



# Oficina - Framework CakePHP

## Desenvolvimento WEB

**Alunos:** Bruno Silveira  
Eduardo Matheus  
Gabriel Victor  
Janaína Araújo  
Kelly Jathay  
Leonardo Soares

**Data:** 18/06/2019



# O que será abordado?

1. Sobre o CakePHP
2. Algumas das principais características
3. Tecnologias usadas
4. Downloads
5. Instalação e configuração
6. Sobre o Projeto (Mão na massa!)
7. Códigos do projeto
8. Referências



# 1. Sobre o CakePHP

- Criado em 2005 pela Cake Software Foundation.
- É um software de código livre, podendo ser baixada livremente no GitHub.
- Também, por ser de código livre, pode ser mantido por pessoas de todo o mundo.
- Sobrevive de doações feitas por mantenedores.
- O principal objetivo é permitir que o desenvolvedor trabalhe de forma estruturada e rápida sem perder a flexibilidade.



# 1. Sobre o CakePHP (Cont.)

- Algumas das marcas que usam o CakePHP





## 2. Algumas das principais características

- Licença flexível.
- Compatibilidade com PHP 5.2.6 e superior.
- Geração de CRUD para interação com banco de dados;
- Arquitetura MVC.
- Validações embutidas.
- Templates rápidos e flexíveis.
- Ferramentas que auxiliam gerar JavaScript, AJAX, Forms HTML, etc.
- Comunidade ativa.
- Funciona em qualquer subdiretório web, com pouca configuração Apache.



### 3. Tecnologias usadas

- Como o nome sugere, ele é feito na linguagem PHP, assim como o Laravel, CodeIgniter, Yii, etc. (usaremos o PHP 7).
- O Cake Está atualmente na versão 3.7 (PHP 7).
- Utiliza-se da arquitetura de interface MVC.
- Para o projeto, iremos utilizar o XAMPP, pois já contém o Apache e MySQL.
- Editor de texto de sua preferência (aqui iremos usar o Sublime Text 3).
- Usa o gerenciador de dependências Composer.
- Para o projeto, iremos utilizar as ferramentas de Bootstrap e jQuery.



## 4. Downloads

- Para o S.O. Windows x64:

1. **MySQL:** <https://dev.mysql.com/downloads/installer/>

2. **PHP:** <https://windows.php.net/download#php-7.3>

3. **Composer:** <https://getcomposer.org/download/>

4. **Servidor HTTP:**

<https://httpd.apache.org/docs/current/platform/windows.html#down>

5. **jQuery:** <https://code.jquery.com/jquery-3.4.1.min.js>

6. **Bootstrap:** <https://getbootstrap.com/docs/4.0/getting-started/download/>

7. **Sublime:** <https://www.sublimetext.com/3>



## 5. Instalação e configuração

- O CakePHP requer um servidor HTTP, como o Apache
- PHP 5.6.0 ou acima deste (vamos utilizar o PHP 7)
- Extensão mbstring para o PHP
- Extensão intl para o PHP
- Extensão simplexml para o PHP
- Extensão PDO para o PHP
- MySQL, MariaDB, PostgreSQL, Microsoft SQL Server ou SQLite 3





## 5. Instalação e configuração (Cont.)

- Muita coisa? Relaxa que o XAMPP já possui todos estes recursos.





## 5. Instalação e configuração (Cont.)

- Antes, precisamos configurar a extensão intl para o CakePHP funcionar normalmente na sua máquina.
- Após instalado o XAMPP, abra-o e procure pela linha onde o Apache se encontra.
- Siga os botões e clique em 'Config' e logo após em 'PHP (php.ini)'
- Busque por "intl"
- Na linha ";extension=intl", remova o ';' e salve o arquivo.



## 5. Instalação e configuração (Cont.)

XAMPP Control Panel v3.2.3 [ Compiled: Mar 7th 2019 ]

**XAMPP Control Panel v3.2.3**

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin <b>Config</b> Logs
<input type="checkbox"/>	MySQL			Start Admin Config
<input type="checkbox"/>	FileZilla			Start Admin Config
<input type="checkbox"/>	Mercury			Start Admin Config
<input type="checkbox"/>	Tomcat			Start Admin Config

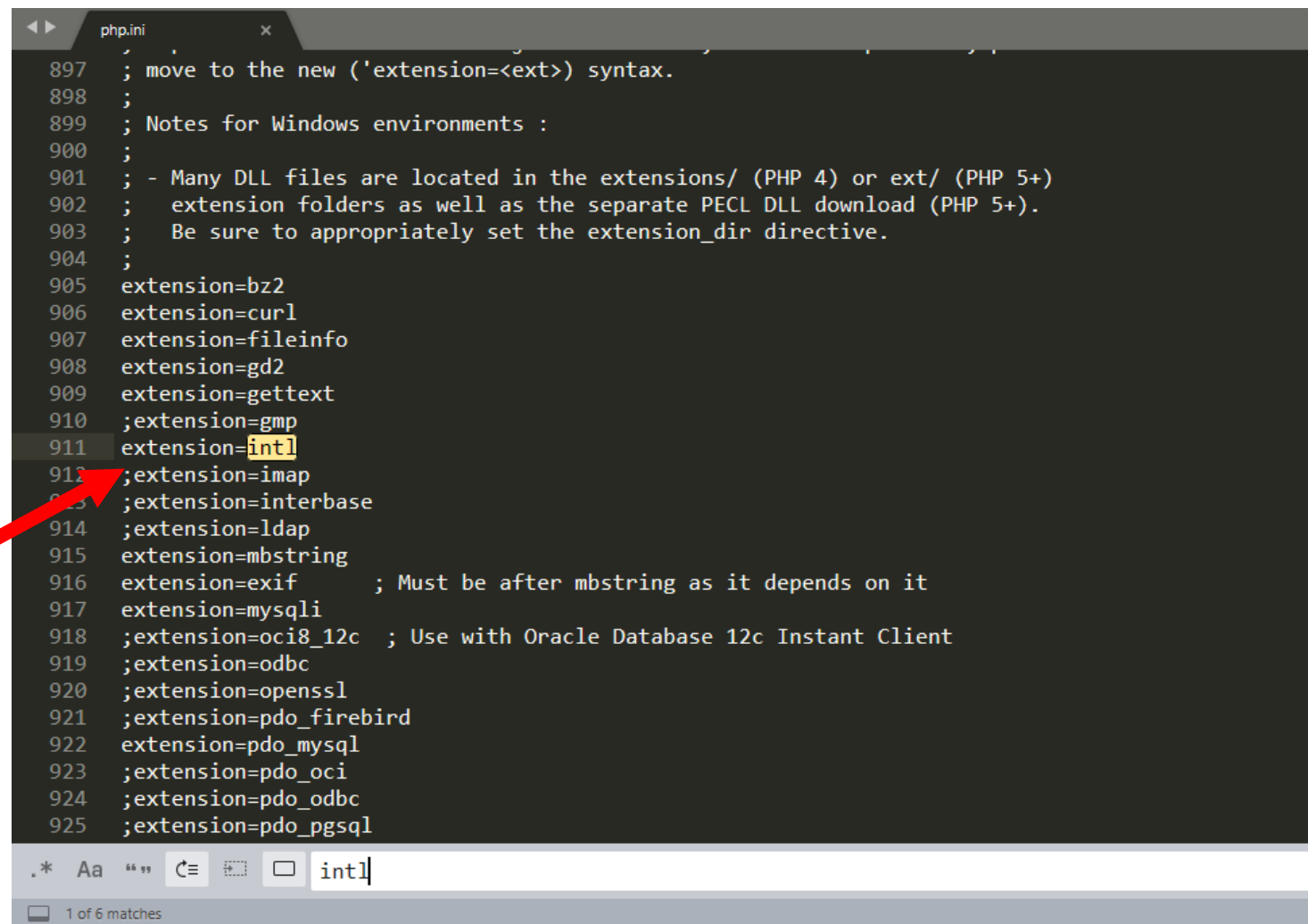
Config Netstat Shell

Apache (httpd.conf)  
Apache (httpd-ssl.conf)  
Apache (httpd-xampp.conf)  
PHP (php.ini)  
phpMyAdmin (config.inc.php)  
<Browse> [Apache]  
<Browse> [PHP]  
<Browse> [phpMyAdmin]

21:12:36 [main] there will be a security dialogue or things will break! So t  
21:12:36 [main] about running this application with administrator rights!  
21:12:36 [main] XAMPP Installation Directory: "c:\xampp\  
21:12:37 [main] Checking for prerequisites  
21:12:37 [main] All prerequisites found  
21:12:37 [main] Initializing Modules  
21:12:37 [main] Starting Check-Timer  
21:12:37 [main] Control Panel Ready



## 5. Instalação e configuração (Cont.)



```
php.ini
897 ; move to the new ('extension=<ext>') syntax.
898 ;
899 ; Notes for Windows environments :
900 ;
901 ; - Many DLL files are located in the extensions/ (PHP 4) or ext/ (PHP 5+)
902 ;   extension folders as well as the separate PECL DLL download (PHP 5+).
903 ;   Be sure to appropriately set the extension_dir directive.
904 ;
905 extension=bz2
906 extension=curl
907 extension=fileinfo
908 extension=gd2
909 extension=gettext
910 ;extension=gmp
911 extension=intl
912 ;extension=imap
913 ;extension=interbase
914 ;extension=ldap
915 extension=mbstring
916 extension=exif ; Must be after mbstring as it depends on it
917 extension=mysqli
918 ;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
919 ;extension=odbc
920 ;extension=openssl
921 ;extension=pdo_firebird
922 extension=pdo_mysql
923 ;extension=pdo_oci
924 ;extension=pdo_odbc
925 ;extension=pdo_pgsql
```

.\* Aa " " ☰ ☒ intl |

1 of 6 matches



## 5. Instalação e configuração (Cont.)

- Para facilitar, vamos utilizar o executável do PHP do XAMPP como base para os comandos a seguir.
- Esta aplicação, normalmente, localiza-se em C:\xampp\php



## 5. Instalação e configuração (Cont.)

- Feito isso, iremos entrar no Prompt de comando do Windows.
- Mude o caminho para a pasta HTDOCS do XAMPP com o comando `cd C:\xampp\htdocs`.
- Com o composer instalado, utilizaremos o comando `composer create-project --prefer-dist cakephp/app agenda_series`.



## 5. Instalação e configuração (Cont.)

```
OS Prompt de Comando - composer create-project --prefer-dist cakephp/app agenda_series

C:\xampp\htdocs>composer create-project --prefer-dist cakephp/app agenda_series
Installing cakephp/app (3.7.1)
- Installing cakephp/app (3.7.1): Loading from cache
Created project in agenda_series
Loading composer repositories with package information
Updating dependencies (including require-dev)
```



## 5. Instalação e configuração (Cont.)

- Quando o download estiver concluído, abra o XAMPP e ative o Apache e o MySQL.
- Feito isso, vá ao navegador de sua preferência e digite [http://localhost/agenda\\_series](http://localhost/agenda_series)
- A página inicial do CakePHP irá aparecer.





## 5. Instalação e configuração (Cont.)

The screenshot shows a web browser window with the address bar displaying 'localhost/agenda\_series/'. The page features the CakePHP logo and the text 'Welcome to CakePHP 3.7.8 Red Velvet. Build fast. Grow solid.' Below this, a yellow warning box states: 'Please be aware that this page will not be shown if you turn off debug mode unless you replace src/Template/Pages/home.ctp with your own version.' The page is divided into two columns: 'Environment' and 'Filesystem'. The 'Environment' column lists four green checkmarks indicating successful PHP configurations. The 'Filesystem' column lists three green checkmarks indicating successful directory and caching configurations.

**Environment**

- ✔ Your version of PHP is 5.6.0 or higher (detected 7.3.4).
- ✔ Your version of PHP has the mbstring extension loaded.
- ✔ Your version of PHP has the openssl extension loaded.
- ✔ Your version of PHP has the intl extension loaded.

**Filesystem**


- ✔ Your tmp directory is writable.
- ✔ Your logs directory is writable.
- ✔ The *FileEngine* is being used for core caching. To change the config edit config/app.php



## 5. Instalação e configuração (Cont.)

- Você poderá notar que esse alerta está aparecendo:

### Database

 CakePHP is NOT able to connect to the database.  
Connection to database could not be established: SQLSTATE[HY000]  
[1045] Access denied for user 'my\_app'@'localhost' (using password:  
YES)

- Isto ocorre pois o CakePHP ainda não foi conectado ao banco de dados MySQL da forma apropriada.



## 5. Instalação e configuração (Cont.)



- Primeiramente, iremos criar o banco de dados.
- Entre em: <http://localhost/phpmyadmin>
- Crie um novo banco de dados clicando em '+New' no menu lateral esquerdo.
- Crie um bando de dados chamado 'agenda\_series' com o esquema 'utf8-general-ci'



## 5. Instalação e configuração (Cont.)



### Base de Dados

 Criar base de dados 



## 5. Instalação e configuração (Cont.)

- Agora, abra o projeto no editor de textos e procure pelo arquivo `config/app.php`
- Encontre a chave `'Datasources'`, onde iremos fazer a conexão com o banco de dados que criamos no phpMyAdmin
- Coloque as credenciais para o banco de dados (o padrão é: `'username' => 'root'`, `'password' => ''`), e na chave interna `'database'`, coloque o nome do banco criado `'agenda_series'`



## 5. Instalação e configuração (Cont.)

```
app.php
244 * other RDBMS.
245 */
246 'Datasources' => [
247     'default' => [
248         'className' => 'Cake\Database\Connection',
249         'driver' => 'Cake\Database\Driver\Mysql',
250         'persistent' => false,
251         'host' => 'localhost',
252     ],
253     /*
254      * CakePHP will use the default DB port based on the driver selected
255      * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
256      * the following line and set the port accordingly
257      */
258     //'port' => 'non_standard_port_number',
259     'username' => 'root',
260     'password' => '',
261     'database' => 'agenda_series',
262     /*
263      * You do not need to set this flag to use full utf-8 encoding (internal default since CakePHP 3.6).
264      */
265     //'encoding' => 'utf8mb4',
266     'timezone' => 'UTC',
267     'flags' => [],
268     'cacheMetadata' => true,
269     'log' => false,
270 ]
```



## 5. Instalação e configuração (Cont.)

- Pronto! Atualize a página [http://localhost/agenda\\_series](http://localhost/agenda_series)
- O alerta estará mudado.

Database

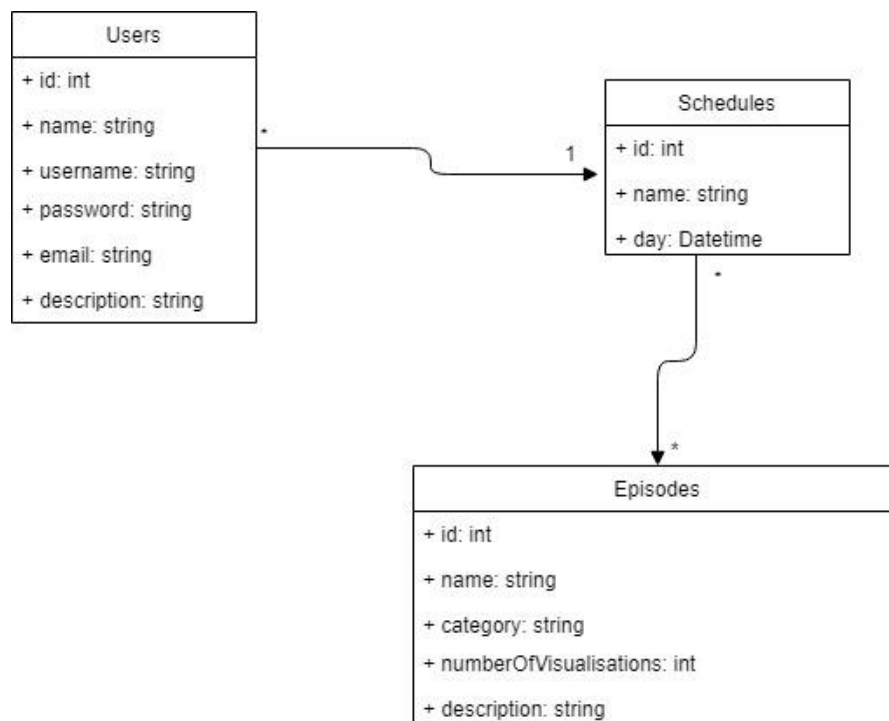


CakePHP is able to connect to the database.



## 6. Sobre o Projeto

- Iremos montar um simples calendário de episódios de séries
- Para isso, seguiremos o modelo a seguir:







## 7. Códigos do projeto

- **Model:** Usuário (UsersTable.php)

```
UsersTable.php x
1  <?php
2  namespace App\Model\Table;
3
4  use Cake\ORM\Table;
5
6  class UsersTable extends Table
7  {
8
9  }
```



## 7. Códigos do projeto (Cont.)

- **View: Usuário**  
(index.ctp)

```
index.ctp
1 <h1>
2   Usuários <?= $this->Html->link('Adicionar', ['action' => 'add'], ['class' => 'btn
      btn-primary btn-sm']) ?>
3 </h1>
4 <div class="table-responsive">
5   <table class="table table-striped">
6     <thead>
7       <tr>
8         <th scope="col">Nome</th>
9         <th scope="col">Usuário</th>
10        <th scope="col">Email</th>
11        <th scope="col"></th>
12      </tr>
13    </thead>
14    <tbody>
15      <?php
16        foreach ($lista as $item) { ?>
17        <tr>
18          <th><?= $this->Html->link($item->name, ['action'=>'edit', $item->id]) ?></th>
19          <td><?= $item->username ?></td>
20          <td><?= $item->email ?></td>
21          <td><?= $this->Form->postLink('Apagar', ['action' => 'delete', $item->id], ['confirm'
            => __('Esse registro será apagado. Deseja continuar?'), 'class' => 'btn btn-sm
            btn-danger']) ?></td>
22        </tr>
23      <?php } ?>
24    </tbody>
25  </table>
26 </div>
```



## 7. Códigos do projeto (Cont.)

- **View: Usuário**  
(add.ctp)

```
add.ctp
1 <h1>
2   Cadastro de usuários
3 </h1>
4 <hr/>
5 <?= $this->Form->create(null,['type'=>'POST'])?>
6 <div class="form-group">
7   <div class="col-12">
8     <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','
9       escape'=>false]) ?>
10   </div>
11   <div class="col-12">
12     <?= $this->Form->control('username',['label'=>'<b>Nome do usuário</b>','class'=>'
13       form-control','escape'=>false]) ?>
14   </div>
15   <div class="col-12">
16     <?= $this->Form->control('password',['label'=>'<b>Senha</b>','type'=>'password','
17       class'=>'form-control','escape'=>false]) ?>
18   </div>
19   <div class="col-12">
20     <?= $this->Form->control('email',['label'=>'<b>Email</b>','class'=>'form-control','
21       escape'=>false]) ?>
22   </div>
23   <div class="col-12">
24     <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'
25       form-control','escape'=>false]) ?>
26   </div>
27   <div class="col-12">
28     <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
29   </div>
30 </div>
31 <?= $this->Form->end() ?>
```



## 7. Códigos do projeto (Cont.)

- **View: Usuário**  
(edit.ctp)

```
edit.ctp
1 <h1>
2     Editar usuário
3 </h1>
4 <hr/>
5 <?= $this->Form->create($user,['type'=>'PUT'])?>
6 <div class="form-group">
7     <div class="col-12">
8         <?= $this->Form->control('name',['label'=><b>Nome</b>','class'=>'form-control','
          escape'=>false]) ?>
9     </div>
10    <div class="col-12">
11        <?= $this->Form->control('username',['label'=><b>Nome do usuário</b>','class'=>'
          form-control','escape'=>false]) ?>
12    </div>
13    <div class="col-12">
14        <?= $this->Form->control('password',['label'=><b>Senha</b>','type'=>'password','
          class'=>'form-control','escape'=>false]) ?>
15    </div>
16    <div class="col-12">
17        <?= $this->Form->control('email',['label'=><b>Email</b>','class'=>'form-control','
          escape'=>false]) ?>
18    </div>
19    <div class="col-12">
20        <?= $this->Form->control('description',['label'=><b>Descrição</b>','class'=>'
          form-control','escape'=>false]) ?>
21    </div>
22    <div class="col-12">
23        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
24    </div>
25 </div>
26 <?= $this->Form->end() ?>
```



## 7. Códigos do projeto (Cont.)

- **Controller:** Usuário  
(UsersController.ctp)

```
UsersController.php
1 <?php
2 namespace App\Controller;
3
4 class UsersController extends AppController{
5
6     public function index(){
7         $lista = $this->Users->find('all')
8                 ->order(['name'=>'ASC'])
9                 ->toArray();
10
11         $this->set(compact('lista'));
12     }
13
14     public function add(){
15         if ($this->request->is('POST')) {
16             $save = $this->Users->newEntity();
17             $save = $this->Users->patchEntity($save,$this->request->data);
18             if ($this->Users->save($save)) {
19                 $this->Flash->success('Registro completado com sucesso.');
```



## 7. Códigos do projeto (Cont.)

- **Controller: Usuário**  
(UsersController.ctp)

cont.

```
UsersController.php x
25
26 public function edit($id) {
27
28     $user = $this->Users->get($id);
29
30     if ($this->request->is('PUT')) {
31         $user = $this->Users->patchEntity($user, $this->request->data);
32         if ($this->Users->save($user)) {
33             $this->Flash->success('Registro editado com sucesso.');
```

```

34             return $this->redirect(['action' => 'index']);
35         }
36         $this->Flash->error('Um erro ocorreu.');
```

```

37     }
38
39     $this->set(compact('user'));
40 }
41
42 public function delete($id) {
43
44     $this->viewBuilder()->layout('ajax');
```

```

45     $this->request->allowMethod(['post', 'delete']);
46     $user = $this->Users->get($id);
47
48     if ($this->Users->delete($user)){
49         $this->Flash->success('Registro deletado com sucesso.');
```

```

50         return $this->redirect(['action' => 'index']);
51     }
52     $this->Flash->error('Um erro ocorreu.');
```

```

53 }
54 }
55 ?>
```



## 7. Códigos do projeto (Cont.)

- **Model: Series**  
(SeiresTable.php)

```
SeriesTable.php
1 <?php
2 namespace App\Model\Table;
3
4 use Cake\ORM\Query;
5 use Cake\ORM\RulesChecker;
6 use Cake\ORM\Table;
7 use Cake\Validation\Validator;
8
9 /**
10  * Series Model
11  *
12  * @method \App\Model\Entity\Series get($primaryKey, $options = [])
13  * @method \App\Model\Entity\Series newEntity($data = null, array $options = [])
14  * @method \App\Model\Entity\Series[] newEntities(array $data, array $options = [])
15  * @method \App\Model\Entity\Series|bool save(\Cake\Datasource\EntityInterface $entity,
16  * $options = [])
17  * @method \App\Model\Entity\Series saveOrFail(\Cake\Datasource\EntityInterface $entity,
18  * $options = [])
19  * @method \App\Model\Entity\Series patchEntity(\Cake\Datasource\EntityInterface $entity,
20  * array $data, array $options = [])
21  * @method \App\Model\Entity\Series[] patchEntities($entities, array $data, array $options =
22  * [])
23  * @method \App\Model\Entity\Series findOrCreate($search, callable $callback = null,
24  * $options = [])
25  */
26 class SeriesTable extends Table
27 {
28     /**
29      * Initialize method
30      *
31      * @param array $config The configuration for the Table.
32      * @return void
33      */
34 }
```



## 7. Códigos do projeto (Cont.)

- **Model: Series**  
(SeiresTable.php)

cont.

```
SeriesTable.php
29 | public function initialize(array $config)
30 | {
31 |     parent::initialize($config);
32 |
33 |     $this->setTable('series');
34 |     $this->setDisplayField('name');
35 |     $this->setPrimaryKey('id');
36 | }
37 |
38 | /**
39 |  * Default validation rules.
40 |  *
41 |  * @param \Cake\Validation\Validator $validator Validator instance.
42 |  * @return \Cake\Validation\Validator
43 |  */
44 | public function validationDefault(Validator $validator)
45 | {
46 |     $validator
47 |         ->integer('id')
48 |         ->allowEmptyString('id', 'create');
49 |
50 |     $validator
51 |         ->scalar('name')
52 |         ->maxLength('name', 255)
53 |         ->requirePresence('name', 'create')
54 |         ->allowEmptyString('name', false);
55 |
56 |     $validator
57 |         ->scalar('category')
58 |         ->maxLength('category', 255)
59 |         ->requirePresence('category', 'create')
60 |         ->allowEmptyString('category', false);
61 | }
```





## 7. Códigos do projeto (Cont.)

- **Model: Series**  
(SeiresTable.php)

cont.

```
61  
62 |     $validator  
63 |         ->scalar('description')  
64 |         ->maxLength('description', 255)  
65 |         ->requirePresence('description', 'create')  
66 |         ->allowEmptyString('description', false);  
67 |  
68 |         return $validator;  
69 |     }  
70 }  
71
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(index.ctp)

```
index.ctp
1 <?php
2 /**
3  * @var \App\View\AppView $this
4  * @var \App\Model\Entity\Series[]|\Cake\Collection\CollectionInterface $series
5  */
6 ?>
7
8 <h1>Minhas Séries <?= $this->Html->link('Adicionar Nova Série',['action' => 'add'],['class'
   => 'btn btn-primary btn-sm']) ?></h1>
9 <div class="table-responsive">
10     <table class="table table-striped">
11         <thead>
12             <tr>
13                 <th scope="col"><?= $this->Paginator->sort('name',['label'=>'Nome']) ?></th>
14                 <th scope="col"><?= $this->Paginator->sort('category',['label'=>'Categoria'])
                   ?></th>
15                 <th scope="col"><?= $this->Paginator->sort('number_of_visualizations',['label'
                   =>'Visualizações']) ?></th>
16                 <th scope="col"><?= $this->Paginator->sort('description',['label'=>'Descrição
                   ']) ?></th>
17                 <th scope="col" class="actions"></th>
18             </tr>
19         </thead>
20         <tbody>
21             <?php foreach ($series as $serie): ?>
22                 <tr>
23                     <td><?= $this->Html->link($serie->name, ['action' => 'edit', $serie->id]) ?>
                       </td>
24                     <td><?= h($serie->category) ?></td>
25                     <td><?= $this->Number->format($serie->number_of_visualizations) ?></td>
26                     <td><?= h($serie->description) ?></td>
27                     <td class="actions">
28                         <?= $this->Form->postLink(__('Delete'), ['action' => 'delete', $serie->id
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(index.ctp)

cont.

```
index.ctp
29         ], ['confirm' => __('Esse registro será apagado. Deseja continuar?'),
30         'class' => 'btn btn-sm btn-danger']] ?>
31     </td>
32 </tr>
33 <?php endforeach; ?>
34 </tbody>
35 </table>
36 </div>
37 <div class="paginator">
38     <ul class="pagination">
39         <?=$this->Paginator->first('<< ' . __('first')) ?>
40         <?=$this->Paginator->prev('< ' . __('previous')) ?>
41         <?=$this->Paginator->numbers() ?>
42         <?=$this->Paginator->next(__('next') . ' >') ?>
43         <?=$this->Paginator->last(__('last') . ' >>') ?>
44     </ul>
45     <p><?=$this->Paginator->counter(['format' => __('Page {{page}} of {{pages}}, showing
46         {{current}} record(s) out of {{count}} total')]) ?></p>
47 </div>
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(add.ctp)

```
add.ctp
1 |<?php
2 |/**
3 | * @var \App\View\AppView $this
4 | * @var \App\Model\Entity\Series $series
5 | */
6 |?>
7 |<h1>
8 |    Cadastro de séries
9 |</h1>
10 |<hr/>
11 |<div class="form-group">
12 |    <?= $this->Form->create($series,['type'=>'POST']) ?>
13 |    <div class="col-12">
14 |        <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','
15 |            escape'=>false]) ?>
16 |    </div>
17 |    <div class="col-12">
18 |        <?= $this->Form->control('category',['label'=>'<b>Categoria</b>','class'=>'
19 |            form-control','escape'=>false]) ?>
20 |    </div>
21 |    <div class="col-12">
22 |        <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'
23 |            form-control','escape'=>false]) ?>
24 |    </div>
25 |    <div class="col-12">
26 |        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
27 |    </div>
28 |</div>
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(edit.ctp)

```
edit.ctp
1
2 <h1>
3   Editar de série
4 </h1>
5 <hr/>
6 <div class="form-group">
7   <?= $this->Form->create($series) ?>
8   <div class="col-12">
9     <?= $this->Form->control('name', ['label'=>'<b>Nome</b>', 'class'=>'form-control', '
      escape'=>false]) ?>
10  </div>
11  <div class="col-12">
12    <?= $this->Form->control('category', ['label'=>'<b>Categoria</b>', 'class'=>'
      form-control', 'escape'=>false]) ?>
13  </div>
14  <div class="col-12">
15    <?= $this->Form->control('description', ['label'=>'<b>Descrição</b>', 'class'=>'
      form-control', 'escape'=>false]) ?>
16  </div>
17  <div class="col-12">
18    <?= $this->Form->button('Enviar', ['class'=>'btn btn-primary mt-3']) ?>
19  </div>
20  <?= $this->Form->end() ?>
21 </div>
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(view.ctp)

```
view.ctp
1 <?php
2 /**
3  * @var \App\View\AppView $this
4  * @var \App\Model\Entity\Series $series
5  */
6 ?>
7 <nav class="large-3 medium-4 columns" id="actions-sidebar">
8     <ul class="side-nav">
9         <li class="heading"><?php __('Actions') ?></li>
10        <li><?php $this->Html->link(__('Edit Series'), ['action' => 'edit', $series->id]) ?> </li>
11        <li><?php $this->Form->postLink(__('Delete Series'), ['action' => 'delete', $series->id], ['confirm' => __('Are you sure you want to delete # {0}?', $series->id)]) ?> </li>
12        <li><?php $this->Html->link(__('List Series'), ['action' => 'index']) ?> </li>
13        <li><?php $this->Html->link(__('New Series'), ['action' => 'add']) ?> </li>
14    </ul>
15 </nav>
16 <div class="series view large-9 medium-8 columns content">
17     <h3><?php h($series->name) ?></h3>
18     <table class="vertical-table">
19         <tr>
20             <th scope="row"><?php __('Name') ?></th>
21             <td><?php h($series->name) ?></td>
22         </tr>
23         <tr>
24             <th scope="row"><?php __('Category') ?></th>
25             <td><?php h($series->category) ?></td>
26         </tr>
27         <tr>
28             <th scope="row"><?php __('Description') ?></th>
29             <td><?php h($series->description) ?></td>
30         </tr>
```



## 7. Códigos do projeto (Cont.)

- **View: Series**  
(view.ctp)

cont.

```
view.ctp
31      <tr>
32          <th scope="row"><?= __('Id') ?></th>
33          <td><?= $this->Number->format($series->id) ?></td>
34      </tr>
35      <tr>
36          <th scope="row"><?= __('Number Of Visualizations') ?></th>
37          <td><?= $this->Number->format($series->number_of_visualizations) ?></td>
38      </tr>
39  </table>
40 </div>
```



## 7. Códigos do projeto (Cont.)

- **Controller: Series**  
(SeriesController.php)

```
SeriesController.php x
1 |<?php
2 namespace App\Controller;
3
4 use App\Controller\AppController;
5
6 /**
7  * Series Controller
8  *
9  * @property \App\Model\Table\SeriesTable $Series
10 *
11 * @method \App\Model\Entity\Series[]|Cake\Datasource\ResultSetInterface paginate($object
    = null, array $settings = [])
12 */
13 class SeriesController extends AppController
14 {
15     /**
16      * Index method
17      *
18      * @return \Cake\Http\Response|void
19      */
20     public function index()
21     {
22         $series = $this->paginate($this->Series);
23
24         $this->set(compact('series'));
25     }
26
27     /**
28      * View method
29      *
30      * @param string|null $id Series id.
31      * @return \Cake\Http\Response|void
32      * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
```





## 7. Códigos do projeto (Cont.)

- **Controller: Series**  
(SeriesController.php)

cont.

```
SeriesController.php x
33  */
34  public function view($id = null)
35  {
36      $series = $this->Series->get($id, [
37          'contain' => []
38      ]);
39
40      $this->set('series', $series);
41  }
42
43  /**
44   * Add method
45   *
46   * @return \Cake\Http\Response|null Redirects on successful add, renders view otherwise.
47   */
48  public function add()
49  {
50      $series = $this->Series->newEntity();
51      if ($this->request->is('post')) {
52          $series = $this->Series->patchEntity($series, $this->request->getData());
53          // var_dump($series);
54          // exit();
55          if ($this->Series->save($series)) {
56              $this->Flash->success(__('The series has been saved.'));
57
58              return $this->redirect(['action' => 'index']);
59          }
60          $this->Flash->error(__('The series could not be saved. Please, try again.'));
61      }
62      $this->set(compact('series'));
63  }
64
65  /**
```



## 7. Códigos do projeto (Cont.)

- **Controller: Series**  
(SeriesController.php)

cont.

```
SeriesController.php x
66  * Edit method
67  *
68  * @param string|null $id Series id.
69  * @return \Cake\Http\Response|null Redirects on successful edit, renders view
    otherwise.
70  * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
71  */
72  public function edit($id = null)
73  {
74      $series = $this->Series->get($id, [
75          'contain' => []
76      ]);
77      if ($this->request->is(['patch', 'post', 'put'])) {
78          $series = $this->Series->patchEntity($series, $this->request->getData());
79          if ($this->Series->save($series)) {
80              $this->Flash->success(__('The series has been saved.'));
81
82              return $this->redirect(['action' => 'index']);
83          }
84          $this->Flash->error(__('The series could not be saved. Please, try again.'));
85      }
86      $this->set(compact('series'));
87  }
88
89  /**
90   * Delete method
91   *
92   * @param string|null $id Series id.
93   * @return \Cake\Http\Response|null Redirects to index.
94   * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
95   */
96  public function delete($id = null)
97  {
```



## 7. Códigos do projeto (Cont.)

- **Controller: Series**  
(SeriesController.php)

cont.

```
98     $this->request->allowMethod(['post', 'delete']);
99     $series = $this->Series->get($id);
100     if ($this->Series->delete($series)) {
101         $this->Flash->success(__('The series has been deleted.));
102     } else {
103         $this->Flash->error(__('The series could not be deleted. Please, try again.));
104     }
105
106     return $this->redirect(['action' => 'index']);
107 }
108 }
```



## 7. Códigos do projeto (Cont.)

- **Model: Schedules**  
(SchedulesTable.php)

```
SchedulesTable.php x
1  <?php
2  namespace App\Model\Table;
3
4  use Cake\ORM\Query;
5  use Cake\ORM\RulesChecker;
6  use Cake\ORM\Table;
7  use Cake\Validation\Validator;
8
9  /**
10   * Schedules Model
11   *
12   * @property \App\Model\Table\UsersTable|\Cake\ORM\Association\BelongsTo $Users
13   * @property \App\Model\Table\SeriesTable|\Cake\ORM\Association\BelongsTo $Series
14   *
15   * @method \App\Model\Entity\Schedule get($primaryKey, $options = [])
16   * @method \App\Model\Entity\Schedule newEntity($data = null, array $options = [])
17   * @method \App\Model\Entity\Schedule[] newEntities(array $data, array $options = [])
18   * @method \App\Model\Entity\Schedule|bool save(\Cake\Datasource\EntityInterface $entity,
19   * $options = [])
20   * @method \App\Model\Entity\Schedule saveOrFail(\Cake\Datasource\EntityInterface $entity,
21   * $options = [])
22   * @method \App\Model\Entity\Schedule patchEntity(\Cake\Datasource\EntityInterface $entity,
23   * array $data, array $options = [])
24   * @method \App\Model\Entity\Schedule[] patchEntities($entities, array $data, array $options
25   * = [])
26   * @method \App\Model\Entity\Schedule findOrCreate($search, callable $callback = null,
27   * $options = [])
28   */
29
30 class SchedulesTable extends Table
31 {
32     /**
33      * Initialize method
34      *
35      * @return void
36      */
37     public function initialize()
38     {
39         $this->addBehavior('Timestampable');
40
41         $this->belongsTo('Users', [
42             'foreignKey' => 'user_id',
43             'cascadeDelete' => true,
44             'dependent' => true,
45             'propertyName' => 'user'
46         ]);
47
48         $this->belongsTo('Series', [
49             'foreignKey' => 'series_id',
50             'cascadeDelete' => true,
51             'dependent' => true,
52             'propertyName' => 'series'
53         ]);
54     }
55
56     /**
57      * Default validation rules.
58      *
59      * @param \Cake\Validation\Validator $validator Validator object.
60      * @return \Cake\Validation\Validator Validation rules.
61      */
62     public function validationDefault(Validator $validator)
63     {
64         $validator->allowEmpty('title');
65
66         $validator->require('title');
67
68         $validator->require('description');
69
70         $validator->require('start_date');
71
72         $validator->require('end_date');
73
74         $validator->require('user_id');
75
76         $validator->require('series_id');
77
78         $validator->require('status');
79
80         $validator->require('created');
81
82         $validator->require('modified');
83
84         $validator->require('deleted');
85
86         $validator->require('deleted_at');
87
88         $validator->require('deleted_at_timestamp');
89
90         $validator->require('deleted_at_timestamp');
91
92         $validator->require('deleted_at_timestamp');
93
94         $validator->require('deleted_at_timestamp');
95
96         $validator->require('deleted_at_timestamp');
97
98         $validator->require('deleted_at_timestamp');
99
100        $validator->require('deleted_at_timestamp');
101    }
102}
```



## 7. Códigos do projeto (Cont.)

- **Model: Schedules**  
(SchedulesTable.php)

cont.

```
SchedulesTable.php x
29      * @param array $config The configuration for the Table.
30      * @return void
31      */
32      public function initialize(array $config)
33      {
34          parent::initialize($config);
35
36          $this->setTable('schedules');
37          $this->setDisplayField('id');
38          $this->setPrimaryKey('id');
39
40          $this->belongsTo('Users', [
41              'foreignKey' => 'user_id',
42              'joinType' => 'INNER'
43          ]);
44          $this->belongsTo('Series', [
45              'foreignKey' => 'serie_id',
46              'joinType' => 'INNER'
47          ]);
48      }
49
50      /**
51       * Default validation rules.
52       *
53       * @param \Cake\Validation\Validator $validator Validator instance.
54       * @return \Cake\Validation\Validator
55       */
56      public function validationDefault(Validator $validator)
57      {
58          $validator
59              ->integer('id')
60              ->allowEmptyString('id', 'create');
61      }
```



## 7. Códigos do projeto (Cont.)

- **Model:** Schedules  
(SchedulesTable.php)

cont.

```
SchedulesTable.php x
62         return $validator;
63     }
64
65     /**
66      * Returns a rules checker object that will be used for validating
67      * application integrity.
68      *
69      * @param \Cake\ORM\RulesChecker $rules The rules object to be modified.
70      * @return \Cake\ORM\RulesChecker
71      */
72     public function buildRules(RulesChecker $rules)
73     {
74         $rules->add($rules->existsIn(['user_id'], 'Users'));
75         $rules->add($rules->existsIn(['serie_id'], 'Series'));
76
77         return $rules;
78     }
79 }
80
```



## 7. Códigos do projeto (Cont.)

- **View: Schedules**  
(index.ctp)

```
index.ctp
1 |<?php
2 |/**
3 | * @var \App\View\AppView $this
4 | * @var \App\Model\Entity\Schedule[]|\Cake\Collection\CollectionInterface $schedules
5 | */
6 |?>
7 |<h1>Meus Agendamentos <?= $this->Html->link('Criar novo agendamento', ['action' => 'add'], ['
   |class' => 'btn btn-primary btn-sm']) ?></h1>
8 |<div class="table-responsive">
9 |    <table class="table table-striped">
10 |        <thead>
11 |            <tr>
12 |                <th scope="col"><?= $this->Paginator->sort('user_id', ['label'=>'Usuário']) ?>
   |                </th>
13 |                <th scope="col"><?= $this->Paginator->sort('serie_id', ['label'=>'Série']) ?>
   |                </th>
14 |                <th scope="col"><?= $this->Paginator->sort('date', ['label'=>'Dia']) ?></th>
15 |                <th scope="col" class="actions"> </th>
16 |            </tr>
17 |        </thead>
18 |        <tbody>
19 |            <?php foreach ($schedules as $schedule): ?>
20 |                <tr>
21 |                    <td><?= $schedule->has('user') ? $this->Html->link($schedule->user->name, ['
   |action' => 'edit', $schedule->id]) : '' ?></td>
22 |                    <td><?= $schedule->has('series') ? $this->Html->link($schedule->series->name,
   |['controller' => 'Series', 'action' => 'edit', $schedule->series->id])
   |: '' ?></td>
23 |                    <td><?= h($schedule->date) ?></td>
24 |                    <td class="actions">
25 |                        <?= $this->Form->postLink(__('Delete'), ['action' => 'delete', $schedule
   |->id], ['confirm' => __('Esse registro será apagado. Deseja continuar?
   |'), 'class' => 'btn btn-sm btn-danger']) ?>
```



## 7. Códigos do projeto (Cont.)

- **View:** Schedules (index.ctp)

cont.

```
index.ctp
26         </td>
27     </tr>
28     <?php endforeach; ?>
29 </tbody>
30 </table>
31 <div class="paginator">
32     <ul class="pagination">
33         <?= $this->Paginator->first('<< ' . __('first')) ?>
34         <?= $this->Paginator->prev('< ' . __('previous')) ?>
35         <?= $this->Paginator->numbers() ?>
36         <?= $this->Paginator->next(__('next') . ' >') ?>
37         <?= $this->Paginator->last(__('last') . ' >>') ?>
38     </ul>
39     <p><?= $this->Paginator->counter(['format' => __('Page {{page}} of {{pages}},
        showing {{current}} record(s) out of {{count}} total')) ?></p>
40 </div>
41 </div>
```





## 7. Códigos do projeto (Cont.)

- **View: Schedules**  
(add.ctp)

```
add.ctp
1 <h1>
2   Criar agendamento
3 </h1>
4 <hr/>
5 <div class="form-group">
6   <?= $this->Form->create($schedule,['type' => 'POST'] ) ?>
7   <div class="col-12">
8     <?= $this->Form->control('user_id', ['options' => $users,'label'=><b>Usuário</b>','
9       class'=>'form-control','escape'=>false]); ?>
10   </div>
11   <div class="col-12">
12     <?= $this->Form->control('serie_id', ['options' => $series, 'label'=><b>Série</b>','
13       class'=>'form-control','escape'=>false]) ?>
14   </div>
15   <div class="col-12">
16     <?= $this->Form->control('date', ['label'=><b>Dia</b>','escape'=>false,
17       'templates' => ['dateWidget' => '
18         {{day}}{{month}}{{year}}'],
19       'day' => ['class' => "form-control-sm ml-2 mt-2
20         mr-2"],
21       'month' => ['class' => "form-control-sm mt-2 mr-2"]
22       ,
23       'year' => ['class' => "form-control-sm mt-2 mr-2"],
24     ]) ?>
25   </div>
26   <div class="col-12">
27     <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
28   </div>
29   <?= $this->Form->end() ?>
30 </div>
```



## 7. Códigos do projeto (Cont.)

- **View: Schedules**  
(edit.ctp)

```
edit.ctp
1 <h1>
2   Editar agendamento
3 </h1>
4 <hr/>
5 <div class="form-group">
6   <?= $this->Form->create($schedule,['type' => 'PUT'] ) ?>
7   <div class="col-12">
8     <?= $this->Form->control('user_id', ['options' => $users,'label'=><b>Usuário</b>','
9       class'=>'form-control','escape'=>false]); ?>
10   </div>
11   <div class="col-12">
12     <?= $this->Form->control('serie_id', ['options' => $series, 'label'=><b>Série</b>','
13       class'=>'form-control','escape'=>false]) ?>
14   </div>
15   <div class="col-12">
16     <?= $this->Form->control('date', ['label'=><b>Dia</b>','escape'=>false,
17       'templates' => ['dateWidget' => '
18         {{day}}{{month}}{{year}}'],
19       'day' => ['class' => "form-control-sm ml-2 mt-2
20         mr-2"],
21       'month' => ['class' => "form-control-sm mt-2 mr-2"]
22       ,
23       'year' => ['class' => "form-control-sm mt-2 mr-2"],
24     ]) ?>
25   </div>
26   <div class="col-12">
27     <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
28   </div>
29   <?= $this->Form->end() ?>
30 </div>
```



## 7. Códigos do projeto (Cont.)

- **View: Schedules**  
(view.ctp)

```
view.ctp
1 |<?php
2 |/**
3 | * @var \App\View\AppView $this
4 | * @var \App\Model\Entity\Schedule $schedule
5 | */
6 |?>
7 |<nav class="large-3 medium-4 columns" id="actions-sidebar">
8 |    <ul class="side-nav">
9 |        <li class="heading"><? = __('Actions') ?></li>
10 |        <li><? = $this->Html->link(__('Edit Schedule'), ['action' => 'edit', $schedule->id])
11 |            ?> </li>
12 |        <li><? = $this->Form->postLink(__('Delete Schedule'), ['action' => 'delete', $schedule
13 |            ->id], ['confirm' => __('Are you sure you want to delete # {0}?', $schedule->id)]
14 |            ) ?> </li>
15 |        <li><? = $this->Html->link(__('List Schedules'), ['action' => 'index']) ?> </li>
16 |        <li><? = $this->Html->link(__('New Schedule'), ['action' => 'add']) ?> </li>
17 |        <li><? = $this->Html->link(__('List Users'), ['controller' => 'Users', 'action' => '
18 |            index']) ?> </li>
19 |        <li><? = $this->Html->link(__('New User'), ['controller' => 'Users', 'action' => 'add'
20 |            ]) ?> </li>
21 |        <li><? = $this->Html->link(__('List Series'), ['controller' => 'Series', 'action' => '
22 |            index']) ?> </li>
23 |        <li><? = $this->Html->link(__('New Series'), ['controller' => 'Series', 'action' => '
24 |            add']) ?> </li>
25 |    </ul>
26 |</nav>
27 |<div class="schedules view large-9 medium-8 columns content">
28 |    <h3><? = h($schedule->id) ?></h3>
29 |    <table class="vertical-table">
30 |        <tr>
31 |            <th scope="row"><? = __('User') ?></th>
32 |            <td><? = $schedule->has('user') ? $this->Html->link($schedule->user->name, ['
33 |                controller' => 'Users', 'action' => 'view', $schedule->user->id]) : '' ?></td>
```



## 7. Códigos do projeto (Cont.)

- **View: Schedules**  
(view.ctp)

cont.

```
view.ctp
<div>
  <table>
    <tr>
      <th scope="row"><?= __('Series') ?></th>
      <td><?= $schedule->has('series') ? $this->Html->link($schedule->series->name, [
        'controller' => 'Series', 'action' => 'view', $schedule->series->id]) : '' ?>
      </td>
    </tr>
    <tr>
      <th scope="row"><?= __('Id') ?></th>
      <td><?= $this->Number->format($schedule->id) ?></td>
    </tr>
    <tr>
      <th scope="row"><?= __('Date') ?></th>
      <td><?= h($schedule->date) ?></td>
    </tr>
  </table>
</div>
```



## 7. Códigos do projeto (Cont.)

- **Controller: Schedules**  
(SchedulesController.php)

```
SchedulesController.php x
1 <?php
2 namespace App\Controller;
3
4 use App\Controller\AppController;
5
6 /**
7  * Schedules Controller
8  *
9  * @property \App\Model\Table\SchedulesTable $Schedules
10  *
11  * @method \App\Model\Entity\Schedule[]|\Cake\Datasource\ResultSetInterface
12  * paginate($object = null, array $settings = [])
13  */
14 class SchedulesController extends AppController
15 {
16     /**
17      * Index method
18      *
19      * @return \Cake\Http\Response|void
20      */
21     public function index()
22     {
23         $this->paginate = [
24             'contain' => ['Users', 'Series']
25         ];
26         $schedules = $this->paginate($this->Schedules);
27
28         $this->set(compact('schedules'));
29     }
30
31     /**
32      * View method
```



## 7. Códigos do projeto (Cont.)

- **Controller: Schedules**  
(SchedulesController.php)

cont.

```
SchedulesController.php x
33  * @param string|null $id Schedule id.
34  * @return \Cake\Http\Response|void
35  * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
36  */
37  public function view($id = null)
38  {
39      $schedule = $this->Schedules->get($id, [
40          'contain' => ['Users', 'Series']
41      ]);
42
43      $this->set('schedule', $schedule);
44  }
45
46  /**
47   * Add method
48   *
49   * @return \Cake\Http\Response|null Redirects on successful add, renders view otherwise.
50   */
51  public function add()
52  {
53      $schedule = $this->Schedules->newEntity();
54      if ($this->request->is('post')) {
55          $schedule = $this->Schedules->patchEntity($schedule, $this->request->getData());
56          if ($this->Schedules->save($schedule)) {
57              $this->Flash->success(__('The schedule has been saved.'));
58
59              return $this->redirect(['action' => 'index']);
60          }
61          $this->Flash->error(__('The schedule could not be saved. Please, try again.'));
62      }
63      $users = $this->Schedules->Users->find('list', ['limit' => 200]);
64      $series = $this->Schedules->Series->find('list', ['limit' => 200]);
65      $this->set(compact('schedule', 'users', 'series'));
```



## 7. Códigos do projeto (Cont.)

- **Controller: Schedules**  
(SchedulesController.php)

cont.

```
SchedulesController.php x
66     }
67
68     /**
69      * Edit method
70      *
71      * @param string|null $id Schedule id.
72      * @return \Cake\Http\Response|null Redirects on successful edit, renders view otherwise.
73      * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
74      */
75     public function edit($id = null)
76     {
77         $schedule = $this->Schedules->get($id, [
78             'contain' => []
79         ]);
80         if ($this->request->is(['patch', 'post', 'put'])) {
81             $schedule = $this->Schedules->patchEntity($schedule, $this->request->getData());
82             if ($this->Schedules->save($schedule)) {
83                 $this->Flash->success(__('The schedule has been saved.));
84
85                 return $this->redirect(['action' => 'index']);
86             }
87             $this->Flash->error(__('The schedule could not be saved. Please, try again.));
88         }
89         $users = $this->Schedules->Users->find('list', ['limit' => 200]);
90         $series = $this->Schedules->Series->find('list', ['limit' => 200]);
91         $this->set(compact('schedule', 'users', 'series'));
92     }
93
94     /**
95      * Delete method
96      *
97      * @param string|null $id Schedule id.
```



## 7. Códigos do projeto (Cont.)

- **Controller: Schedules**  
(SchedulesController.php)

cont.

```
SchedulesController.php x
98      * @return \Cake\Http\Response|null Redirects to index.
99      * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
100     */
101     public function delete($id = null)
102     {
103         $this->request->allowMethod(['post', 'delete']);
104         $schedule = $this->Schedules->get($id);
105         if ($this->Schedules->delete($schedule)) {
106             $this->Flash->success(__('The schedule has been deleted.'));
107         } else {
108             $this->Flash->error(__('The schedule could not be deleted. Please, try again.'));
109         }
110
111         return $this->redirect(['action' => 'index']);
112     }
113 }
114
```





## 7. Códigos do projeto (Cont.)

- **Pages:** Home (home.ctp)

```
home.ctp
1 <h1>
2   Próximas séries a assistir
3 </h1>
4 <hr/>
5 <div class="table-responsive">
6   <table class="table table-striped">
7     <thead>
8       <tr>
9         <th scope="col">Usuário</th>
10        <th scope="col">Série</th>
11        <th scope="col">Dia</th>
12      </tr>
13    </thead>
14    <tbody>
15      <?php
16        foreach ($schedules as $item) { ?>
17        <tr>
18          <td><?= $item->user->username ?></td>
19          <td><?= $item->series->name ?></td>
20          <td><?= $item->date->format('d/m/Y') ?></td>
21        </tr>
22        <?php } ?>
23      </tbody>
24    </table>
25  </div>
```



## 7. Códigos do projeto (Cont.)

- **Layout:** Default (default.ctp)

```
default.ctp
1 <?php
2 /**
3  * CakePHP(tm) : Rapid Development Framework (https://cakephp.org)
4  * Copyright (c) Cake Software Foundation, Inc. (https://cakefoundation.org)
5  *
6  * Licensed under The MIT License
7  * For full copyright and license information, please see the LICENSE.txt
8  * Redistributions of files must retain the above copyright notice.
9  *
10 * @copyright Copyright (c) Cake Software Foundation, Inc. (https://cakefoundation.org)
11 * @link https://cakephp.org CakePHP(tm) Project
12 * @since 0.10.0
13 * @license https://opensource.org/licenses/mit-license.php MIT License
14 */
15
16 $cakeDescription = 'CakePHP: the rapid development php framework';
17 ?>
18 <!DOCTYPE html>
19 <html>
20 <head>
21     <?= $this->Html->charset() ?>
22     <meta name="viewport" content="width=device-width, initial-scale=1.0">
23     <title>
24         Agenda Séries
25     </title>
26     <?= $this->Html->css('bootstrap-grid.css') ?>
27     <?= $this->Html->css('bootstrap-reboot.css') ?>
28     <?= $this->Html->css('bootstrap.css') ?>
29     <?= $this->Html->script('jquery-3.4.1.min.js') ?>
30     <?= $this->Html->script('bootstrap.bundle.js') ?>
31     <?= $this->Html->script('bootstrap.js') ?>
32
33     <?= $this->fetch('meta') ?>
```



## 7. Códigos do projeto (Cont.)

- **Layout:** Default (default.ctp)

cont.

```
default.ctp
34 <?= $this->fetch('css') ?>
35 <?= $this->fetch('script') ?>
36 </head>
37 <body>
38 <nav class="navbar navbar-expand-lg navbar-light bg-light">
39 <a class="navbar-brand" href="#"><?= $this->Html->image('agenda-series-logo-sml.png', ['
  alt'=>'Agenda Séries']) ?></a>
40 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
  #navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
  aria-label="Toggle navigation">
41 <span class="navbar-toggler-icon"></span>
42 </button>
43
44 <div class="collapse navbar-collapse" id="navbarSupportedContent">
45 <ul class="navbar-nav mr-auto">
46 <li class="nav-item active">
47 <?= $this->Html->link('Home', ['controller'=>'Pages', 'action'=>'display'], ['class'
  =>'nav-link']) ?>
48 </li>
49 <li class="nav-item">
50 <?= $this->Html->link('Usuários', ['controller'=>'Users', 'action'=>'index'], ['
  class'=>'nav-link']) ?>
51 </li>
52 <li class="nav-item">
53 <?= $this->Html->link('Minhas Séries', ['controller'=>'Series', 'action'=>'index'],
  ['class'=>'nav-link']) ?>
54 </li>
55 <li class="nav-item">
56 <?= $this->Html->link('Meus Agendamentos', ['controller'=>'Schedules', 'action'=>'
  index'], ['class'=>'nav-link']) ?>
57 </li>
58 </ul>
59 </div>
```



## 7. Códigos do projeto (Cont.)

- **Layout:** Default (default.ctp)

cont.

```
default.ctp
60     </nav>
61     <?= $this->Flash->render() ?>
62     <div class="container clearfix">
63         <?= $this->fetch('content') ?>
64     </div>
65     <footer>
66     </footer>
67 </body>
68 </html>
69
```



## 8. Referências

- Análise sobre a utilização de frameworks em PHP. Slideshare. Disponível em: < <https://pt.slideshare.net/ThiagoSinsio/analise-sobre-a-utilizacao-de-frameworks-em-php-cakephp-codeigniter-e-zend> >. Acessado em: 07/06/2019.
- CakePHP. Disponível em: < <https://cakephp.org/> >. Acessado em: 07/06/2019.
- CakePHP. GitHub. Disponível em: < <https://github.com/cakephp> >. Acessado em: 07/06/2019.
- Como criar um CRUD com o bake do CakePHP 3. Celke. Disponível em: < <https://celke.com.br/artigo/como-criar-crud-com-bake-do-cakephp> >. Acessado em: 11/06/2019.