Instituto Federal de Educação, Ciência e Tecnologia do Amazonas
Campus Manaus Centro

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

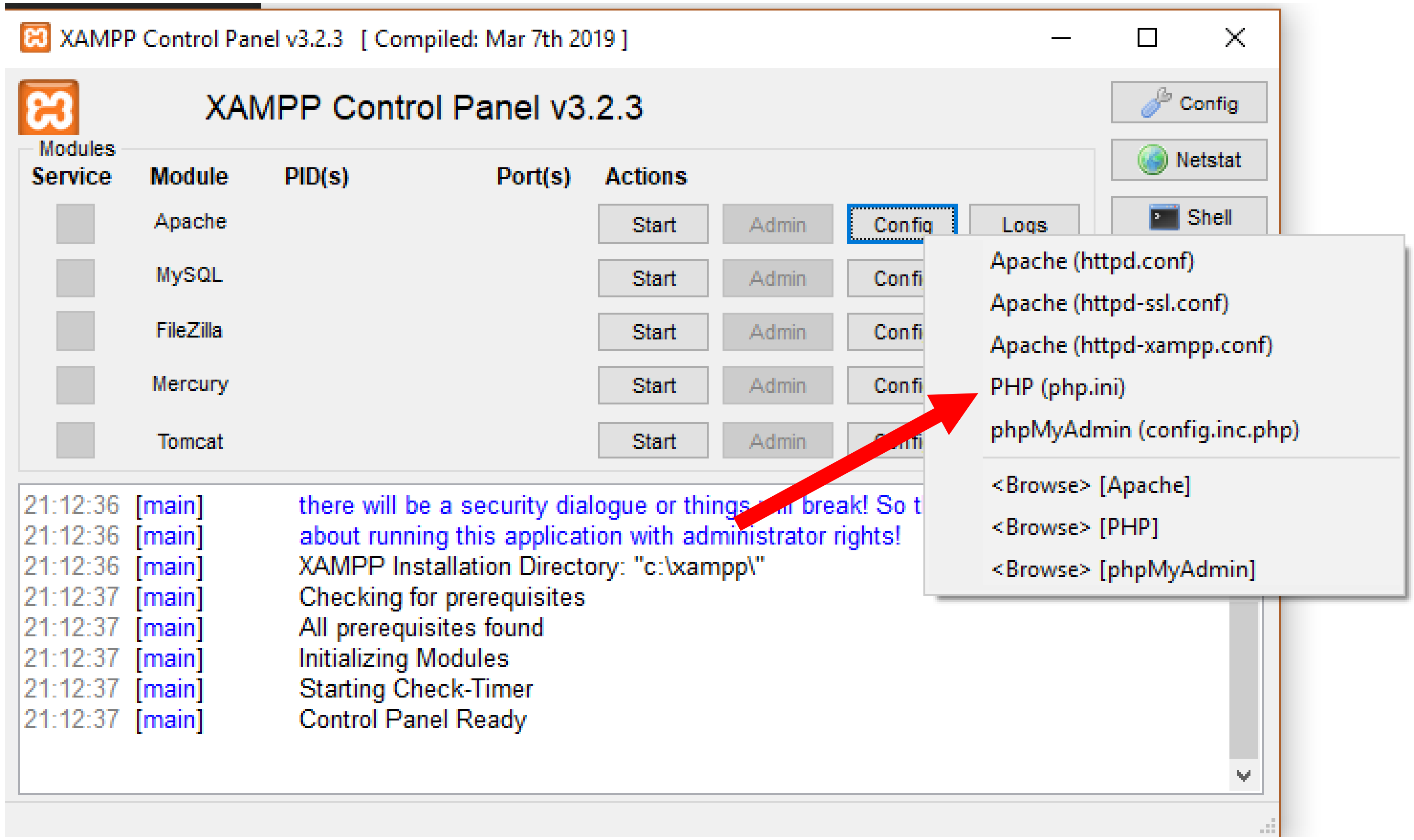
5. Instalação e configuração (Cont.)

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

5. Instalação e configuração (Cont.)

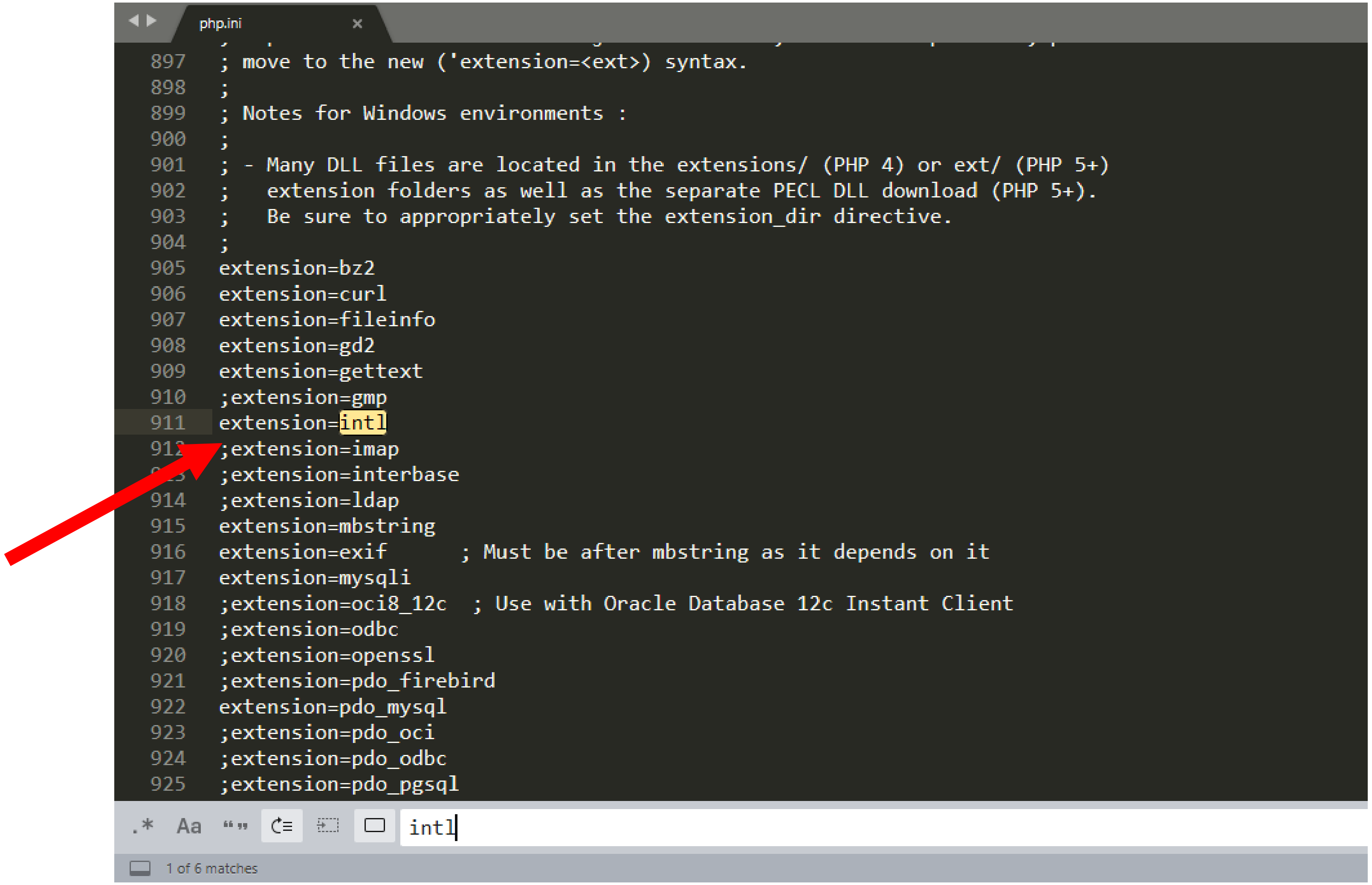
- Antes, precisamos configurar a extensão intl para o CakePHP funcionar normalmente na sua máquina.
- Após instalado o XAMPP, abra-o e procure pela linha onde o Apache se encontra.
- Siga os botões e clique em ‘Config’ e logo após em ‘PHP (php.ini)’
- Busque por “intl”
- Na linha “;extension=intl”, remova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

5. Instalação e configuração (Cont.)



```
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.)



```
; move to the new ('extension=<ext>) syntax.
;
; Notes for Windows environments :
;
; - Many DLL files are located in the extensions/ (PHP 4) or ext/ (PHP 5+)
;   extension folders as well as the separate PECL DLL download (PHP 5+).
;   Be sure to appropriately set the extension_dir directive.
;
extension=bz2
extension=curl
extension=fileinfo
extension=gd2
extension=gettext
;extension=gmp
extension=intl
;extension=imap
;extension=interbase
;extension=ldap
extension=mbstring
extension=exif      ; Must be after mbstring as it depends on it
extension=mysqli
;extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
```

1 of 6 matches

5. Instalação e configuração (Cont.)

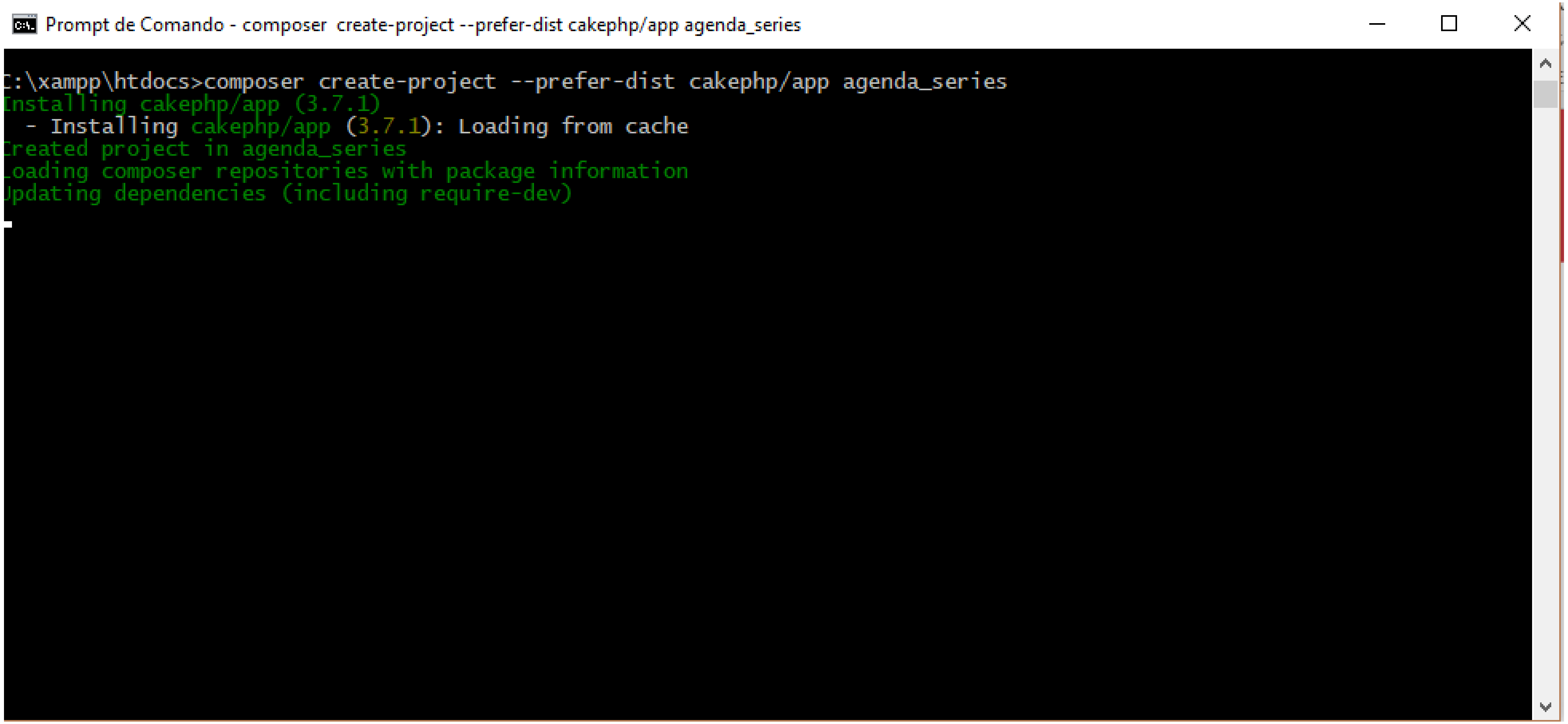
- Para facilitar, vamos utilizar o executável do PHP do XAMPP como base para os comandos a seguir.

- 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.)



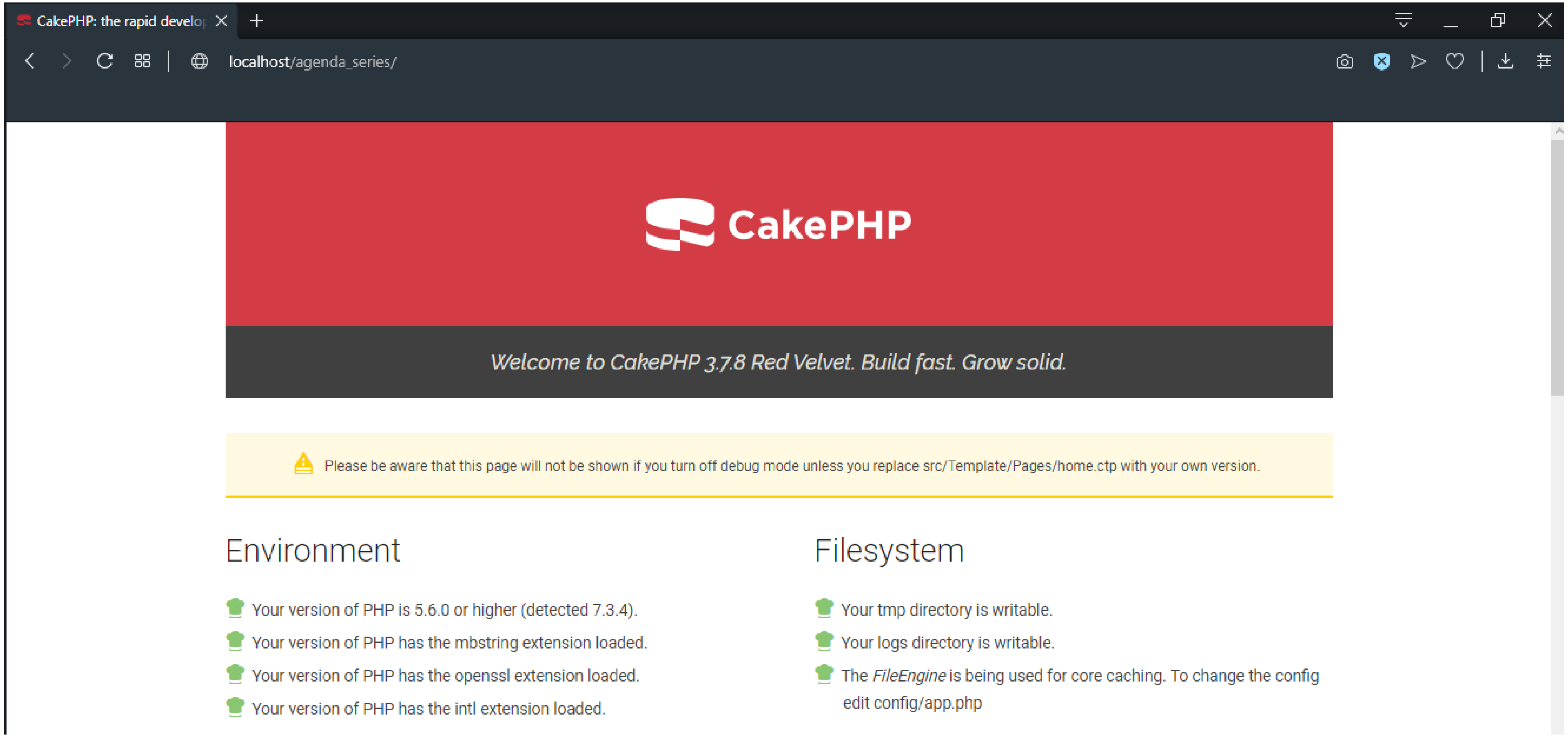
Prompt de Comando - composer create-project --prefer-dist cakephp/app agenda_series

```
:\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
- A página inicial do CakePHP irá aparecer.

5. Instalação e configuração (Cont.)



Welcome to CakePHP 3.7.8 Red Velvet. Build fast. Grow solid.

⚠ 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.

Environment

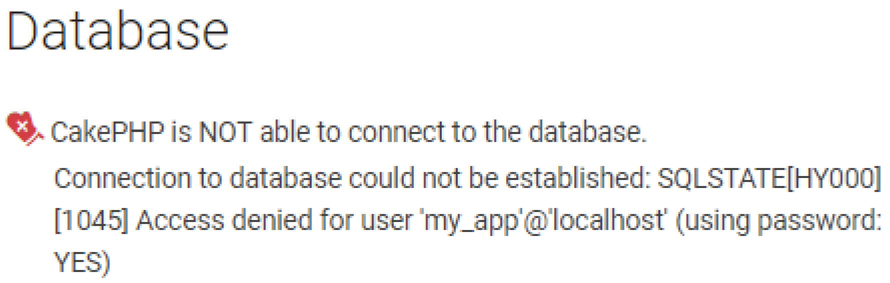
- Your version of PHP is 5.6.0 or higher (detected 7.3.4).
- Your version of PHP has the mbstring extension loaded.
- Your version of PHP has the openssl extension loaded.
- Your version of PHP has the intl extension loaded.

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.

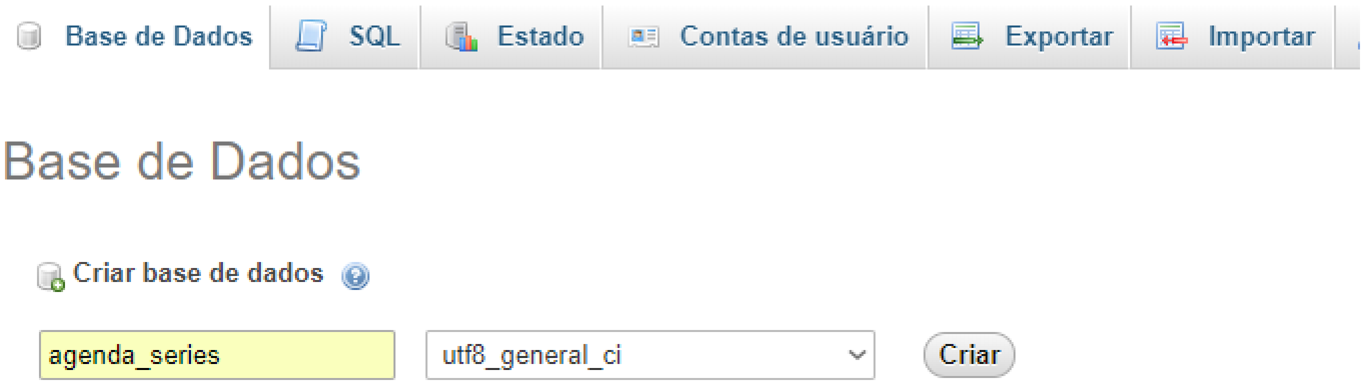
Connection to database could not be established: SQLSTATE[HY000] [1045] Access denied for user 'my_app'@'localhost' (using password: YES)

- 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

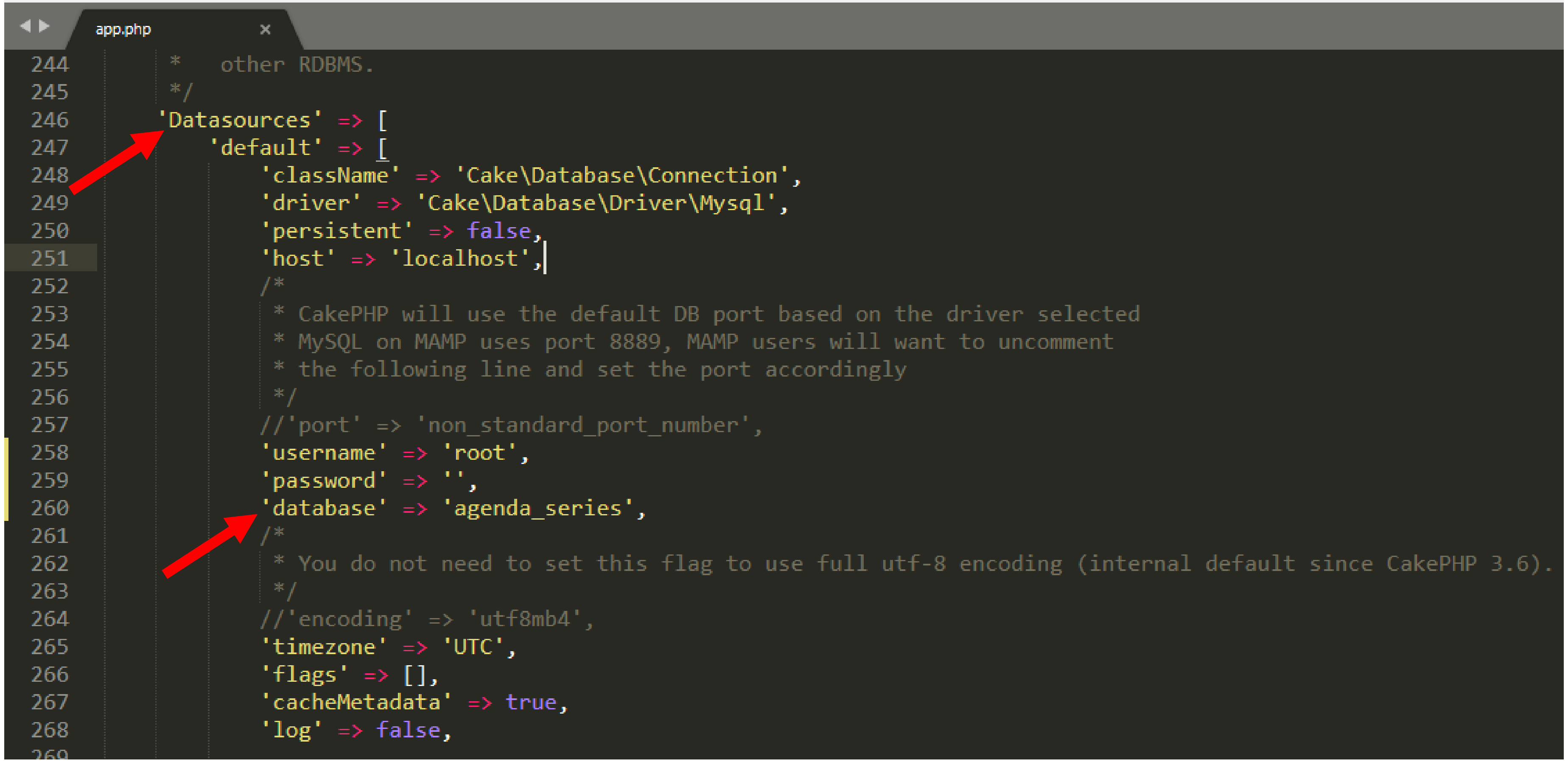
agenda_series

utf8_general_ci

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’

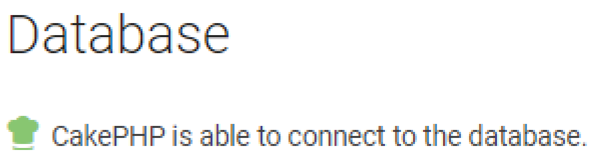
5. Instalação e configuração (Cont.)



```
     *   other RDBMS.
     */
    'Datasources' => [
        'default' => [
            'className' => 'Cake\Database\Connection',
            'driver' => 'Cake\Database\Driver\Mysql',
            'persistent' => false,
            'host' => 'localhost',
            /*
             * CakePHP will use the default DB port based on the driver selected
             * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
             * the following line and set the port accordingly
             */
            //'port' => 'non_standard_port_number',
            'username' => 'root',
            'password' => '',
            'database' => 'agenda_series',
            /*
             * You do not need to set this flag to use full utf-8 encoding (internal default since CakePHP 3.6).
             */
            //'encoding' => 'utf8mb4',
            'timezone' => 'UTC',
            'flags' => [],
            'cacheMetadata' => true,
            'log' => false,
```

5. Instalação e configuração (Cont.)

- Pronto! Atualize a página http://localhost/agenda_series
- O alerta estará mudado.

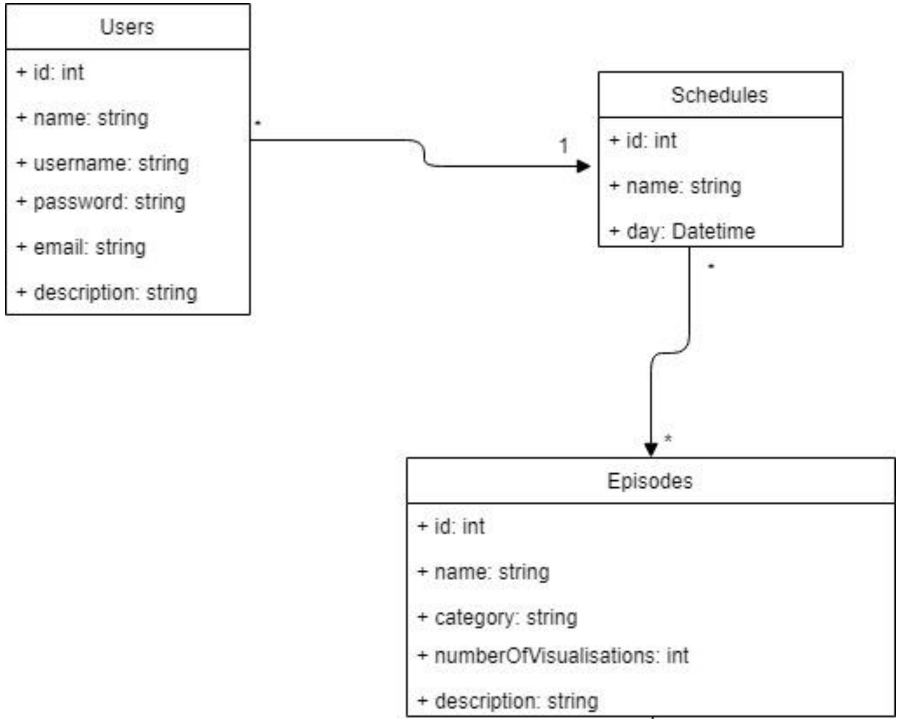


Database

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:



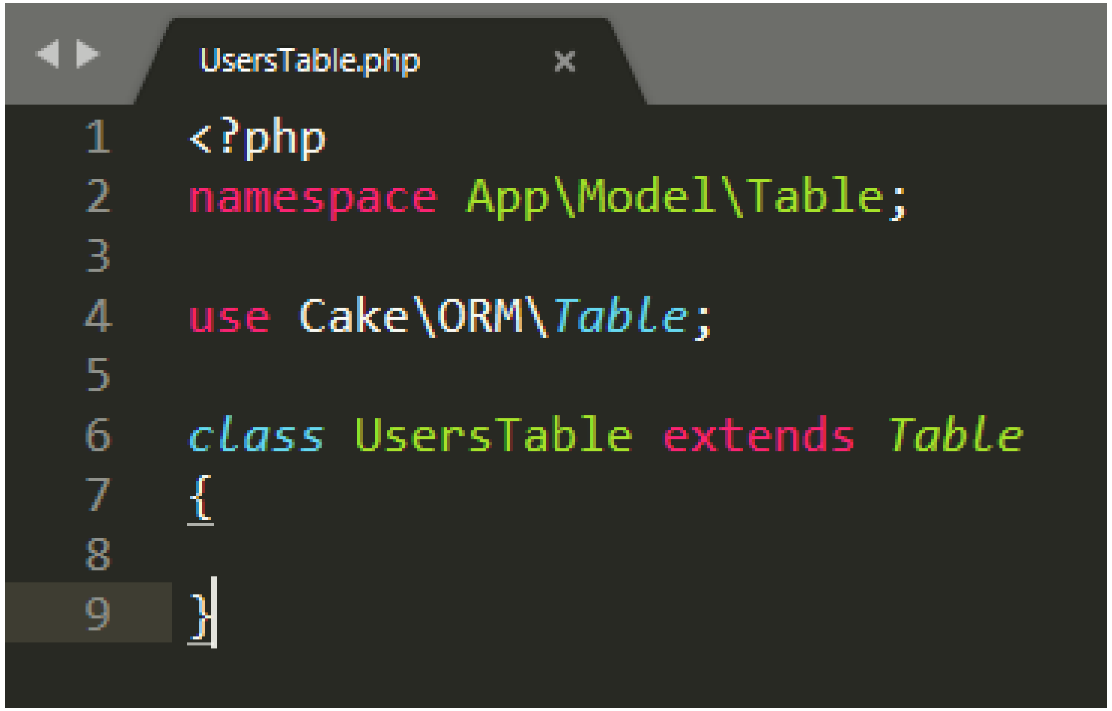
Users
+ id: int
+ name: string
+ username: string
+ password: string
+ email: string
+ description: string

Schedules
+ id: int
+ name: string
+ day: Datetime

Episodes
+ id: int
+ name: string
+ category: string
+ numberOfVisualisations: int
+ description: string

7. Códigos do projeto

- **Model:** Usuário (UsersTable.php)

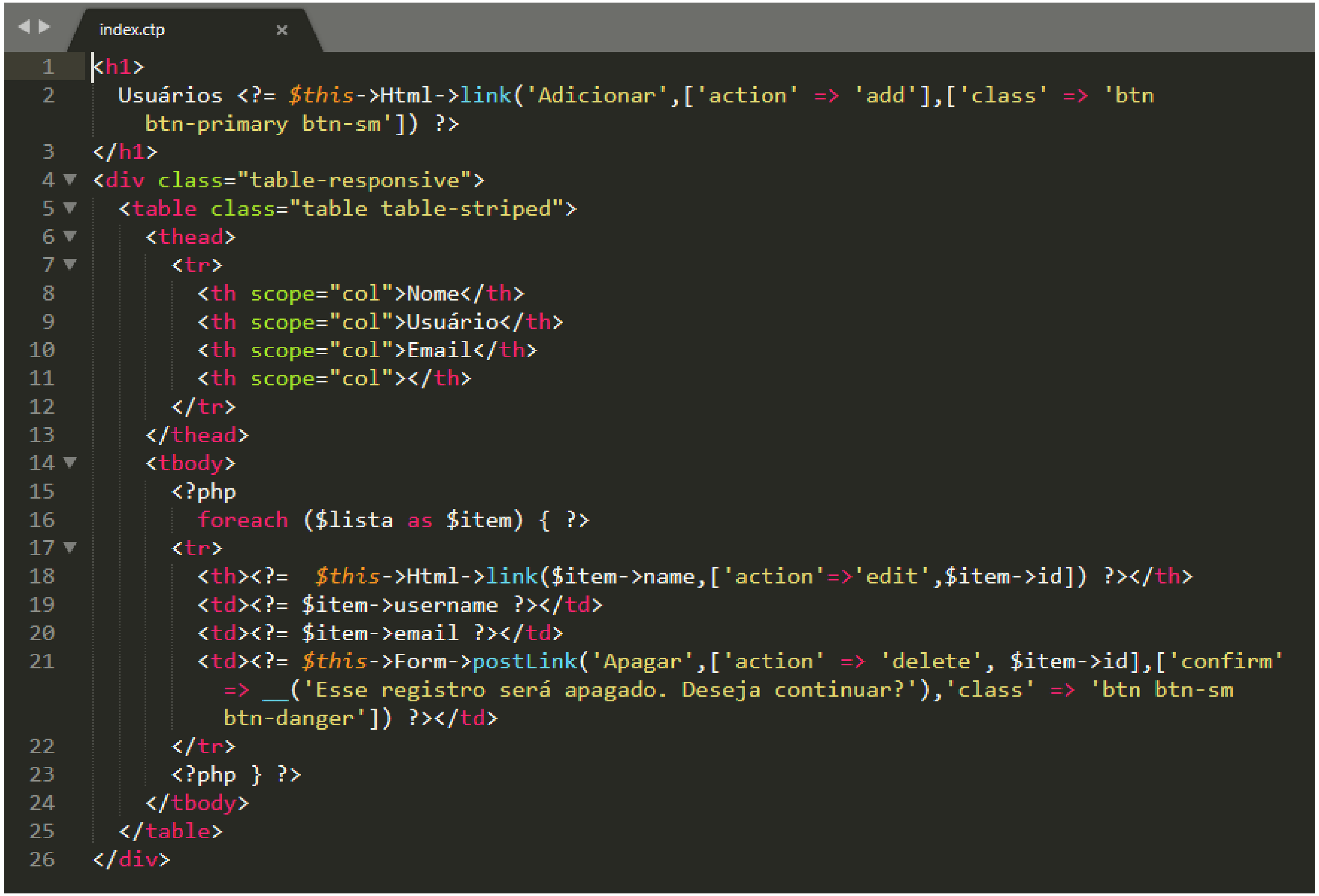


```php
<?php
namespace App\Model\Table;

use Cake\ORM\Table;

class UsersTable extends Table
{

}
```

C:\xampp\htdocs\agenda_series\src\Model\Table

7. Códigos do projeto (Cont.)

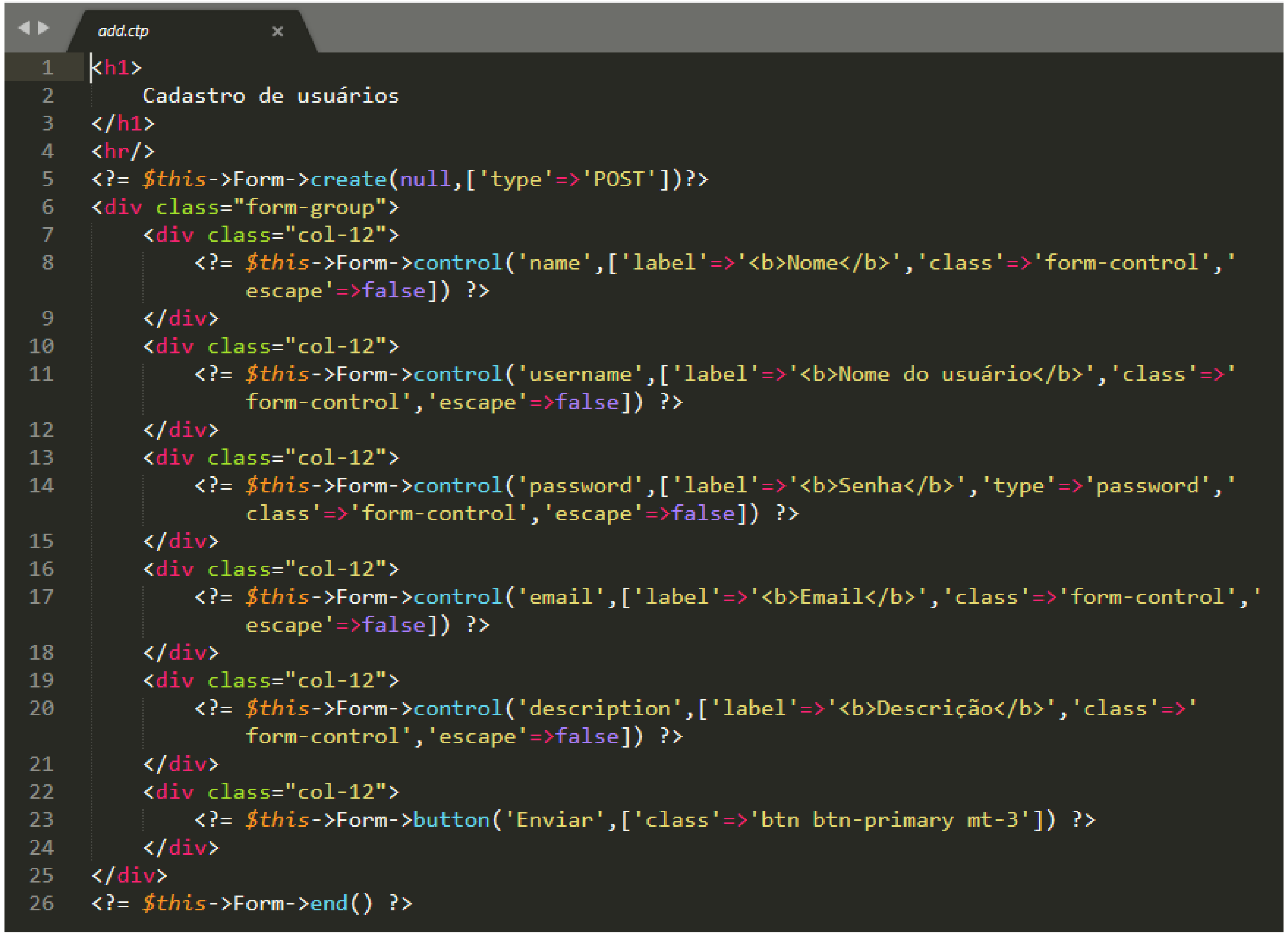
- **View:** Usuário (index.ctp)



```
<h1>
  Usuários <?= $this->Html->link('Adicionar',['action' => 'add'],['class' => 'btn btn-primary btn-sm']) ?>
</h1>
<div class="table-responsive">
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">Nome</th>
        <th scope="col">Usuário</th>
        <th scope="col">Email</th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>
      <?php
        foreach ($lista as $item) { ?>
      <tr>
        <th><?=  $this->Html->link($item->name,['action'=>'edit',$item->id]) ?></th>
        <td><?= $item->username ?></td>
        <td><?= $item->email ?></td>
        <td><?= $this->Form->postLink('Apagar',['action' => 'delete', $item->id],['confirm' => __('Esse registro será apagado. Deseja continuar?'),'class' => 'btn btn-sm btn-danger']) ?></td>
      </tr>
      <?php } ?>
    </tbody>
  </table>
</div>
```

7. Códigos do projeto (Cont.)

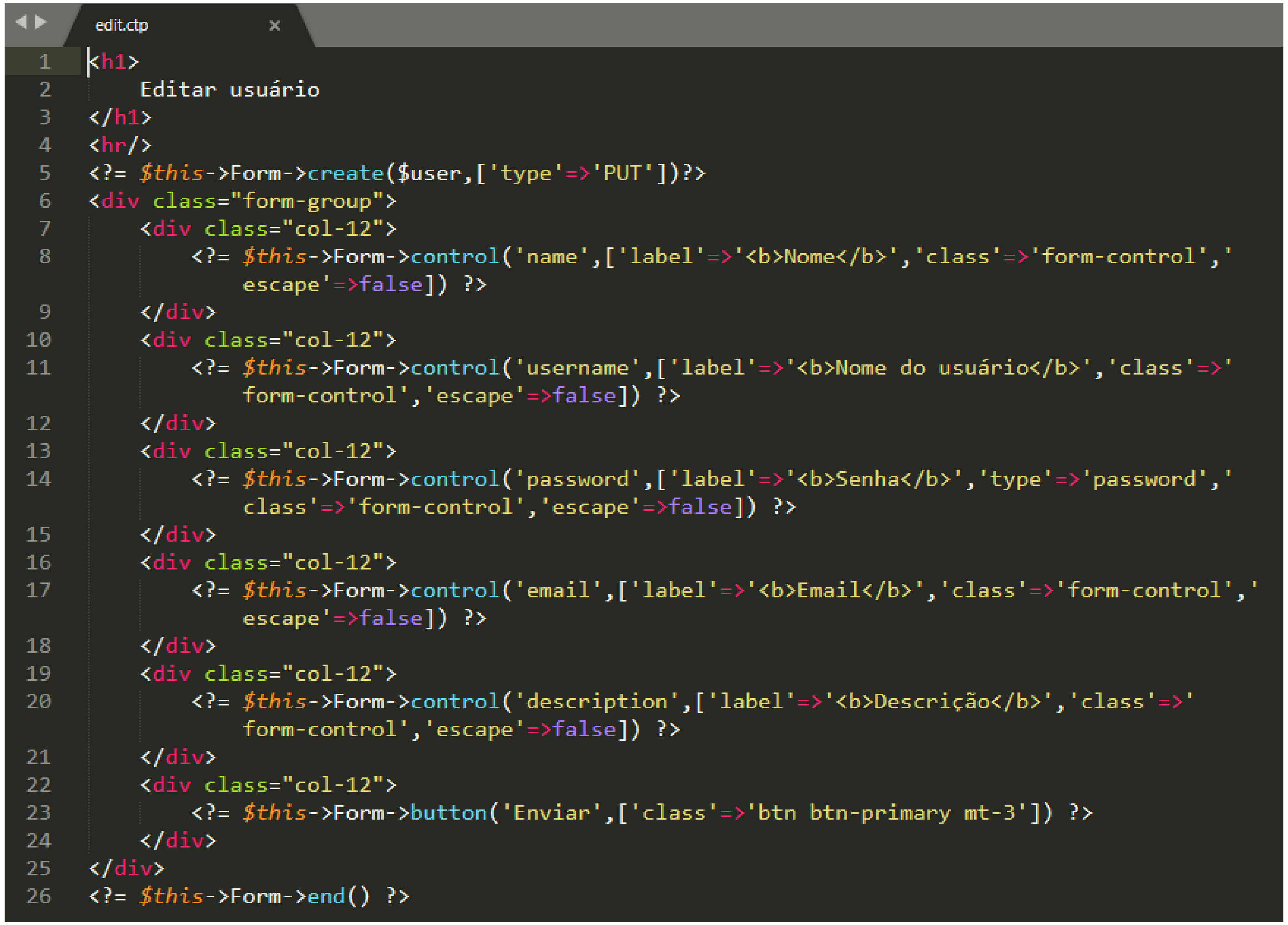
- **View:** Usuário (add.ctp)



```
<h1>
    Cadastro de usuários
</h1>
<hr/>
<?= $this->Form->create(null,['type'=>'POST'])?>
<div class="form-group">
    <div class="col-12">
        <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('username',['label'=>'<b>Nome do usuário</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('password',['label'=>'<b>Senha</b>','type'=>'password','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('email',['label'=>'<b>Email</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
</div>
<?= $this->Form->end() ?>
```

7. Códigos do projeto (Cont.)

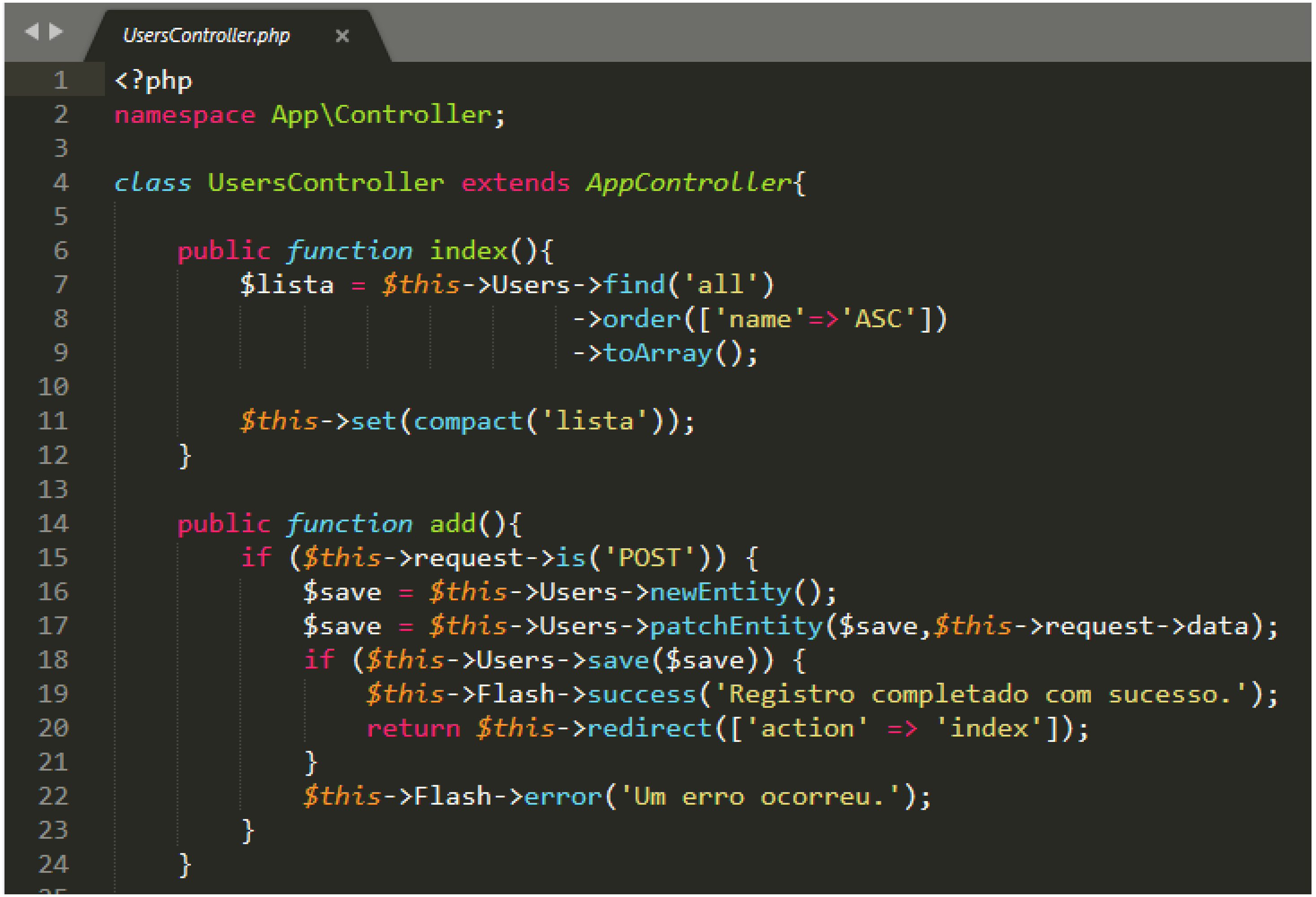
- **View:** Usuário (edit.ctp)



```
<h1>
    Editar usuário
</h1>
<hr/>
<?= $this->Form->create($user,['type'=>'PUT'])?>
<div class="form-group">
    <div class="col-12">
        <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('username',['label'=>'<b>Nome do usuário</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('password',['label'=>'<b>Senha</b>','type'=>'password','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('email',['label'=>'<b>Email</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
</div>
<?= $this->Form->end() ?>
```

7. Códigos do projeto (Cont.)

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

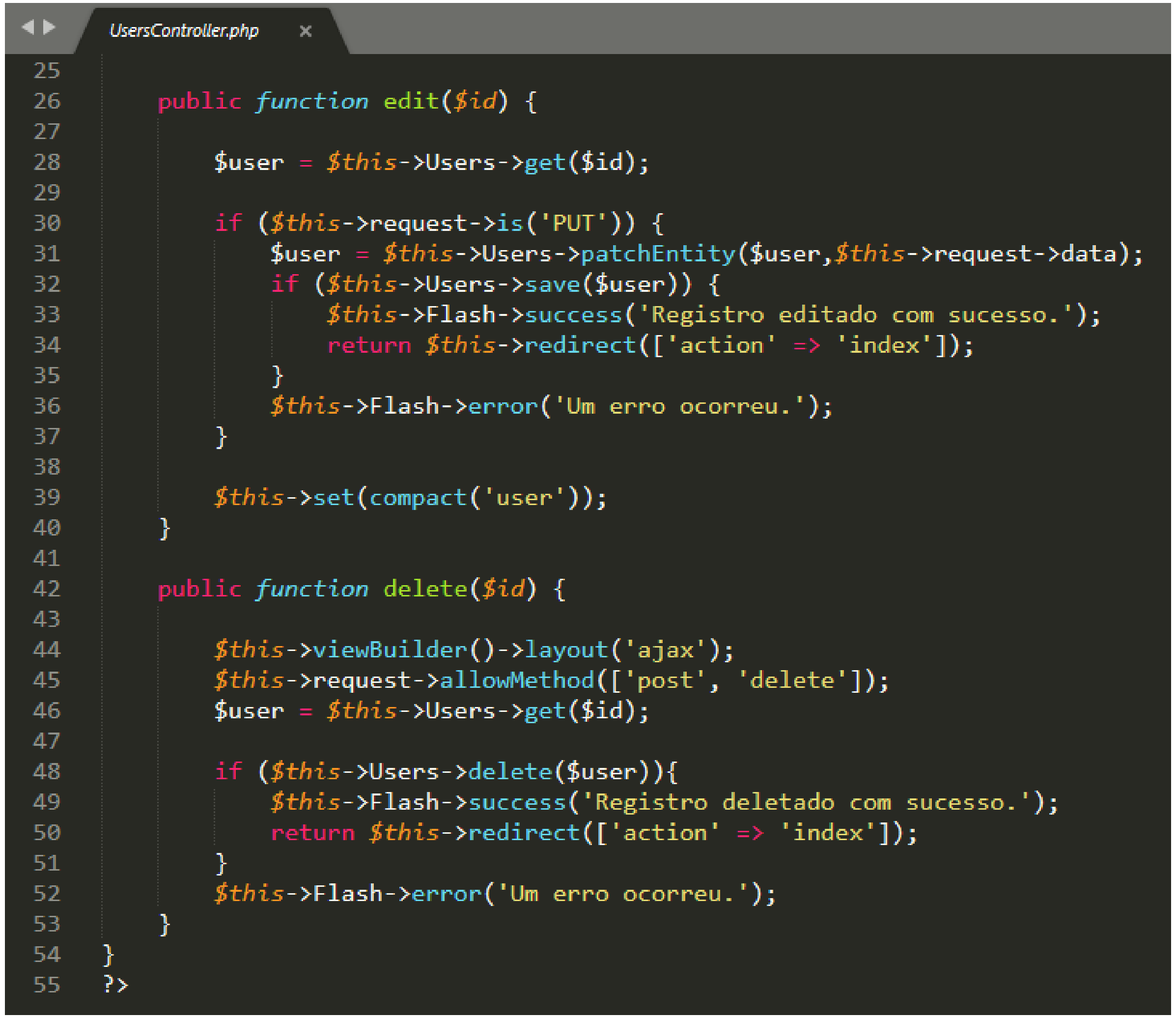


```php
<?php
namespace App\Controller;

class UsersController extends AppController{

    public function index(){
        $lista = $this->Users->find('all')
                             ->order(['name'=>'ASC'])
                             ->toArray();

        $this->set(compact('lista'));
    }

    public function add(){
        if ($this->request->is('POST')) {
            $save = $this->Users->newEntity();
            $save = $this->Users->patchEntity($save,$this->request->data);
            if ($this->Users->save($save)) {
                $this->Flash->success('Registro completado com sucesso.');
                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error('Um erro ocorreu.');
        }
    }
```

7. Códigos do projeto (Cont.)

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

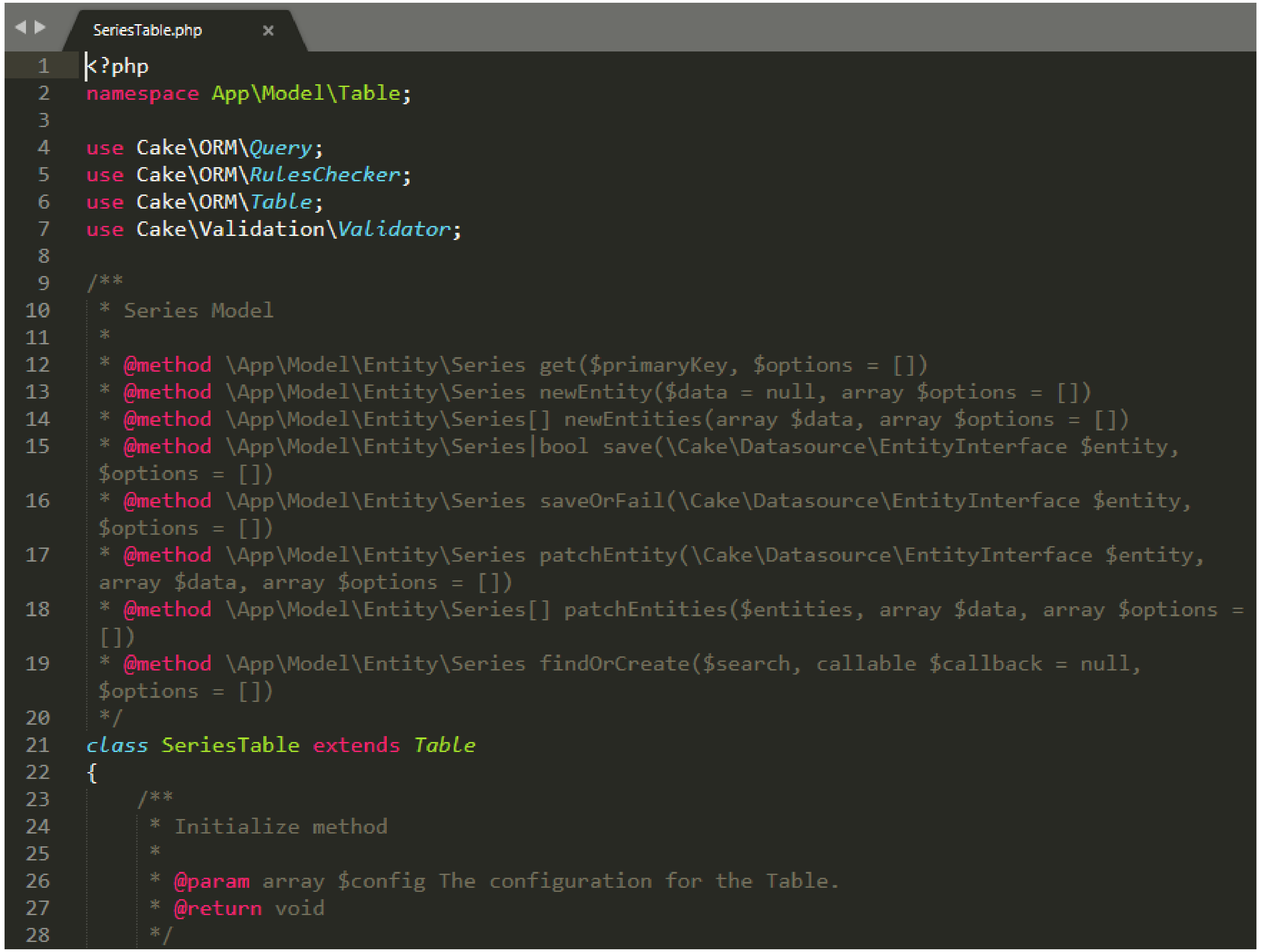
  cont.



```php

    public function edit($id) {

        $user = $this->Users->get($id);

        if ($this->request->is('PUT')) {
            $user = $this->Users->patchEntity($user,$this->request->data);
            if ($this->Users->save($user)) {
                $this->Flash->success('Registro editado com sucesso.');
                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error('Um erro ocorreu.');
        }

        $this->set(compact('user'));
    }

    public function delete($id) {

        $this->viewBuilder()->layout('ajax');
        $this->request->allowMethod(['post', 'delete']);
        $user = $this->Users->get($id);

        if ($this->Users->delete($user)){
            $this->Flash->success('Registro deletado com sucesso.');
            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error('Um erro ocorreu.');
    }
}
?>
```

7. Códigos do projeto (Cont.)

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

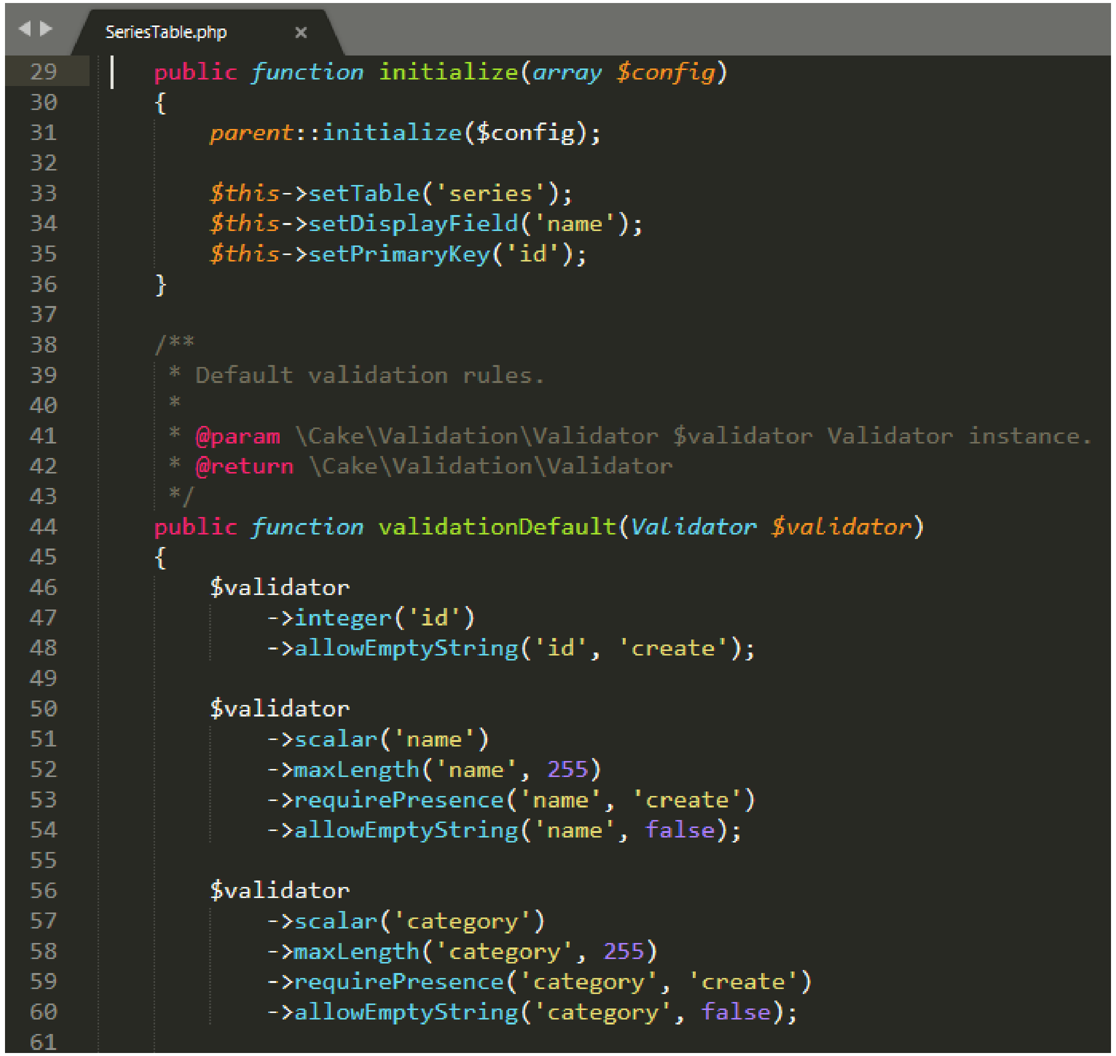


```php
<?php
namespace App\Model\Table;

use Cake\ORM\Query;
use Cake\ORM\RulesChecker;
use Cake\ORM\Table;
use Cake\Validation\Validator;

/**
 * Series Model
 *
 * @method \App\Model\Entity\Series get($primaryKey, $options = [])
 * @method \App\Model\Entity\Series newEntity($data = null, array $options = [])
 * @method \App\Model\Entity\Series[] newEntities(array $data, array $options = [])
 * @method \App\Model\Entity\Series|bool save(\Cake\Datasource\EntityInterface $entity, $options = [])
 * @method \App\Model\Entity\Series saveOrFail(\Cake\Datasource\EntityInterface $entity, $options = [])
 * @method \App\Model\Entity\Series patchEntity(\Cake\Datasource\EntityInterface $entity, array $data, array $options = [])
 * @method \App\Model\Entity\Series[] patchEntities($entities, array $data, array $options = [])
 * @method \App\Model\Entity\Series findOrCreate($search, callable $callback = null, $options = [])
 */
class SeriesTable extends Table
{
    /**
     * Initialize method
     *
     * @param array $config The configuration for the Table.
     * @return void
     */
```

7. Códigos do projeto (Cont.)

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

  cont.

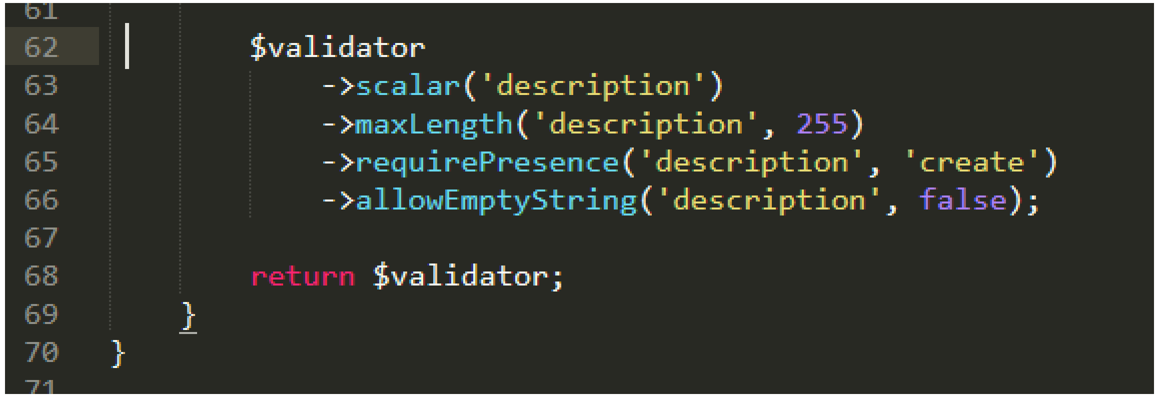


```
    public function initialize(array $config)
    {
        parent::initialize($config);

        $this->setTable('series');
        $this->setDisplayField('name');
        $this->setPrimaryKey('id');
    }

    /**
     * Default validation rules.
     *
     * @param \Cake\Validation\Validator $validator Validator instance.
     * @return \Cake\Validation\Validator
     */
    public function validationDefault(Validator $validator)
    {
        $validator
            ->integer('id')
            ->allowEmptyString('id', 'create');

        $validator
            ->scalar('name')
            ->maxLength('name', 255)
            ->requirePresence('name', 'create')
            ->allowEmptyString('name', false);

        $validator
            ->scalar('category')
            ->maxLength('category', 255)
            ->requirePresence('category', 'create')
            ->allowEmptyString('category', false);

```

7. Códigos do projeto (Cont.)

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

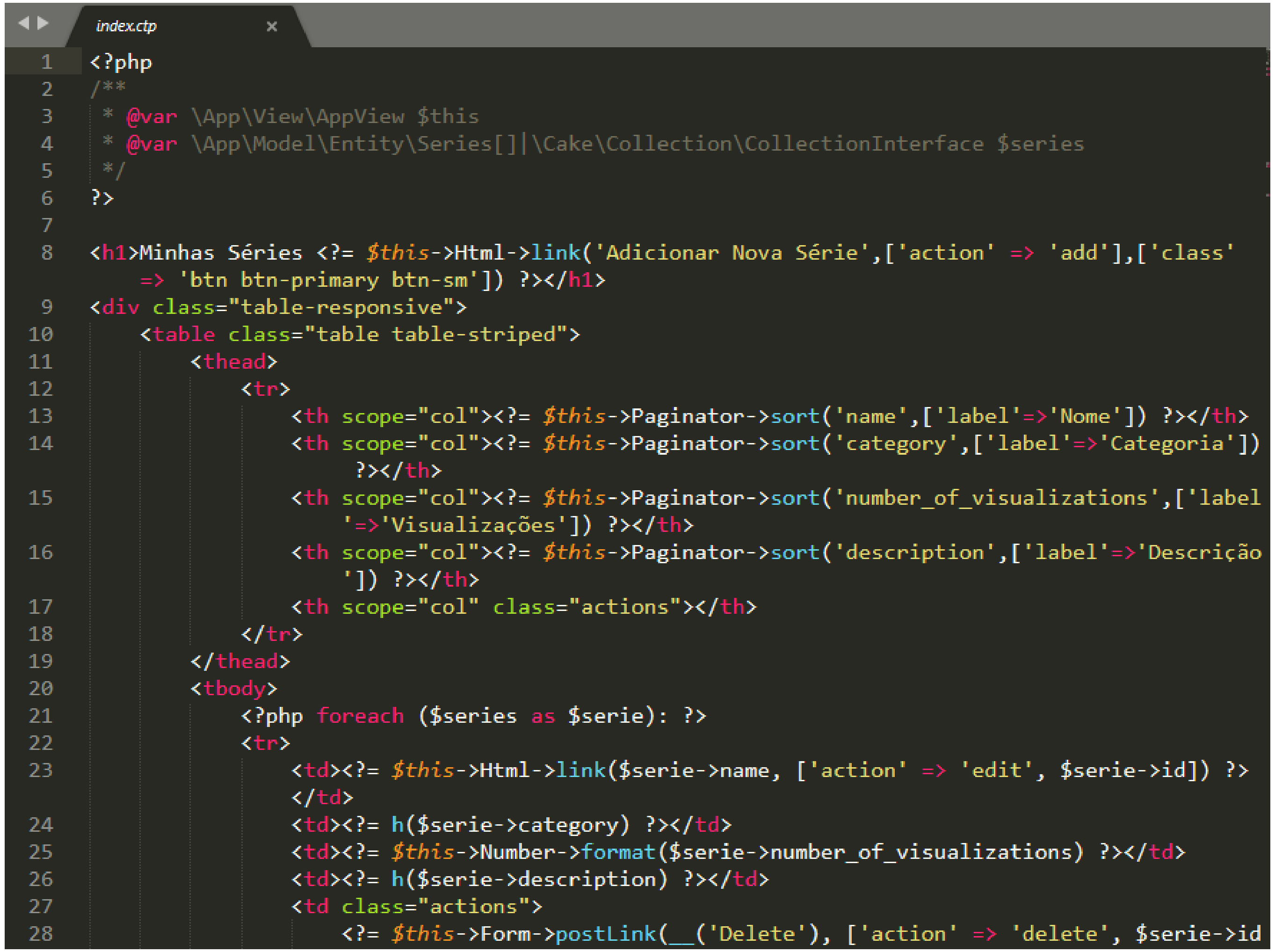
  cont.



```php
        $validator
            ->scalar('description')
            ->maxLength('description', 255)
            ->requirePresence('description', 'create')
            ->allowEmptyString('description', false);

        return $validator;
    }
}
```

7. Códigos do projeto (Cont.)

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

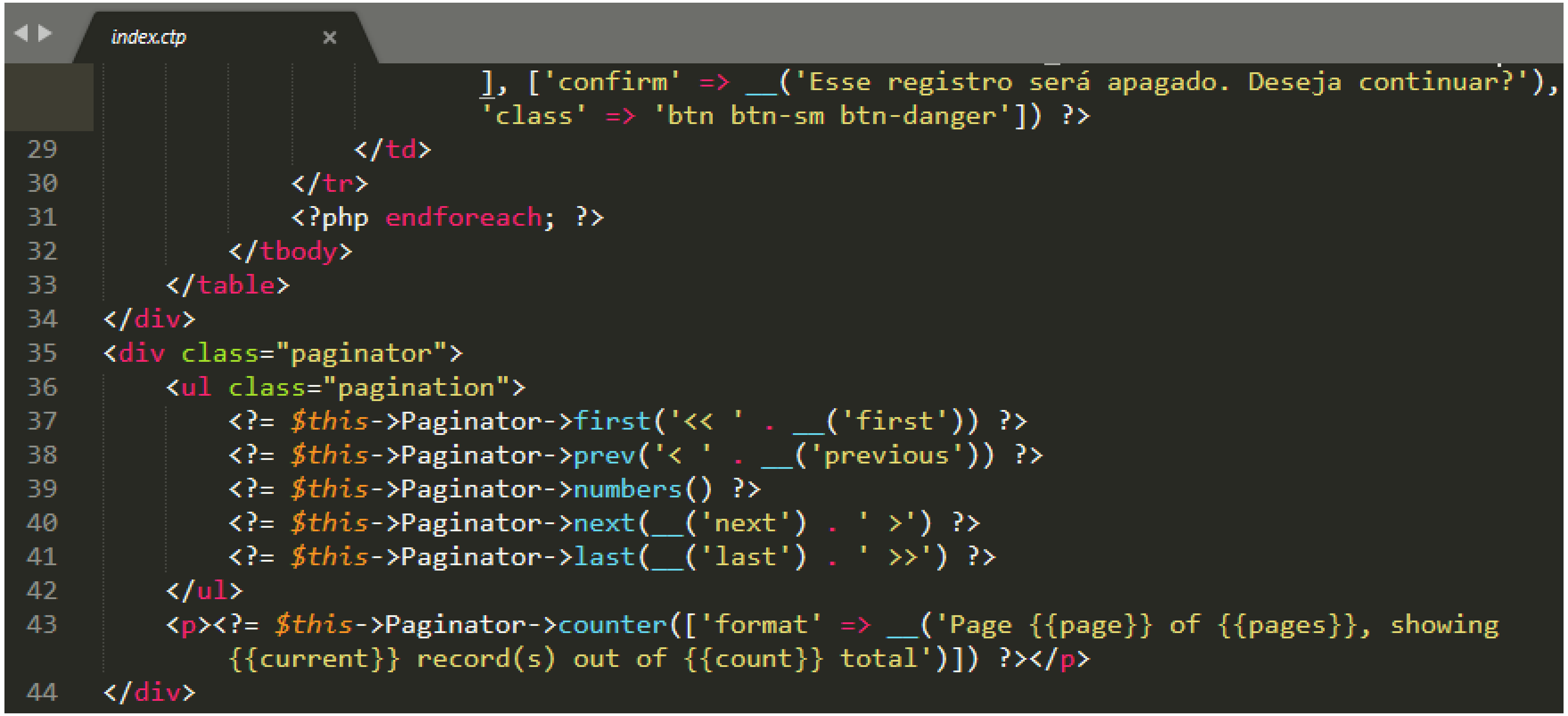


```
<?php
/**
 * @var \App\View\AppView $this
 * @var \App\Model\Entity\Series[]|\Cake\Collection\CollectionInterface $series
 */
?>

<h1>Minhas Séries <?= $this->Html->link('Adicionar Nova Série',['action' => 'add'],['class' => 'btn btn-primary btn-sm']) ?></h1>
<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col"><?= $this->Paginator->sort('name',['label'=>'Nome']) ?></th>
                <th scope="col"><?= $this->Paginator->sort('category',['label'=>'Categoria']) ?></th>
                <th scope="col"><?= $this->Paginator->sort('number_of_visualizations',['label'=>'Visualizações']) ?></th>
                <th scope="col"><?= $this->Paginator->sort('description',['label'=>'Descrição']) ?></th>
                <th scope="col" class="actions"></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($series as $serie): ?>
            <tr>
                <td><?= $this->Html->link($serie->name, ['action' => 'edit', $serie->id]) ?></td>
                <td><?= h($serie->category) ?></td>
                <td><?= $this->Number->format($serie->number_of_visualizations) ?></td>
                <td><?= h($serie->description) ?></td>
                <td class="actions">
                    <?= $this->Form->postLink(__('Delete'), ['action' => 'delete', $serie->id
```

7. Códigos do projeto (Cont.)

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

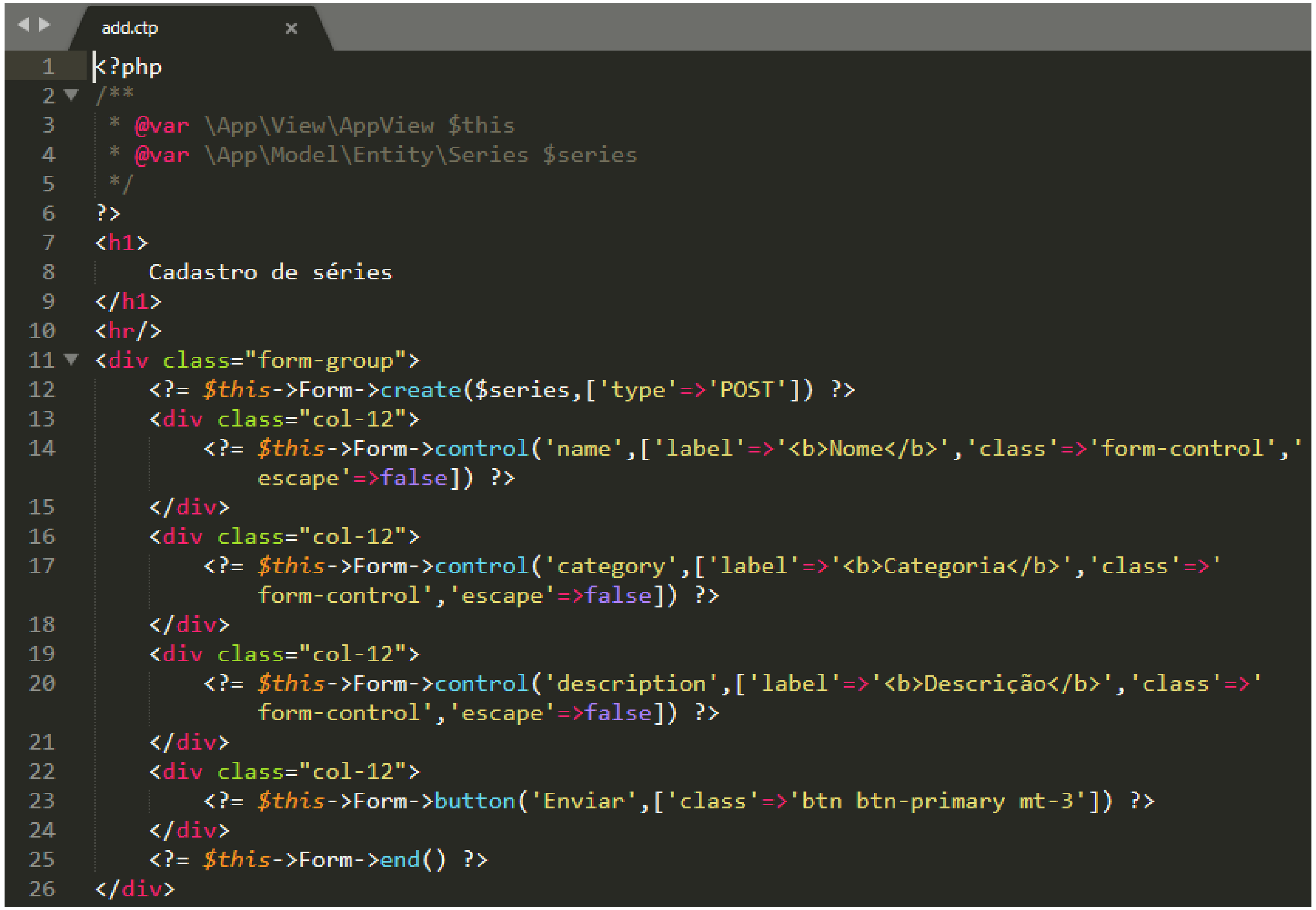
  cont.



```
                        ], ['confirm' => __('Esse registro será apagado. Deseja continuar?'),
                        'class' => 'btn btn-sm btn-danger']) ?>
                </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
</div>
<div class="paginator">
    <ul class="pagination">
        <?= $this->Paginator->first('<< ' . __('first')) ?>
        <?= $this->Paginator->prev('< ' . __('previous')) ?>
        <?= $this->Paginator->numbers() ?>
        <?= $this->Paginator->next(__('next') . ' >') ?>
        <?= $this->Paginator->last(__('last') . ' >>') ?>
    </ul>
    <p><?= $this->Paginator->counter(['format' => __('Page {{page}} of {{pages}}, showing
        {{current}} record(s) out of {{count}} total')]) ?></p>
</div>
```

7. Códigos do projeto (Cont.)

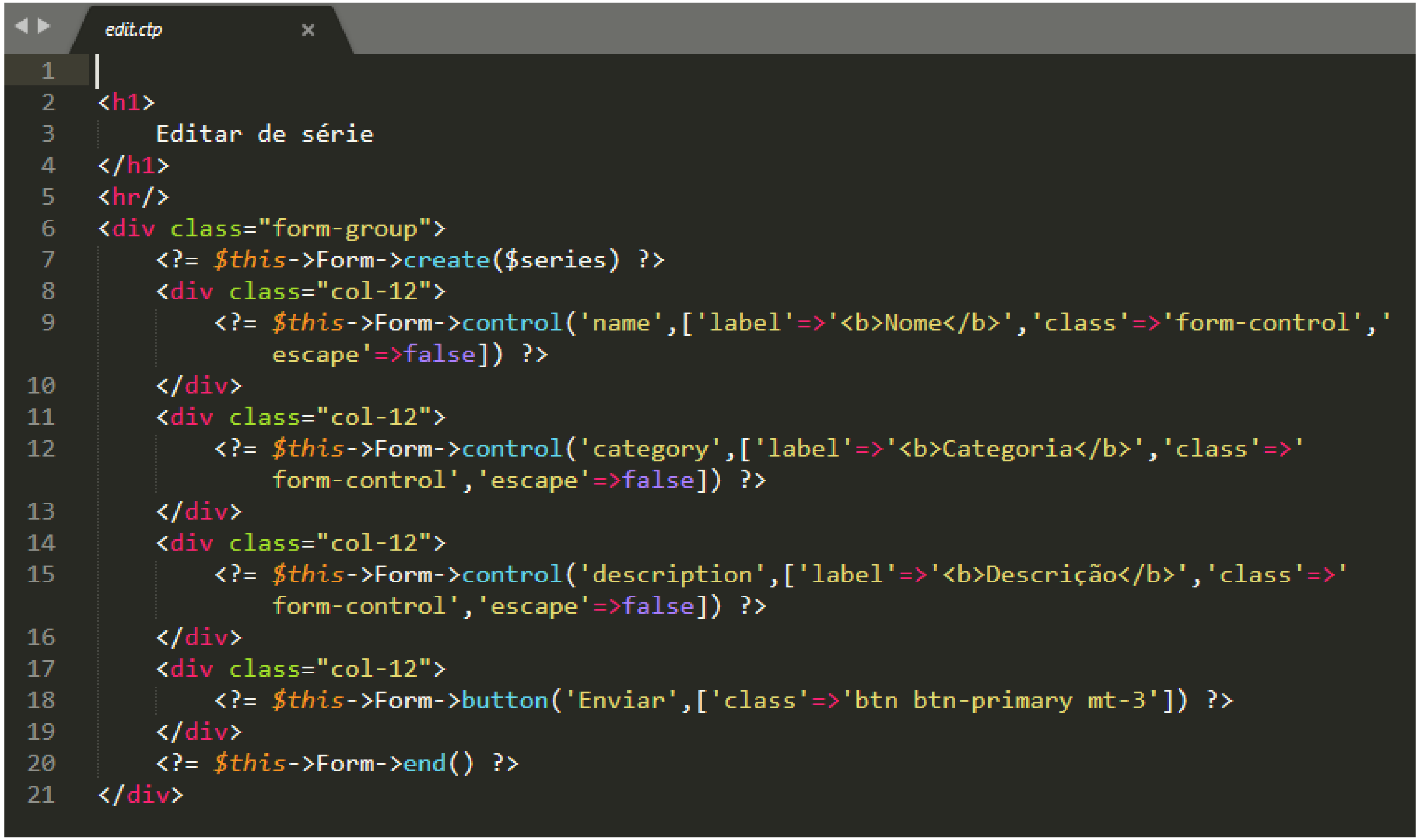
- **View:** Series (add.ctp)



```
<?php
/**
 * @var \App\View\AppView $this
 * @var \App\Model\Entity\Series $series
 */
?>
<h1>
    Cadastro de séries
</h1>
<hr/>
<div class="form-group">
    <?= $this->Form->create($series,['type'=>'POST']) ?>
    <div class="col-12">
        <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('category',['label'=>'<b>Categoria</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
    <?= $this->Form->end() ?>
</div>
```

7. Códigos do projeto (Cont.)

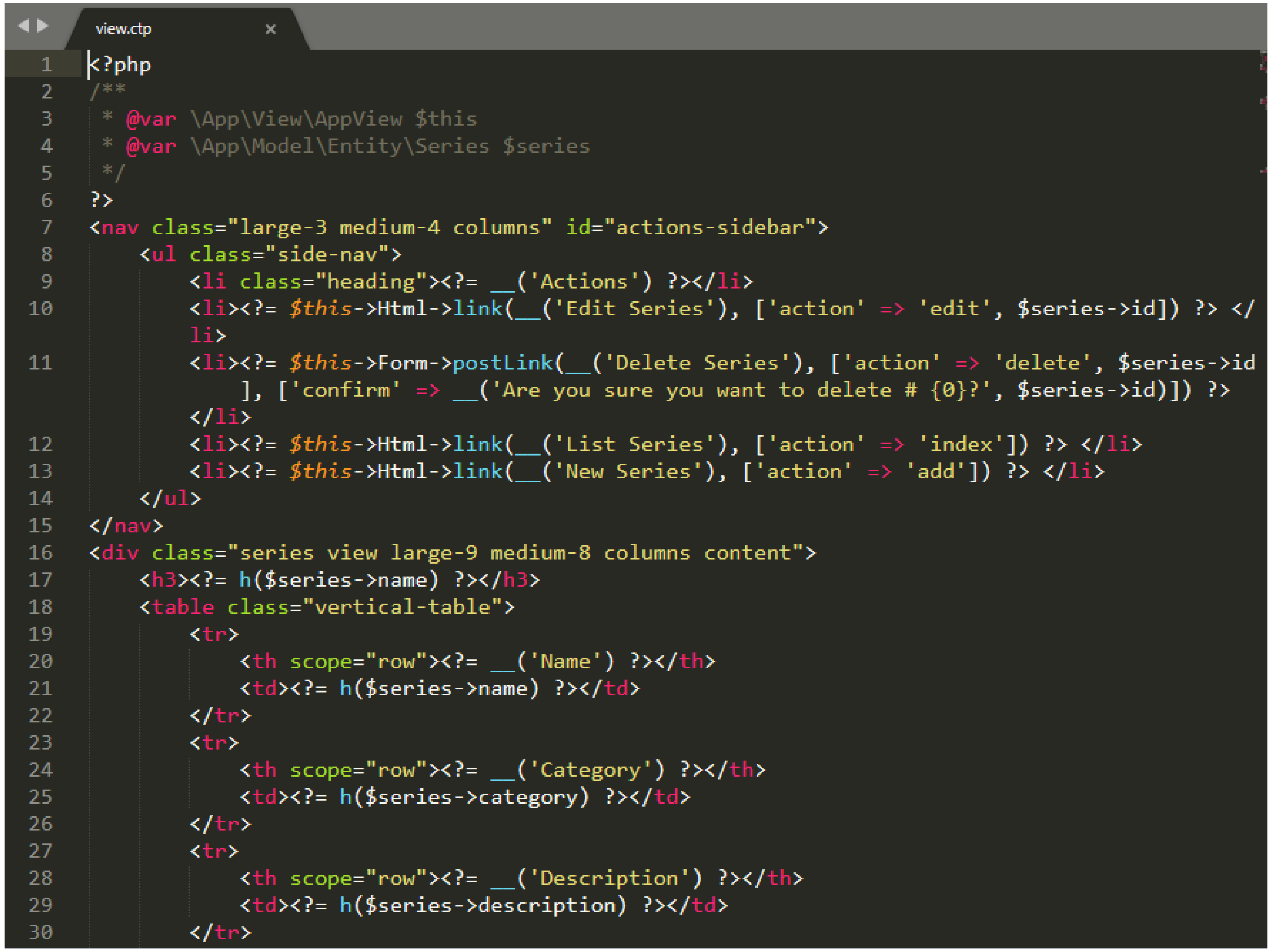
- **View:** Series (edit.ctp)



```

<h1>
    Editar de série
</h1>
<hr/>
<div class="form-group">
    <?= $this->Form->create($series) ?>
    <div class="col-12">
        <?= $this->Form->control('name',['label'=>'<b>Nome</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('category',['label'=>'<b>Categoria</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('description',['label'=>'<b>Descrição</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
    <?= $this->Form->end() ?>
</div>
```

7. Códigos do projeto (Cont.)

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

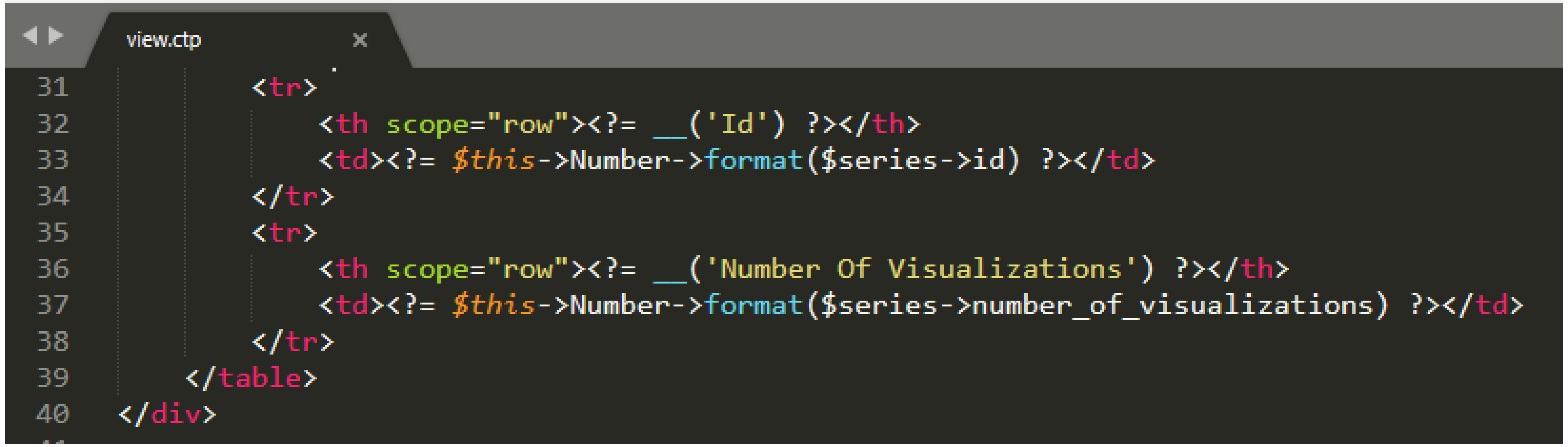


```php
<?php
/**
  * @var \App\View\AppView $this
  * @var \App\Model\Entity\Series $series
  */
?>
<nav class="large-3 medium-4 columns" id="actions-sidebar">
    <ul class="side-nav">
        <li class="heading"><?= __('Actions') ?></li>
        <li><?= $this->Html->link(__('Edit Series'), ['action' => 'edit', $series->id]) ?> </li>
        <li><?= $this->Form->postLink(__('Delete Series'), ['action' => 'delete', $series->id], ['confirm' => __('Are you sure you want to delete # {0}?', $series->id)]) ?> </li>
        <li><?= $this->Html->link(__('List Series'), ['action' => 'index']) ?> </li>
        <li><?= $this->Html->link(__('New Series'), ['action' => 'add']) ?> </li>
    </ul>
</nav>
<div class="series view large-9 medium-8 columns content">
    <h3><?= h($series->name) ?></h3>
    <table class="vertical-table">
        <tr>
            <th scope="row"><?= __('Name') ?></th>
            <td><?= h($series->name) ?></td>
        </tr>
        <tr>
            <th scope="row"><?= __('Category') ?></th>
            <td><?= h($series->category) ?></td>
        </tr>
        <tr>
            <th scope="row"><?= __('Description') ?></th>
            <td><?= h($series->description) ?></td>
        </tr>
```

7. Códigos do projeto (Cont.)

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

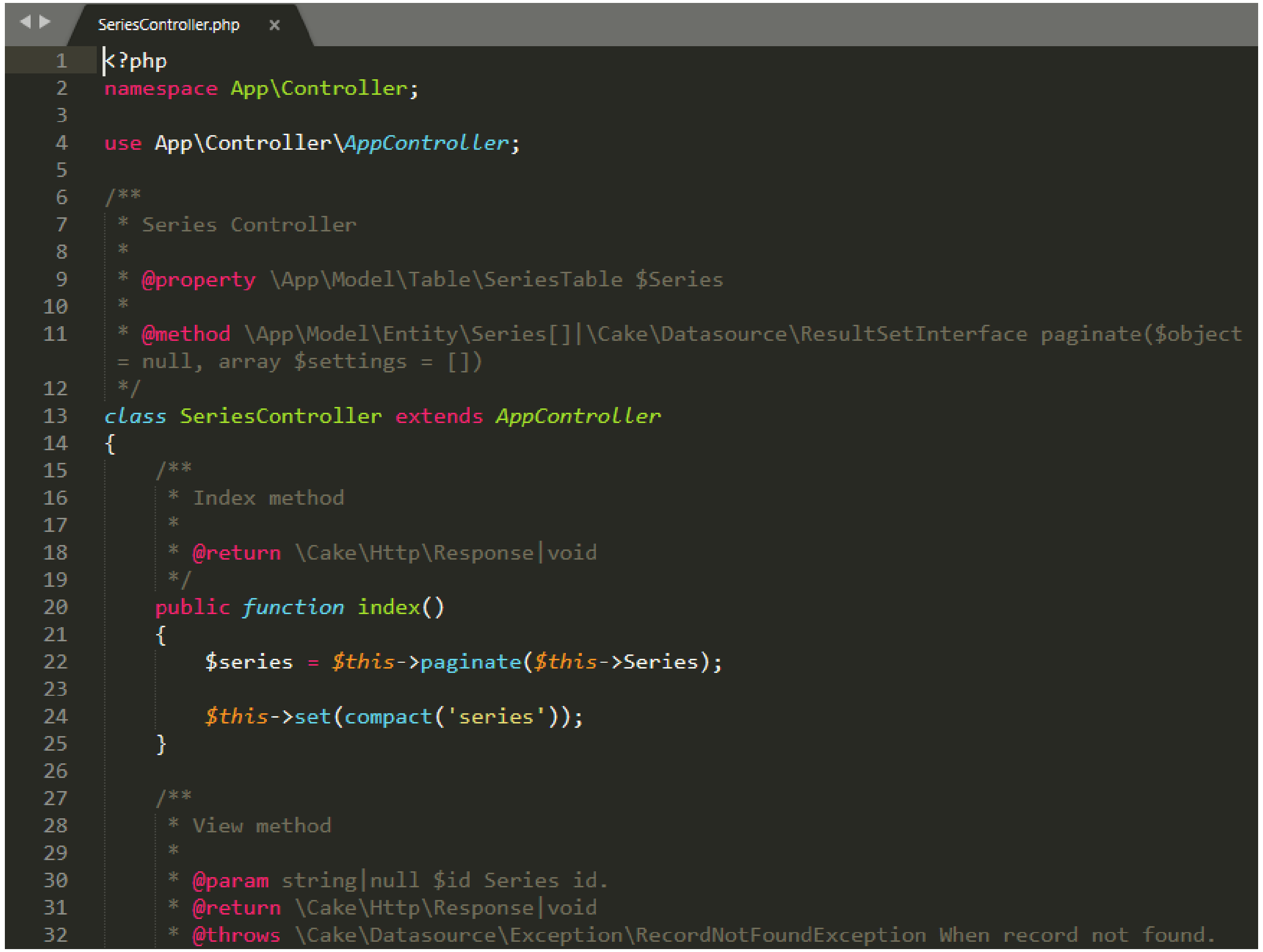
cont.



```
        <tr>
            <th scope="row"><?= __('Id') ?></th>
            <td><?= $this->Number->format($series->id) ?></td>
        </tr>
        <tr>
            <th scope="row"><?= __('Number Of Visualizations') ?></th>
            <td><?= $this->Number->format($series->number_of_visualizations) ?></td>
        </tr>
    </table>
</div>
```

7. Códigos do projeto (Cont.)

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

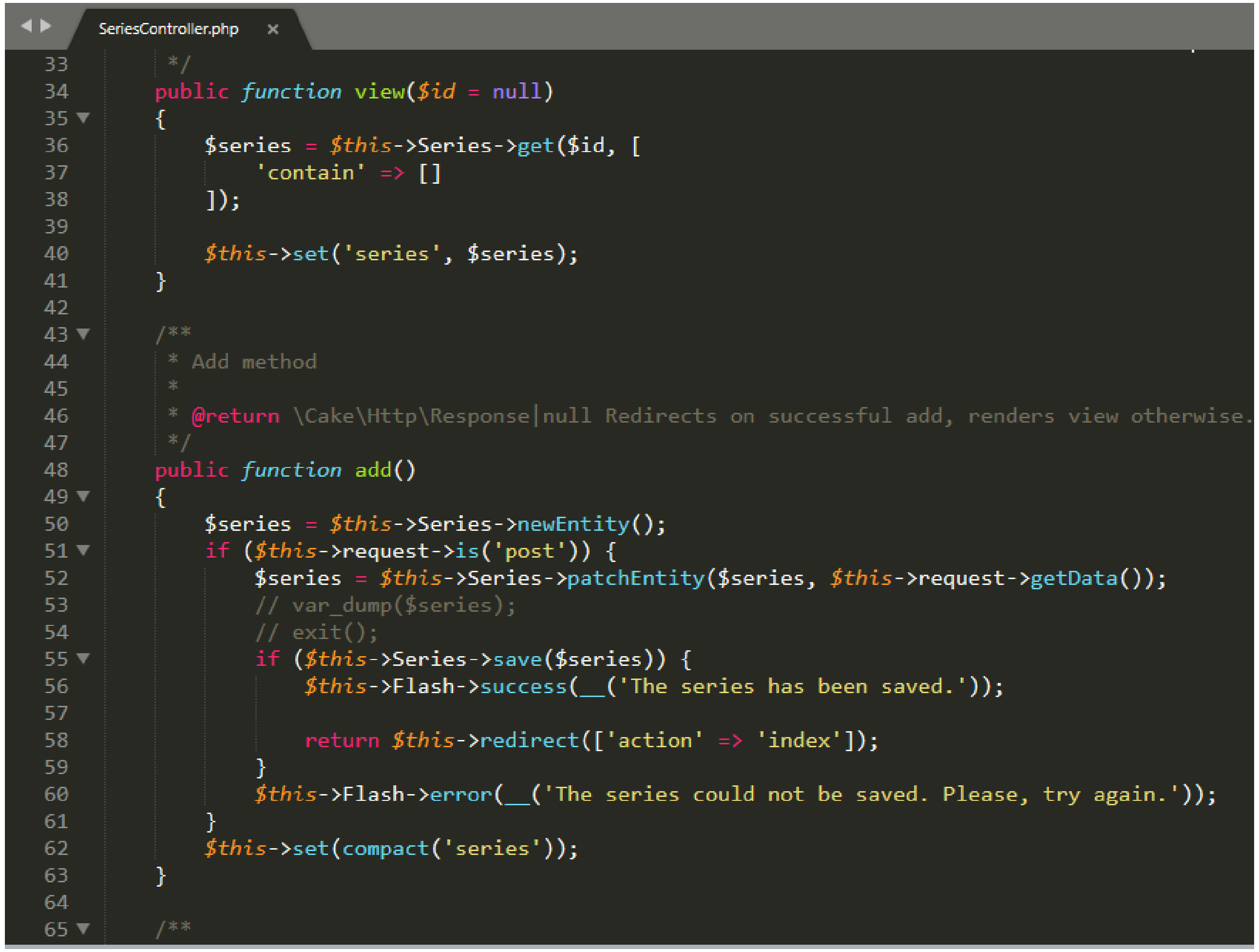


```php
<?php
namespace App\Controller;

use App\Controller\AppController;

/**
 * Series Controller
 *
 * @property \App\Model\Table\SeriesTable $Series
 *
 * @method \App\Model\Entity\Series[]|\Cake\Datasource\ResultSetInterface paginate($object = null, array $settings = [])
 */
class SeriesController extends AppController
{
    /**
     * Index method
     *
     * @return \Cake\Http\Response|void
     */
    public function index()
    {
        $series = $this->paginate($this->Series);

        $this->set(compact('series'));
    }

    /**
     * View method
     *
     * @param string|null $id Series id.
     * @return \Cake\Http\Response|void
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
```

7. Códigos do projeto (Cont.)

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

  cont.

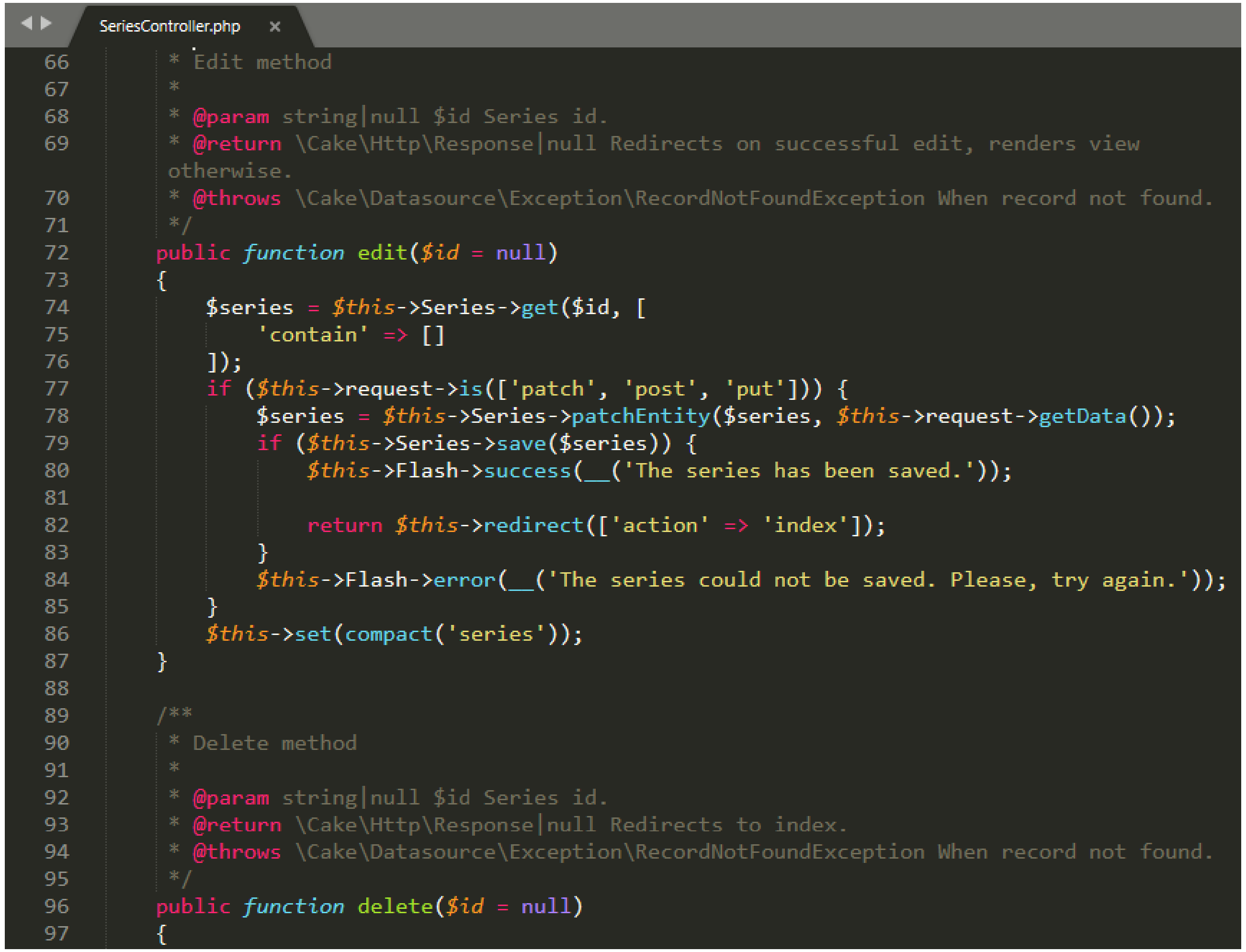


```
     */
    public function view($id = null)
    {
        $series = $this->Series->get($id, [
            'contain' => []
        ]);

        $this->set('series', $series);
    }

    /**
     * Add method
     *
     * @return \Cake\Http\Response|null Redirects on successful add, renders view otherwise.
     */
    public function add()
    {
        $series = $this->Series->newEntity();
        if ($this->request->is('post')) {
            $series = $this->Series->patchEntity($series, $this->request->getData());
            // var_dump($series);
            // exit();
            if ($this->Series->save($series)) {
                $this->Flash->success(__('The series has been saved.'));

                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error(__('The series could not be saved. Please, try again.'));
        }
        $this->set(compact('series'));
    }

    /**
```

7. Códigos do projeto (Cont.)

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

  cont.

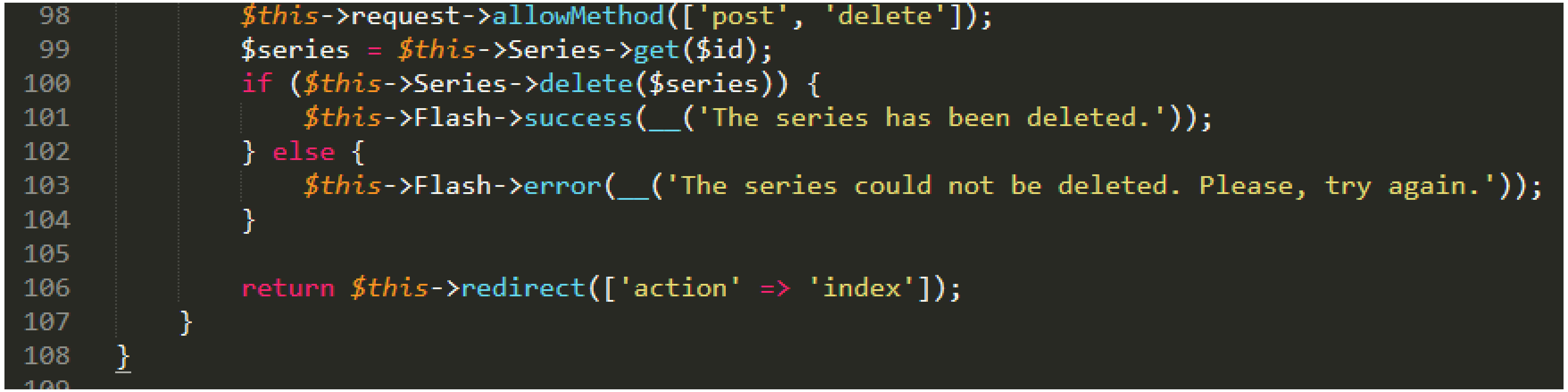


```
     * Edit method
     *
     * @param string|null $id Series id.
     * @return \Cake\Http\Response|null Redirects on successful edit, renders view otherwise.
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
     */
    public function edit($id = null)
    {
        $series = $this->Series->get($id, [
            'contain' => []
        ]);
        if ($this->request->is(['patch', 'post', 'put'])) {
            $series = $this->Series->patchEntity($series, $this->request->getData());
            if ($this->Series->save($series)) {
                $this->Flash->success(__('The series has been saved.'));

                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error(__('The series could not be saved. Please, try again.'));
        }
        $this->set(compact('series'));
    }

    /**
     * Delete method
     *
     * @param string|null $id Series id.
     * @return \Cake\Http\Response|null Redirects to index.
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
     */
    public function delete($id = null)
    {
```

7. Códigos do projeto (Cont.)

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

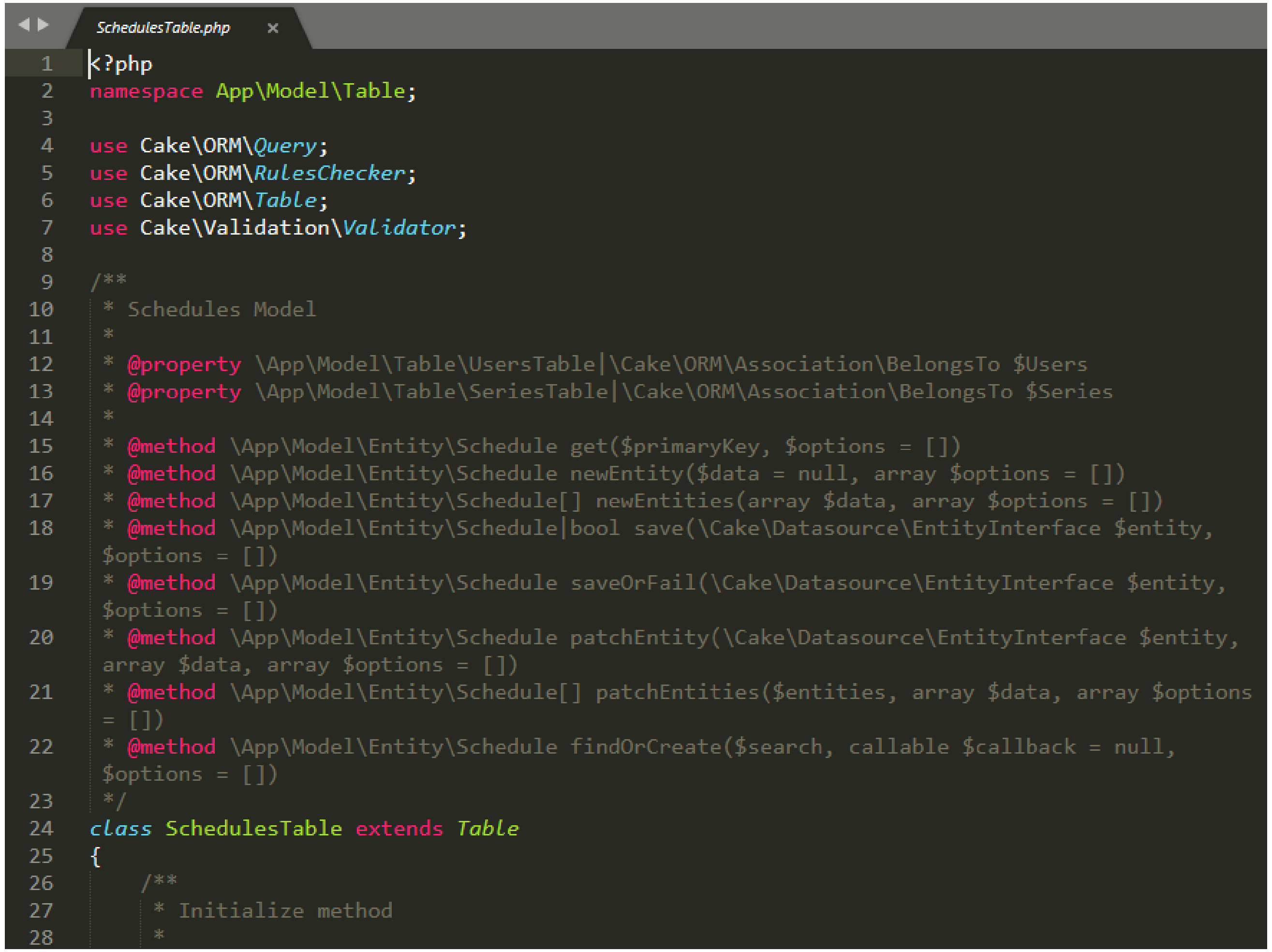
cont.



```php
        $this->request->allowMethod(['post', 'delete']);
        $series = $this->Series->get($id);
        if ($this->Series->delete($series)) {
            $this->Flash->success(__('The series has been deleted.'));
        } else {
            $this->Flash->error(__('The series could not be deleted. Please, try again.'));
        }

        return $this->redirect(['action' => 'index']);
    }
}
```

7. Códigos do projeto (Cont.)

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

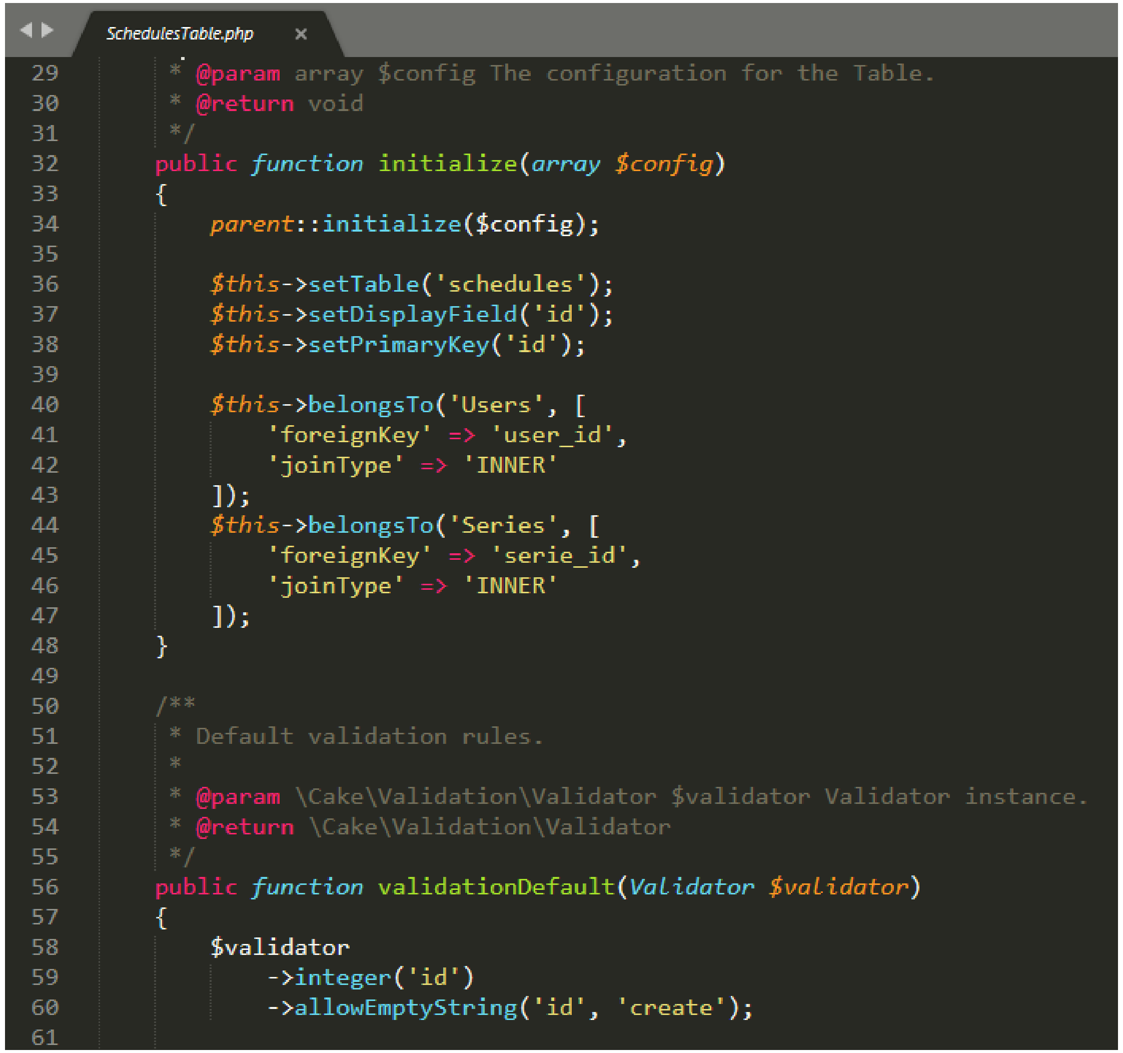


```php
<?php
namespace App\Model\Table;

use Cake\ORM\Query;
use Cake\ORM\RulesChecker;
use Cake\ORM\Table;
use Cake\Validation\Validator;

/**
 * Schedules Model
 *
 * @property \App\Model\Table\UsersTable|\Cake\ORM\Association\BelongsTo $Users
 * @property \App\Model\Table\SeriesTable|\Cake\ORM\Association\BelongsTo $Series
 *
 * @method \App\Model\Entity\Schedule get($primaryKey, $options = [])
 * @method \App\Model\Entity\Schedule newEntity($data = null, array $options = [])
 * @method \App\Model\Entity\Schedule[] newEntities(array $data, array $options = [])
 * @method \App\Model\Entity\Schedule|bool save(\Cake\Datasource\EntityInterface $entity, $options = [])
 * @method \App\Model\Entity\Schedule saveOrFail(\Cake\Datasource\EntityInterface $entity, $options = [])
 * @method \App\Model\Entity\Schedule patchEntity(\Cake\Datasource\EntityInterface $entity, array $data, array $options = [])
 * @method \App\Model\Entity\Schedule[] patchEntities($entities, array $data, array $options = [])
 * @method \App\Model\Entity\Schedule findOrCreate($search, callable $callback = null, $options = [])
 */
class SchedulesTable extends Table
{
    /**
     * Initialize method
     *
```

7. Códigos do projeto (Cont.)

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

  cont.

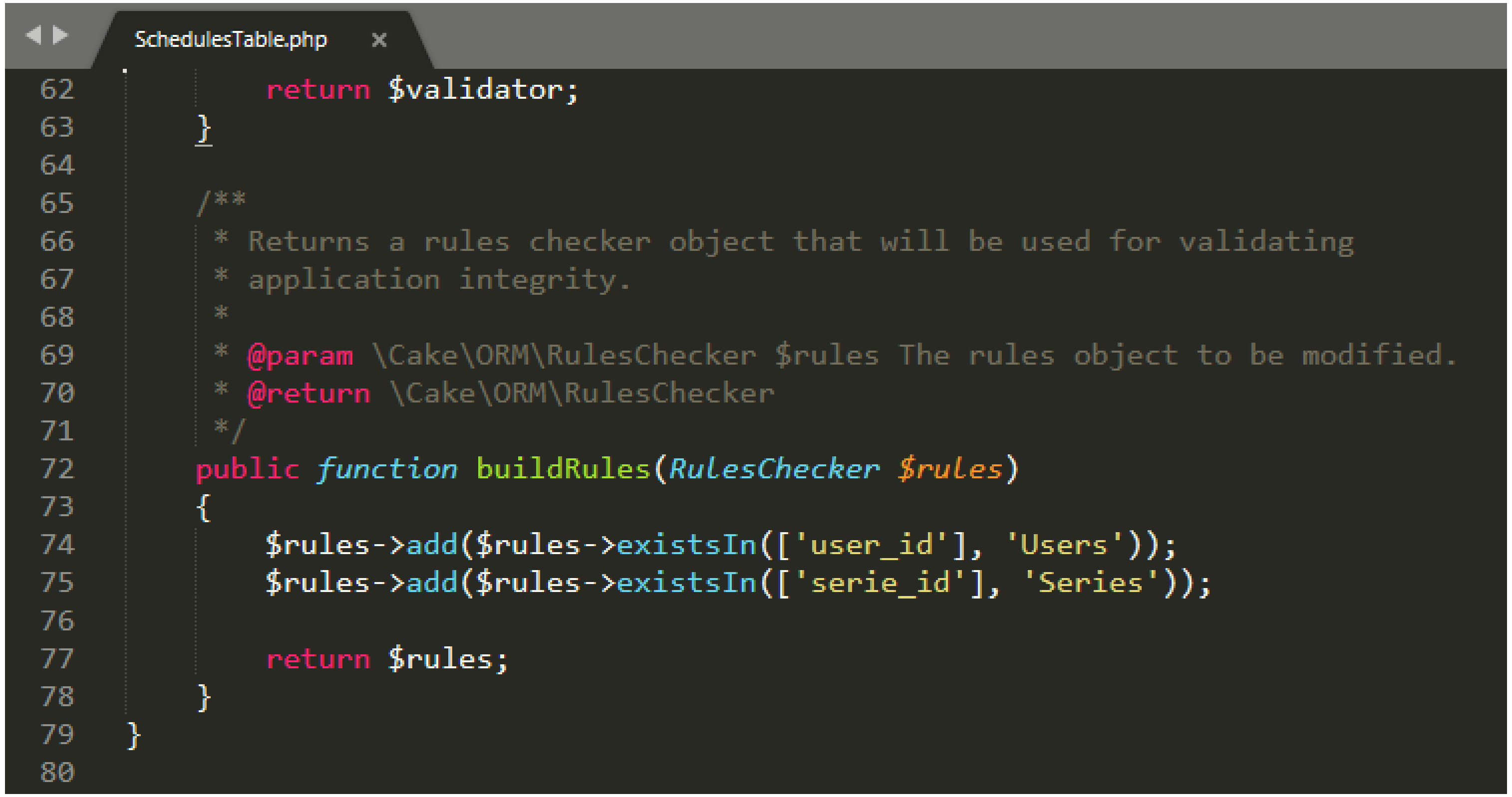


```php
     * @param array $config The configuration for the Table.
     * @return void
     */
    public function initialize(array $config)
    {
        parent::initialize($config);

        $this->setTable('schedules');
        $this->setDisplayField('id');
        $this->setPrimaryKey('id');

        $this->belongsTo('Users', [
            'foreignKey' => 'user_id',
            'joinType' => 'INNER'
        ]);
        $this->belongsTo('Series', [
            'foreignKey' => 'serie_id',
            'joinType' => 'INNER'
        ]);
    }

    /**
     * Default validation rules.
     *
     * @param \Cake\Validation\Validator $validator Validator instance.
     * @return \Cake\Validation\Validator
     */
    public function validationDefault(Validator $validator)
    {
        $validator
            ->integer('id')
            ->allowEmptyString('id', 'create');

```

7. Códigos do projeto (Cont.)

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

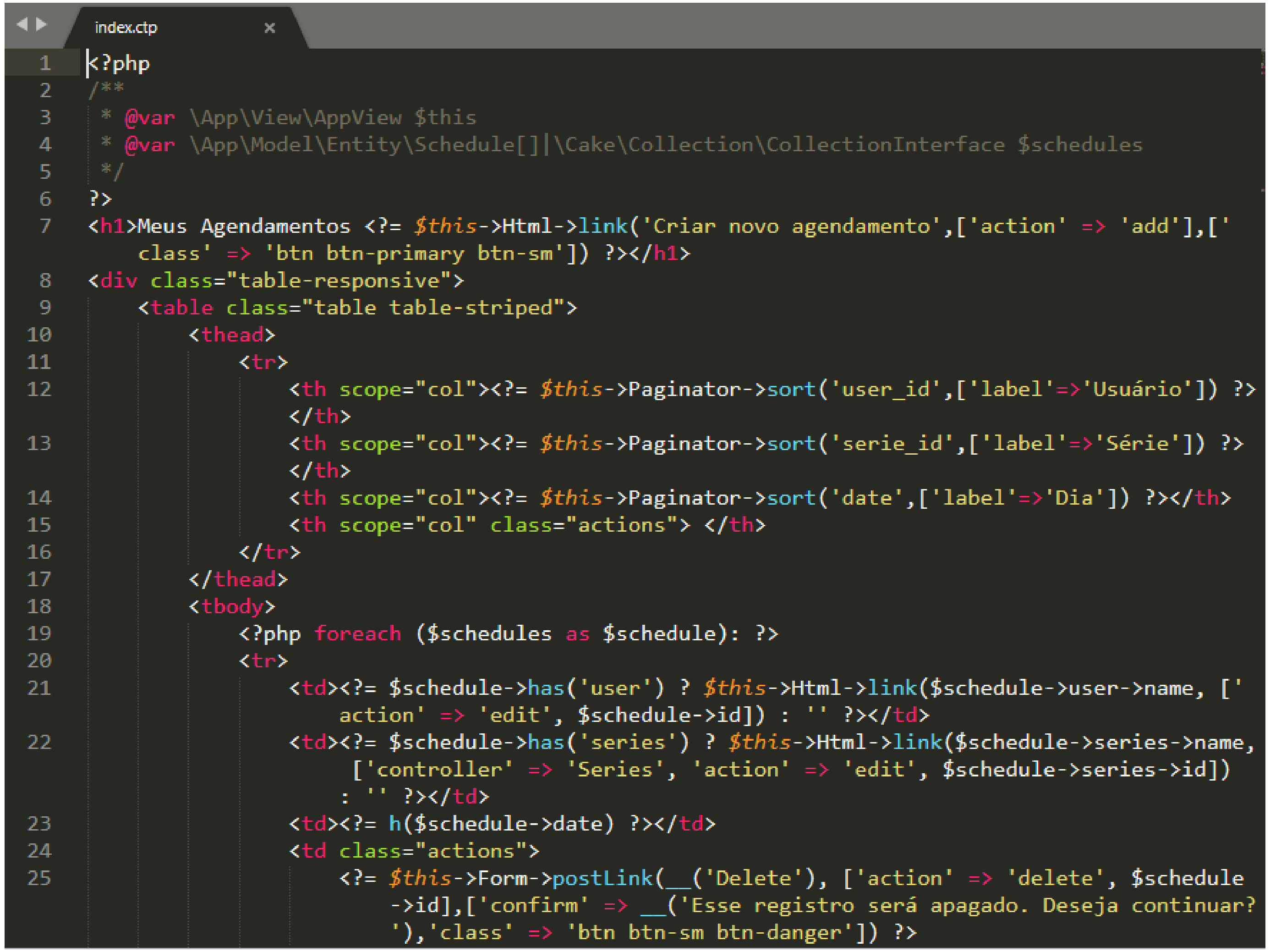
cont.



```php
        return $validator;
    }

    /**
     * Returns a rules checker object that will be used for validating
     * application integrity.
     *
     * @param \Cake\ORM\RulesChecker $rules The rules object to be modified.
     * @return \Cake\ORM\RulesChecker
     */
    public function buildRules(RulesChecker $rules)
    {
        $rules->add($rules->existsIn(['user_id'], 'Users'));
        $rules->add($rules->existsIn(['serie_id'], 'Series'));

        return $rules;
    }
}
```

7. Códigos do projeto (Cont.)

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

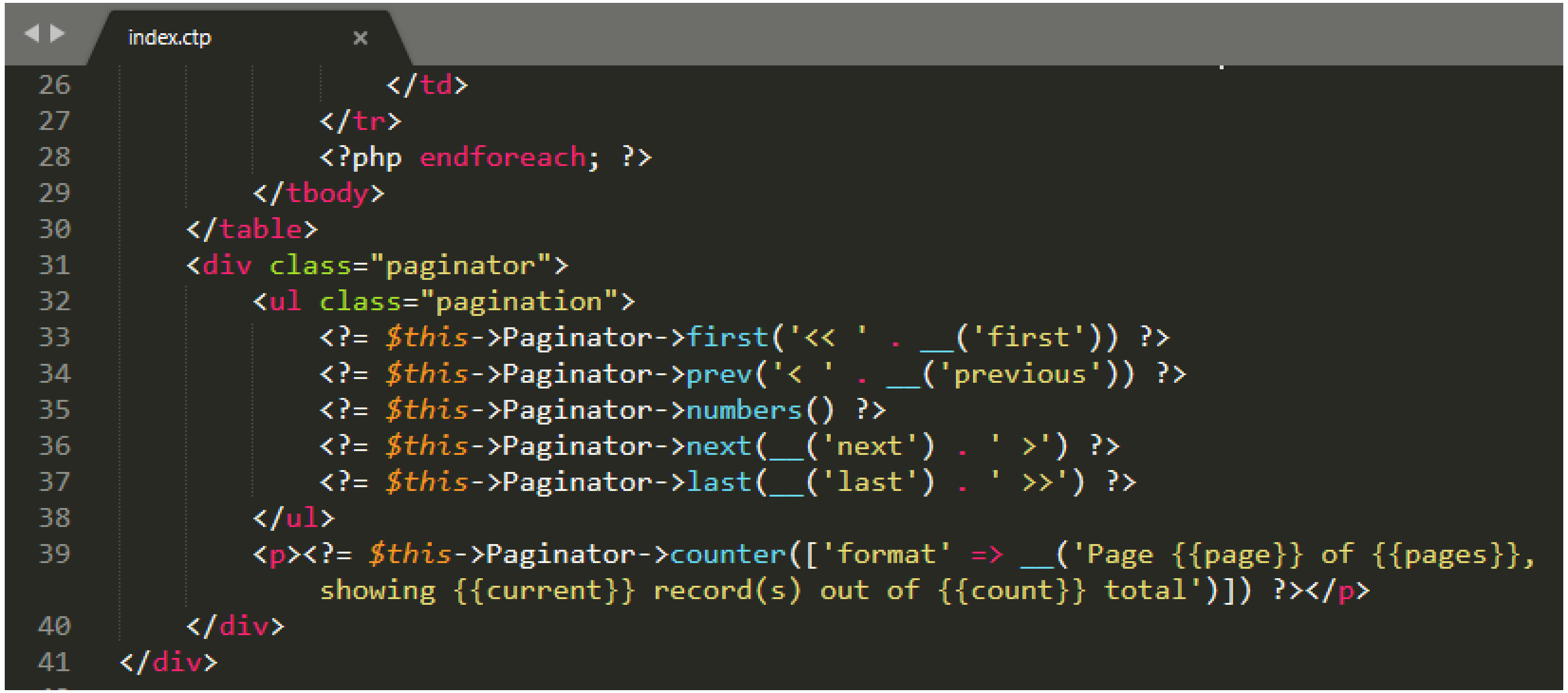


```
<?php
/**
 * @var \App\View\AppView $this
 * @var \App\Model\Entity\Schedule[]|\Cake\Collection\CollectionInterface $schedules
 */
?>
<h1>Meus Agendamentos <?= $this->Html->link('Criar novo agendamento',['action' => 'add'],['class' => 'btn btn-primary btn-sm']) ?></h1>
<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col"><?= $this->Paginator->sort('user_id',['label'=>'Usuário']) ?></th>
                <th scope="col"><?= $this->Paginator->sort('serie_id',['label'=>'Série']) ?></th>
                <th scope="col"><?= $this->Paginator->sort('date',['label'=>'Dia']) ?></th>
                <th scope="col" class="actions"> </th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($schedules as $schedule): ?>
            <tr>
                <td><?= $schedule->has('user') ? $this->Html->link($schedule->user->name, ['action' => 'edit', $schedule->id]) : '' ?></td>
                <td><?= $schedule->has('series') ? $this->Html->link($schedule->series->name, ['controller' => 'Series', 'action' => 'edit', $schedule->series->id]) : '' ?></td>
                <td><?= h($schedule->date) ?></td>
                <td class="actions">
                    <?= $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)

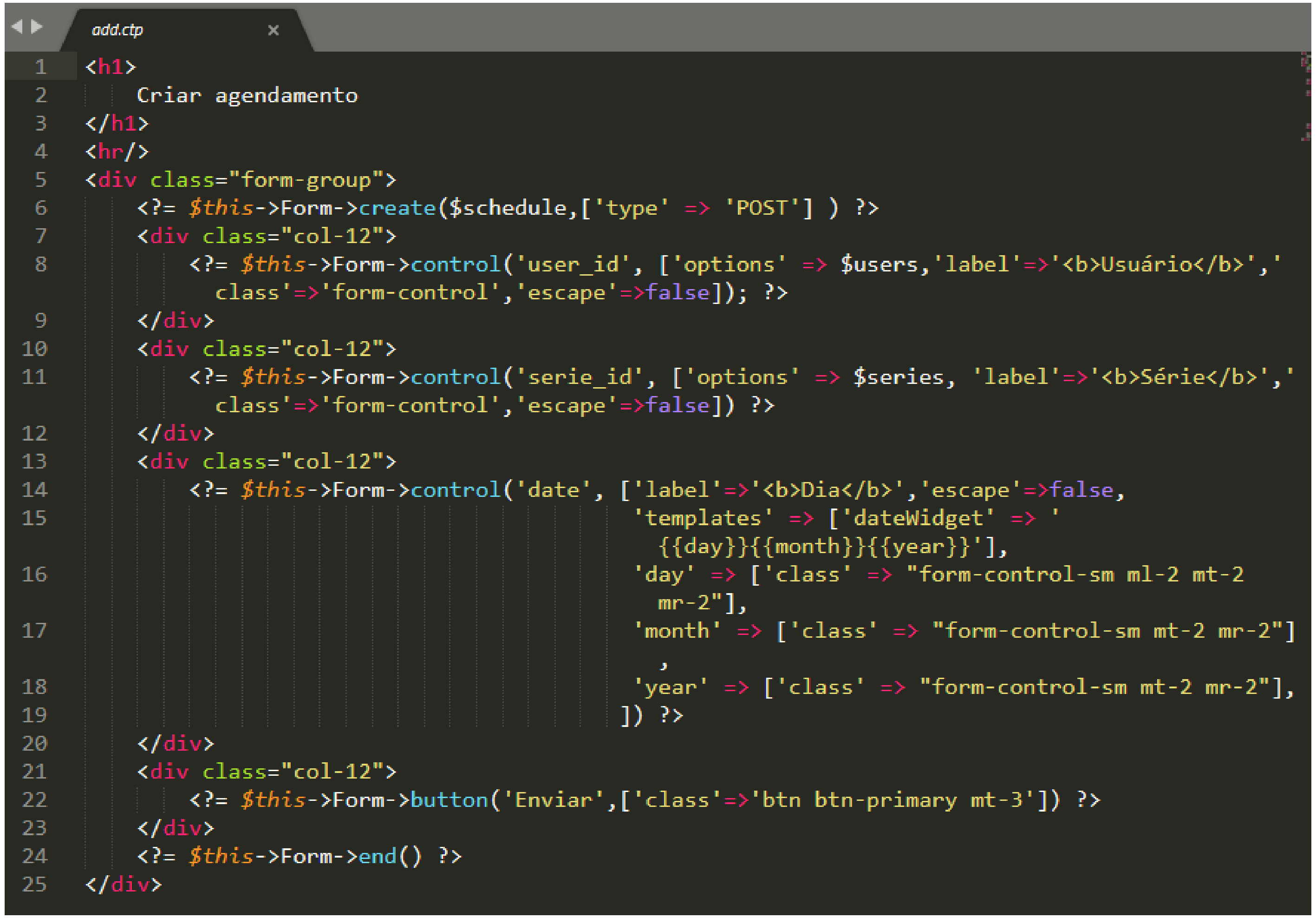
cont.



```
                </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
    <div class="paginator">
        <ul class="pagination">
            <?= $this->Paginator->first('<< ' . __('first')) ?>
            <?= $this->Paginator->prev('< ' . __('previous')) ?>
            <?= $this->Paginator->numbers() ?>
            <?= $this->Paginator->next(__('next') . ' >') ?>
            <?= $this->Paginator->last(__('last') . ' >>') ?>
        </ul>
        <p><?= $this->Paginator->counter(['format' => __('Page {{page}} of {{pages}}, showing {{current}} record(s) out of {{count}} total')]) ?></p>
    </div>
</div>
```

7. Códigos do projeto (Cont.)

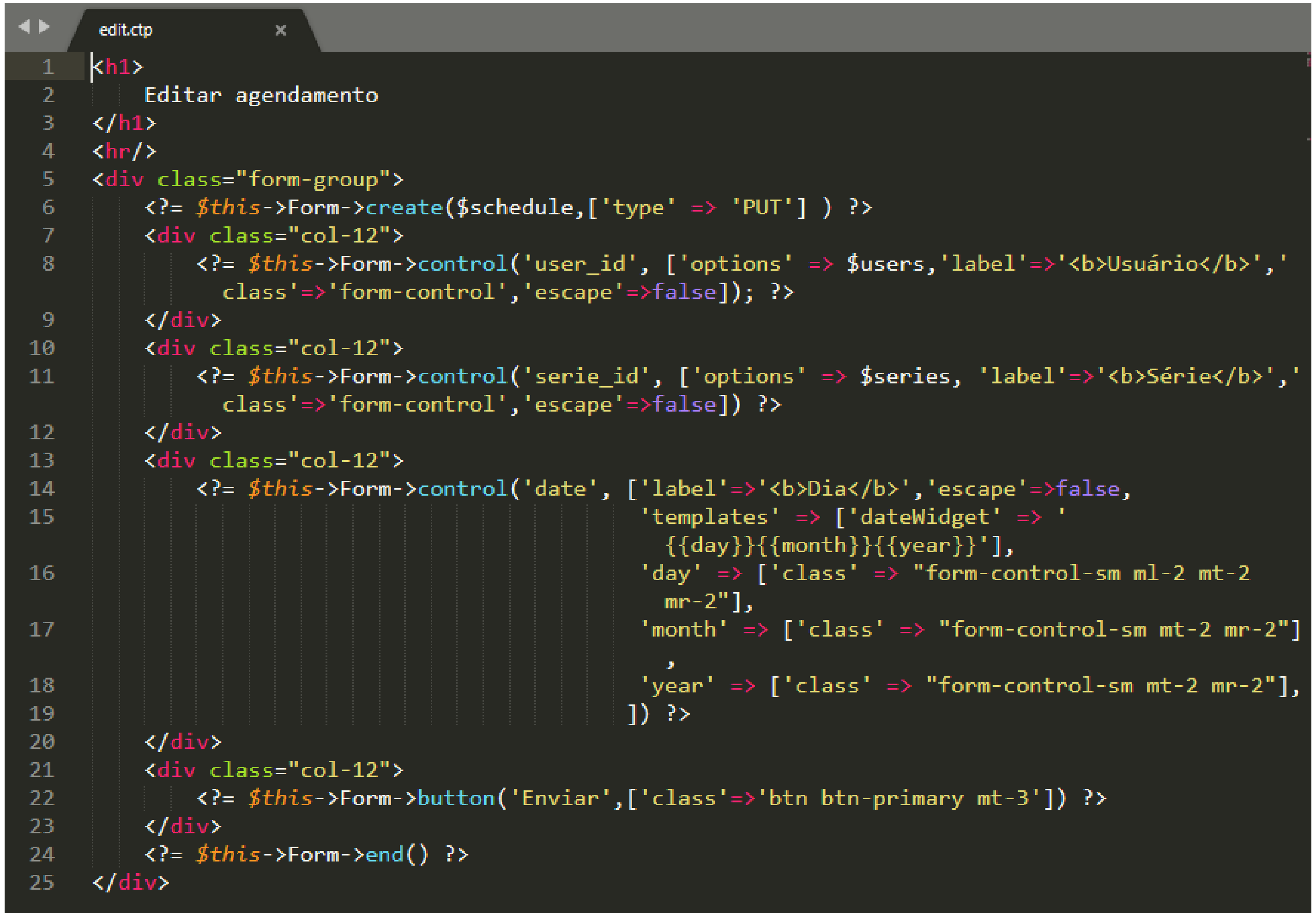
- **View:** Schedules (add.ctp)



```
<h1>
    Criar agendamento
</h1>
<hr/>
<div class="form-group">
    <?= $this->Form->create($schedule,['type' => 'POST'] ) ?>
    <div class="col-12">
        <?= $this->Form->control('user_id', ['options' => $users,'label'=>'<b>Usuário</b>','class'=>'form-control','escape'=>false]); ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('serie_id', ['options' => $series, 'label'=>'<b>Série</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('date', ['label'=>'<b>Dia</b>','escape'=>false,
                                          'templates' => ['dateWidget' => '{{day}}{{month}}{{year}}'],
                                          'day' => ['class' => "form-control-sm ml-2 mt-2 mr-2"],
                                          'month' => ['class' => "form-control-sm mt-2 mr-2"],
                                          'year' => ['class' => "form-control-sm mt-2 mr-2"],
                                         ]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
    <?= $this->Form->end() ?>
</div>
```

7. Códigos do projeto (Cont.)

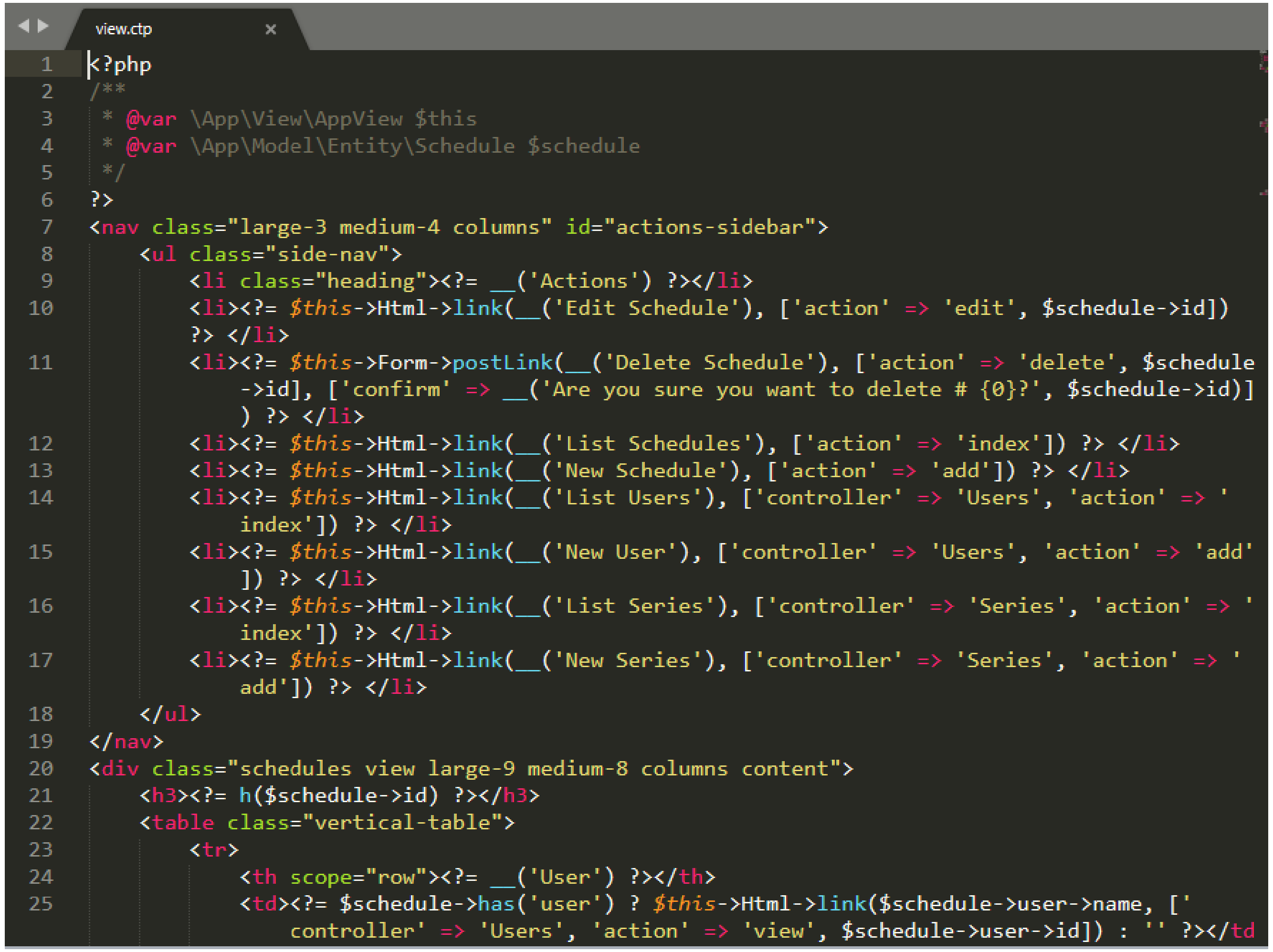
- **View:** Schedules (edit.ctp)



```
<h1>
    Editar agendamento
</h1>
<hr/>
<div class="form-group">
    <?= $this->Form->create($schedule,['type' => 'PUT'] ) ?>
    <div class="col-12">
        <?= $this->Form->control('user_id', ['options' => $users,'label'=>'<b>Usuário</b>','class'=>'form-control','escape'=>false]); ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('serie_id', ['options' => $series, 'label'=>'<b>Série</b>','class'=>'form-control','escape'=>false]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->control('date', ['label'=>'<b>Dia</b>','escape'=>false,
                                          'templates' => ['dateWidget' => '{{day}}{{month}}{{year}}'],
                                          'day' => ['class' => "form-control-sm ml-2 mt-2 mr-2"],
                                          'month' => ['class' => "form-control-sm mt-2 mr-2"],
                                          'year' => ['class' => "form-control-sm mt-2 mr-2"],
                                         ]) ?>
    </div>
    <div class="col-12">
        <?= $this->Form->button('Enviar',['class'=>'btn btn-primary mt-3']) ?>
    </div>
    <?= $this->Form->end() ?>
</div>
```

7. Códigos do projeto (Cont.)

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

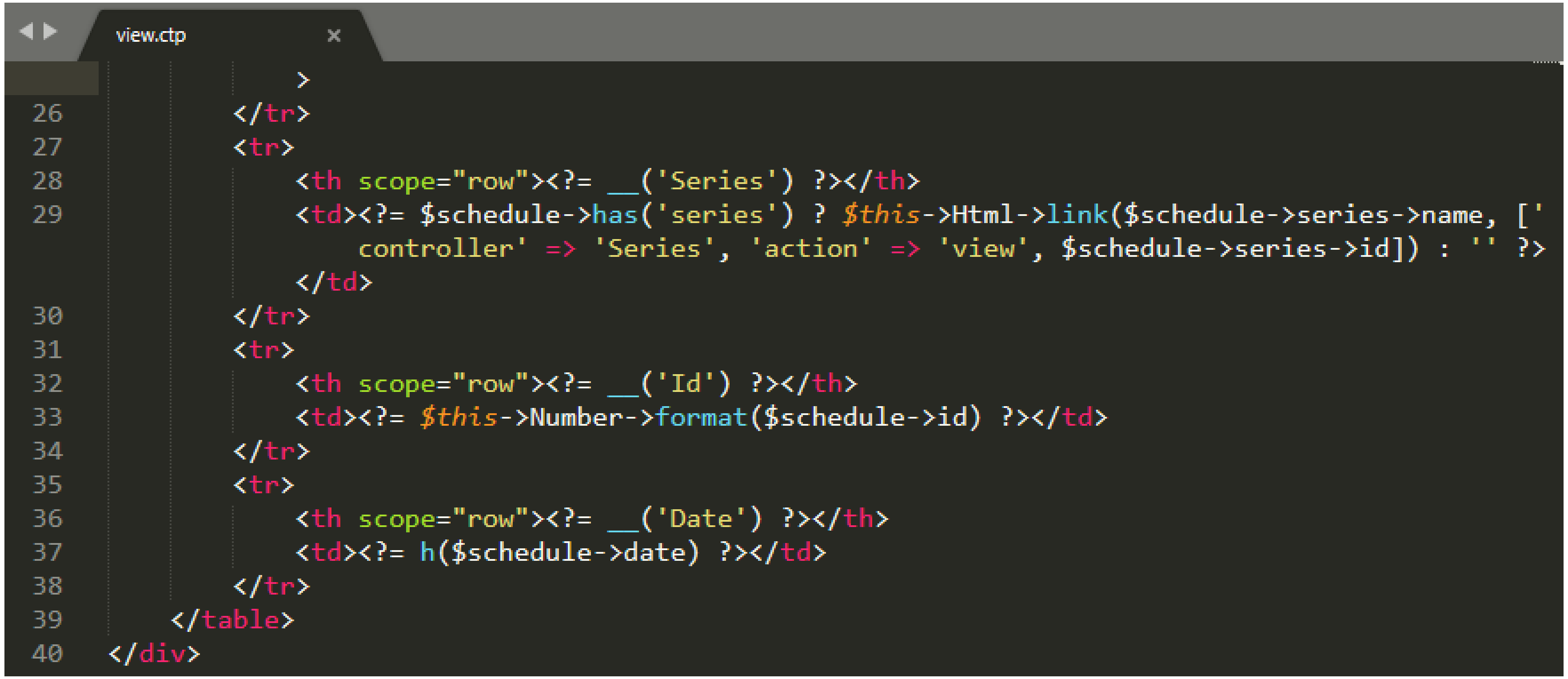


```
<?php
/**
 * @var \App\View\AppView $this
 * @var \App\Model\Entity\Schedule $schedule
 */
?>
<nav class="large-3 medium-4 columns" id="actions-sidebar">
    <ul class="side-nav">
        <li class="heading"><?= __('Actions') ?></li>
        <li><?= $this->Html->link(__('Edit Schedule'), ['action' => 'edit', $schedule->id]) ?> </li>
        <li><?= $this->Form->postLink(__('Delete Schedule'), ['action' => 'delete', $schedule->id], ['confirm' => __('Are you sure you want to delete # {0}?', $schedule->id)]) ?> </li>
        <li><?= $this->Html->link(__('List Schedules'), ['action' => 'index']) ?> </li>
        <li><?= $this->Html->link(__('New Schedule'), ['action' => 'add']) ?> </li>
        <li><?= $this->Html->link(__('List Users'), ['controller' => 'Users', 'action' => 'index']) ?> </li>
        <li><?= $this->Html->link(__('New User'), ['controller' => 'Users', 'action' => 'add']) ?> </li>
        <li><?= $this->Html->link(__('List Series'), ['controller' => 'Series', 'action' => 'index']) ?> </li>
        <li><?= $this->Html->link(__('New Series'), ['controller' => 'Series', 'action' => 'add']) ?> </li>
    </ul>
</nav>
<div class="schedules view large-9 medium-8 columns content">
    <h3><?= h($schedule->id) ?></h3>
    <table class="vertical-table">
        <tr>
            <th scope="row"><?= __('User') ?></th>
            <td><?= $schedule->has('user') ? $this->Html->link($schedule->user->name, ['controller' => 'Users', 'action' => 'view', $schedule->user->id]) : '' ?></td
```

7. Códigos do projeto (Cont.)

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

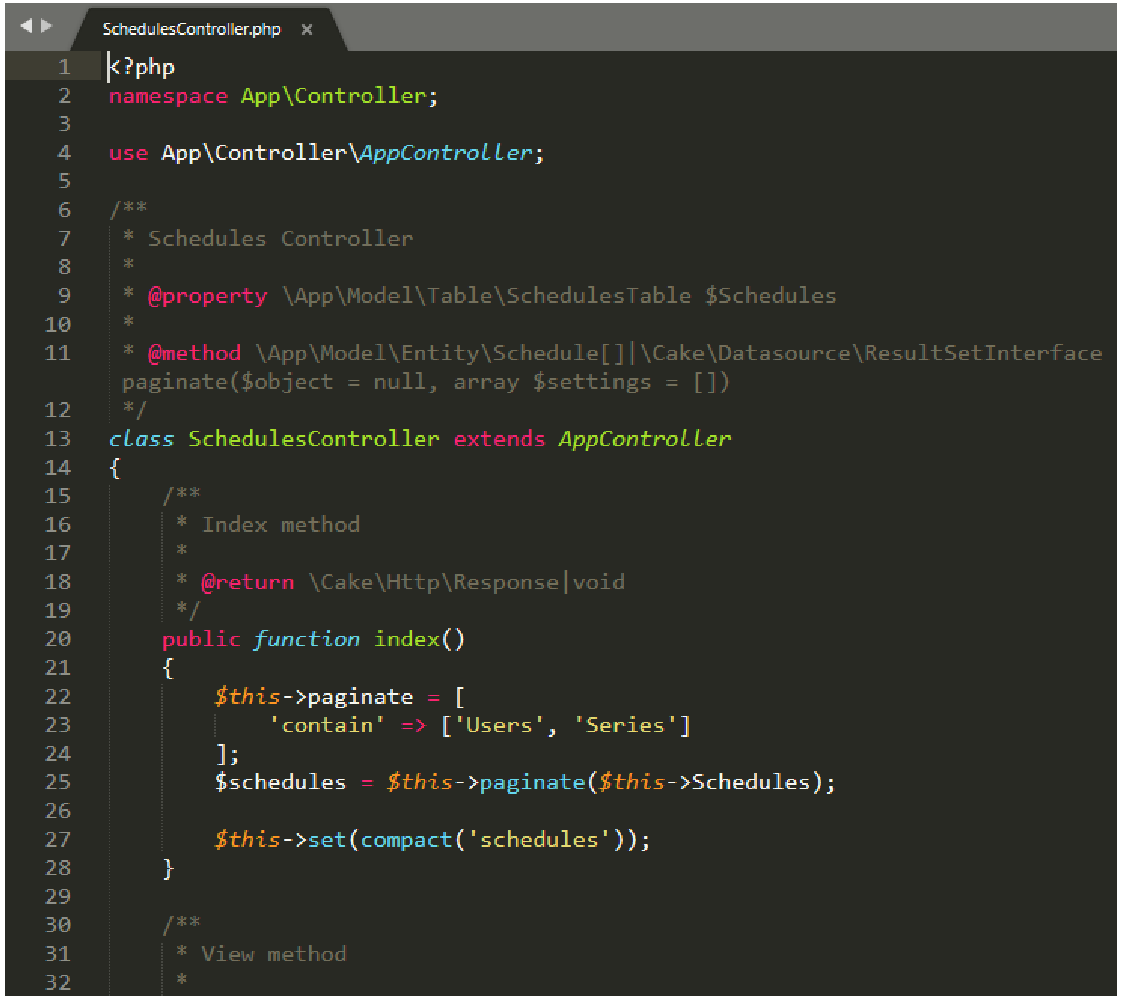
cont.



```
            >
        </tr>
        <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)

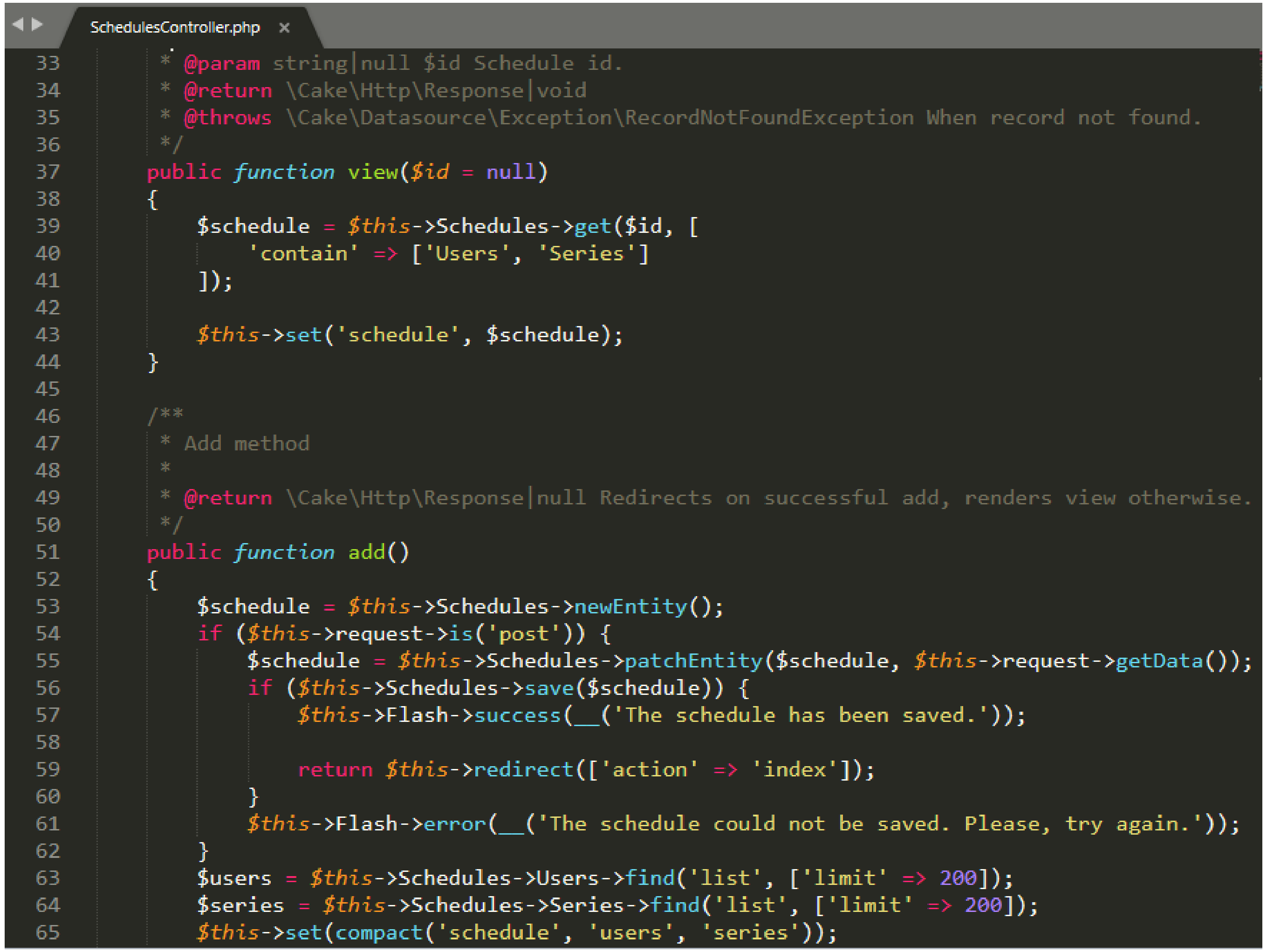


```php
<?php
namespace App\Controller;

use App\Controller\AppController;

/**
 * Schedules Controller
 *
 * @property \App\Model\Table\SchedulesTable $Schedules
 *
 * @method \App\Model\Entity\Schedule[]|\Cake\Datasource\ResultSetInterface paginate($object = null, array $settings = [])
 */
class SchedulesController extends AppController
{
    /**
     * Index method
     *
     * @return \Cake\Http\Response|void
     */
    public function index()
    {
        $this->paginate = [
            'contain' => ['Users', 'Series']
        ];
        $schedules = $this->paginate($this->Schedules);

        $this->set(compact('schedules'));
    }

    /**
     * View method
     *
```

7. Códigos do projeto (Cont.)

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

  cont.

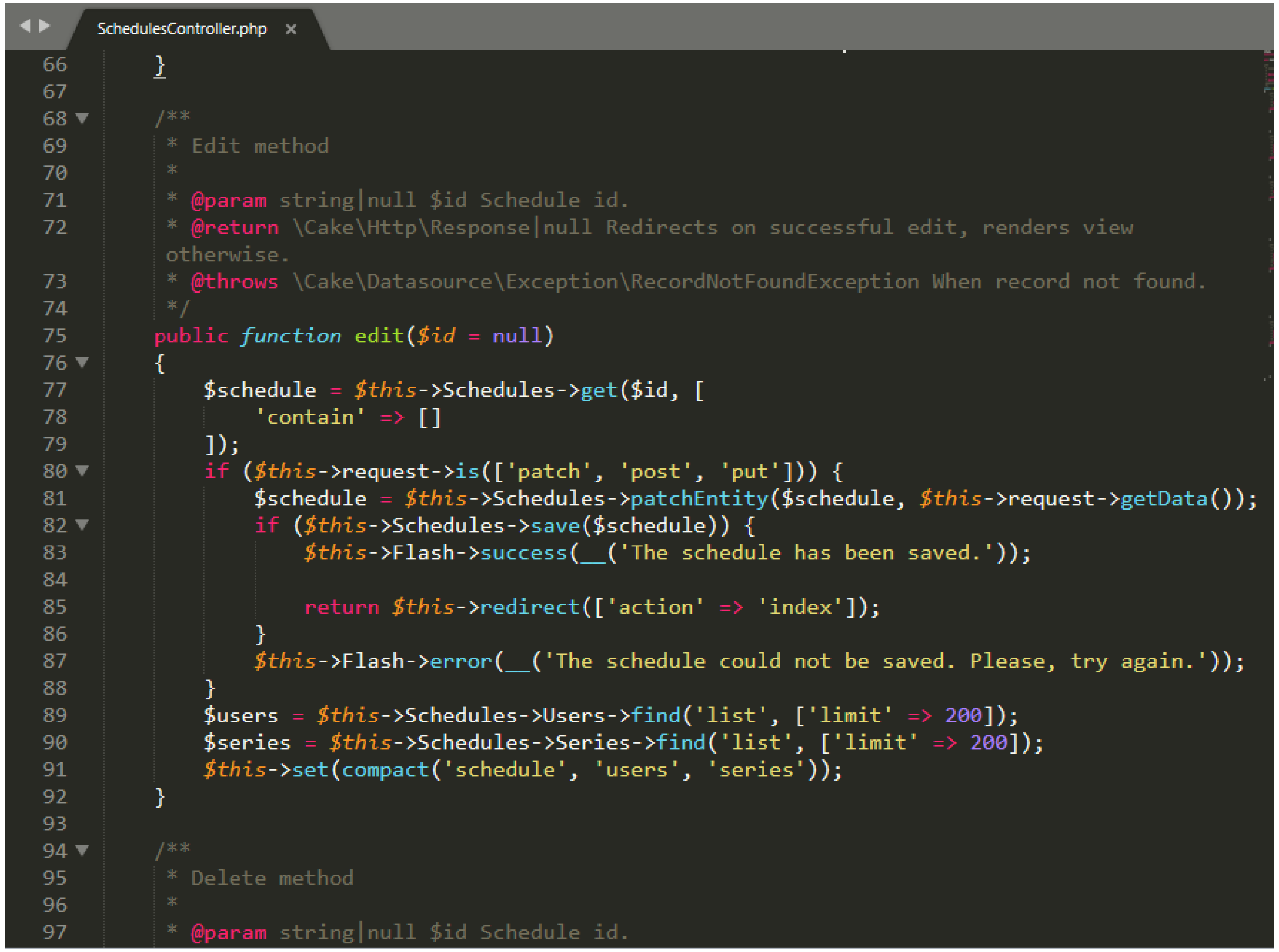


```php
     * @param string|null $id Schedule id.
     * @return \Cake\Http\Response|void
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
     */
    public function view($id = null)
    {
        $schedule = $this->Schedules->get($id, [
            'contain' => ['Users', 'Series']
        ]);

        $this->set('schedule', $schedule);
    }

    /**
     * Add method
     *
     * @return \Cake\Http\Response|null Redirects on successful add, renders view otherwise.
     */
    public function add()
    {
        $schedule = $this->Schedules->newEntity();
        if ($this->request->is('post')) {
            $schedule = $this->Schedules->patchEntity($schedule, $this->request->getData());
            if ($this->Schedules->save($schedule)) {
                $this->Flash->success(__('The schedule has been saved.'));

                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error(__('The schedule could not be saved. Please, try again.'));
        }
        $users = $this->Schedules->Users->find('list', ['limit' => 200]);
        $series = $this->Schedules->Series->find('list', ['limit' => 200]);
        $this->set(compact('schedule', 'users', 'series'));
```

7. Códigos do projeto (Cont.)

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

  cont.

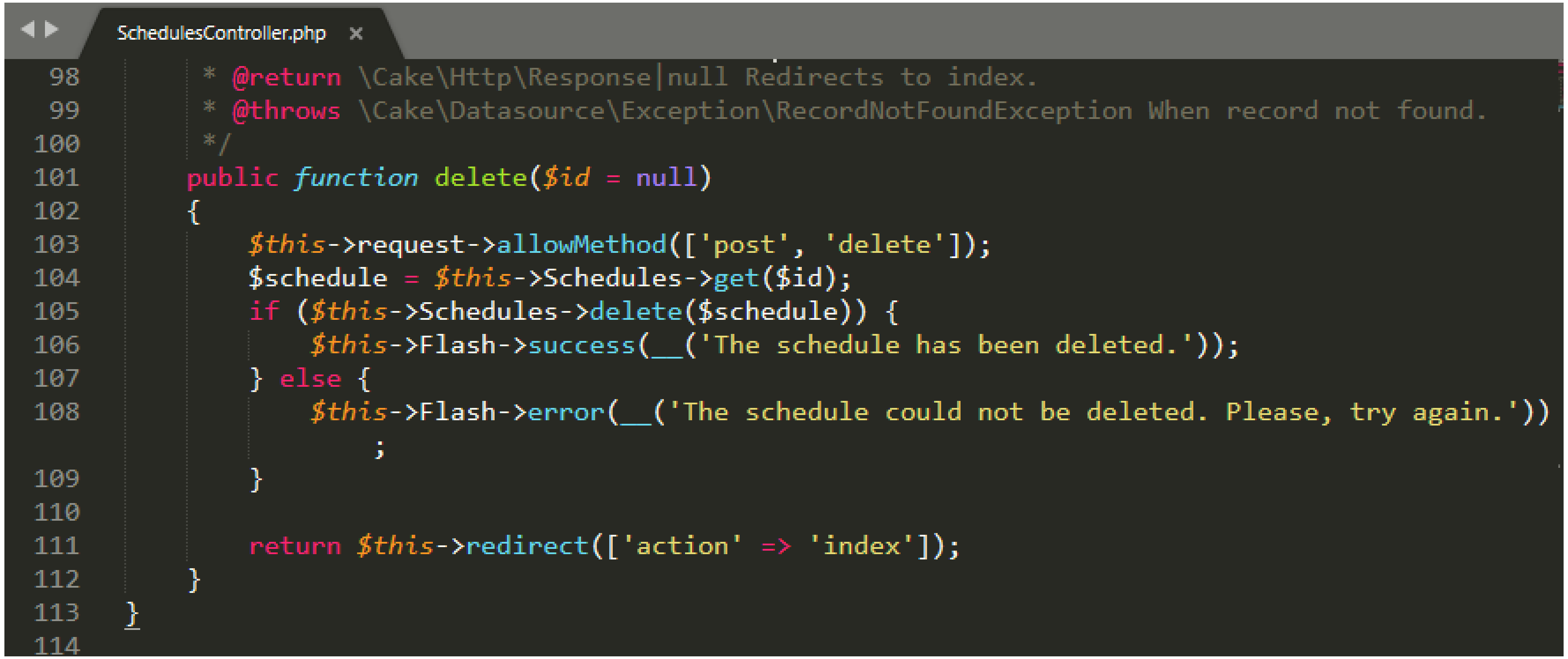


```php
    }

    /**
     * Edit method
     *
     * @param string|null $id Schedule id.
     * @return \Cake\Http\Response|null Redirects on successful edit, renders view otherwise.
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
     */
    public function edit($id = null)
    {
        $schedule = $this->Schedules->get($id, [
            'contain' => []
        ]);
        if ($this->request->is(['patch', 'post', 'put'])) {
            $schedule = $this->Schedules->patchEntity($schedule, $this->request->getData());
            if ($this->Schedules->save($schedule)) {
                $this->Flash->success(__('The schedule has been saved.'));

                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error(__('The schedule could not be saved. Please, try again.'));
        }
        $users = $this->Schedules->Users->find('list', ['limit' => 200]);
        $series = $this->Schedules->Series->find('list', ['limit' => 200]);
        $this->set(compact('schedule', 'users', 'series'));
    }

    /**
     * Delete method
     *
     * @param string|null $id Schedule id.
```

7. Códigos do projeto (Cont.)

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

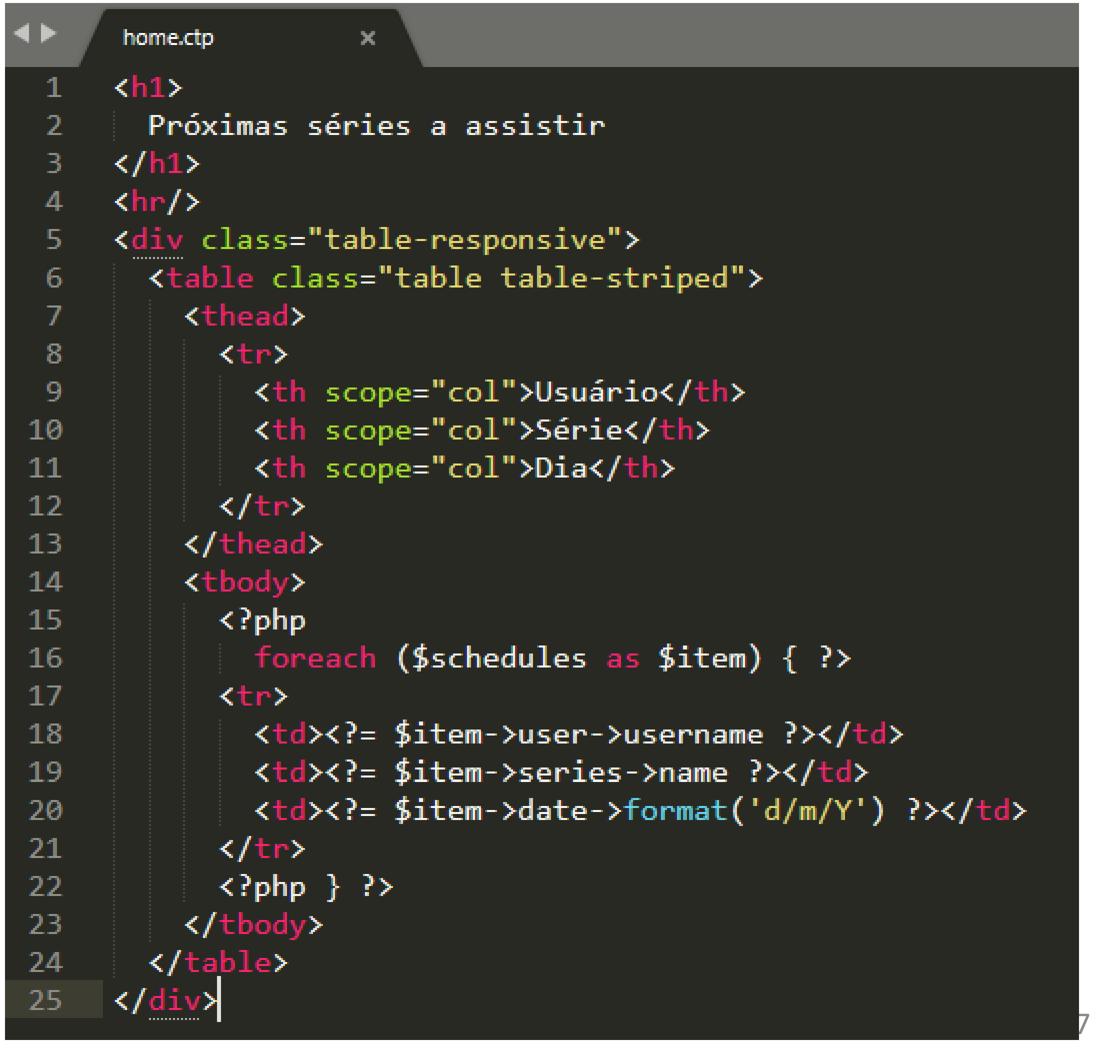
cont.



```php
     * @return \Cake\Http\Response|null Redirects to index.
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
     */
    public function delete($id = null)
    {
        $this->request->allowMethod(['post', 'delete']);
        $schedule = $this->Schedules->get($id);
        if ($this->Schedules->delete($schedule)) {
            $this->Flash->success(__('The schedule has been deleted.'));
        } else {
            $this->Flash->error(__('The schedule could not be deleted. Please, try again.'));
        }

        return $this->redirect(['action' => 'index']);
    }
}
```

7. Códigos do projeto (Cont.)

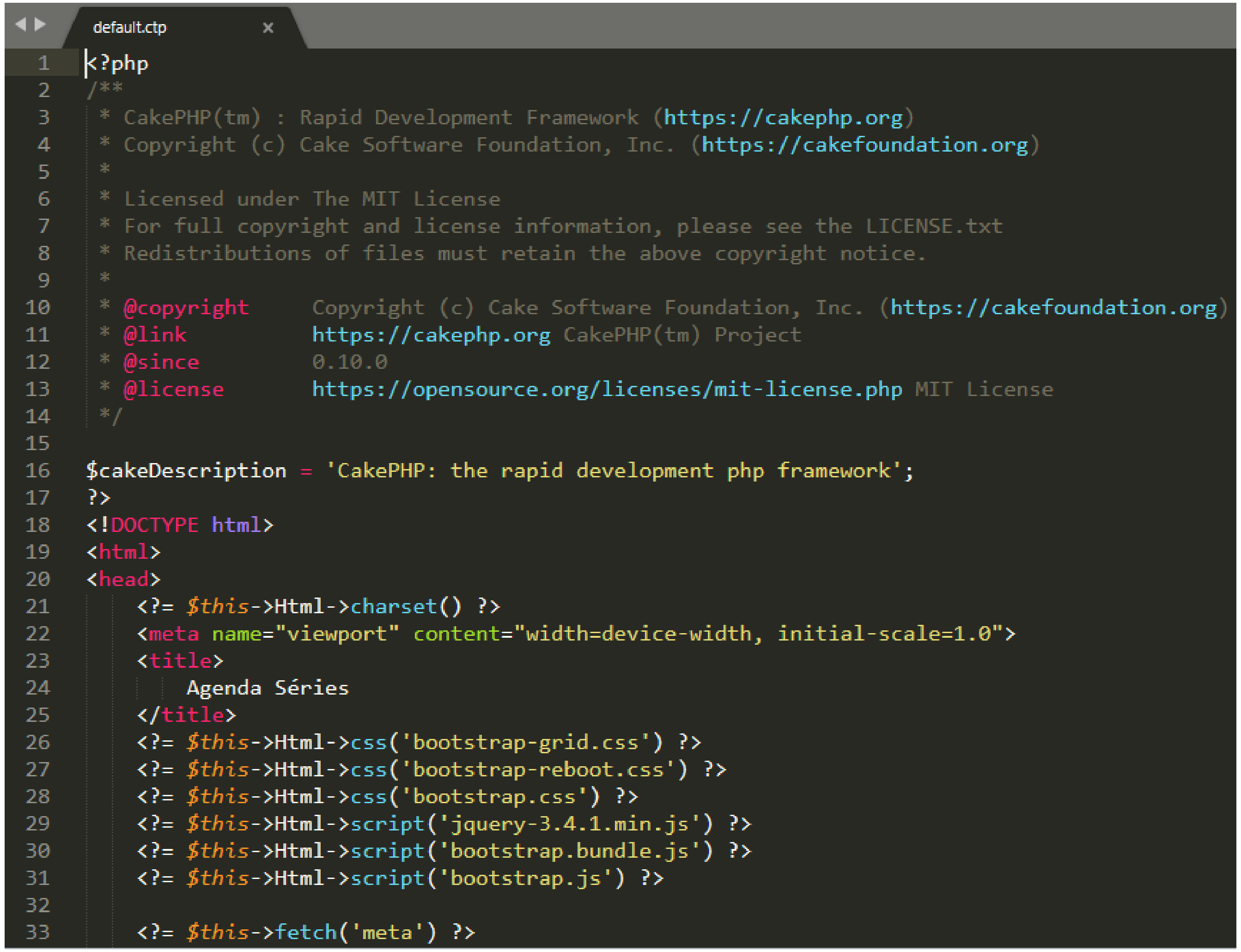
- **Pages:** Home (home.ctp)



```
<h1>
  Próximas séries a assistir
</h1>
<hr/>
<div class="table-responsive">
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">Usuário</th>
        <th scope="col">Série</th>
        <th scope="col">Dia</th>
      </tr>
    </thead>
    <tbody>
      <?php
        foreach ($schedules as $item) { ?>
      <tr>
        <td><?= $item->user->username ?></td>
        <td><?= $item->series->name ?></td>
        <td><?= $item->date->format('d/m/Y') ?></td>
      </tr>
      <?php } ?>
    </tbody>
  </table>
</div>
```

7. Códigos do projeto (Cont.)

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

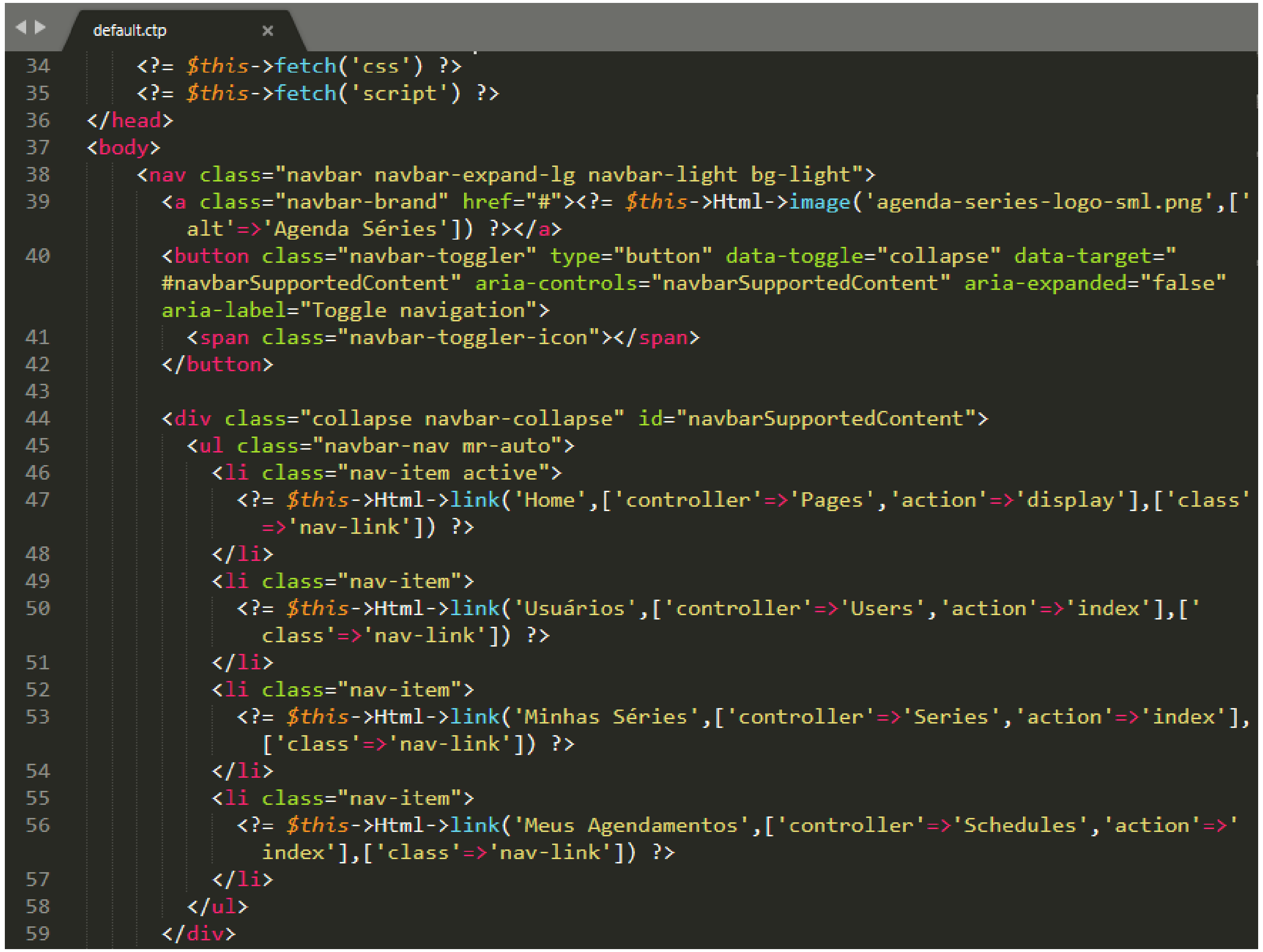


```
<?php
/**
 * CakePHP(tm) : Rapid Development Framework (https://cakephp.org)
 * Copyright (c) Cake Software Foundation, Inc. (https://cakefoundation.org)
 *
 * Licensed under The MIT License
 * For full copyright and license information, please see the LICENSE.txt
 * Redistributions of files must retain the above copyright notice.
 *
 * @copyright     Copyright (c) Cake Software Foundation, Inc. (https://cakefoundation.org)
 * @link          https://cakephp.org CakePHP(tm) Project
 * @since         0.10.0
 * @license       https://opensource.org/licenses/mit-license.php MIT License
 */

$cakeDescription = 'CakePHP: the rapid development php framework';
?>
<!DOCTYPE html>
<html>
<head>
    <?= $this->Html->charset() ?>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>
        Agenda Séries
    </title>
    <?= $this->Html->css('bootstrap-grid.css') ?>
    <?= $this->Html->css('bootstrap-reboot.css') ?>
    <?= $this->Html->css('bootstrap.css') ?>
    <?= $this->Html->script('jquery-3.4.1.min.js') ?>
    <?= $this->Html->script('bootstrap.bundle.js') ?>
    <?= $this->Html->script('bootstrap.js') ?>

    <?= $this->fetch('meta') ?>
```

7. Códigos do projeto (Cont.)

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

  cont.

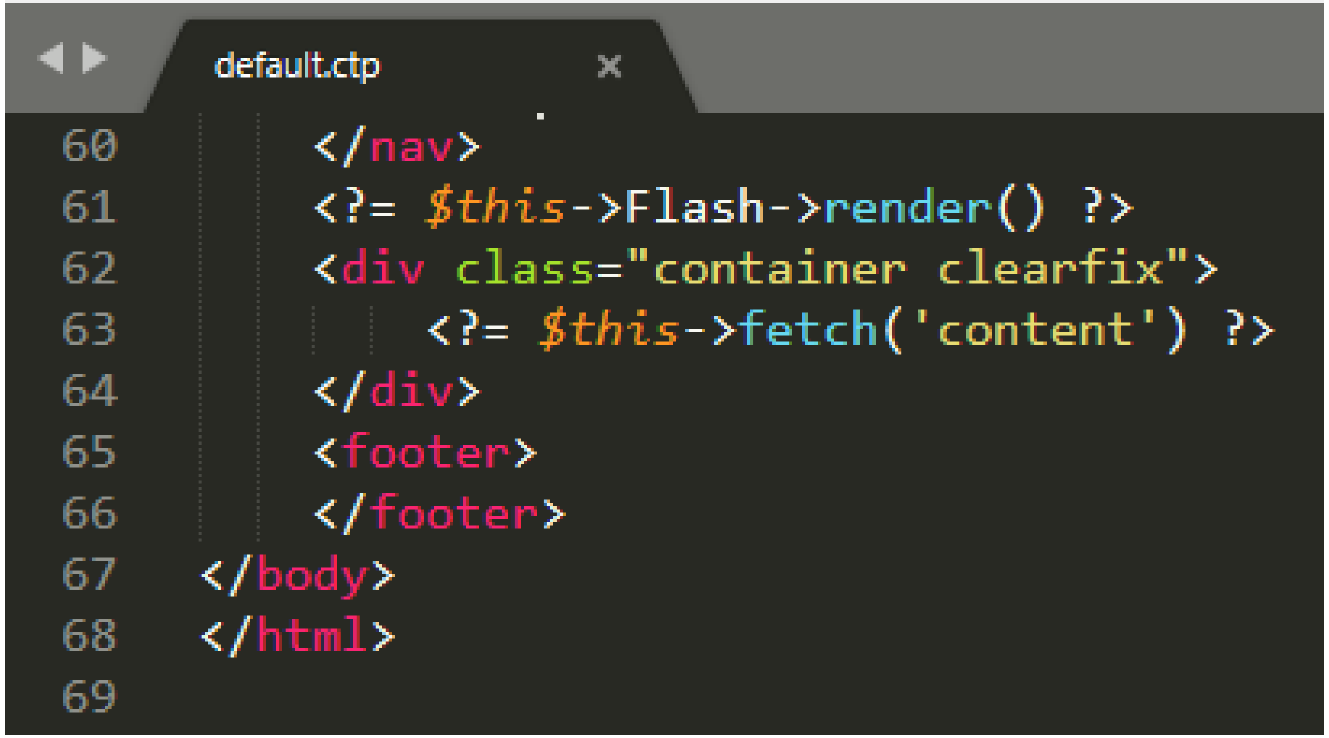


```
  <?= $this->fetch('css') ?>
  <?= $this->fetch('script') ?>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#"><?= $this->Html->image('agenda-series-logo-sml.png',['alt'=>'Agenda Séries']) ?></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <?= $this->Html->link('Home',['controller'=>'Pages','action'=>'display'],['class'=>'nav-link']) ?>
        </li>
        <li class="nav-item">
          <?= $this->Html->link('Usuários',['controller'=>'Users','action'=>'index'],['class'=>'nav-link']) ?>
        </li>
        <li class="nav-item">
          <?= $this->Html->link('Minhas Séries',['controller'=>'Series','action'=>'index'],['class'=>'nav-link']) ?>
        </li>
        <li class="nav-item">
          <?= $this->Html->link('Meus Agendamentos',['controller'=>'Schedules','action'=>'index'],['class'=>'nav-link']) ?>
        </li>
      </ul>
    </div>
```

7. Códigos do projeto (Cont.)

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

cont.



```
        </nav>
        <?= $this->Flash->render() ?>
        <div class="container clearfix">
                <?= $this->fetch('content') ?>
        </div>
        <footer>
        </footer>
</body>
</html>

```

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.