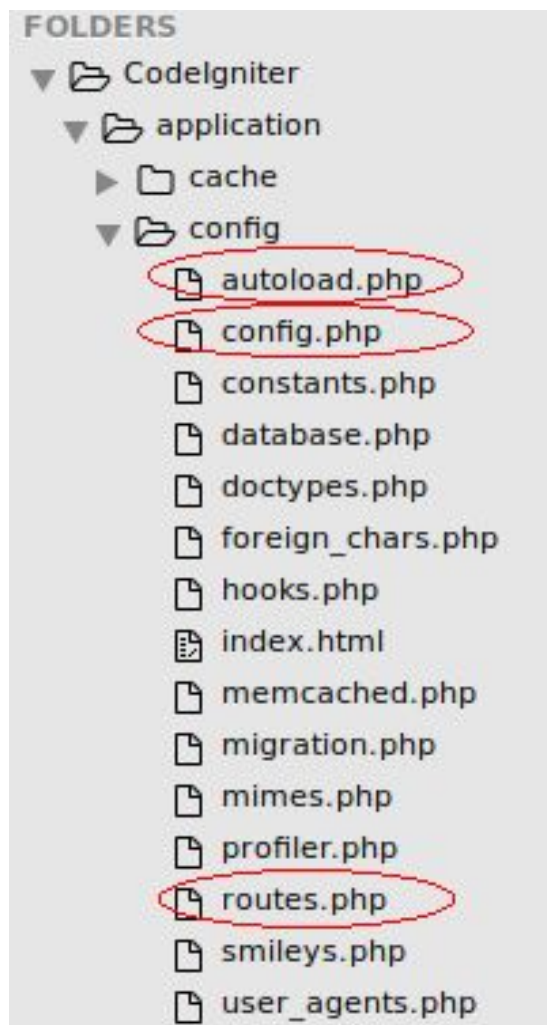
2) CodeIgniter : Rotas



As configurações de rotas ficam no arquivo routes.php. O CI define 3 rotas reservadas:

- default_controller
 - O controller 'inicial'
- 404_override
 - Rotas não definidas
- translate_uri_dashes
 - Troca '-' por '_' se TRUE

2) CodeIgniter : Rotas



Vamos criar um rota nova:

```
$route['user-session']['GET'] = 'sessions';  
$route['user-session']['POST'] = 'sessions/set_auth';
```

Quando o usuário acessar essa rota iremos exibir uma página para iniciar uma sessão em nosso site.

Ele será direcionado ao controller welcome e o método session será chamado.

2) CodeIgniter : Rotas

Para que nossas rotas funcionem vamos criar o controller responsável:

```
class Sessions extends CI_Controller {  
  
    public function __construct()  
    {  
        parent::__construct();  
        $this->load->library('session');  
    }  
}
```

Para facilitar edite a pagina inicial para o seguinte:

`<a href="<?=site_url('user-session')?>">página de sessão`

2) CodeIgniter : Rotas

```
public function index()
{
    $args = array();

    if($this->session->userdata('auth') === NULL) {
        $args['has_session'] = 'iniciar sessão';
    } else {
        $args['has_session'] = 'finalizar sessão';
    }

    $this->load->view('session_page',$args);
}
```

2) CodeIgniter : Rotas

```
public function set_auth() {  
    if($this->session->userdata('auth') == NULL) {  
        $this->session->set_userdata(['auth'=>true]);  
    } else {  
        $this->session->unset_userdata('auth');  
        $this->session->sess_destroy();  
    }  
  
    $this->index();  
}  
}
```

2) CodeIgniter : Rotas

Por fim crie a view session_page.php que será carregada pelo controller

```
<!DOCTYPE html>
<html>
<head>
  <title>Controle de sessão</title>
  <meta charset="utf-8">
</head>
<body>
  <form method="POST" action="<?=current_url()?>">
    <input type="submit" value="<?=$has_session?>">
  </form>
  <a href="<?=site_url()?>">pagina inicial</a>
</body>
</html>
```


2) CodeIgniter : Controle de Acesso

Agora que podemos saber quem possui ou não uma sessão, vamos usar isso para 'proteger' algumas rotas. Para isso usaremos um controller que só poderá ser acessado se **auth**(em session) estiver definida.

```
$route['produtos/(:num)'] = 'produtos/index/$1';
```

Crie um controller Produtos.php

2) CodeIgniter : Controle de Acesso

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

class Produtos extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->library('session');
        if($this->session->userdata('auth') === NULL) {
            redirect('user-session','GET');
        }
    }
}
```

Agora se um usuário sem sessão tentar acessar essa rota ele irá ser direcionado para a pagina de sessão

2) CodeIgniter : Banco de Dados

Vamos começar a usar uma base de dados. Crie um BD chamado 'CI' e a tabela:

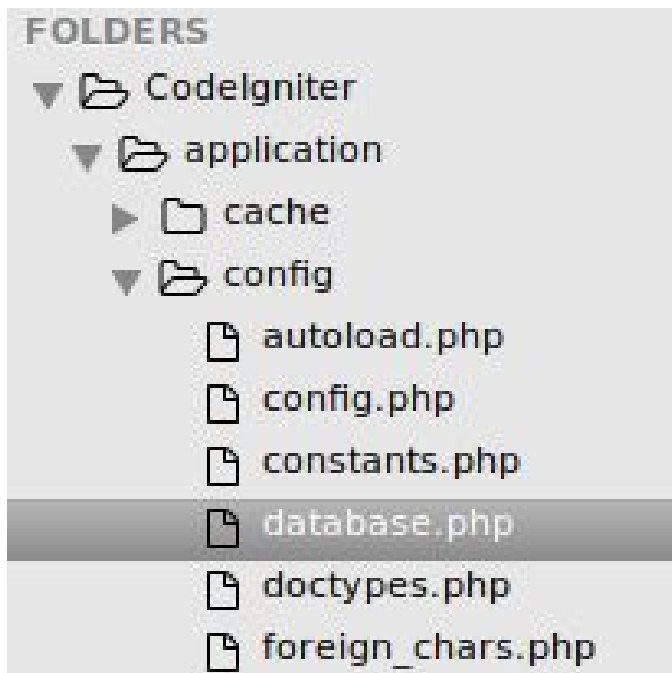
```
CREATE TABLE produtos(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(30) NOT NULL)
```

Coloque algumas tuplas:

```
INSERT INTO produtos(nome) VALUES  
( 'pao'), ('leite'), ('arroz'), ('batata'), ('cafe');
```

2) CodeIgniter : Banco de Dados

Vamos configurar o carregamento do banco.



username=>'root'
password=>''
database=>'CI'

Adicione o link na página inicial: <a href="<?=site_url('produtos')?>">Produtos

2) CodeIgniter : Banco de Dados

Adicione as seguintes linhas no construtor de Produtos:

```
$this->load->model('produto');  
$this->load->database();
```

Se o usuário possuir uma sessão válida o controller irá carregar o model e a conexão com o banco.

Adicione o link na página inicial:

```
<a href="<?=site_url('produtos')?>">Produtos</a>
```

2) CodeIgniter : Banco de Dados

Para facilitar a interação com o banco criaremos uma model:

```
class Produto extends CI_Model {  
    public $id;  
    public $nome;  
}
```

Esse model não irá possuir métodos pois o CI pode preencher todos os valores de acordo com a query executada

2) CodeIgniter : Banco de Dados

Agora vamos a função index de Produtos:

```
public function index($id = NULL)
{
    $args = array();
    if($id === NULL){
        $query = $this->db->get('produtos');

        $args['produtos'] =
        $query->custom_result_object('produto');
        $args['rows'] = $query->num_rows();
    } else {
```

2) CodeIgniter : Banco de Dados

```
$query = $this->db->get_where('produtos',['id' => $id]);

$args['produtos'] =
$query->custom_result_object('produto');
$args['rows'] = $query->num_rows();
}

$this->load->view('produtos_page',$args);
}
```


2) CodeIgniter : Banco de Dados

Criaremos também uma view para mostrar os resultados em uma tabela:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<!DOCTYPE html>
<head>
    <meta charset="utf-8">
    <title>Produtos</title>
</head>
<body>
<?php if($rows > 0): ?>
<table>
```

2) CodeIgniter : Banco de Dados

```
<tr>
    <th>ID</th>
    <th>Nome</th>
</tr>
<?php
foreach ($produtos as $produto) {?>
    <tr>
        <td><?=$produto->id ?></td>
        <td><?=$produto->nome ?></td>
    </tr>

    <?php
    }
    ?>
</table>
```

2) CodeIgniter : Banco de Dados

```
<?php else: ?>
    <h5>Não há produto com o id pedido</h5>
<?php endif;?>

<a href="<?=site_url()?>">pagina inicial</a>

</body>
</html>
```

Com essa url podemos carregar produtos pelo ID também, vamos fazer alguns testes.

2) CodeIgniter : Banco de Dados

Agora iremos criar uma página para cadastrar novos produtos:

```
$route['produtos/register']['GET'] = 'produtos/register_page';  
$route['produtos/register']['POST'] = 'produtos/register';
```

Vamos criar os métodos agora:

```
public function register_page()  
{  
    $this->load->view('register_page');  
}
```

2) CodeIgniter : Banco de Dados

```
public function register()  
{  
    $produto = new produto;  
    $produto->nome = $this->input->post('nomeProduto');  
  
    $result = $this->db->insert('produtos', $produto);  
    redirect('produtos','GET');  
}
```

O CI irá mapear os valores para o produto através do objeto produto, e inserir no banco.

2) CodeIgniter : Banco de Dados

```
<!-- register_page.php -->
<!DOCTYPE html>
<head>
    <meta charset="utf-8">
    <title>Produtos - Registrar</title>
</head>
<body>
<form method="POST" action="<?=current_url()?>">
    Nome:<input type="text" name="nomeProduto">
    <input type="submit">
</form>

<a href="<?=site_url()?>">pagina inicial</a>

</body>
</html>
```

3) Exercício

1. Adicione o campo quantidade na tabela de produtos.
2. Termine as operações de CRUD para os produtos,
 - a. O produto só deve ser deletado quando sua quantidade for zero, ou seja a operação de DELETE deve ser executada caso um UPDATE faça a quantidade ficar em zero.

3) Exercício - 1 semana

3. Implemente um sistema de login para substituir a criação de sessão pelo usuário, ou seja, crie a sessão se o login for válido. Não se esqueça do registro do usuário

DICA: Use a tabela do exercício da aula de rotas para cadastro de usuário

3) Exercício - 1 semana

Cabeçalho do email: Exercicio CI

Corpo: Matricula - Nome

Arquivo: envie um .zip!