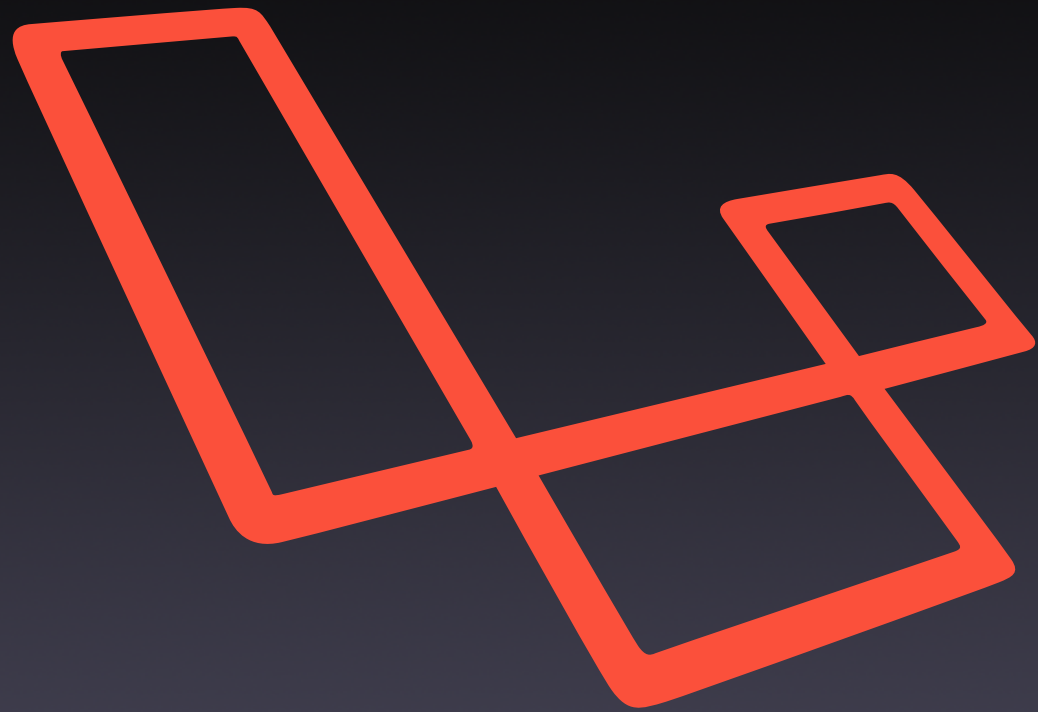
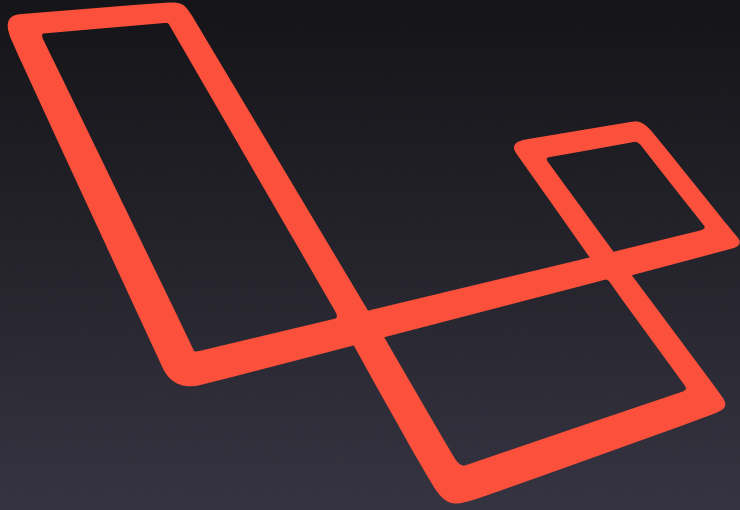


WordPress y la API REST:  
usando WordPress como gestor de  
contenidos en aplicaciones externas







# Laravel

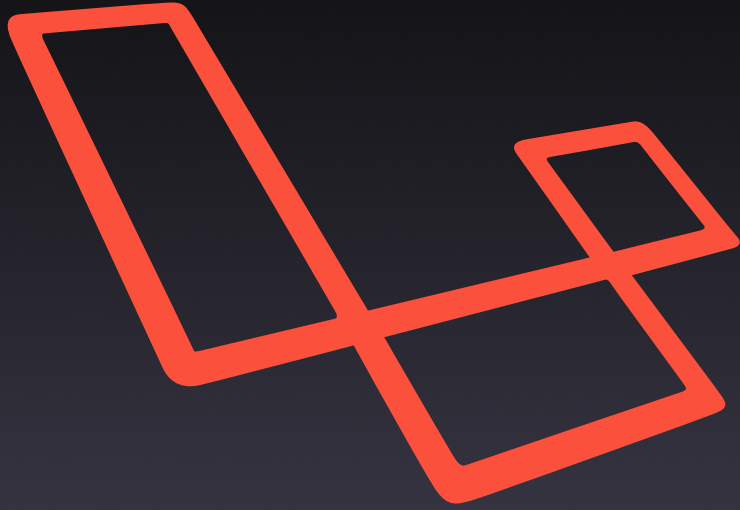
- Framework PHP más popular
- Curva de aprendizaje pequeña
- MVC
- Template engine (Blade)
- ORM (Eloquent)
- Database migrations

# Laravel (II)

- CLI: Artisan
- Desacoplado
- Tests unitarios y de integración sencillos
- Seguridad
- Comunidad: soporte y componentes
- Libre (FLOSS)

# WordPress

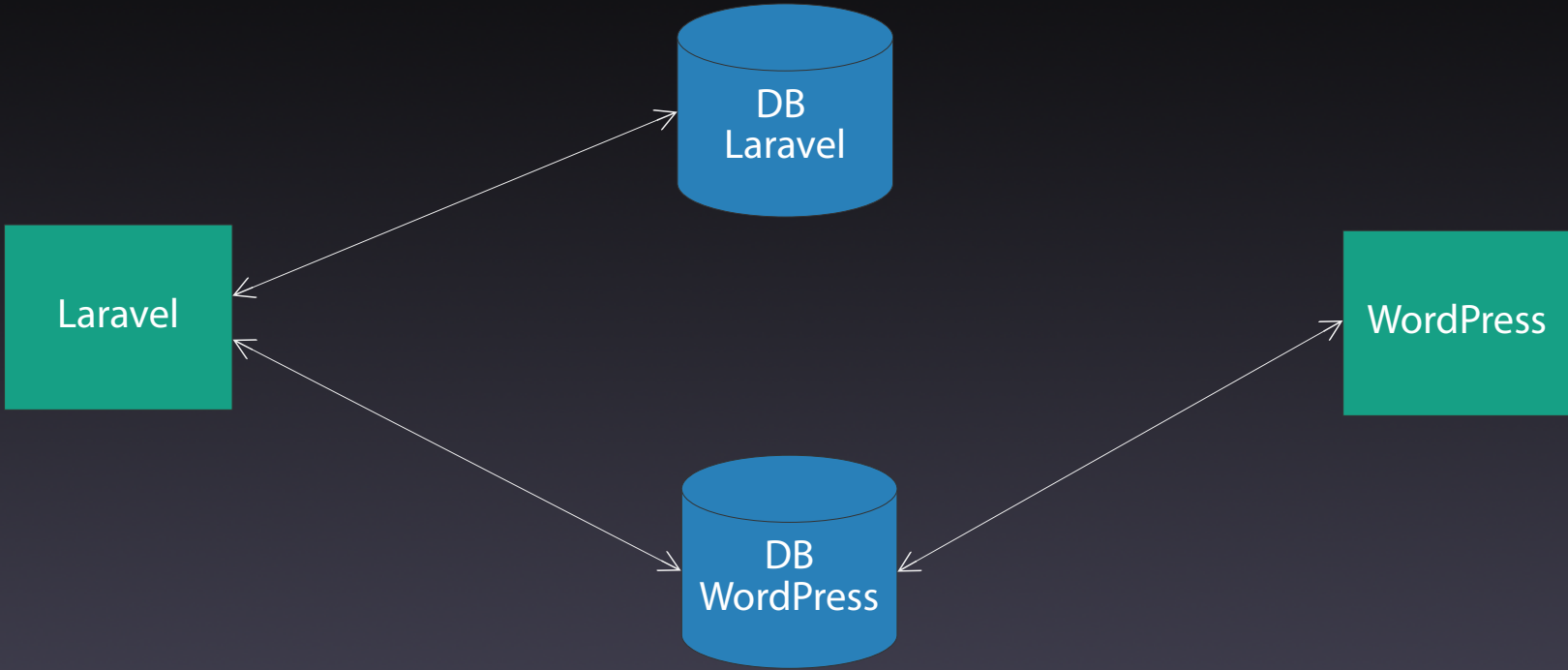
- CMS más popular:
  - 63,6 % CMS
  - 43,1 % web
- Simple y flexible
- Comunidad
- Libre (FLOSS)





1.

Base de datos



2.

Paquetes  
(Corcel)

# Corcel

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

# Corcel (II)

- Formatos de post
- Adjuntos y revisión
- Thumbnails
- Opciones
- Menú
- Usuarios y autenticación

# Corcel (III)

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

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

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

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

3.

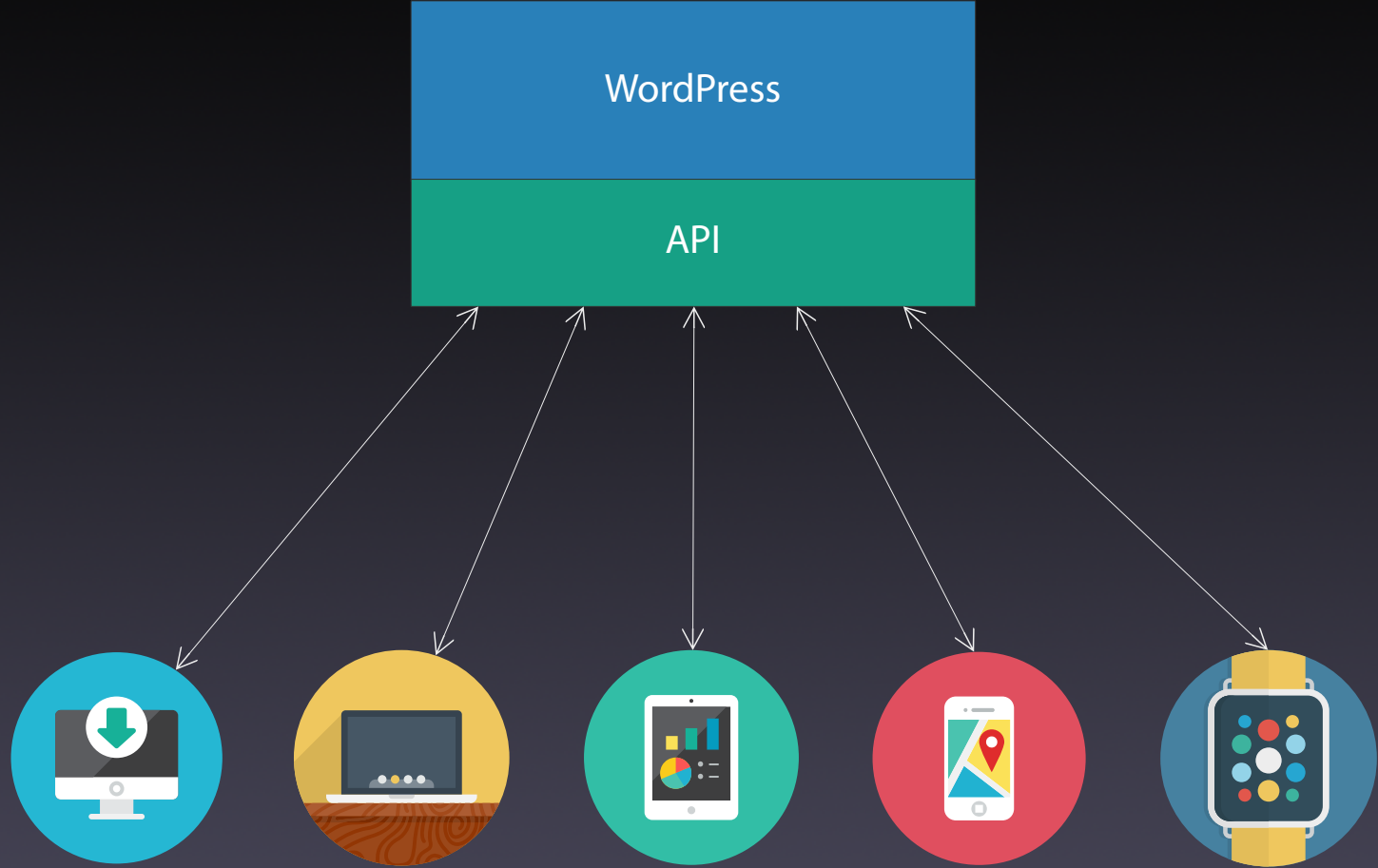
# WP REST API

WordPress 4.7.0 – 6 de diciembre de 2016

# API

Conjunto de métodos que ofrece un software (WordPress) para que otro software (Laravel) lo use





# REST

REST: REpresentational State Transfer

- Protocolo cliente/servidor sin estado
- HTTP(S)
- Basado en URI: identificación unívoca
- JSON: interoperatividad

# REST (II)

- 4 operaciones principales (verbos):
  - POST (create)
  - GET (read)
  - PATCH (update)
  - DELETE (delete)
- Equivalentes a CRUD

# Usos

- Aplicaciones para móvil/tablet
- Aplicaciones web
- Aplicaciones de escritorio
- Importaciones/exportaciones complejas
- Redes sociales
- IoT

# Endpoints

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

`https://www.example.com/wp-json/  
[ruta]?[parametros]`

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

# Rutas

- `/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`

# Postman

<https://www.postman.com/product/rest-client/>

Ejercicio



# Documentación

[https://developer.wordpress.org/rest-api/  
reference/posts/](https://developer.wordpress.org/rest-api/reference/posts/)

# Instalar WordPress

```
$ wp core download --locale=es_ES
```

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

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

# Instalar WordPress (II)

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

```
$ wp theme activate twentytwentyone
```

# Instalar WordPress (III)

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

# Namespaces

`http://wordpress.test/wp-json/`

```
name:          "Sitio WordPress de test"
description:   ""
url:          "http://wordpress.test"
home:         "http://wordpress.test"
gmt_offset:   "0"
timezone_string: ""
▼ namespaces:
  0:          "oembed/1.0"
  1:          "wp/v2"
  2:          "wp-site-health/v1"
  3:          "wp-block-editor/v1"
```

# Namespaces (II)

```
wp plugin install woocommerce --activate
```

```
name: "Sitio WordPress de test"
description: ""
url: "http://wordpress.test"
home: "http://wordpress.test"
gmt_offset: "0"
timezone_string: ""
namespaces:
  0: "oembed/1.0"
  1: "wc-admin"
  2: "wc-analytics"
  3: "wc/store"
  4: "wc/store/v1"
  5: "wc/v1"
  6: "wc/v2"
  7: "wc/v3"
  8: "wc-telemetry"
  9: "wccom-site/v1"
  10: "wp/v2"
  11: "wp-site-health/v1"
  12: "wp-block-editor/v1"
```

```
wp plugin deactivate woocommerce
```

# Rutas

`http://wordpress.test/wp-json/`

```
▼ routes:
  ▶ /: {}
  ▶ /batch/v1: {}
  ▶ /oembed/1.0: {}
  ▶ /oembed/1.0/embed: {}
  ▶ /oembed/1.0/proxy: {}
  ▶ /wp/v2: {}
  ▶ /wp/v2/posts: {}
  ▶ /wp/v2/posts/(?P<id>[\d]+): {}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions: {}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+): {}
  ▶ /wp/v2/posts/(?P<id>[\d]+)/autosaves: {}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/autosaves/(?P<id>[\d]+): {}
  ▶ /wp/v2/pages: {}
  ▶ /wp/v2/pages/(?P<id>[\d]+): {}
  ▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions: {}
  ▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+): {}
  ▶ /wp/v2/pages/(?P<id>[\d]+)/autosaves: {}
```

# Rutas (II)

<http://wordpress.test/wp-json/wp/v2/>

```
namespace: "wp/v2"
▼ routes:
  ▶ /wp/v2: {...}
  ▶ /wp/v2/posts: {...}
  ▶ /wp/v2/posts/(?P<id>[\d]+): {...}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions: {...}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+): {...}
  ▶ /wp/v2/posts/(?P<id>[\d]+)/autosaves: {...}
  ▶ /wp/v2/posts/(?P<parent>[\d]+)/autosaves/(?P<id>[\d]+): {...}
  ▶ /wp/v2/pages: {...}
  ▶ /wp/v2/pages/(?P<id>[\d]+): {...}
  ▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions: {...}
  ▶ /wp/v2/pages/(?P<parent>[\d]+)/revisions/(?P<id>[\d]+): {...}
  ▶ /wp/v2/pages/(?P<id>[\d]+)/autosaves: {...}
  ▶ /wp/v2/pages/(?P<parent>[\d]+)/autosaves/(?P<id>[\d]+): {...}
```



## Entrada 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ad corpus dices pertinere-, sed ea, quae dixi, ad corpusne refers? Legimus tamen Diogenem, Antipatrum, Mnesarchum, Panaetium, multos alios in primisque familiarem nostrum Posidonium. Graecum enim hunc versum nostis omnes-: Suavis laborum est praeteritorum memoria. Laelius clamores sofòw ille so lebat Edere compellans gumias ex ordine nostros.... [Seguir leyendo](#)

Publicada el 15 de enero de 2023 [Editar](#)

Categorizado como [Sin categoría](#)

---

## Entrada 3

- Escritorio
- Entradas
- Medios
- Páginas
- Comentarios
- Apariencia
- Plugins
- Usuarios**
- Todos los usuarios
- Añadir nuevo
- Perfil
- Herramientas
- Ajustes
- Cerrar menú

## Acerca de ti

### Información biográfica

Incluye alguna información biográfica en tu perfil. Podrá mostrarse públicamente.

### Imagen de perfil



[Puedes cambiar la imagen de tu perfil en Gravatar.](#)

## Gestión de la cuenta

### Nueva contraseña

### Sesiones

Solo estás conectado en esta ubicación.

## Contraseñas de aplicación

Las contraseñas de aplicación permiten la identificación a través de sistemas no interactivos, como XML-RPC o la API REST, sin proporcionar tu contraseña real. Las contraseñas de aplicación pueden ser anuladas fácilmente. No se pueden usar para los accesos...

### Nombre de la nueva contraseña de la aplicación

Necesario para crear una contraseña de aplicación, pero no para actualizar al usuario.

# wp-admin.php

```
define( 'WP_ENVIRONMENT_TYPE', 'local' );
```

- Escritorio
- Entradas
- Medios
- Páginas
- Comentarios
- Apariencia
- Plugins
- Usuarios**
- Todos los usuarios
- Añadir nuevo
- Perfil
- Herramientas
- Ajustes
- Cerrar menú

### Imagen de perfil



[Puedes cambiar la imagen de tu perfil en Gravatar.](#)

### Gestión de la cuenta

#### Nueva contraseña

[Establecer una nueva contraseña](#)

#### Sesiones

[Desconectar del resto de sitios](#)

Solo estás conectado en esta ubicación.

### Contraseñas de aplicación

Las contraseñas de aplicación permiten la identificación a través de sistemas no interactivos, como XML-RPC o la API REST, sin proporcionar tu contraseña real. Las contraseñas de aplicación pueden ser anuladas fácilmente. No se pueden usar para los accesos tradicionales a tu web.

#### Nombre de la nueva contraseña de la aplicación

Laravel APP

Necesario para crear una contraseña de aplicación, pero no para actualizar al usuario.

[Añadir una nueva contraseña de aplicación](#)

[Actualizar perfil](#)

- Escritorio
- Entradas
- Medios
- Páginas
- Comentarios
- Apariencia
- Plugins
- Usuarios**
- Todos los usuarios
- Añadir nuevo
- Perfil
- Herramientas
- Ajustes
- Cerrar menú

### Imagen de perfil



Puedes cambiar la imagen de tu perfil en [Gravatar](#).

### Gestión de la cuenta

#### Nueva contraseña

[Establecer una nueva contraseña](#)

#### Sesiones

[Desconectar del resto de sitios](#)

Solo estás conectado en esta ubicación.

### Contraseñas de aplicación

Las contraseñas de aplicación permiten la identificación a través de sistemas no interactivos, como XML-RPC o la API REST, sin proporcionar tu contraseña real. Las contraseñas de aplicación pueden ser anuladas fácilmente. No se pueden usar para los accesos tradicionales a tu web.

#### Nombre de la nueva contraseña de la aplicación

Necesario para crear una contraseña de aplicación, pero no para actualizar al usuario.

[Añadir una nueva contraseña de aplicación](#)

Tu nueva contraseña para **Laravel APP** es: `pfU1 j6pC 2d9o k719 jM3N UtqL`

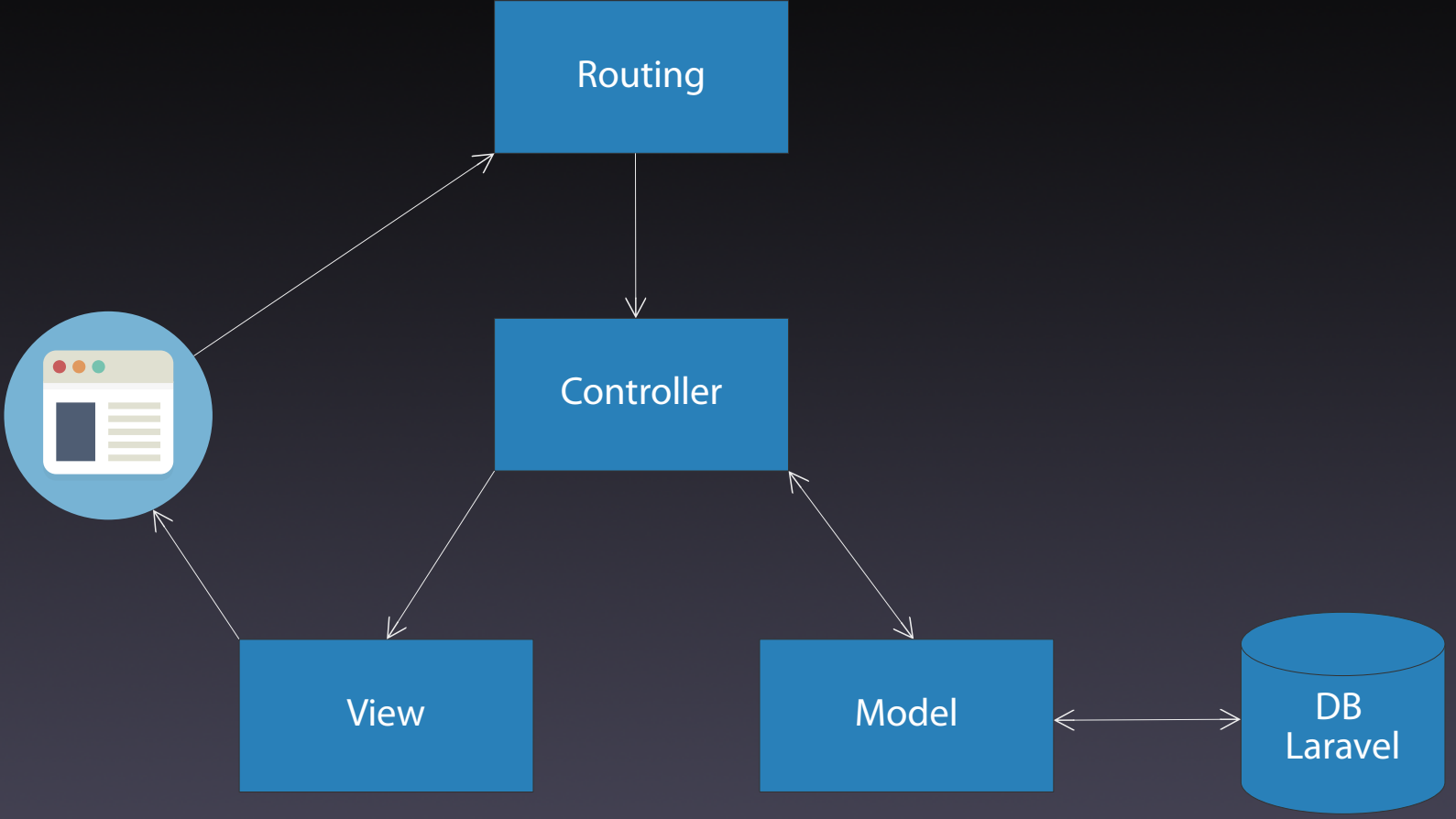
Asegúrate de guardarla en un lugar seguro. No podrás recuperarla.

# Instalar Laravel

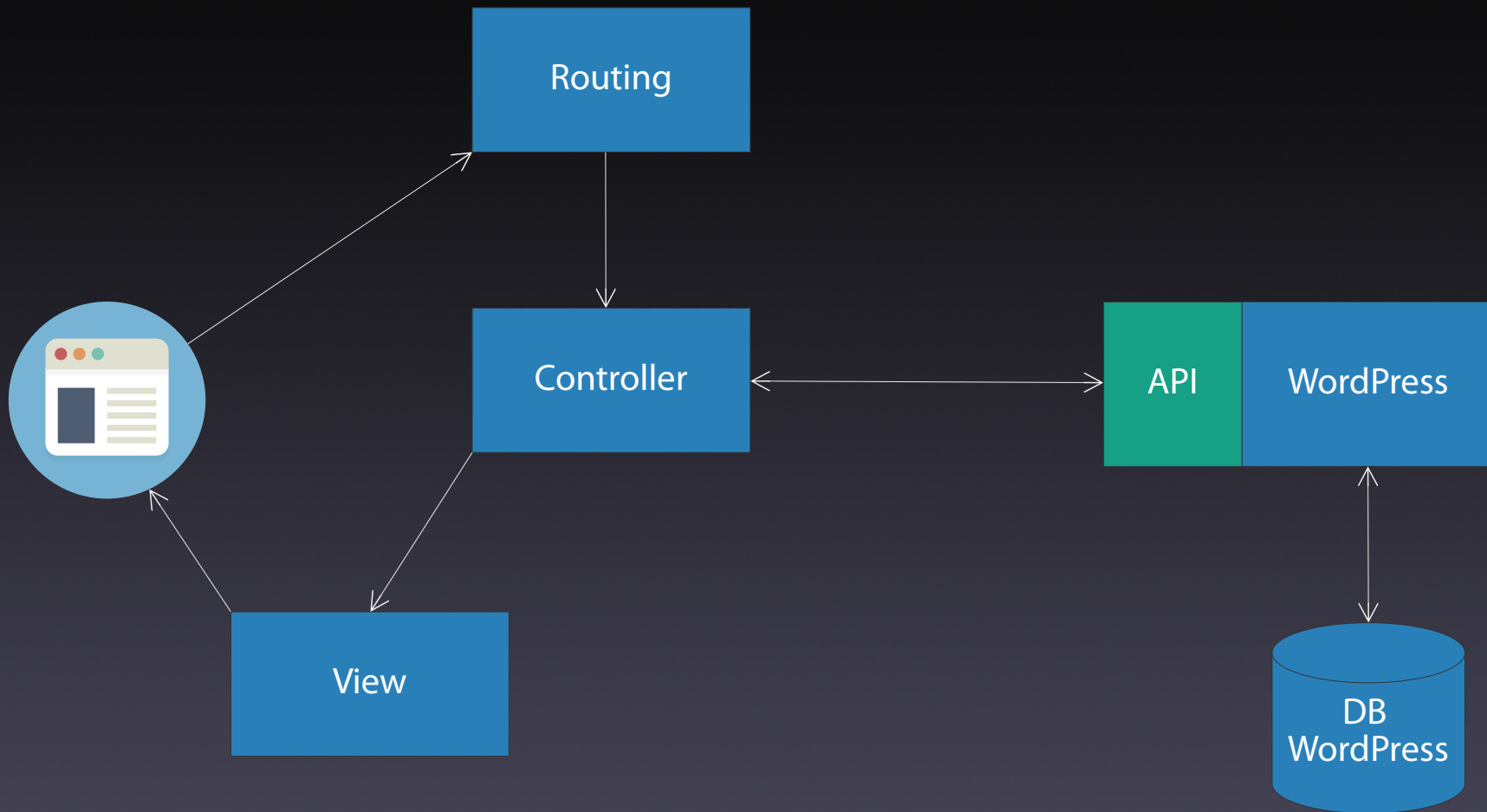
```
$ composer create-project \  
laravel/laravel laravel-zgz
```

# Guzzle

<https://docs.guzzlephp.org>







# Bootstrap

```
wget
```

```
https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css -P public/css
```

# Archivo .env

```
APP_NAME="WordPress REST API Zaragoza"
```

```
APP_URL=http://laravel-zgz.test
```

# Controlador

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

# Archivo de rutas

```
/routes/web.php
```

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

- 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

```
namespace App\Http\Controllers;
```

```
use GuzzleHttp\Client;
```

```
use GuzzleHttp\RequestOptions;
```

# Controlador

```
protected string $url = 'http://wordpress.test/wp-json/wp/v2/posts';
protected string $username = 'admin';
protected string $password = 'La contraseña generada previamente';
protected array $auth;

public function __construct() {
    $this->auth = [
        $this->username,
        $this->password,
    ];
}
```

# Controlador. GET /posts

```
public function index()
{
    $params = array(
        'per_page' => 100,
    );
    $client = new Client();
    $response = $client->request('GET', $this->url, [
        RequestOptions::JSON => $params
    ]);
    $posts = json_decode($response->getBody()->getContents());
    return view('posts.index', compact('posts'));
}
```



Vista layouts/app.blade.php

Vista posts/index.blade.php

Posts <a href="#">Add post</a>		
<a href="#">Entrada 10</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 9</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 8</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 7</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 6</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 5</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 4</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 3</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 2</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 1</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">¡Hola, mundo!</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

# Controlador. GET /posts/{id}

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

Vista posts/show.blade.php

## Entrada 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Torquatus, is qui consul cum Cn. Conferam tecum, quam cuique verso rem subicias; Nam si propter voluptatem, quae est ista laus, quae possit e macello peti? Cum id fugiunt, re eadem defendunt, quae Peripatetici, verba. Duo Reges: constructio interrete. Ergo ita: non posse honeste vivi, nisi honeste vivatur? Tum Torquatus: Prorsus, inquit, assentior; Qui autem diffidet perpetuitati bonorum suorum, timeat necesse est, ne aliquando amissis illis sit miser.

Sed ne, dum huic obsequor, vobis molestus sim. Satis est ad hoc responsum. Ergo hoc quidem apparet, nos ad agendum esse natos. Fortasse id optimum, sed ubi illud: Plus semper voluptatis? Occultum facinus esse potuerit, gaudebit; Summus dolor plures dies manere non potest?

Tria genera bonorum; Sed ad haec, nisi molestum est, habeo quae velim. Rhetorice igitur, inquam, nos mavis quam dialectice disputare? Beatus autem esse in maximarum rerum timore nemo potest. Non igitur potestis voluptate omnia dirigentes aut tueri aut retinere virtutem. Quia dolori non voluptas contraria est, sed doloris privatio. Quem Tiberina descensio festo illo die tanto gaudio affecit, quanto L. Iam doloris medicamenta illa Epicurea tamquam de narthecio proment: Si gravis, brevis; Idemque diviserunt naturam hominis in animum et corpus. Si quidem, inquit, tollerem, sed relinquo.

Quid de Platone aut de Democrito loquar? Quare hoc videndum est, possitne nobis hoc ratio philosophorum dare. Sic, et quidem diligentius saepiusque ista loquemur inter nos agemusque communiter. Quis Aristidem non mortuum diligit? Inrandum est igitur in rerum naturam et penitus quid ea postulet pervidendum; At multis malis affectus. Quod quidem iam fit etiam in Academia. Eiuo, inquit adridens, iniquum, hac quidem de re; Quamquam id quidem licebit iis existimare, qui legerint. Quod dicit Epicurus etiam de voluptate, quae minime sint voluptates, eas obscurari saepe et obrui. Compensabatur, inquit, cum summis doloribus laetitia.

Aufert enim sensus actionemque tollit omnem. Hoc etsi multimodis reprehendi potest, tamen accipio, quod dant. Istic sum, inquit. At quicum ioca seria, ut dicitur, quicum arcana, quicum occulta omnia? Quid est enim aliud esse versutum? Si enim ad populum me vocas, eum. Vitae autem degendae ratio maxime quidem illis placuit quieta.

# Controlador. GET /posts/create

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

Vista posts/create.blade.php



## Crear un nuevo post

Título

Entrada creada usando la REST API. Título

Contenido

Entrada creada usando la REST API. Contenido



Crear post

# Controlador. POST /posts

```
public function store(Request $request)
{
    $params = array(
        'title' => $request['titulo'],
        'content' => $request['contenido'],
        'status' => 'publish'
    );
    $client = new Client([
        'auth' => $this->auth,
    ]);
}
```

# Controlador. POST /posts

```
$response = $client->post($this->url, [  
    RequestOptions::JSON => $params  
]);  
  
if ($response->getStatusCode() == 201) {  
    return redirect()->route('posts.index');  
}  
  
}
```

Posts

Add post

[Entrada creada usando la REST API. Título](#)

Edit

Delete

[Entrada 10](#)

Edit

Delete

[Entrada 9](#)

Edit

Delete

[Entrada 8](#)

Edit

Delete

[Entrada 7](#)

Edit

Delete

[Entrada 6](#)

Edit

Delete

[Entrada 5](#)

Edit

Delete

[Entrada 4](#)

Edit

Delete

[Entrada 3](#)

Edit

Delete

[Entrada 2](#)

Edit

Delete

[Entrada 1](#)

Edit

Delete

[¡Hola, mundo!](#)

Edit

Delete

# Controlador. GET /posts/{id}/edit

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

Vista posts/edit.blade.php

Editar el post

Título

Entrada creada usando la REST API. Título

Contenido

Entrada creada usando la REST API. Contenido

Actualizar post

## Editar el post

### Título

Entrada actualizada usando la REST API. Título

### Contenido

Entrada actualizada usando la REST API. Contenido



Actualizar post



# Controlador. PUT /posts/{id}

```
public function update(Request $request, $id)
{
    $params = array(
        'title' => $request['titulo'],
        'content' => $request['contenido'],
        'status' => 'publish'
    );
    $client = new Client([
        'auth' => $this->auth,
    ]);
}
```

# Controlador. PUT /posts/{id}

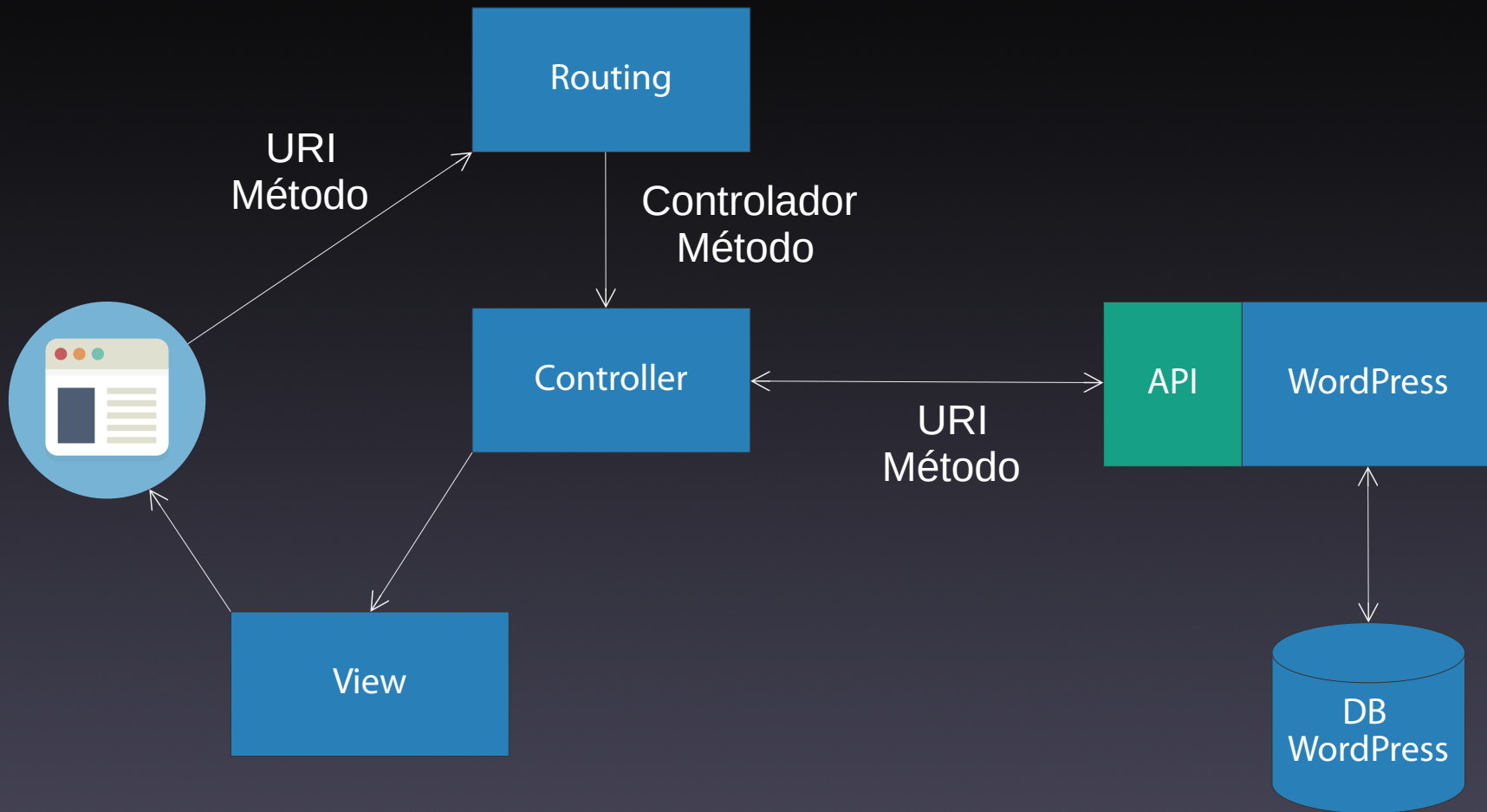
```
$response = $client->put($this->url . '/' . $id, [  
    RequestOptions::JSON => $params  
]);  
  
if ($response->getStatusCode() == 200) {  
    return redirect()->route('posts.index');  
}  
  
}
```

Posts	<a href="#">Add post</a>		
<a href="#">Entrada actualizada usando la REST API. Título</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 10</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 9</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 8</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 7</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 6</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 5</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 4</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 3</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 2</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">Entrada 1</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	
<a href="#">¡Hola, mundo!</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	

# Controlador. DELETE /posts/{id}

```
public function destroy($id)
{
    $client = new Client([
        'auth' => $this->auth,
    ]);
    $response = $client->delete($this->url . '/' . $id);
    if ($response->getStatusCode() == 200) {
        return redirect()->route('posts.index');
    }
}
```

Posts <a href="#">Add post</a>		
<a href="#">Entrada 10</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 9</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 8</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 7</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 6</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 5</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 4</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 3</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 2</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Entrada 1</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">¡Hola, mundo!</a>	<a href="#">Edit</a>	<a href="#">Delete</a>





jesus@jesusamieiro.com

