

DESENVOLVIMENTO PARA WEB II

Modelando o SIGlg



Professor: Alexandre Strapção Guedes Vianna – IFPE Igarassu
alexandrevianna.net

SIGIG

- Sistema Integrado de Gestão Igarassu
 - Gestão de processos internos do campus igarassu
 - Solicitação de Reserva de Veículo
 - Solicitação de Impressão
 - Envio dos Comprovantes de Plano de Saúde

SIGIG

- Papeis (role) de Usuário
 - Servidor
 - Aluno
 - Gestor de Transportes
 - Gestor de Impressão
 - Gestor de Pessoas
 - Administrador

SIGIG

- Modelando a Entidade Usuário
 - Atributos:
 - id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
 - name VARCHAR(150)
 - matricula VARCHAR(11) UNIQUE
 - email VARCHAR(30)
 - username VARCHAR(50)
 - password VARCHAR(255)
 - role VARCHAR(20)
 - created DATETIME DEFAULT NULL
 - modified DATETIME DEFAULT NULL

SIGIG

1º Gere passo crie um projeto

```
$ composer create-project --prefer-dist cakephp/app sigig
```

2º Crie um banco de dados para o projeto

```
$ mysql -u root -p
```

```
mysql> create database sigig_db;
```



3º Configure as credenciais do banco no arquivo **config/app.php**

SIGIG

Criando a tabela users

```
$ mysql -u root -p
```

```
mysql> use sigig_db;
```

3º Rode o código SQL de criação da tabela users

SIGIG

- Código SQL:

```
CREATE TABLE users (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(150),  
    matricula VARCHAR(14),  
    email VARCHAR(30),  
    username VARCHAR(50),  
    password VARCHAR(255),  
    role VARCHAR(200),  
    created DATETIME DEFAULT NULL,  
    modified DATETIME DEFAULT NULL  
);
```

SIGIG

Gerando o esqueleto de users com o bake

```
$ bin/cake bake all Users
```


SIGIG -

Vamos alterar a validação do cadastro de usuários, acesse `src/Model/Table/UsersTable.php` e modifique a função `validationDefault`

- É necessário que o papel seja um dos papéis do nosso sistema: author ou admin

```
public function validationDefault(Validator $validator)
{
    ...
    $validator
        ->requirePresence('role', 'create')
        ->notEmpty('role')
        ->add('role', 'inList', [
            'rule' => ['inList', ['admin', 'gestor_transportes', 'gestor_impressao', 'gestor_pessoas', 'servidor']],
            'message' => 'Please enter a valid role']);
    return $validator;
}
```

SIGIG -

Acesse [src/Template/Users/add.ctp](#) e altere o campo de entrada do role (papel)

```
echo $this->Form->control('role');
```

para que fique desta forma:

```
echo $this->Form->control('role', [  
    'options' => ['admin' => 'Administrador', 'gestor_pessoas' =>  
'Gestor de Pessoas', 'gestor_transportes' => 'Gestor de Transportes',  
'gestor_impressao' => 'Gestor de Impressao'] ]);
```

O Cake irá colocar um componente para seleccionar entre os papeis disponíveis.

SIGIG -

Adicionando a camada de autenticação

No CakePHP a autenticação e a autorização são gerenciadas pelo componente `vendo\cakephp\cakephp\src\Controller\Component\AuthComponent`

Para incluir o este componente na tua aplicação altere o arquivo `src/Controller/AppController.php` conforme as instruções a seguir....

Autenticação (Login e Logout)

Adicione o carregamento do componente de autenticação na função `initialize()` [src/Controller/AppController.php](#)

```
public function initialize()
{
    parent::initialize();
    $this->loadComponent('RequestHandler');
    $this->loadComponent('Flash');
    $this->loadComponent('Auth', [
        'loginRedirect' => [
            'controller' => 'Users',
            'action' => 'index'
        ],
        'logoutRedirect' => [
            'controller' => 'Pages',
            'action' => 'display',
            'home'
        ]
    ]);
}
```

Ao logar, redirecionar para a listagem de usuarios

Ao sair, redirecionar para a home

Autenticação (Login e Logout)

Agora vamos modificar o arquivo para adicionar a função `beforeFilter`, e as actions de login e logout [src/Controller/UsersController.php](#)

Comece incluindo o uso da classe `Event` no início do arquivo:

```
use Cake\Event\Event;
```

Adicione a função `beforeFilter` ao `UserController`

```
public function beforeFilter(Event $event)
{
    parent::beforeFilter($event);
    $this->Auth->allow(['add', 'login', 'logout',]);
}
```

Autenticação (Login e Logout)

Action login em src/Controller/UsersController.php

Adicione a função login ao UsersController

```
public function login()
{
    if ($this->request->is('post')) {
        $user = $this->Auth->identify();
        if ($user) {
            $this->Auth->setUser($user);
            return $this->redirect($this->Auth->redirectUrl());
        }
        $this->Flash->error(__('Invalid username or password, try
again'));
    }
}
```

Autenticação (Login e Logout)

Action logout em src/Controller/UsersController.php

Adicione a função logout ao UsersController

```
public function logout()  
{  
    return $this->redirect($this->Auth->logout());  
}
```

Hash da Senha

O nosso password ainda esta sendo armazenado de forma bruta, vamos realizar um hash da senha em [src/Model/Entity/User.php](#)

Antes inclua o uso classe `Event` e `DefaultPasswordHasher` no início do arquivo:

```
use Cake\Auth\DefaultPasswordHasher;
```

Depois inclua a função

```
protected function _setPassword($password)
{
    return (new DefaultPasswordHasher)->hash($password);
}
```


Autenticação (Login e Logout)

Criando uma tela de login

Crie o arquivo login.ctp em src/Template/Users/login.ctp

Inclua o conteúdo:

```
<div class="users form">
<?= $this->Flash->render('auth') ?>
<?= $this->Form->create() ?>
    <fieldset>
        <legend><?= __('Please enter your username and password') ?
    ></legend>
        <?= $this->Form->control('username') ?>
        <?= $this->Form->control('password') ?>
    </fieldset>
<?= $this->Form->button(__('Login')); ?>
<?= $this->Form->end() ?>
</div>
```

Autorização

Ao arquivo src/Controller/AppController.php adicione a função `isAuthorized`

```
public function isAuthorized($user)
{
    // Admin can access every action
    if (isset($user['role']) && $user['role'] === 'admin') {
        return true;
    }

    // Default deny
    return false;
}
```

Esta função retorna TRUE se uma determinada ação é autorizada e FALSE caso contrário

O usuário admin pode realizar qualquer action em qualquer controller

Demais casos FALSE, ou seja, negue acesso

Teste o seu sistema de login

Agora realize teste na sua aplicação!!

- Crie um usuário em `/users/add`
- Faça login `/users/login`
- Teste as actions add, view, edit e delete
- Verifique que como os usuários estão sendo salvos no banco
- Verifique se o password esta sendo armazenado com hash

SIGIG

- Modelando a Entidade Solitação Impressão
 - Atributos:
 - id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
 - Solicitante INT UNSIGNED – Chave estrangeira com id users
 - Arquivo INT(11) – Chave estrangeira com id files
 - num_copias INT(3)
 - frente_verso BOOLEAN
 - observacao TEXT
 - status ENUM('pendente', 'impressa', 'retornada')
 - retorno TEXT
 - created DATETIME DEFAULT NULL
 - modified DATETIME DEFAULT NULL

SIGIG

- Antes de criar a tabela solicitação de impressão precisamos criar a tabela files Código SQL:

```
CREATE TABLE `files` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `path` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `created` datetime NOT NULL,  
  `modified` datetime NOT NULL,  
  `status` tinyint(1) NOT NULL DEFAULT '1' COMMENT '1 = Active, 0 =  
Inactive',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

<http://www.codexworld.com/handling-file-uploads-cakephp/>

SIGIG

- Código SQL:

```
CREATE TABLE impressions (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    solicitante INT UNSIGNED,  
    arquivo INT(11),  
    num_copias INT(3),  
    frente_verso BOOLEAN,  
    observacao TEXT,  
    status ENUM('pendente', 'impressa', 'retornada'),  
    retorno TEXT,  
    created DATETIME DEFAULT NULL,  
    modified DATETIME DEFAULT NULL,  
    FOREIGN KEY (solicitante) REFERENCES users(id),  
    FOREIGN KEY (arquivo) REFERENCES files(id)  
);
```

SIGIG

Gerando o model e controller de files com o bake

```
$ bin/cake bake model Files  
$ bin/cake bake controller Files
```

Files não tem view

Gerando o esqueleto de impressions com o bake

```
$ bin/cake bake all Impressions
```