Mansori Amar Sara

smansori@uoc.edu

Abstract

Desarrollo de web de noticias utilizando el framework de PHP Laravel

PEC 3 – dESARROLLO BACKEND CON PHP

LARAVEL

1. Contenidos

[1. Paso 1 1](#_Toc105760700)

[1.1 Heading 2 1](#_Toc105760701)

[1.2 Heading 2.2 1](#_Toc105760702)

[2.1 Heading 2.1 1](#_Toc105760703)

1. Instalación de Laravel de forma local

## Instalación a través de composer

Tras comprobar que el equipo tiene tanto PHP como Composer instalados, se pueden crear proyectos de Laravel utilizando el CLI de Artisan, e inicializar el servidor local de desarrollo con el comando “serve”:

“composer create-project laravel/laravel name-of-the-app”

“php artisan serve” -> Hace que esté disponible en http://localhost:8000

## Comparativa entre Laravel y CodeIgniter o gestores de contenidos

Laravel me ha parecido más parecido a CodeIgniter que a gestores de contenidos como Wordpress o Laravel, ya que no es tan opinionado como estos y permite una mayor capacidad de customización (además de que ambos siguen el Modelo Vista Controlador).

Por un lado, la parte positiva de Laravel es este nivel de customización o capacidad de desarrollar al gusto del programador, aunque por otro lado también se hace más complejo debido a que debe hacerse por linea de comandos y requiere cierto conocimiento de programación.

En cuanto a las diferencias entre Laravel y CodeIgniter en términos de su instalación, me ha sorprendido la cantidad de opciones disponibles en Laravel, pudiendo instalarse de forma local o dockerizada, y habiendo diferentes maneras de hacerlo dentro de cada una de las opciones mencionadas, lo que me hace considerar que Laravel se pueda adaptar mejor a las necesidades de cada desarrollador.

1. Crear las migraciones y modelos necesarios

En primer lugar deberemos crear la base de datos (en nuestro caso se llamará news) en la que posteriormente añadiremos las tablas, filas y columnas necesarias. Para eso, podemos hacerlo desde la interfaz gráfica o inicializando desde la terminal SQL y usando los siguientes comandos:

* mysql -uroot -p
* create database news

Una vez la base de datos esté creada y corroboremos que tiene el mismo nombre que la base de datos referenciadas en el archivo de entorno (.env) de nuestro proyecto de Laravel, necesitaremos crear tres migraciones y dos modelos para el ejercicio propuesto: modelo Article con migración create\_article\_table, modelo Category con migración create\_category\_table y migración para crear una tabla que albergue las relaciones many to many entre Category y Article.

Para crear estas migraciones en Laravel, podemos ejecutar los comandos del php artisan por separado, o crear un modelo y una migración relacionada usando un flag.

* php artisan make:model Article -m
* php artisan make:model Article -m
* php artisan make:migration create\_article\_category\_table

Tras crear estos archivos, derebemos realizar ciertas modificaciones para que reflejen los campos y valores que queremos que tenga nuestra base de datos.

En las migraciones (que estarán en el directorio “database/migrations”) añadiremos en la función que se le pasa como segundo parámetro al método create() del Schema, todas las columnas que queremos que tenga nuestra tabla, indicando el tipo de columna en base a los datos que va a albergar (basándonos en los [tipos de columna de Laravel](https://laravel.com/docs/9.x/migrations#available-column-types)).

Text

Description automatically generated

Figure . "create-categories-table"

Text

Description automatically generated

Figure . "create-articles-table"

Text

Description automatically generated

Figure . "create-article-category-table"

En los modelos (que están en el directorio “App/Models/”, teniendo en cuenta que la relación entre los modelos es de many to many (un artículo puede tener muchas categorías y una categoría pertenecer a muchos artículos), desarrollaremos un método para cada modelo que nos permita acceder a las categorías desde el modelo Article y a los artículos que contengan la categoría del modelo Category.

También podemos introducir una variable llamada $fillable que indica qué valores pueden ser asignados a través de lo que se conoce como “mass assignment” o asignación masiva, debido a que esta forma de asignar valores puede presentar vulnerabilidades, esto previene que se asignen campos que no está previsto que cambien).

Text

Description automatically generated

Figure . Model\Article

Text

Description automatically generated

Figure . Model\Category

## Crear elementos con Tinker

Para comprobar que nuestros modelosy migraciones están correctamente creados, podemos utilizar una herramienta llamada Tinker que nos permite acceder al modelo y crear elementos a través de la terminal.

Para ello tendremos que introducir el comando: php artisan tinker

Una vez que estemos dentro del modo de tinker podemos utilizar comandos como los siguientes para popular nuestra base de datos:

* $article = new App\Models\Article; (guardamos una instancia del modelo Article)
* $article->title=’Título de la noticia’; (le asignamos los diferentes valores a las columnas declaradas en la migración)
* $article->author=’Anónimo’;
* $article->content=’Lorem ipsum dolor sit amet.
* $article->save()’ (guardamos la fila en la base de datos)

En caso de que quisiéramos asignarle alguna categoría, tenemos que tener en cuenta la existencia de la pivot table que relaciona los Article con las Category. Para ello haríamos lo siguiente (a continuación del comando anterior o seleccionando el artículo de la base de datos al que le queremos asignar una categoría con por ejemplo Article::find(id):

* $article->categories()->attach(category\_id) -> si queremos asignarle una categoría
* $article->categories()->attach( [ category1\_id, category2\_id… ] ) -> si queremos asignarle varias categorías a la vez

1. Paso 3
2. Instalación de Laravel de forma local