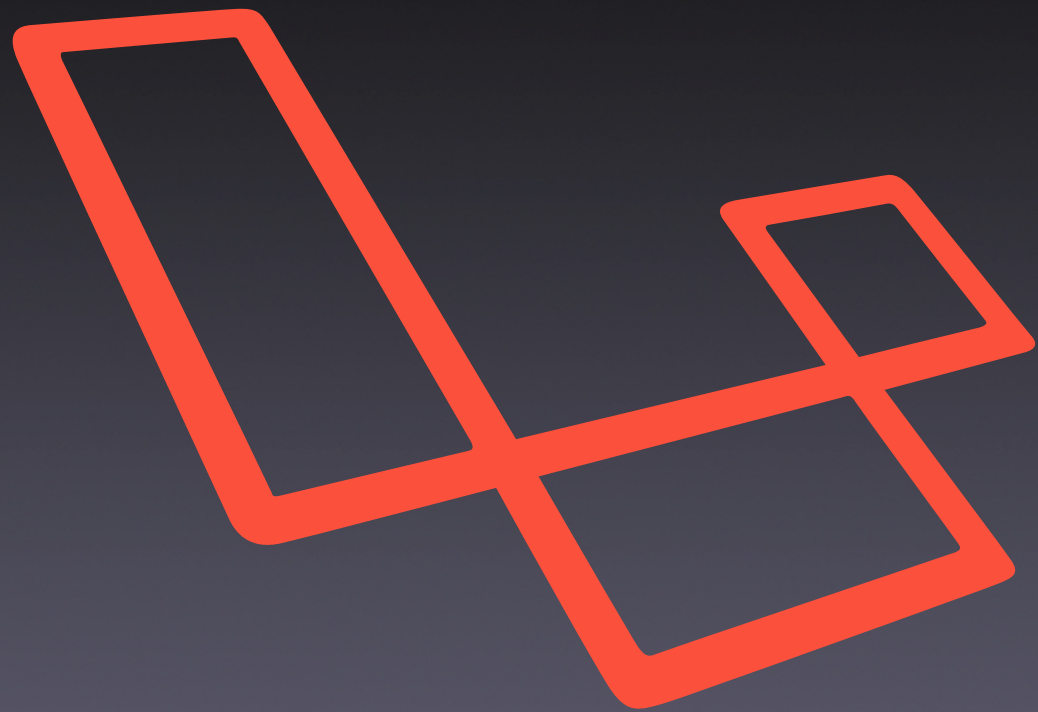
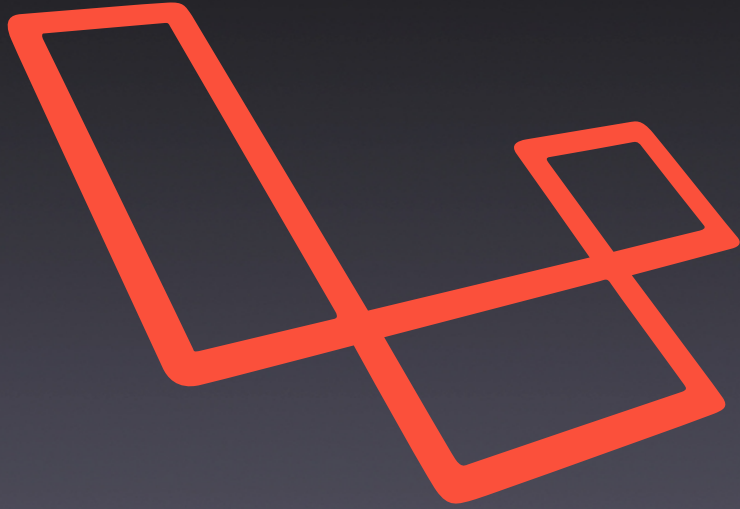


Laravel, WordPress e a API REST:  
usando WordPress como xestor de  
contidos en aplicacións Laravel









# Laravel

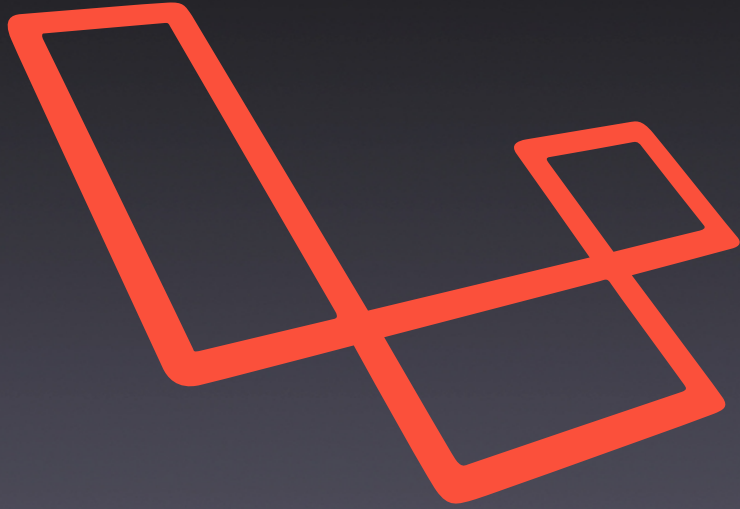
- Framework PHP máis popular
- Curva de aprendizaxe pequena
- MVC
- Template engine (Blade)
- ORM (Eloquent)
- Database migrations

# Laravel (II)

- CLI: Artisan
- Desacoplado
- Tests unitarios e de integración sinxelos
- Seguridade
- Comunidade: soporte e compoñentes
- Libre (FLOSS)

# WordPress

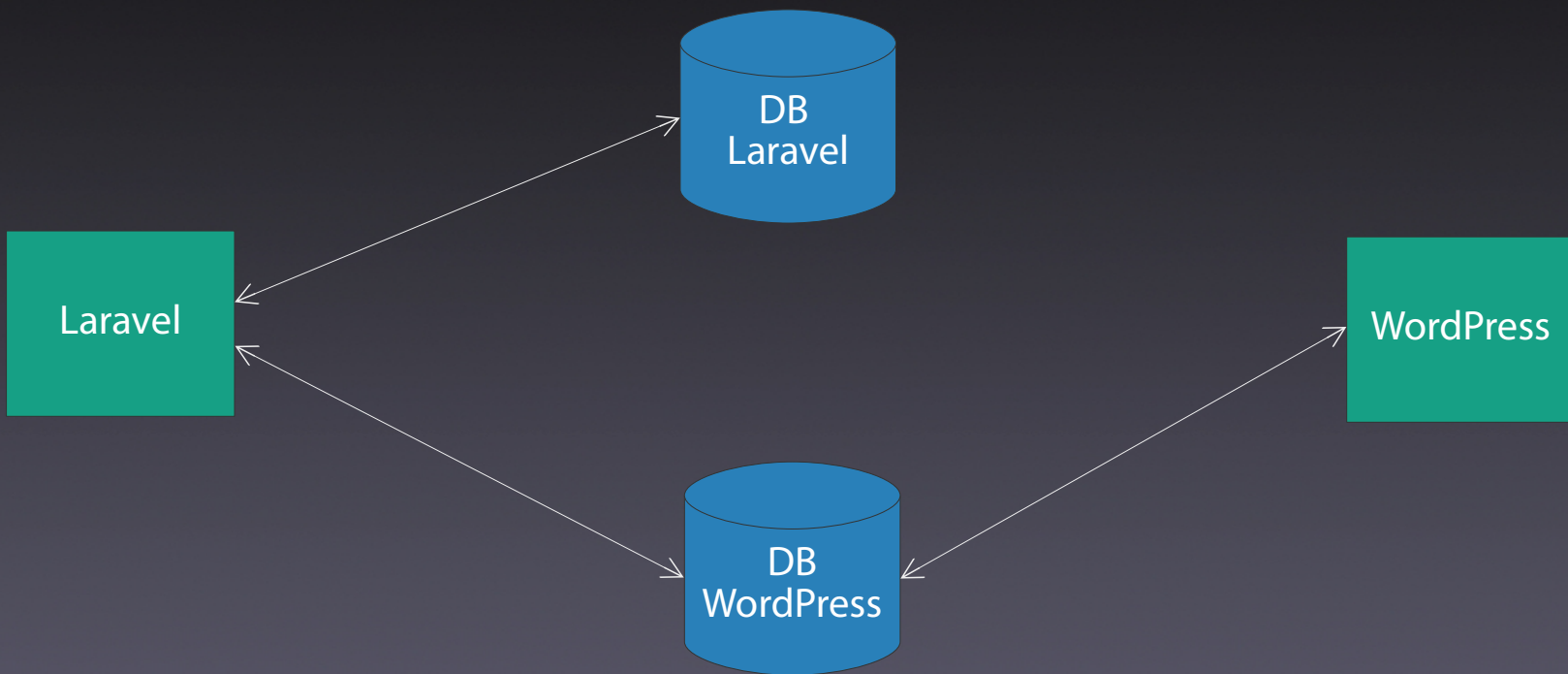
- CMS mais popular:
  - 60,7 % CMS
  - 33,9 % web
- Simple e flexible
- Comunidade
- Libre (FLOSS)





1.

Base de datos



2.

Paquetes  
(Corcel)

# Corcel

- Posts
- Páginas
- ACF
- CPT
- Shortcodes
- Taxonomías



# Corcel (II)

- Formatos de post
- Adxuntos e revisión
- Thumbnails
- Opcións
- Menú
- Usuarios e autenticación

# Corcel (III)

```
$posts = Post::published()->get();
```

```
$posts = Post::status('publish')->get();
```

```
$post = Post::find(31);
```

```
echo $post->post_title;
```

# 3.

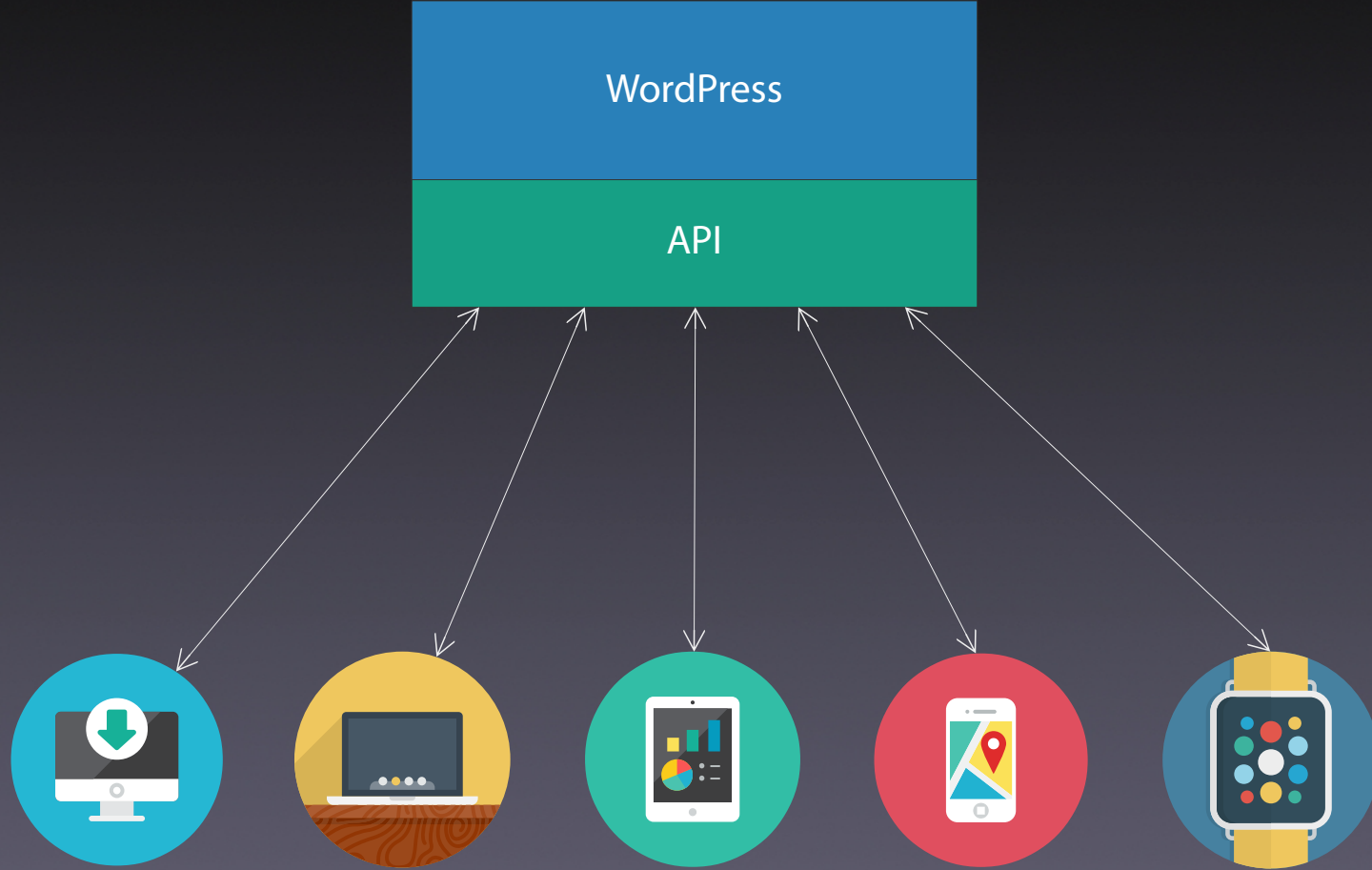
# WP REST API

WordPress 4.7.0 – 4 de xaneiro de 2017

# API

Conxunto de métodos que  
ofrece un software (WordPress)  
para que outro software  
(Laravel) o use





# REST

REST: REpresentational State Transfer

- Protocolo cliente/servidor sen estado
- HTTP(S)
- Baseado en URI: identificación unívoca
- JSON: interoperatividade

# REST (II)

- 4 operaci3ns principais:
  - POST (create)
  - GET (read)
  - PUT (update)
  - DELETE (delete)
- Equivalentes a CRUD

# Usos

- Aplicacións para móbil/tablet
- Aplicacións web
- Aplicacións de escritorio
- Importacións complexas
- Redes sociais
- IoT



# Endpoints

`https://www.example.com/wp-json/wp/v2/  
[endpoint]?[parameters]`

`https://www.example.com/wp-json/wp/v2/posts`

- `/wp/v2/posts`
- `/wp/v2/revisions`
- `/wp/v2/categories`

# Endpoints (II)

- /wp/v2/tags
- /wp/v2/pages
- /wp/v2/comments
- /wp/v2/taxonomies
- /wp/v2/media
- /wp/v2/users
- /wp/v2/types
- /wp/v2/statuses
- /wp/v2/settings

# Exemplo

# Instalar WordPress

```
$ wp core download \  
--locale=gl_ES
```

```
$ wp config create \  
--dbname=wordpress \  
--dbuser=homestead \  
--dbpass=secret
```

```
$ wp db clean --yes
```



# Instalar WordPress (II)

```
$ wp core install \  
--url=wordpress.test \  
--title="Sitio WordPress de test" \  
--admin_user=manager \  
--admin_password=password \  
--admin_email=hello@wordpress.test
```

```
$ wp theme activate twentysixteen
```

# Instalar WordPress (III)

```
$ wp plugin install \
application-passwords --activate
```

```
$ curl http://loripsum.net/api/5 \
| wp post generate --post_content --count=10
```

## Sitio WordPress de test

Outro sitio WordPress máis

# Artigo 10



14 Maio, 2019

Deixa un comentario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eorum enim est haec querela, qui sibi cari sunt seseque diligunt. Negat enim summo bono afferre incrementum diem. Duo Reges: constructio interrete. Terram, mihi crede, ea lanx et maria deprimet. Conferam tecum, quam cuique verso rem subicias; Quid enim est a Chrysippo praetermissum in Stoicis? Est enim tanti philosophi tamque nobilis audacter sua decreta defendere.

Nihil acciderat ei, quod nollet, nisi quod anulum, quo delectabatur, in mari abiecerat. Hoc non est positum in nostra actione. Mihi quidem Homerus huius modi quiddam vidisse videatur in iis, quae de Sirenum cantibus finxerit. Qua igitur re ab deo vincitur, si aeternitate non vincitur? Te ipsum, dignissimum maioribus tuis, voluptasne induxit, ut adolescentulus eriperes P.

---



---

### ARTIGOS RECENTES

- [Artigo 10](#)
- [Artigo 9](#)
- [Artigo 8](#)
- [Artigo 7](#)
- [Artigo 6](#)

---

### COMENTARIOS RECENTES

- [Un comentarista de WordPress en Ola mundo!](#)

- Panel
- Artigos
- Multimedia
- Páxinas
- Comentarios
- Aparencia
- Plugins
- Usuarios**
- Todos os usuarios
- Engadir novo
- O meu perfil
- Ferramentas
- Configuración
- Contraer o menú

## Sobre ti

### Información biográfica

Compartir algo de información biográfica para completar o teu perfil. Isto podería aparecer en público.

### Imaxe de perfil



Podes cambiar a túa foto de perfil en [Gravatar](#).

## Xestión da conta

### Contrasinal novo

Xerar contrasinal

### Sesións

Desconectar do resto de sitios

Perdiches o teu teléfono ou deixaches a túa conta aberta nun ordenador público? Podes desconectarte de calquera lugar e seguir conectado aquí.

## Application Passwords


Application passwords allow authentication via non-interactive systems, such as XMLRPC or the REST API, without providing your actual password. Application passwords can be easily revoked. They cannot be used for traditional logins to your website.


stock


Add New


Actualizar o perfil





 Panel


 Artigos


 Multimedia

 Páxinas

 Comentarios

 Aparencia


 Plugins


 Usuarios


Todos os usuarios

Engadir novo

O meu perfil

 Ferramentas

 Configuración

 Contraer o menú

## Sobre ti

Información biográfica

Compartir algo de información biográfica para completar o teu perfil. Isto podería aparecer en público.

Imaxe de perfil

Podes cambiar a túa foto de perfil aquí.

## Xestión da conta

Contrasinal novo

Xerar contrasinal

Sesións

Desconectar do resto de sitios

Perdiches o teu teléfono ou deixaches a túa conta aberta nun ordenador público? Podes desconectarte de calquera lugar e seguir conectado aquí.

## Application Passwords

Application passwords allow authentication via non-interactive systems, such as XMLRPC or the REST API, without providing your actual password. Application passwords can be easily revoked. They cannot be used for traditional logins to your website.

New Application Password Name

Add New

Nome	Created	Last Used	Last IP	Revoke
stock	15 May, 2019	—	—	<div>Revoke</div>
Nome	Created	Last Used	Last IP	Revoke

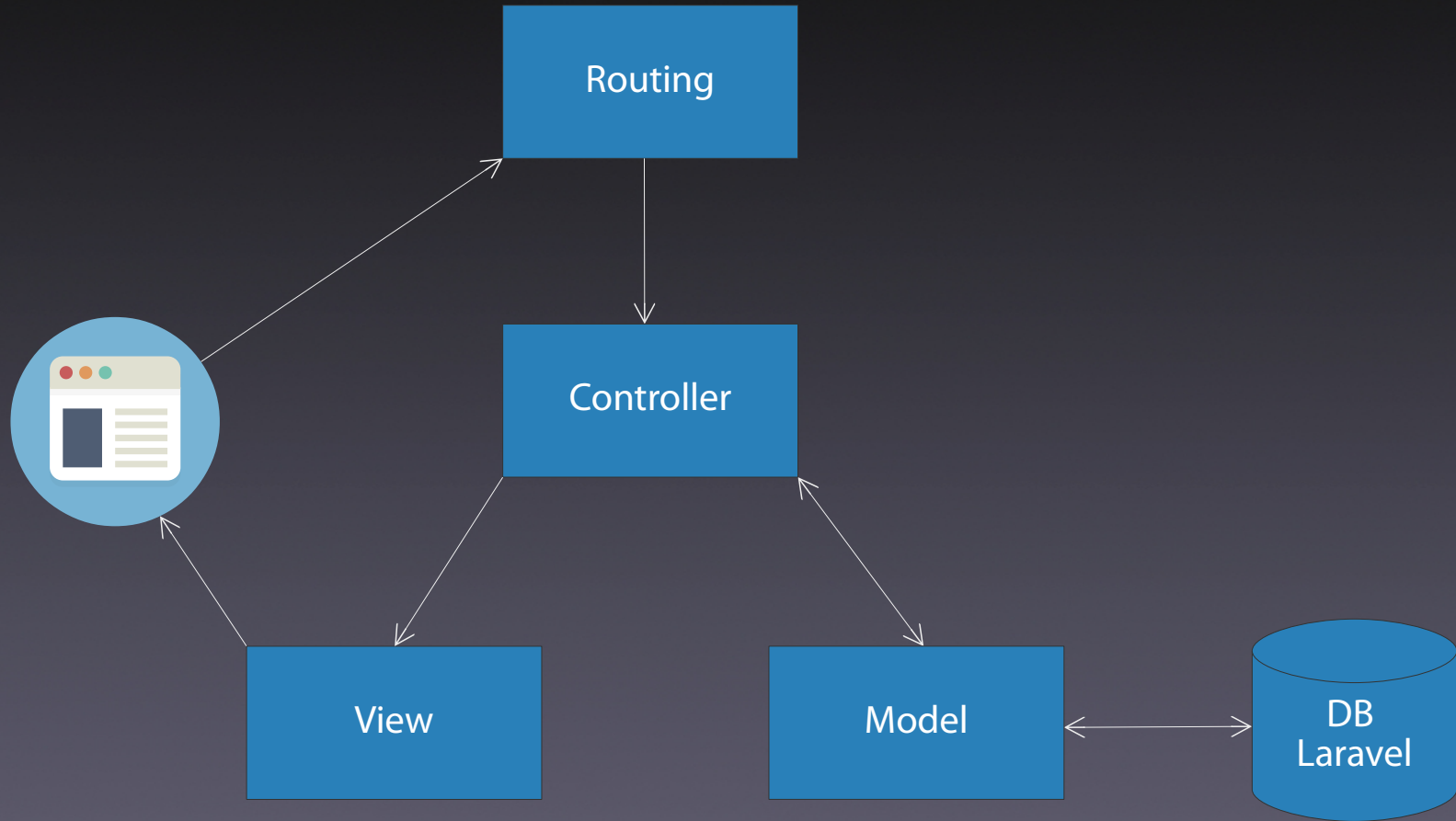
# Instalar Laravel

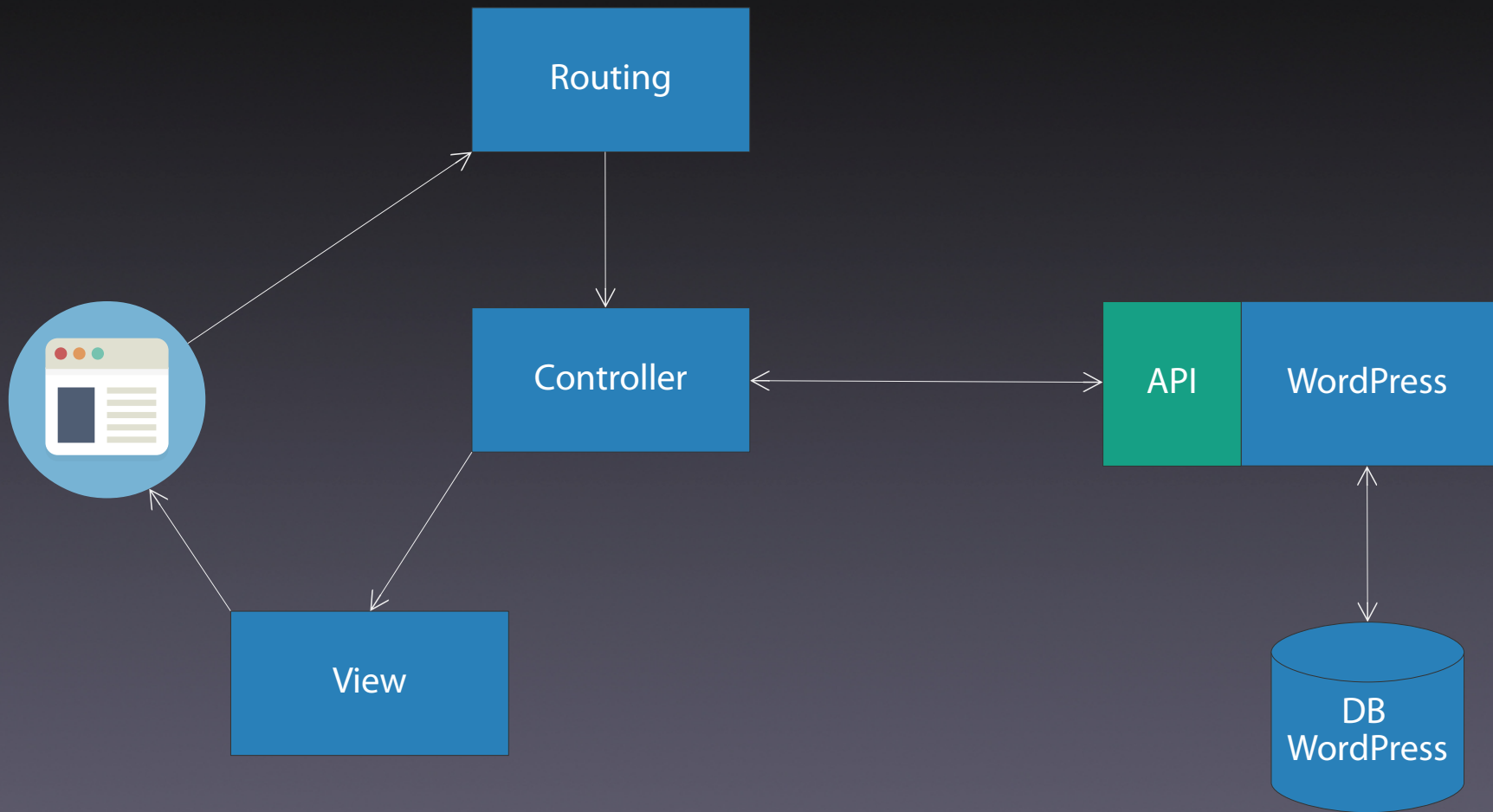
```
$ composer create-project \  
--prefer-dist laravel/laravel stock
```

```
$ composer require guzzlehttp/guzzle
```

```
$ composer require laravelcollective/html
```

```
$ php artisan make:auth
```







# Arquivo de rotas

/routes/web.php

```
Route::resource('posts', 'PostController');
```

- GET /posts index
- GET /posts/{id} show
- GET /posts/create create
- POST /posts store
- GET /posts/{id}/edit edit
- PUT /posts/{id} update
- DELETE /posts/{id} destroy

# Controlador

```
$ php artisan make:controller PostController  
--resource
```

# GET /posts

```
protected $url = 'http://wordpress.test/wp-json/wp/v2/posts';
protected $username = 'manager';
protected $password = 'I6Wn uACu cifk C3it 0M3C ziKV';

public function index()
{
    $client = new Client();
    $response = $client->get($this->url);
    $posts = json_decode($response->getBody());
    return view('posts.index', compact('posts'));
}
```

```
@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">Posts
                    <a class="btn btn-primary float-right" href="{{action('PostController@create')}}">Add post</a>
                    </div>
                    <div class="card-body">
                        @foreach($posts as $post)
                            <table width="100%">
                                <tr>
                                    <td width="70%">
                                        <a href="{{action('PostController@show', $post->id)}}">
                                            {{ $post->title->rendered }}
                                        </a>
                                    </td>
                                    <td width="15%">
                                        <a class="btn btn-info" href="{{action('PostController@edit', $post->id)}}">
                                            Edit
                                        </a>
                                    </td>
                                    <td width="15%">
                                        <form action="{{ action('PostController@destroy', $post->id) }}" method="POST">
                                            {{ method_field('DELETE') }}
                                            {{ csrf_field() }}
                                            <button class="btn btn-danger">Delete</button>
                                        </form>
                                    </td>
                                </tr>
                            </table>
                        @endforeach
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection
```



## Posts

[Add post](#)[Artigo 10](#)[Edit](#)[Delete](#)[Artigo 9](#)[Edit](#)[Delete](#)[Artigo 8](#)[Edit](#)[Delete](#)[Artigo 7](#)[Edit](#)[Delete](#)[Artigo 6](#)[Edit](#)[Delete](#)[Artigo 5](#)[Edit](#)[Delete](#)[Artigo 4](#)[Edit](#)[Delete](#)[Artigo 3](#)[Edit](#)[Delete](#)[Artigo 2](#)[Edit](#)[Delete](#)[Artigo 1](#)[Edit](#)[Delete](#)

# GET /posts/{id}

```
public function show($id)
{
    $client = new Client();
    $response = $client->get( uri: $this->url . '/' . $id);
    $post = json_decode($response->getBody());
    return view( view: 'posts.show', compact( varname: 'post' ));
}
```

```
@extends('layouts.app')

@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ $post->title->rendered }}</div>
                    <div class="card-body">
                        {!! $post->content->rendered !!}
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

## Artigo 10

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Respondent extrema primis, media utrisque, omnia omnibus. Si autem id non concedatur, non continuo vita beata tollitur. Duo Reges: constructio interrete. Quod non faceret, si in voluptate summum bonum poneret.

Quid ergo aliud intellegetur nisi uti ne quae pars naturae neglegatur? Servari enim iustitia nisi a forti viro, nisi a sapiente non potest. Itaque in rebus minime obscuris non multus est apud eos disserendi labor. Cum ageremus, inquit, vitae beatum et eundem supremum diem, scribebamus haec. Qui autem esse poteris, nisi te amor ipse ceperit? Tum ille timide vel potius verecunde: Facio, inquit.

Ut proverbia non nulla veriora sint quam vestra dogmata. Sed in rebus apertissimis nimium longi sumus. Praeterea sublata cognitione et scientia tollitur omnis ratio et vitae degendae et rerum gerendarum. Apparet statim, quae sint officia, quae actiones. De vacuitate doloris eadem sententia erit. Illis videtur, qui illud non dubitant bonum dicere -; Nec lapathi suavitatem acupenseri Galloni Laelius anteponebat, sed suavitatem ipsam neglegebat; Summum enim bonum exposuit vacuitatem doloris; Sin eam, quam Hieronymus, ne fecisset idem, ut voluptatem illam Aristippi in prima commendatione poneret.

Nunc omni virtuti vitium contrario nomine opponitur. Nihil opus est exemplis hoc facere longius. Ego vero isti, inquam, permitto. Nec vero pietas adversus deos nec quanta iis gratia debeatur sine explicatione naturae intellegi potest. Sextilio Rufo, cum is rem ad amicos ita deferret, se esse heredem Q. Traditur, inquit, ab Epicuro ratio neglegendi doloris. Negat enim summo bono afferre incrementum diem. Non quaero, quid dicat, sed quid convenienter possit rationi et sententiae suae dicere. Quia dolori non voluptas contraria est, sed doloris privatio.

Nosti, credo, illud: Nemo pius est, qui pietatem-; Virtutis, magnitudinis animi, patientiae, fortitudinis fomentis dolor mitigari solet. Quo tandem modo? Iam id ipsum absurdum, maximum malum neglegi. Vitae autem degendae ratio maxime quidem illis placuit quieta.



# GET /posts/create

```
public function create()  
{  
    return view('posts.create');  
}
```

```

@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">Crear un nuevo post</div>
                    <div class="card-body">
                        {{ Form::open([
                            'action' => ['PostController@store'],
                            'class' => 'card',
                        ]) }}

                        <div class="col-md-9">
                            <div class="form-group">
                                {{ Form::label('titulo', __('Titulo') . ' (*)', ['class' => 'form-label']) }}
                                {{ Form::text('titulo', null, ['class' => 'form-control', 'roles' => 'form']) }}
                                @error('name')
                                    <div class="invalid-feedback">{{ $errors->first('titulo') }}</div>
                                @enderror
                            </div>
                        </div>

                        <div class="col-md-12">
                            <div class="form-group">
                                {{ Form::label('contido', __('Contido') . ' (*)', ['class' => 'form-label']) }}
                                {{ Form::textarea('contido', null, ['class' => 'form-control', 'roles' => 'form']) }}

                                @error('name')
                                    <div class="invalid-feedback">{{ $errors->first('contido') }}</div>
                                @enderror
                            </div>
                        </div>

                        <div class="card-footer text-right">
                            <button type="submit" class="btn btn-primary">{{ __('Update post') }}</button>
                        </div>
                        {{ Form::close() }}
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection

```

Crear un novo post

Título (\*)

Contido (\*)

Update post

# POST /posts

```
public function store(Request $request)
{
    $params = array(
        'title' => $request['titulo'],
        'content' => $request['contido'],
        'status' => 'publish'
    );
    $client = new Client([
        'auth' => [
            $this->username,
            $this->password
        ]
    ]);
    $response = $client->post($this->url, [
        RequestOptions::JSON => $params
    ]);
    return redirect(action('PostController@index'));
}
```

## Posts

[Add post](#)

Artigo 11

[Edit](#)[Delete](#)

Artigo 10

[Edit](#)[Delete](#)

Artigo 9

[Edit](#)[Delete](#)

Artigo 8

[Edit](#)[Delete](#)

Artigo 7

[Edit](#)[Delete](#)

Artigo 6

[Edit](#)[Delete](#)

Artigo 5

[Edit](#)[Delete](#)

Artigo 4

[Edit](#)[Delete](#)

Artigo 3

[Edit](#)[Delete](#)

Artigo 2

[Edit](#)[Delete](#)



# GET /posts/{id}/edit

```
public function edit($id)
{
    $client = new Client();
    $response = $client->get(uri: $this->url . '/' . $id);
    $post = json_decode($response->getBody());
    return view(view: 'posts.edit', compact(varname: 'post'));
}
```

```

@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ $post->title->rendered }}</div>
                    <div class="card-body">
                        {{ Form::open([
                            'method' => 'PUT',
                            'action' => ['PostController@update', $post->id],
                            'class' => 'card',
                        ]) }}

                        <div class="col-md-9">
                            <div class="form-group">
                                {{ Form::label('titulo', __('Titulo') . ' (*)', ['class' => 'form-label']) }}
                                {{ Form::text('titulo', $post->title->rendered, ['class' => 'form-control', 'roles' => 'form', 'placeholder' => 'Titulo']) }}
                                @error('name')
                                    <div class="invalid-feedback">{{ $errors->first('titulo') }}</div>
                                @enderror
                            </div>
                        </div>

                        <div class="col-md-12">
                            <div class="form-group">
                                {{ Form::label('contenido', __('Contenido') . ' (*)', ['class' => 'form-label']) }}
                                {{ Form::textarea('contenido', $post->content->rendered, ['class' => 'form-control', 'roles' => 'form', 'placeholder' => 'Contenido']) }}

                                @error('name')
                                    <div class="invalid-feedback">{{ $errors->first('contenido') }}</div>
                                @enderror
                            </div>
                        </div>

                        <div class="card-footer text-right">
                            <button type="submit" class="btn btn-primary">{{ __('Update post') }}</button>
                        </div>
                        {{ Form::close() }}
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection

```

## Artigo 11

Título (\*)

Contido (\*)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed scelerisque, nisi non accumsan varius, sem mauris vulputate velit, sit amet euismod ex ante quis sem. Ut condimentum turpis dui, eu malesuada turpis euismod sed. Donec hendrerit in mi vel sollicitudin. Pellentesque quis ex tristique, consequat odio sed, vulputate dolor. Integer rutrum sapien purus, et tempus nulla auctor in. Praesent vitae nulla semper, vestibulum dui vitae, tristique velit. Sed hendrerit ultrices ipsum. Nunc consequat odio ex, in eleifend mi tincidunt sit amet. Suspendisse augue tellus, ultrices quis sem eu, varius faucibus orci.

[Update post](#)

# PUT /posts/{id}

```
public function update(Request $request, $id)
{
    $params = array(
        'title' => $request['titulo'],
        'content' => $request['contido'],
        'status' => 'publish'
    );
    $client = new Client([
        'auth' => [
            $this->username,
            $this->password
        ]
    ]);
    $response = $client->put(uri: $this->url . '/' . $id, [
        RequestOptions::JSON => $params
    ]);
    return redirect(action(name: 'PostController@index'));
}
```

## Posts

[Add post](#)

Artigo 11 modificado

[Edit](#)[Delete](#)

Artigo 10

[Edit](#)[Delete](#)

Artigo 9

[Edit](#)[Delete](#)

Artigo 8

[Edit](#)[Delete](#)

Artigo 7

[Edit](#)[Delete](#)

Artigo 6

[Edit](#)[Delete](#)

Artigo 5

[Edit](#)[Delete](#)

Artigo 4

[Edit](#)[Delete](#)

Artigo 3

[Edit](#)[Delete](#)

Artigo 2

[Edit](#)[Delete](#)



# DELETE /posts/{id}

```
public function destroy($id)
{
    $client = new Client([
        'auth' => [
            $this->username,
            $this->password
        ]
    ]);
    $response = $client->delete( uri: $this->url . '/' . $id);
    return back();
}
```

## Posts

[Add post](#)

Artigo 10

[Edit](#)[Delete](#)

Artigo 9

[Edit](#)[Delete](#)

Artigo 8

[Edit](#)[Delete](#)

Artigo 7

[Edit](#)[Delete](#)

Artigo 6

[Edit](#)[Delete](#)

Artigo 5

[Edit](#)[Delete](#)

Artigo 4

[Edit](#)[Delete](#)

Artigo 3

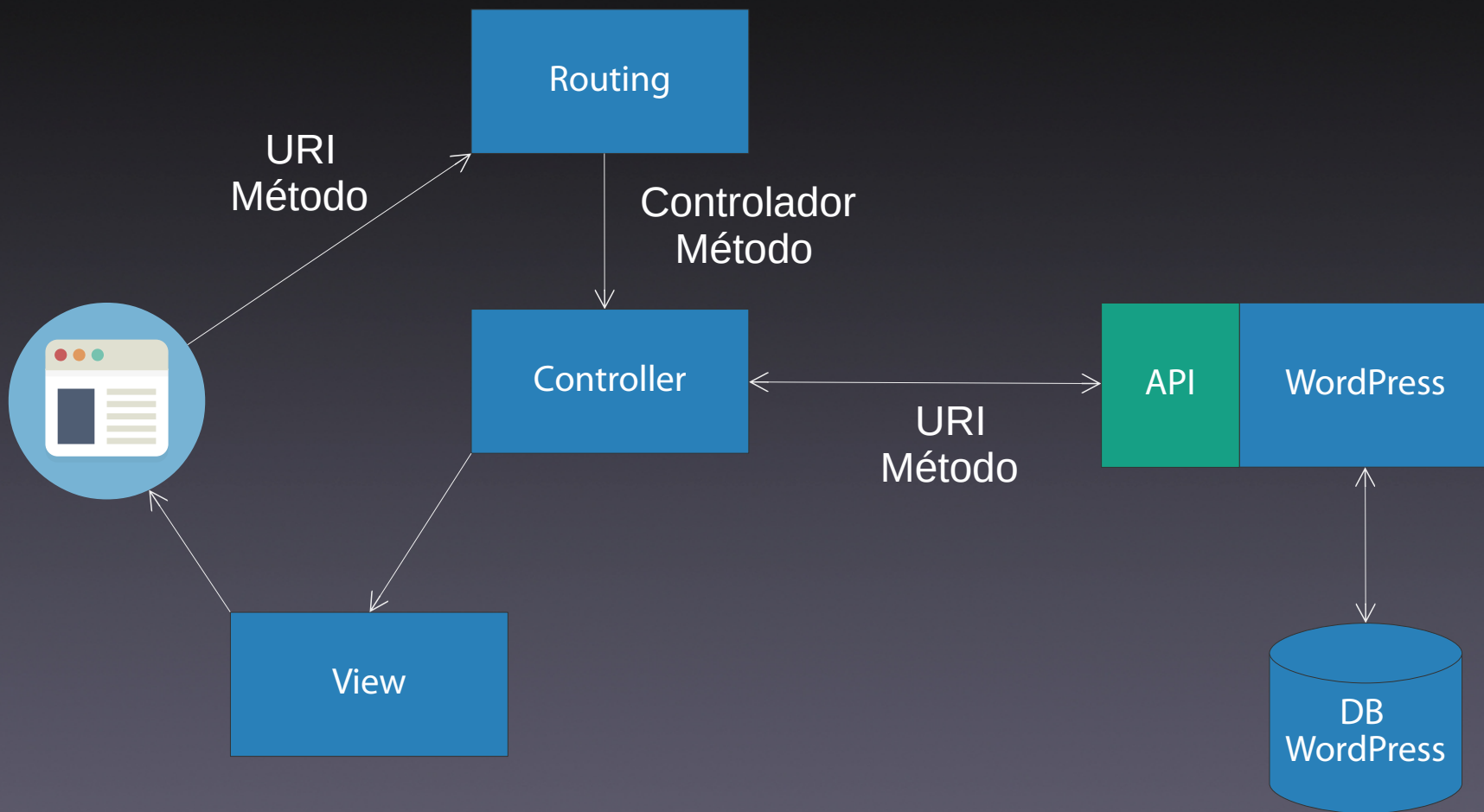
[Edit](#)[Delete](#)

Artigo 2

[Edit](#)[Delete](#)

Artigo 1

[Edit](#)[Delete](#)





jesus@jesusamieiro.com

