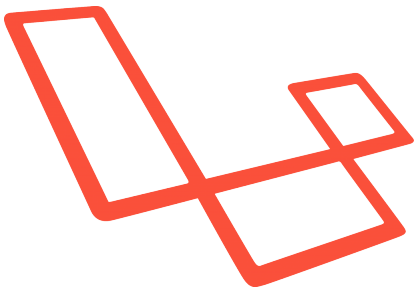
Programadores Chile
WorkShopDay

Workshop
Laravel 5.4



Expositor
Sebastian Zamorano Jara

25 de Marzo de 2017

Revisiones

Editor	Comentarios	Versión	Fecha
Arturo Mantinetti	Diseño Template, estructura inicial y algunos apuntes	1.0	03/04/2017

Índice general

1. Introducción

1.1. Resumen

2. Lo Nuevo en PHP 7

- Permite forzar tipos de entrada y salida de una función
- Implementación del comparador <=> retornando -1 0 1 según sea menor, igual o mayor
- Nuevo Operador ?? Ver Listing 2.1

```php
<?php
// obtener el valor de $_GET['usuario'] y devolver 'nadie'
// si no existe.
$nombre_usuario = $_GET['usuario'] ?? 'nadie';
// Esto equivale a:
$nombre_usuario = isset($_GET['usuario']) ? $_GET['usuario'] : 'nadie';
```

Listing 2.1: isset vs ??

3. Aprendamos PHP Correctamente

3.1. Buenas Practicas en PHP

- Nunca usar echo

4. Conozcamos Laravel

4.1. ¿Por que Utilizarlo?

La principal razón que se utiliza Laravel en el mercado es su facilidad de uso en comparado a otros frameworks. Ademas este tiene una gran comunidad en aumento. En el caso que debamos desarrollar una API, existe una version ligera conocida como Lumen.

4.2. Estructura de Carpetas

4.2.1. .env

Archivo de configuración de Laravel, este se encuentra en la carpeta principal, en este encontraremos las principales configuraciones de la base de datos, del correo electrónico que utilizara laravel y si se encuentra en modo de prueba.

4.2.2. app

En esta carpeta encontraremos los modelos, siendo estos las clases definidas por nosotros como entidades. Dentro de app podemos encontrar la carpeta Console que contiene el archivo Kernel donde definiremos nuestros comandos protegidos de consola y el listado de acciones programadas.
También podemos encontrar la carpeta Controllers dentro de Http, en esta carpeta se sitúan los controladores de la aplicación, es decir donde añadiremos las lineas de código para procesar los datos que necesitamos, el nombre de los controladores por norma debe terminar en "Controller.php".
Dentro de Http encontraremos además una carpeta Middleware, acá van las funciones que nos permitirán filtrar peticiones y agregar distintas condiciones a estas.

4.2.3. config

Acá se encuentran los archivos de configuración, dentro del el archivo app.php podemos configurar el nombre de nuestra aplicación, la zona horaria, el lenguaje, cifrado y los paquetes que utilizaremos dentro de nuestra aplicación. También podremos encontrar archivos de configuración de correo, almacenamiento de archivos, cache, entre otros.

4.2.4. database

Nos centraremos en una carpeta migrations, que esta contenida dentro de esta. En este lugar encontraremos nuestros archivos de creación de la base de datos.

4.2.5. public

Esta carpeta es a la cual accederá el usuario via el navegador, aca el index es el encargado de importar el framework, también podremos agregar dentro de esta carpeta archivos css, js, imagenes, etc. que nosotros queramos que cualquiera tenga acceso sin filtro alguno.

4.2.6. resources

Dentro de esta carpeta encontraremos otra llamada views, acá es donde dejaremos todas las vistas en formato blade.php de nuestra aplicación. También podemos encontrar una carpeta llamada lang, en esta debemos agregar los lenguajes de Laravel que descarguemos.

4.2.7. routes

Acá encontraremos un archivo llamado web.php, este contiene las rutas de la aplicación, y en console.php tenemos la posibilidad de agregar nuestros propios comandos.

4.2.8. storage

Dentro de esta carpeta se almacenaran los archivos que subamos desde Laravel, también podremos encontrar el Log de el framework.

5. Utilicemos Laravel

5.0.1. Instalando Laravel

Para utilizar Laravel necesitamos tener previamente instalado php y composer, este ultimo lo podemos instalar del la siguiente forma:

Composer en Windows

Instalar composer en windows es una tarea simple, tan solo debemos descargar el instalador de *https://getcomposer.org/download/* y ejecutar el exe.

Composer en Linux y OSX

Debemos descargar el instalador de *https://getcomposer.org/installer* y luego de esto lo ejecutamos con *$ sudo php – –install-dir=/usr/local/bin –filename=composer*

Luego de esto podremos crear nuestro proyecto con
composer create-project –prefer-dist laravel/laravel <nombre del proyecto>

5.0.2. Diagrama Funcionamiento

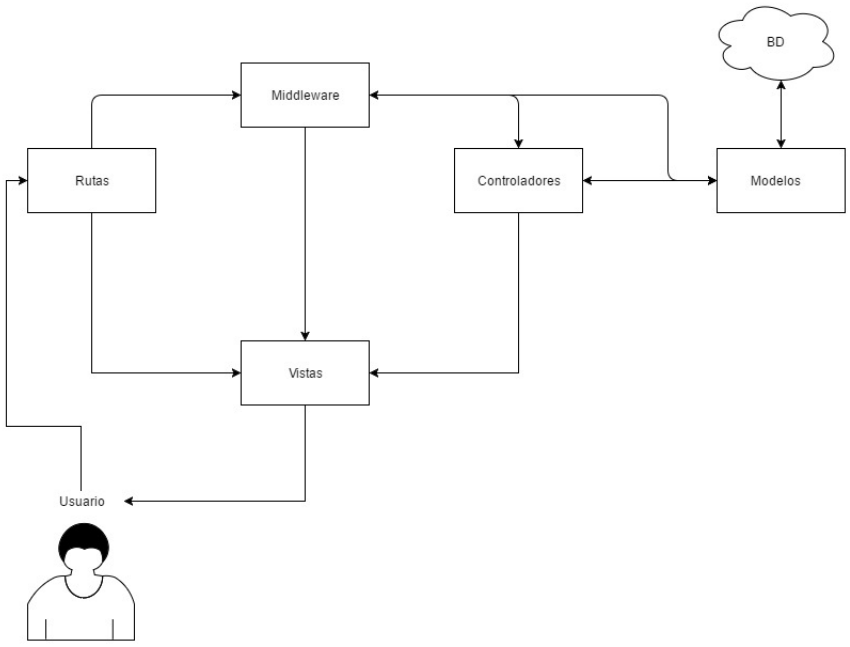


Figura 5.1: Diagrama Funcionamiento Laravel

5.0.3. Modelos

En laravel tenemos acceso a un ORM[1] llamado Eloquent, este hace uso de los Modelos como intermediarios o representantes de la base de datos, estos tienen la capacidad de detectar automáticamente la tabla si es que el nombre del modelo esta en singular y la tabla esta el mismo en plural. Dentro de estos modelos encontraremos y podremos usar una propiedad llamada fillable, donde los valores que se encuentren en esta propiedad parasan a ser un tipo de constructor, ya que podremos asignarlos en conjunto vía un arreglo y nos devolvera un nuevo objeto, con estos atributos. Tambien aca podremos relacionar objetos usando hasOne, hasMany, belongsTo y belongsToMany, es decir tiene uno, tiene muchos, pertenece a uno y pertenece a muchos.

```
//si tenemos un usuario que tiene libros lo podemos definir de la siguiente forma
public function libros()
{
    return $this->belongsTo('App\Libros', 'foreign_key', 'local_key');
}
```

Listing 5.1: Ejemplo de Relaciones

Podemos crear la base de un modelo rápidamente gracias a artisan, para esto en la consola dentro del directorio raíz del proyecto escribimos *php artisan make* : *model nombre − del − modelo*

5.0.4. Controladores

Acá es donde definiremos las peticiones y las manipularemos según lo que requiramos. Estos por convención debe contener Controller al final de su nombre.
Para crear un controlador lo hacemos vía consola con:
php artisan make : *controller nombre − del − controlador*

5.0.5. web.php

En este archivo se encuentran las rutas, estas las definiremos de la siguiente forma:

```
Route::get('/enlace', 'NombreControladorr@Funcion');
Route::post('/insertar', 'NombreControlador@Funcion');
```

Listing 5.2: Ejemplo de rutas

5.0.6. Creando Autenticación de Usuarios

En laravel podemos crear de una forma muy simple un sistema de usuarios con el siguiente comando:
php artisan make : *auth*
Esto nos generara las vistas de ingresar, registrarse y recuperar contraseña, el modelo de usuario, mas sus respectivos controladores.

[1](Object-Relational mapping, Mapeo Objeto-Relacional en español

Índice de figuras