

Desarrollo de aplicaciones web basadas en Laravel 4



Víctor Téllez Lozano

<http://vtellez.es>

Contenido del curso

DÍA 1

- 1. ¿Qué es Laravel?
- 2. Por qué me gusta
- 3. Características
- 4. Instalación y Configuración
 - Dependencias
 - Deploy
- 5. Estructura y normalización

Contenido del curso

DÍA 1

- 6. Nuestro arsenal
- 7. Rutas en Laravel
 - Tipo de protocolos
 - Filtros
 - Binding
 - API RESTFul
 - Creando Servicios Web

Contenido del curso

DÍA 1

- 8. Manejo de peticiones e inputs
- 9. Blade Templating
- 10. Schema Builder
- 11. Migraciones
- 12. Seeds

0. Antes de empezar

Curso de 8+8 horas... Luego necesitarás:

- Documentación **indispensable**
 - <http://laravel.com/docs/>
 - <http://four.laravel.com/docs/>
 - <http://daylerees.com>
 - <https://leanpub.com/codehappy>
 - <http://codehero.co/series/laravel-4-desde-cero/>

1. ¿Qué es Laravel?

- Es un framework PHP utilizado para construir aplicaciones web de forma expresiva y elegante.
- Busca reducir la sintaxis de código.
- Bebe de los mejores frameworks (Ruby on Rails) de los últimos años.
- Nuevo resurgir de PHP.

2. Por qué me gusta

- Facilita las tareas más típicas de cualquier proyecto y las simplifica mucho.
- Enfocado enormemente a reutilizar conocimiento y desarrollo de los demás.
- Cómodo y verdaderamente elegante.
- Ventajas propias de un framework (Seguridad, rendimiento, etc).

3. Características

- Es un framework basado en el paradigma de programación Modelo-Vista-Controlador (**MVC**).
- Basado en el sistema gestor de paquetes **PHP Composer**.
- Aplicaciones basadas en Laravel como paquetes Composer → Empuje de la **comunidad**.
- **Facilidad** de uso.

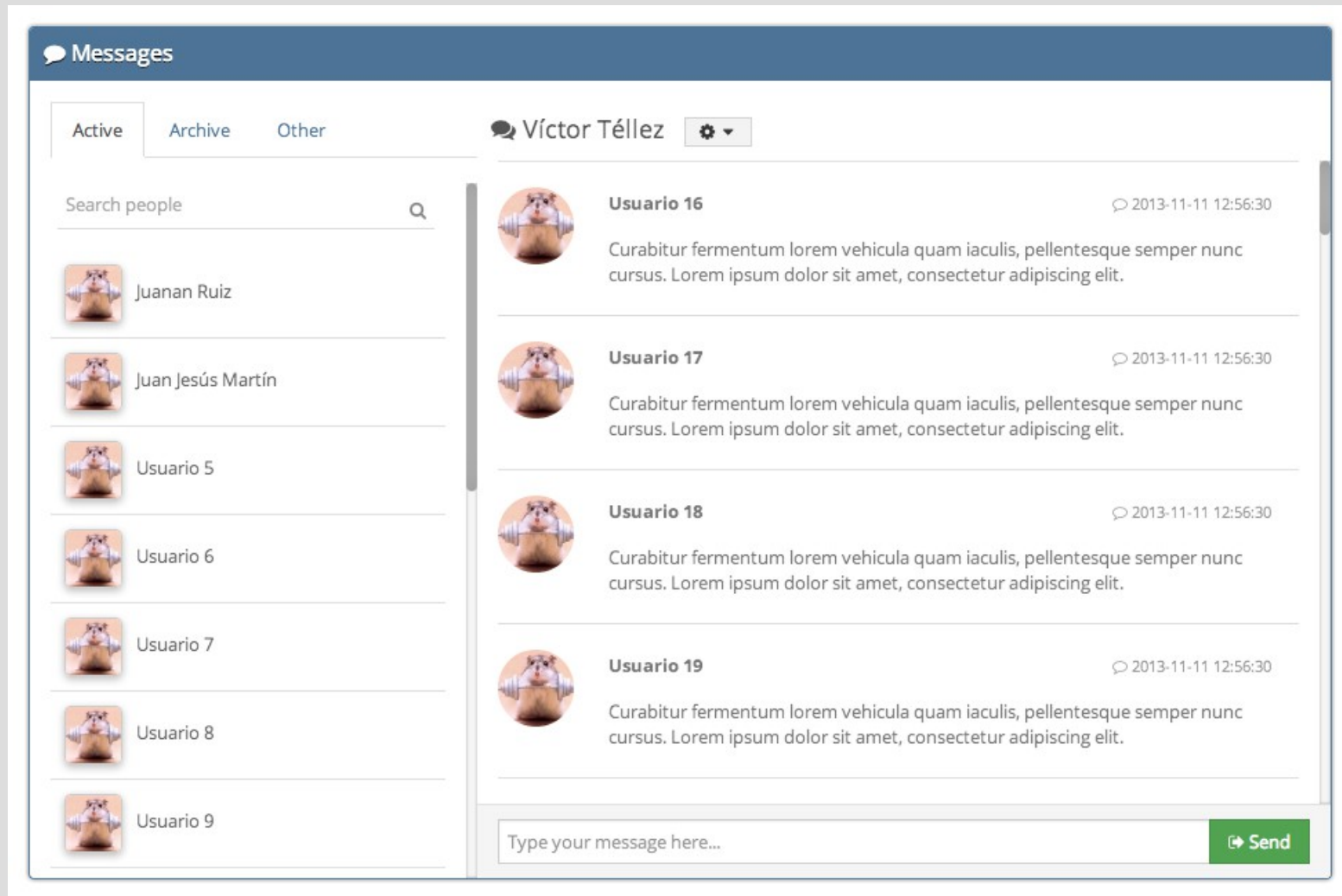
3. Características

- Sistema de **autoload** basado en estandarización de PHP PSR-0
 - <http://www.sitepoint.com/autoloading-and-the-psr-0-standard/>

Nuestra aplicación

- Para ilustrar este curso, nuestro objetivo será construir una aplicación web real basada en Laravel.
 - Se tratará de un sistema de mensajería entre usuarios que...
 - Podrá soportar conversaciones individuales o de grupos.
 - Manejará sesiones y ORM para ssggbbdd.
 - Basado en servicios web.
 - Haremos uso de estos WS desde Ajax.

Nuestra aplicación



4. Instalación y configuración

- Antes de empezar, hablemos de las **dependencias** de Laravel:
 - Composer
 - Librería Mcrypt
 - MAC OS: Xcode, devtools

4.1 Instalación

- Instalando Laravel vía **composer**:
 - `composer create-project laravel/laravel your-project-name --prefer-dist`
- Instalando Laravel vía **git**:
 - `git clone https://github.com/laravel/laravel.git`
- Instalando Laravel vía **zip**:
 - `https://github.com/laravel/laravel/archive/master.zip`

4.1 Instalación

- En cualquier caso, debemos “instalar” Laravel:
 - composer install
 - chmod -R o+w app/storage/

4.2 Deploy como Virtual Host

Utilizaremos apache como servidor Web:

```
<VirtualHost *:80>
```

```
    ServerName curso.dev
```

```
    ServerAlias curso.dev *curso.dev
```

```
    DocumentRoot /var/www/laravel/public
```

```
</VirtualHost>
```

! Añadir entrada a /etc/hosts

4.2 Deploy como Virtual Host

La cosa va bien si...



You have arrived.

5. Estructura y normalización

Laravel nos propone la siguiente organización de trabajo:

- App/ - Carpeta principal de nuestra aplicación. La veremos con más detalle.
- Artisan.php – Script PHP que actuará de línea de comandos de nuestra app.
- Bootstrap – No confundir con BS, se trata de las librerías propias de Laravel
- Composer.json – Definición de dependencias

5. Estructura y normalización

- Vendor/ – Código de paquetes de terceros
- Public/ - Parte pública de nuestro proyecto (js, img, css, y todo tipo de assets). También contiene index.php principal.

5. Estructura y normalización

Centrándonos en app/

- Config/ - Directorio de configuración principal. Lo veremos en detalle
 - <http://laravel.com/docs/configuration>
- Controllers/
- Views/
- Models/
- Database/ - Alojará la definición de esquema, las migraciones y seeds.

5. Estructura y normalización

Centrándonos en app/

- Storage/ - Directorio de cacheado, sesiones, vistas compliadas y logs en servidor.
- Test/ - Directorio para alojar nuestras pruebas unitarias de sistema.
- Lang/ - Directorio para alojar sistema de ficheros para la localización e internacionalización.
- Routes.php y Filters.php

6. Nuestro arsenal

Antes de empezar, repasemos ciertas herramientas que considero imprescindibles:

- Sublime Text 3
 - <http://www.sublimetext.com/3>
- Sistema de control de versiones
 - GIT o SVN
- Producteev – Gestión de tareas
 - <https://www.producteev.com/>

6. Nuestro arsenal

- Laravel schema designer
 - <http://laravelsd.com/>
- Bootstrap
 - <http://getbootstrap.com/>
- Bootsnip
 - <http://bootsnipp.com/>
- ThemeForest
 - <http://themeforest.net/category/site-templates/admin>

7. Rutas en Laravel

- Se definen en `app/routes.php`
- Distingue rutas en función del protocolo.
- Admite reutilización de filtros
 - Filtros after
 - Filtros before
- Binding
- RESTFul
- Distinción de llamadas AJAX y WS

7. Rutas en Laravel

<http://laravel.com/docs/routing>

8. Manejo de peticiones e inputs

<http://laravel.com/docs/requests>

9. Sistemas de plantillas y Blade

- Laravel permite servir las vistas:
 - Como archivos .html
 - Como archivos .php
 - Como archivos blade.php
- Estos últimos en realidad se compilan y cachean como ficheros .php
 - Estilizan el código.
 - Facilitan su lectura y usabilidad.
 - Similar a jekyll

9. Sistemas de plantillas y Blade

- Directivas principales:
 - @extends
 - @section - @stop
 - @yield
 - @include
- Estructuras propias de PHP
 - echo {{ }}
 - Estructuras if, if/else, if/elseif
 - Estructuras for, foreach

9. Sistemas de plantillas y Blade

<http://laravel.com/docs/templates>

10. Schema Builder

- Laravel permite utilizar bases de datos, definidas de forma tradicional (sql a mano, phpmyadmin, etc).
- Además proporciona una Clase, pensada para actuar como Fachada de BBDD:
 - <http://laravel.com/docs/schema>

11. Sistema de Migraciones

<http://laravel.com/docs/migrations>

- Comandos para migraciones:
 - `php artisan migrate`
 - `php artisan migrate:rollback`
 - `php artisan migrate:refresh`

12. Seeding, población de la bbdd

<http://laravel.com/docs/migrations#database-seed>

- Comandos para poblar bbdd:
 - php artisan **db:seed**
 - php artisan migrate --seed
 - php artisan migrate:refresh --seed