

Laboratorio de Redes y Sistemas Operativos

Laravel 5.0

Sergio Woicik
Guillermo Sanchez

16 de Julio de 2015

1 Introducción

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple.

Sus principales características son:

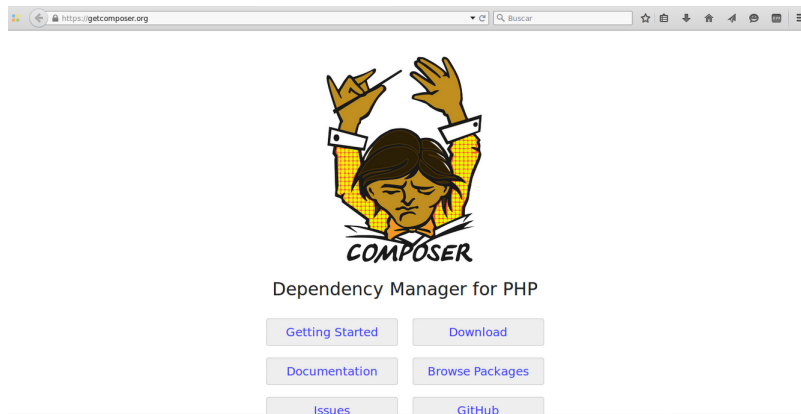
- Sistema de ruteo, también RESTful
- Blade, Motor de plantillas
- Peticiones Fluent
- Eloquent ORM
- Basado en Composer
- Soporte para MVC

2 Instalación

La instalación de Laravel se hace a través de un manejador de dependencias llamado Composer, similar a NPM en node.js y a los bundles de Ruby. Composer nos permite manejar las dependencias de nuestros proyectos, ya sea la descarga de un framework completo como Laravel o componentes más sencillos.

- Paso 1: Instalar Curl (Es un programa para transferencia de archivos con sintaxis URL, que soporta FTP, FTPS, HTTP, HTTPS, etc) para poder instalar composer.

```
sudo apt-get install curl
```



- Paso 2: Instalar el manejador de dependencias Composer:
curl -sS https://getcomposer.org/installer — php
o sin curl:
php -r "readfile('https://getcomposer.org/installer');" — php
- Paso 3: Una vez instalado el Composer, este se instala en una carpeta con el nombre composer.phar pero nosotros queremos que sea una herramienta global entonces para se pueda utilizar en todos los proyectos PHP.
sudo mv composer.phar /usr/local/bin/composer
- Paso 4: Crear un archivo con las siguientes características en formato json que es útil para composer: composer.json

```
composer.json
1 {
2     "name": "Guille/Laravel",
3     "description": "Tp Labo",
4     "type": "project",
5     "authors": [
6         {
7             "name": "Woicik Sergio"
8         },
9         {
10            "name": "Sanchez Guillermo"
11        }
12    ],
13    "require": {
14        "php": ">=5.3.0"
15    },
16    "autoload": {
17        "psr-4": {
18        }
19    }
20 }
21 }
22 }
23 }
24 }
```

Donde:
name: Va la ruta del proyecto.
description: Una descripción del proyecto.
type: Proyecto.
authors: Donde va el nombre y email de los autores.

required: Van los paquetes de los cuales dependen mi aplicación.
autoload: Carga automática de las clases del proyecto.

- Paso 5: Para Instalar Composer dentro de nuestro proyecto y hacer el autoload de nuestras clases, nos situamos sobre el y hacemos:
composer install
- Paso 6: Ahora necesitamos instalar las dependencias del proyecto:
composer global require "laravel/installer= 1.1"

```
ariel@skull1 ~/Galle/Laravel $ composer install
Installing dependencies (including require-dev)
Nothing to install or update

ariel@skull1 ~/Galle/Laravel $ composer global require "laravel/installer=1.1"
Composer (1.10.0) has been checked
Running composer require --global with package information
Running require --global (including require-dev)
- Installing composer/composer (v2.7.1)
  Downloading: 100%
- Installing symfony/console (v2.7.1)
  Downloading: 100%
- Installing symfony/yaml (v2.2.4)
  Downloading: 100%
- Installing symfony/serializer (3.0.0)
  Downloading: 100%
- Installing symfony/http-kernel (3.1.0)
  Downloading: 100%
- Installing symfony/http-foundation (3.1.0)
  Downloading: 100%
- Installing symfony/http-client (v1.2.1)
  Downloading: 100%
symfony/console suggests installing symfony/event-dispatcher ()
symfony/console suggests installing psr/log (for using the console logger)
puuzlehty/ringphp suggests installing ext-curl (Guzzle will use specific adapters if cURL is present)
Warning: This file is deprecated.
Warning: This file is deprecated.
```

- Paso 7: Crear proyecto:
composer create-project laravel/laravel nombre 5.0.*

3 Dependencias, Problemas y Soluciones

Laravel es un framework que trabaja con php5 por lo tanto hace falta instalarlo para esto hacemos:

```
sudo apt-get install php5
```

Un problema que nos surgió fue que larevel requiere ext.mcrypt. Mcrypt es un método de cifrado de archivos que utiliza técnicas seguras para el intercambio de datos.

La solución a esto la encontramos mediante el siguiente comando:

```
sudo ln -s /etc/php5/mods-available/mcrypt.ini /etc/php5/cli/conf.d/
```

o sino instalando los siguientes paquetes:

```
sudo apt-get install mcrypt
```

```
sudo apt-get install php5-mcrypt
```

```
ariel@skull1 ~/Guille/Laravel $ composer create-project laravel/laravel tpLabo 5.0.*
Installing laravel/laravel (v5.0.22)
- Installing laravel/laravel (v5.0.22)
  Loading from cache
  2) Edit your test profile
Created project in tpLabo
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Your requirements could not be resolved to an installable set of packages.

Problem 1
- Installation request for laravel/framework v5.0.16 -> satisfiable by laravel/framework[v5.0.16].
- laravel/framework v5.0.16 requires ext-mcrypt * -> the requested PHP extension mcrypt is missing from your system.
```

4 Esquema de un proyecto

La estructura de archivos de laravel 5 es la siguiente:

- El modelo se encuentra dentro de app/ en la forma que tenga más sentido para tu aplicación.
- Las vistas se ubican en /resources/views/
- Los controladores se ubican en app/Http/Controllers.
- Los comandos personalizados que crees estarán dentro de /app/Console/.
- Los repositorios de la base de datos se encuentran en database/migrations
- Los archivos css se encuentran en public/css
- Los test se encuentran en la carpeta tests
- Los servicios se encuentran en app/Services

Acá una foto de como se vería el proyecto:

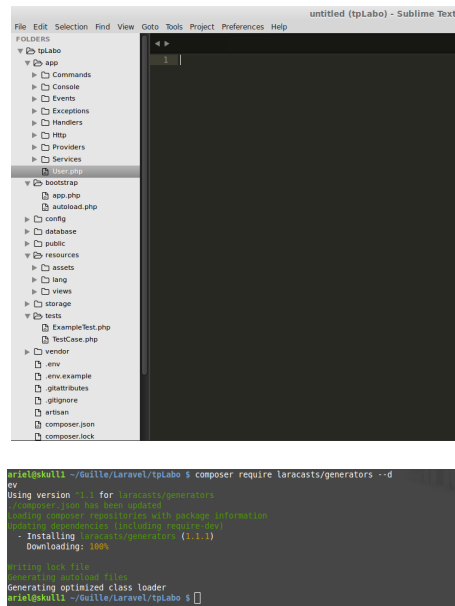
5 Generadores

Los generadores son pequeños comandos de consola, que nos permiten generar código listo para usar, evitándonos escribir código que casi siempre hacemos repetidas veces y realizan trabajos minimos.

Para utilizar los generadores de Laravel 5 hace falta hacer falta ejecutar el siguiente comando:

```
composer require laracasts/generators --dev
```

Algunos generadores de laravel 5 son:



- **make:model**
Crear una nueva clase del modelo y su tabla en la base de datos
- **make:controller**
Crear una nueva clase controlador de recursos
- **make:migration**
Crear un nuevo archivo de migración
- **make:middleware**
Crear una nueva clase de middleware
- **make:request**
Crear una nueva clase de petición
- **make:provider**
Crear una nueva clase de servicios
- **make:event**
Crear una nueva clase de evento
- **make:console**
Crear un nuevo comando Artisan
- **make:command**
Crear una nueva clase de comandos

Estos se ejecutan haciendo php artisan make:...

Para mas información sobre los generadores de laravel:

```
php artisan list
```

6 Panel Administrador

Es un generador de una interfaz para el panel administrativo, basado en la configuración de las aplicaciones de Laravel 5.

- Instalación

Hay que incluir el paquete necesario a Composer, agregando al archivo composer.json de nuestro proyecto:

```
"require": "vivify-ideas/admin-panel-generator": "dev-master"
```

```
"name": "laravel/laravel",
"description": "The Laravel Framework.",
"keywords": ["framework", "laravel"],
"license": "MIT",
"type": "project",
"require": {
    "laravel/framework": "5.0.*",
    "vivify-ideas/admin-panel-generator": "dev-master"
},
"require-dev": {
    "phpunit/phpunit": "~4.0",
    "phpspec/phpspec": "~2.1",
    "laracasts/generators": "1.1"
},
```

Luego hay que actualizar el compose, ejecutando en la terminal "composer update". Una vez realizada esa operación habrá que agregar el proveedor del servicio en la aplicación. Para ello abrimos el archivo "app.php" que se encuentra en la carpeta "config" del proyecto y se añade un nuevo elemento a "providers":

```
'providers' => [
    /*
     * Admin Panel Generator
     */
    'VivifyIdeas\AdminPanelGenerator\Providers\AdminPanelGeneratorProvider',
    /*
     * Laravel Framework Service Providers...
     */
    'Illuminate\Foundation\Providers\ArtisanServiceProvider',
    'Illuminate\Auth\AuthServiceProvider',
    'Illuminate\Bus\BusServiceProvider',
    'Illuminate\Cache\CacheServiceProvider',
```

El último paso en el proceso de instalación es publicar las dependencias de los paquetes, ejecutando en la terminal: php artisan vendor:publish

```
sergio@linuxlite-VirtualBox:~/Desktop/project$ php artisan vendor:publish
   Command "vendor:publish" is not defined.

panel_generator.php | (config/vivifyideas/admin-panel-generator.php)
[...]
```

De este modo se creará un nuevo archivo de configuración en la dirección: " config/vivifyideas/admin-panel-generador.php", el cual se tendrá que editar para configurar nuestro panel de administración.

- Configuración

El nuevo archivo para administrar creado anteriormente tendrá el siguiente aspecto:

Las diferentes secciones de configuración:

```
<?php
return [
    'prefix' => 'admin',
    'authMiddleware' => 'auth',
    'tables' => [],
    'rowsPerPage' => 15,
    'columns' => [],
    'filters' => [],
    'forms' => [],
    'validationRules' => []
];
```

- prefix: representa el prefijo URL de nuestro panel de administración. Por ejemplo, si se establece "admin", la URL será similar a esta: <http://localhost:8000/admin>.
- authMiddleware: se utiliza para autorizar a los usuarios admin. De forma predeterminada se ajusta para Laravel default "auth" middleware, pero se puede usar nuestro propio "middleware"
- tables: Representan la lista de nombres de la tabla que se quiere administrar a través de nuestro panel de administración. Ejemplo: 'tables' => ['alumno', 'materias']
- rowsPerPage: establece el número de filas por página que se quiere mostrar en la lista. De forma predeterminada se establece en 15.
- columns: define las columnas que se quieren mostrar para cada tabla. En primer lugar se definen para que cuadro se quiere configurar filtros. Para cada filtro se pueden configurar 3 parámetros: "label", "type" (ambos requeridos) y "compare".
- forms: se utiliza para crear y editar algunos modelos. Representa la lista de campos que se quiere permitir introducir al usuario administrador al crear/modificar ciertos modelos.
- validationRules: aquí se pueden configurar las reglas de validación para cada tabla. Será llamado cuando el administrador quiera crear o editar alguna entidad.

Para correr el servidor:

php artisan serve
y nos conectamos a localhost:8000/admin

7 Fuentes

- <https://getcomposer.org/>
- <https://github.com/Vivify-Ideas/admin-panel-generator>