# Guia del curso Laravel + Angular

# Composer

Composer es el manejador de dependencias para PHP. Una dependencia puede ser tanto un framework (como Laravel o Symfony) así como paquetes o componentes, por ejemplo para la generación de reportes en Excel o PDF, aceptar pagos con tarjetas de crédito y mucho más.

**Instalación**

Requiere PHP 5.3.2+ para funcionar.

**En MacOSX y Linux**

Fuente <https://getcomposer.org/download/>

**- Opción 1:**

Escribir o copiar y pegar en el terminal:

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php -r "if (hash\_file('sha384', 'composer-setup.php') === '8a6138e2a05a8c28539c9f0fb361159823655d7ad2deecb371b04a83966c61223adc522b0189079e3e9e277cd72b8897') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"

php composer-setup.php

php -r "unlink('composer-setup.php');"

**- Opción 2:**

Descargar la version estable <https://getcomposer.org/composer-stable.phar>

**Configuración Global**

Abrir el terminal y situarse en el directorio donde se descargó el archivo composer.phar, ya sea descargado con la opción 1 o 2.

Mover a un directorio global del Sistema operativo OSX

mv composer.phar /usr/local/bin/composer

Si el comando anterior falla por permisos intenta usando “**sudo**” antes de “**mv**”.

Ingresar al directorio bin donde se movió el composer.phar

cd /usr/local/bin/

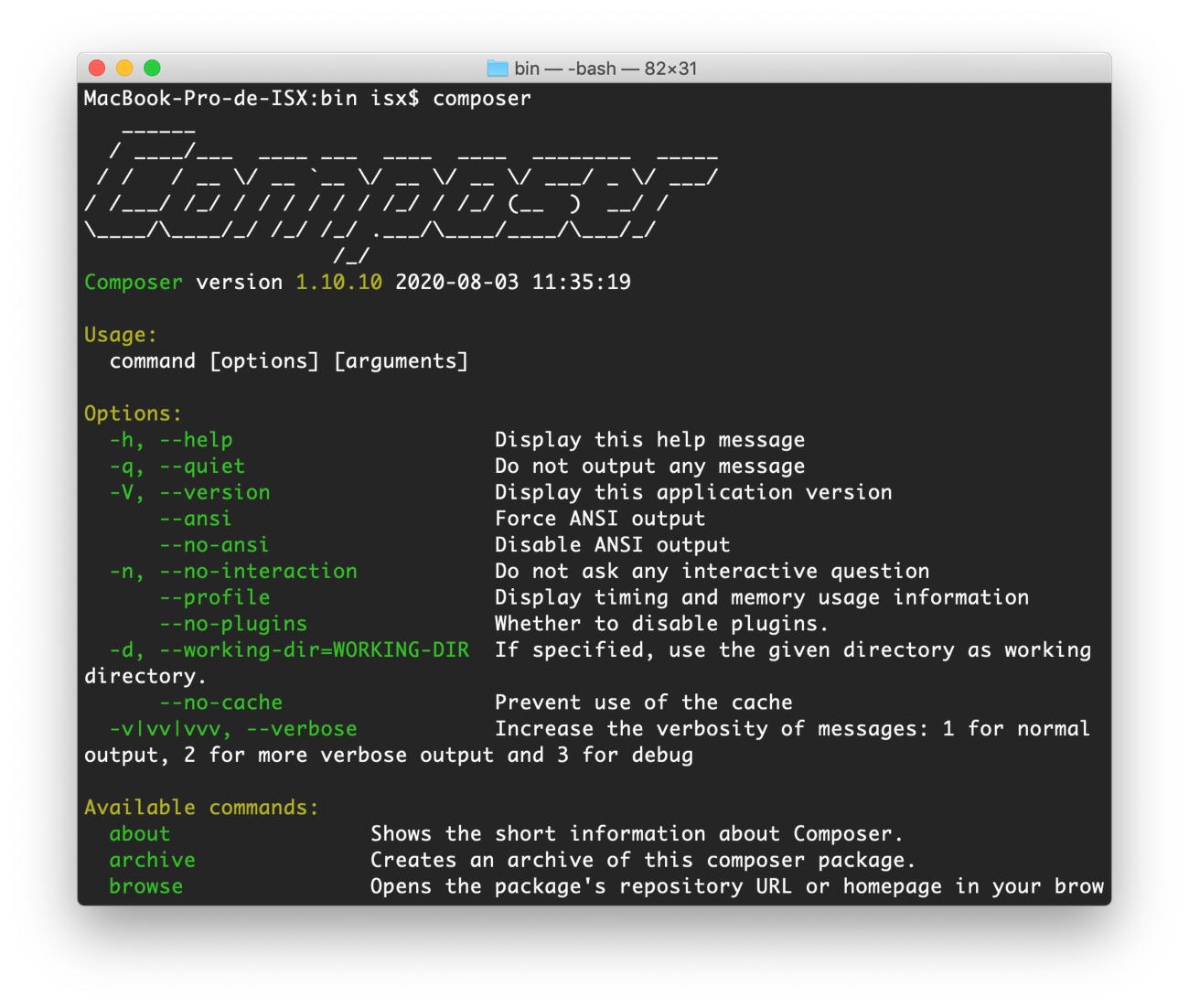
Dar permisos de ejecución

chmod +x composer

Ejecutar el comando composer

composer

y deberia salir la versión instalada, las opciones y comandos disponibles

****

**En Windows**

Es muy sencillo gracias a su instalador que puedes descargar desde aquí: <https://getcomposer.org/Composer-Setup.exe>. Instalará la última versión de Composer y configurará el PATH por tí para que puedas ejecutar composer en tu consola desde cualquier directorio.

# Laravel

**Requisitos**

* PHP >= 7.2.5
* BCMath PHP Extension
* Ctype PHP Extension
* Fileinfo PHP extension
* JSON PHP Extension
* Mbstring PHP Extension
* OpenSSL PHP Extension
* PDO PHP Extension
* Tokenizer PHP Extension
* XML PHP Extension

**Instalando Laravel**

**Via Composer Create-Project**

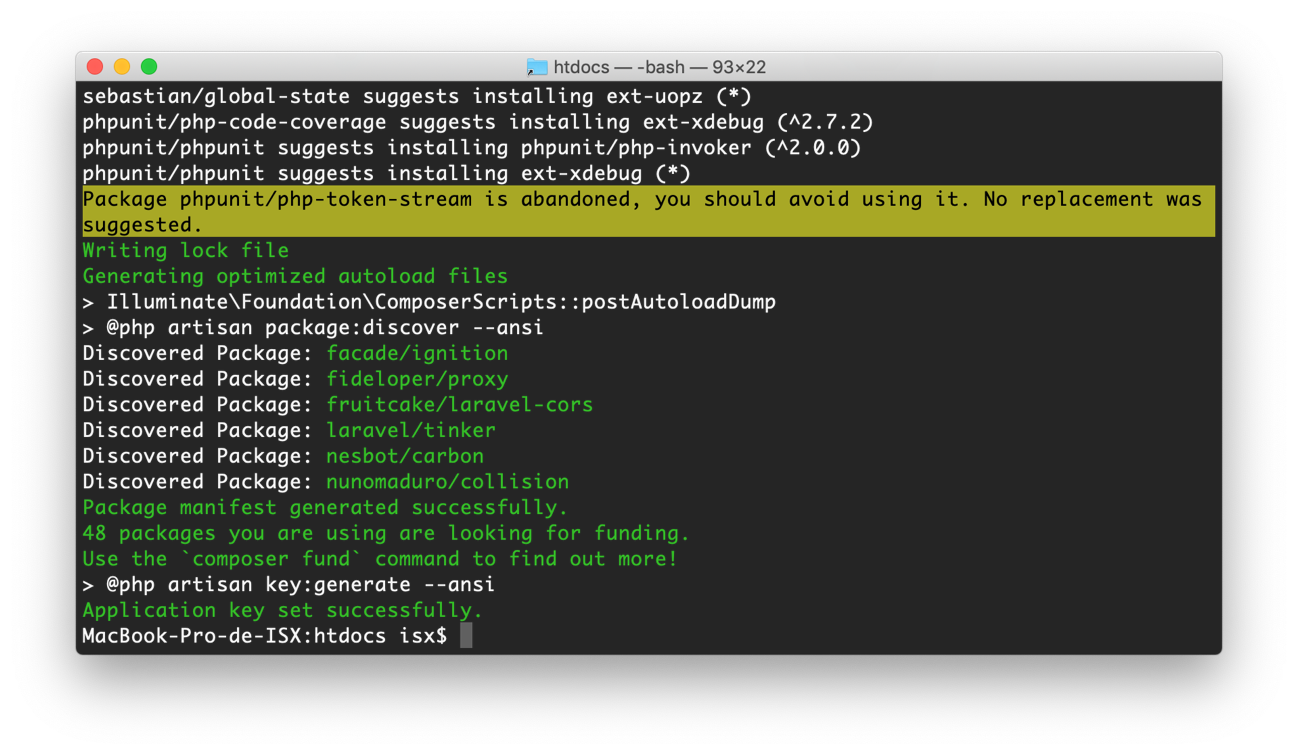
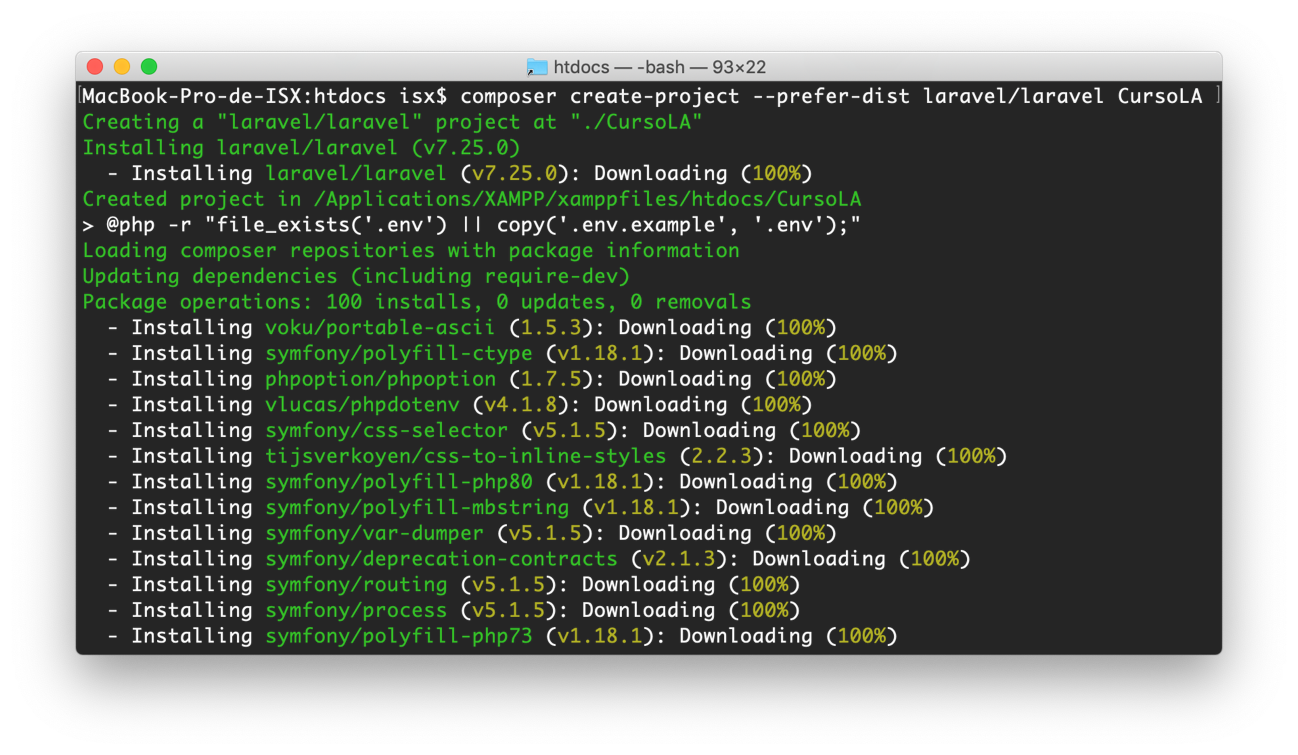
Ingresar al directorio donde se va a crear el proyecto, por ejemplo:

cd /Applications/XAMPP/htdocs/

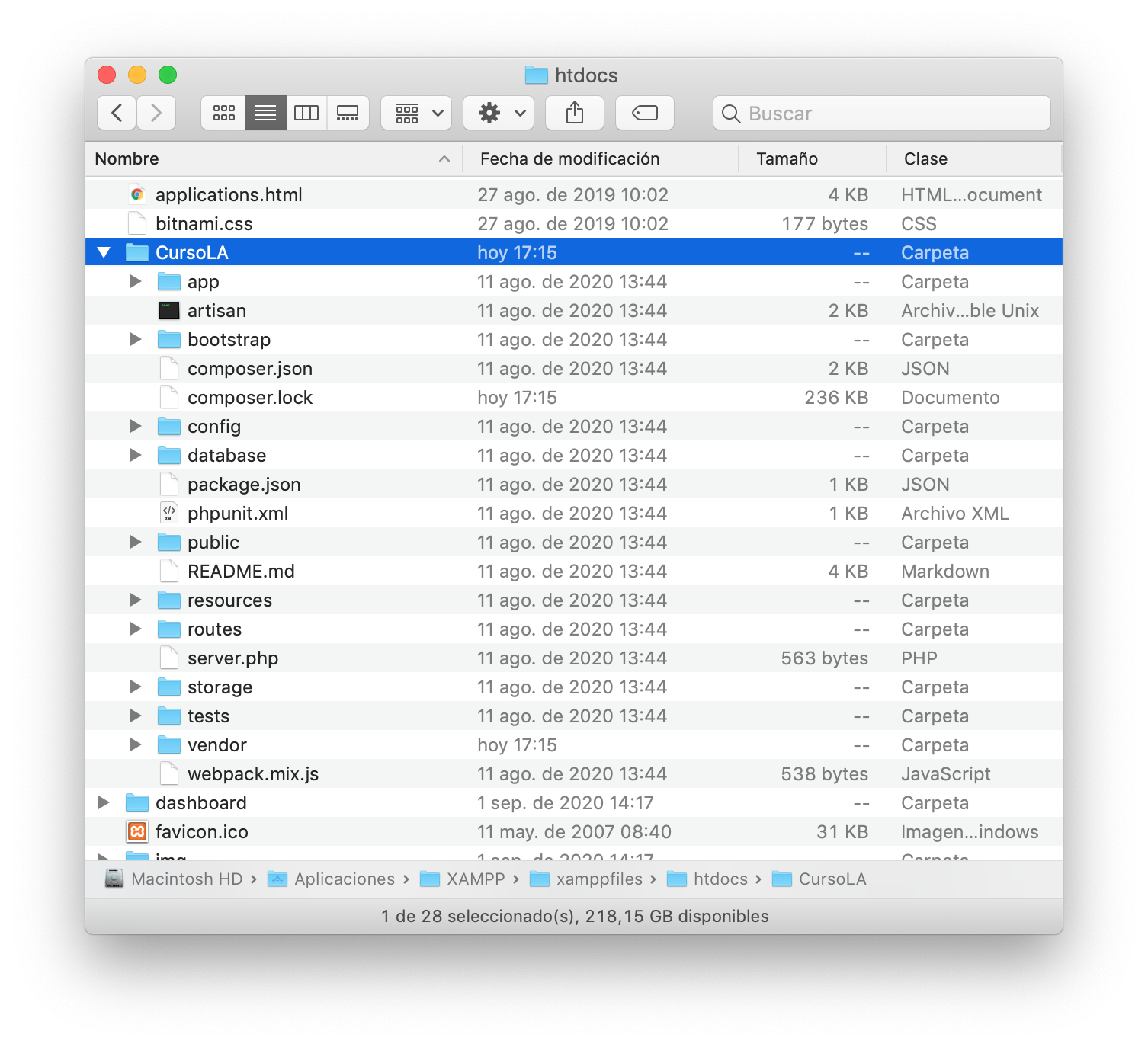
Instalar laravel emitiendo el siguiente comandoen el terminal:

composer create-project --prefer-dist laravel/laravel CursoLA

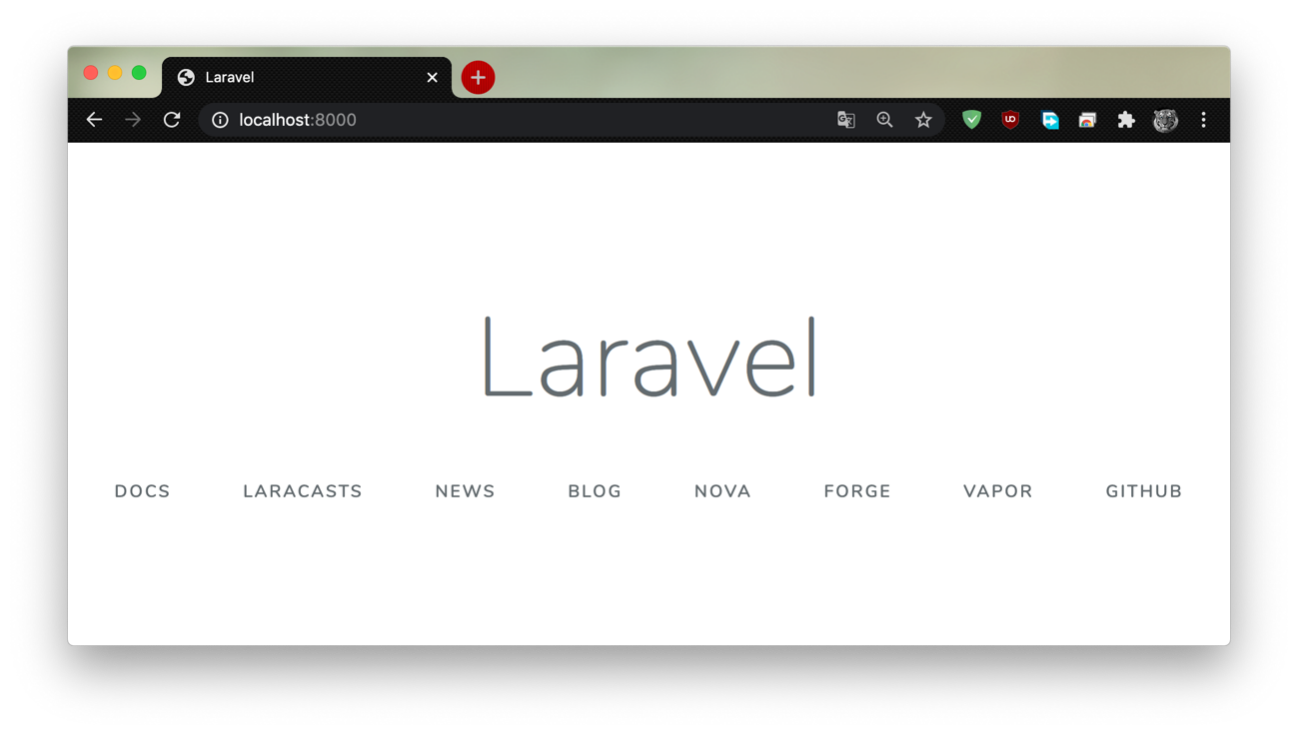
donde **CursoLA** es el nombre del proyecto.

****

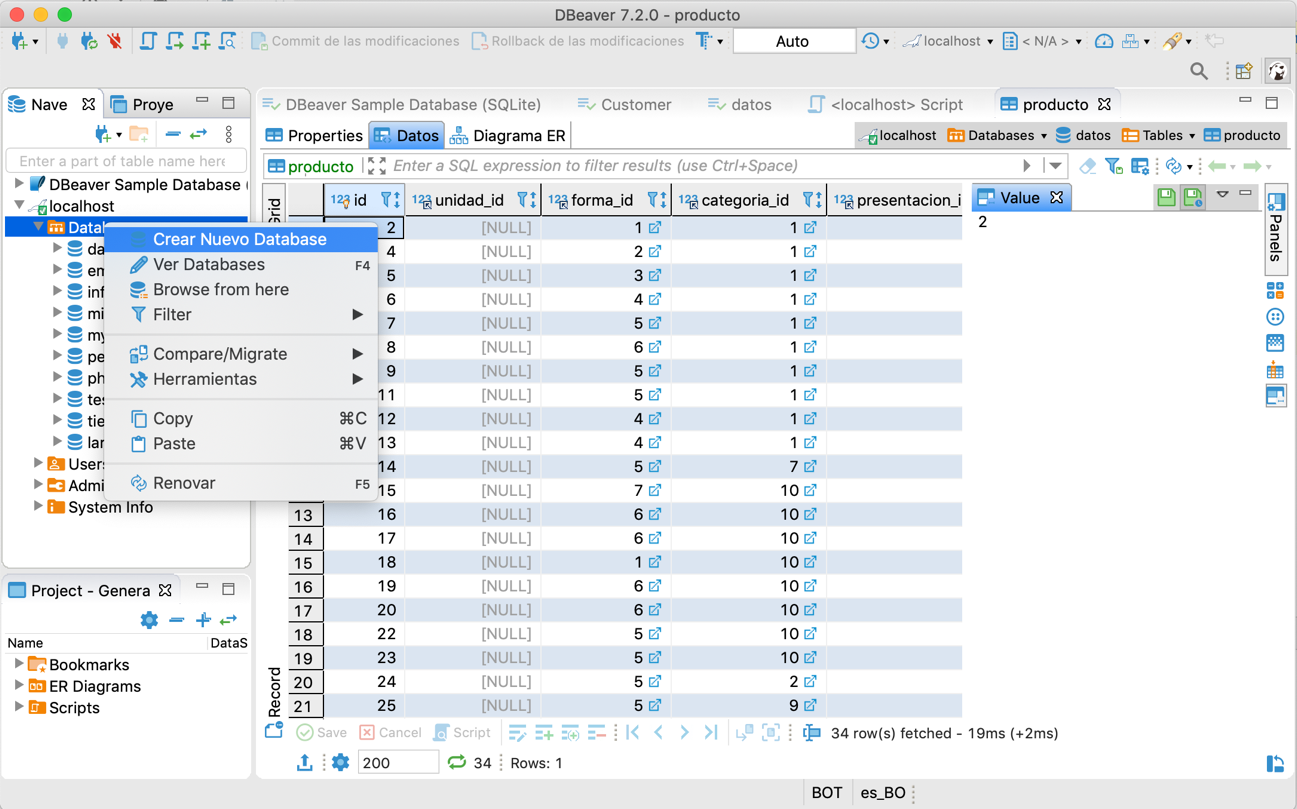
Una vez finalizado satisfactoriamente ya estará listo nuestro proyecto creado en el directorio donde le indicamos.

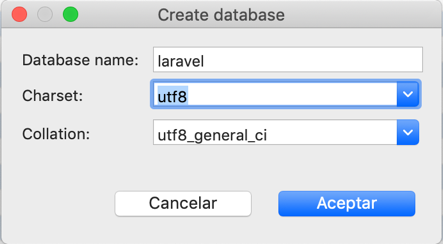
****

Se puede ver funcionando en <http://localhost:8000>

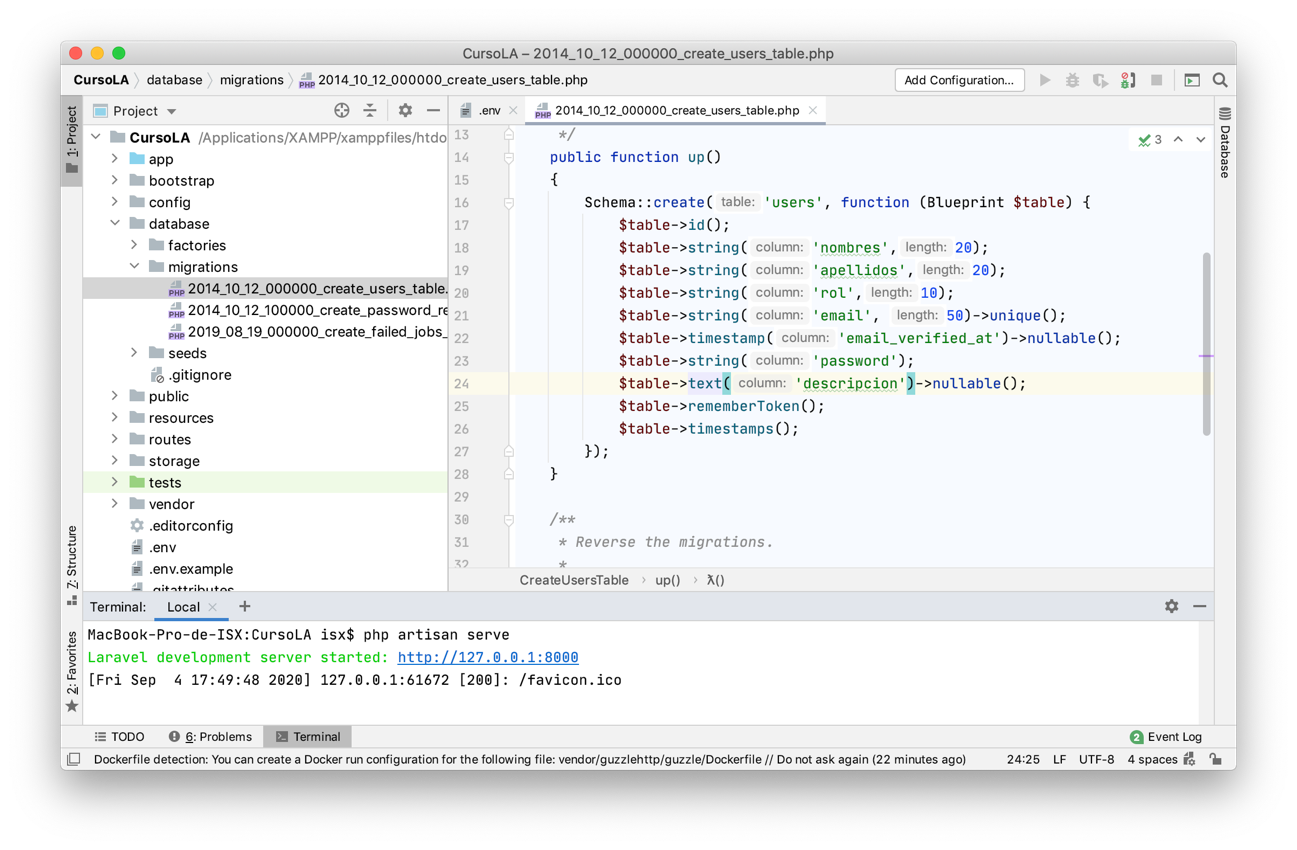


Utilizando el software **DBeaver**, vamos a crear la base de datos, y desde el código php en laravel creamos las tablas.



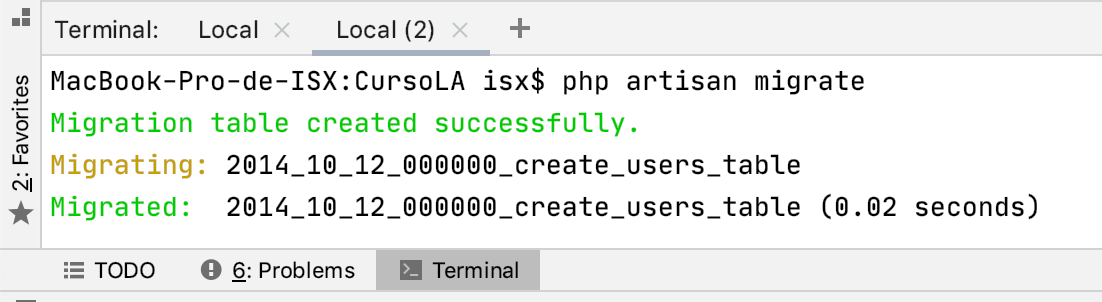
****

Modificar el archivo **2014\_10\_12\_000000\_create\_users\_table** ubicado en el directorio del proyecto CursoLA/database/migrations, de la siguiente manera:

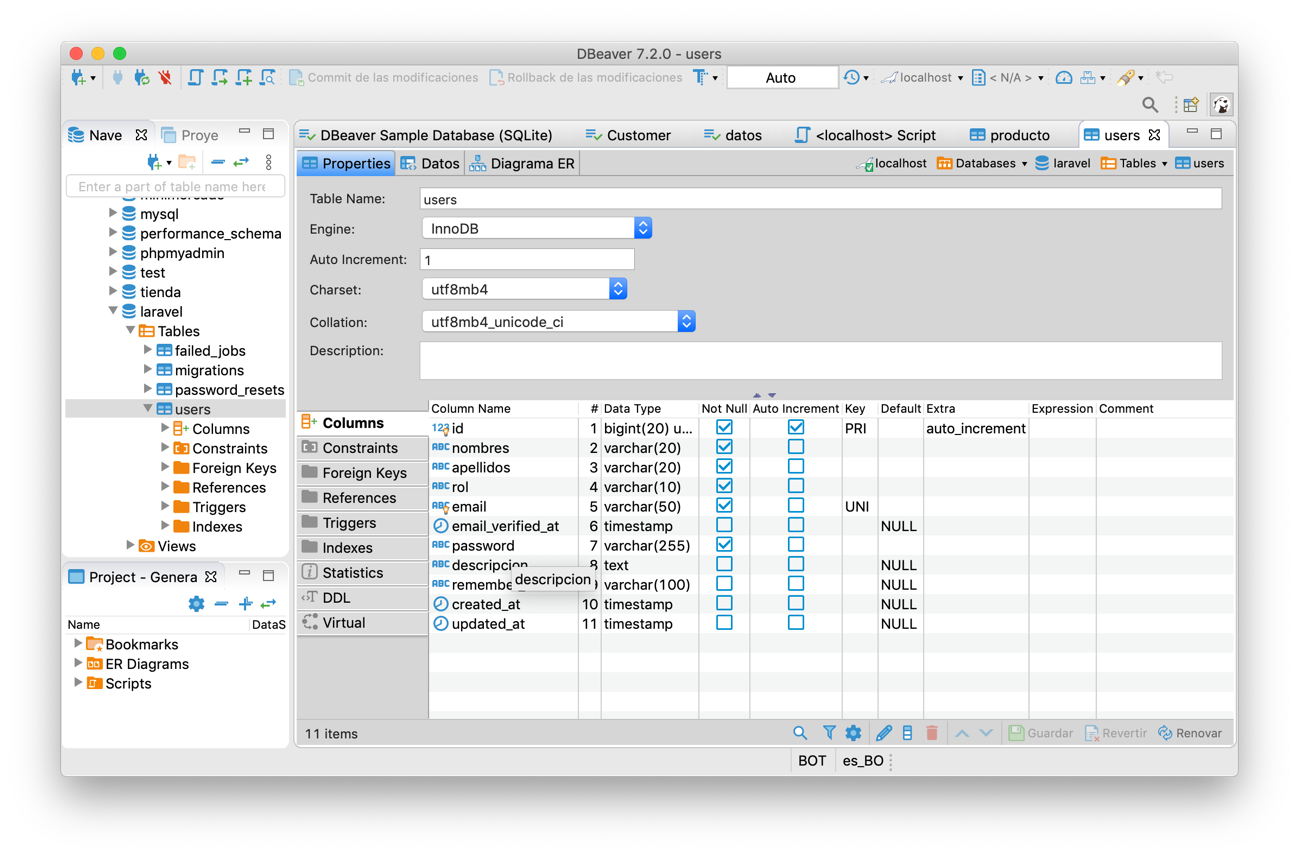


Para ejecutar abrimos otra terminal desde phpStorm y ejecutamos el siguiente comando:

php artisan migrate



Verificamos en la base de datos que efectivamente se creó la tabla con sus atributos.



# API REST

**API**: interfaz de comunicación entre dos software.

Para este intercambio de información normalmente se utilizan dos formatos **JSON** y **XML.**

**API REST:** interfaz entre dos sistemas que utiliza HTTP para intercambiar datos o realizar operaciones.

Un **cliente** **REST**: aplicación que permite hacer consultas a una **API REST**.

El uso mas inmediato, es para realizar pruebas de funcionamiento de una API que uno ha realizado, o bien, es contra la que se va a conectar.

**INSOMNIA**

Es un **cliente** **REST** multiplataforma. Con funcionalidades que facilitan enormemente el trabajo. Algunas funcionalidades son:

* **Personalización**.
* **Atajos de teclado**.
* **Importar y exportar**.
* **Configuración proxy**.
* Admite **complementos**.
* **Ayudas para la autentificación**.
* **Variables generales** o de entorno.

**Instalación de Insomnia**

**En Linux**

Se encuentra en los repositorios oficiales de Ubuntu. Sin embargo, la puedes instalar ejecutando las siguientes órdenes en un terminal:

echo "deb https://dl.bintray.com/getinsomnia/Insomnia /" | sudo tee -a /etc/apt/sources.list.d/insomnia.list

wget --quiet -O - https://insomnia.rest/keys/debian-public.key.asc | sudo apt-key add -

sudo apt update

sudo apt install insomnia

Alternativamente si se tiene snap en ubuntu:

sudo snap install insomnia

**En MacOSX**

**En Windows**

**PROBLEMAS DE USO**

Al parecer con versiones anteriores de **Insomnia** ha habido algunos problemas con dos asuntos en particular, y que se resuelven fácilmente.

* El directorio /tmp tiene que permitir ejecución
* Falta la dependencia libcurl.

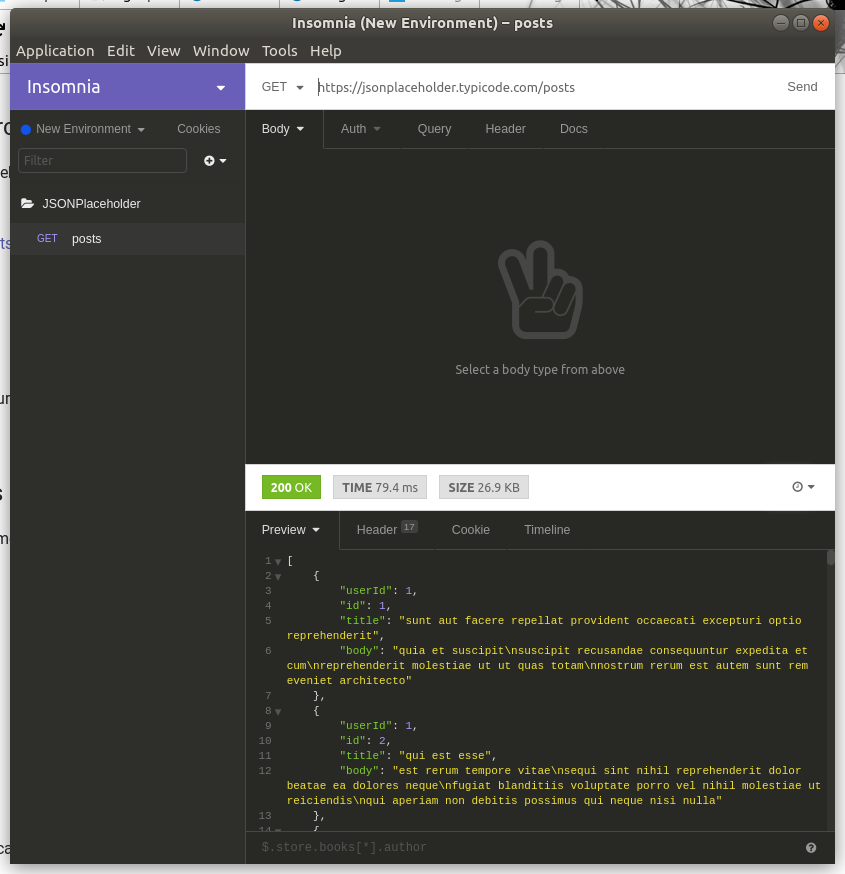
Dicho esto, en caso de que tengas algún problema, revisa estos dos conceptos.

**CREAR UNA CONSULTA CON INSOMNIA**

Como ejemplo, para que veas las posibilidades de un cliente REST como Insomnia, voy a utilizar el fantástico sitio [JSON Placeholder](https://jsonplaceholder.typicode.com/). Esta página, nos permite realizar consultas, para comprobar el correcto funcionamiento de nuestra aplicación.

Para tenerlo todo organizado, vamos a crear una carpeta que se llamará  JSONPlaceholder . Para ello pulsaremos el botón  +  y seleccionaremos la opción  New Folder .

A partir de aquí vamos a crear una consulta para obtener todos los  posts . Para ello solo tenemos que seleccionar la opción clic to add first request..., le damos un nombre,  posts  y seleccionamos el método  GET . En la parte superior escribimos la dirección y pulsamos  Send  y ya lo tenemos,



Ahora podríamos hacer el resto con los otros recursos que *JSON Placeholder*, pone a nuestra disposición,  comments ,  albums ,  photos ,  todos  y  users . Sin embargo, vamos a crear una variable para la carpeta  JSONPlaceholder  que nos ahorrará algo de trabajo, pero sobre todo mejorará la legibilidad de nuestro trabajo.

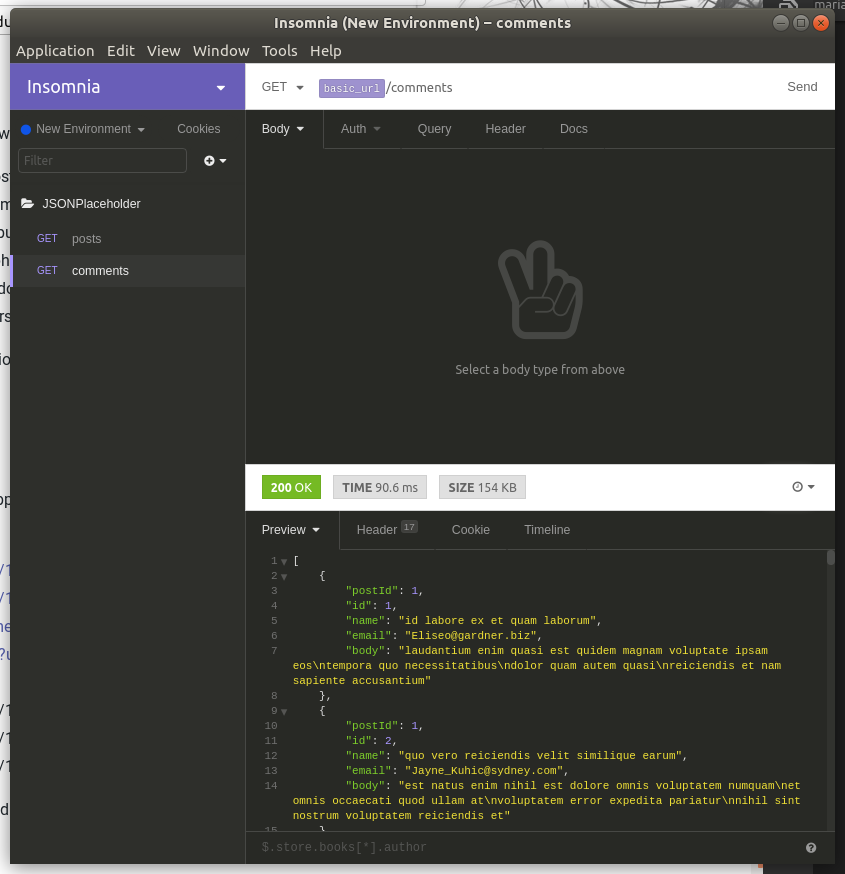
Si hacemos clic con el botón derecho del ratón sobre  JSONPlaceholder  y seleccionamos la entrada de menú  Environment , entraremos dentro de la configuración de nuestra carpeta. Allí podemos crear una variable global, que llamaremos  basic\_url , tal y como ves a continuación,

{

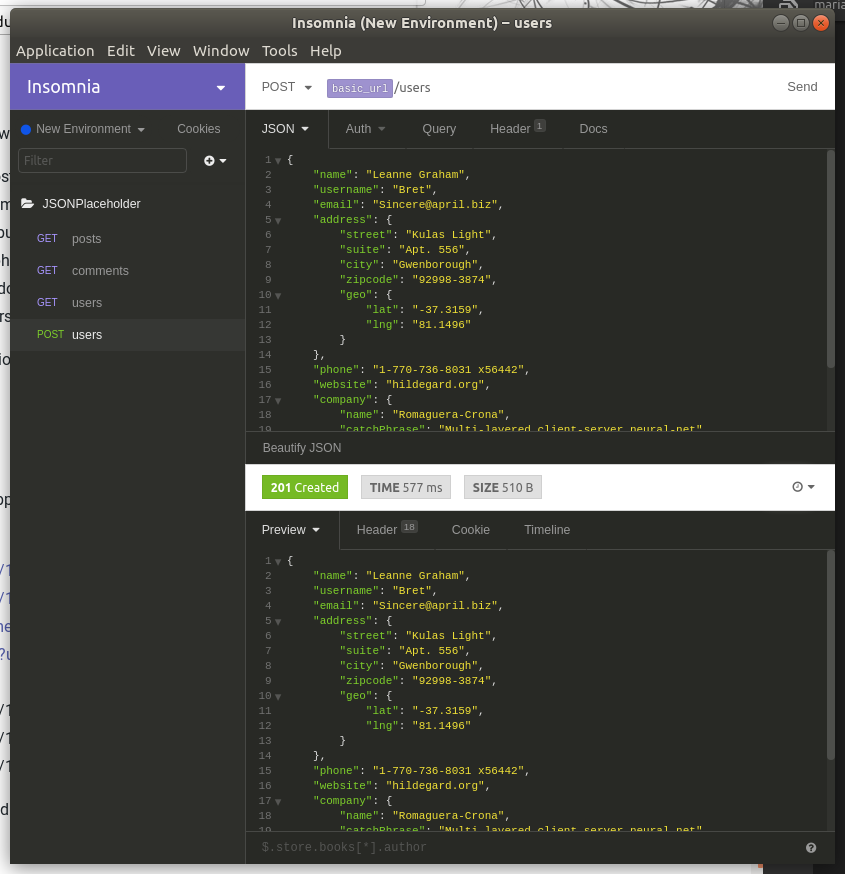
"basic\_url": "https://jsonplaceholder.typicode.com"

}

Esto permite utilizarlo posteriormente en nuestras consultas,



Si ahora quisiéramos hacer una consulta tipo POST, para añadir un nuevo elemento, por ejemplo un user, seleccionaremos el método POST. Además debemos añadir el contenido, que se corresponde con la primera pestaña debajo justo de la ruta, que seleccionaremos en este caso a JSON, tal y como ves en la siguiente captura de pantalla,



Instalacion de JWT

Json Web Token

Dado los credenciales genera un token

composer require tymon/jwt-auth

Config/app

Adicionamos en el providers

Tymon\JWTAuth\Providers\LaravelServiceProvider::class,

Adicionar en los alias

'JWTAuth' => Tymon\JWTAuth\Facades\JWTAuth::class,

'JWTFactory' => Tymon\JWTAuth\Facades\JWTFactory::class,

Ejecutar el comando

php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"

Generamos un token

php artisan jwt:secret

Generamos un controller

php artisan make:controller UserController

Generamos un middlerware

php artisan make:middleware JwtMiddleware

Registramos middleware app/http/kernel.php

'jwt.verify' => \App\Http\Middleware\JwtMiddleware::class,

Creamos rutas

Route::post('register', 'UserController@register');

Route::post('login', 'UserController@authenticate');

Agregamos control de rutas

Route::group(['middleware' => ['jwt.verify']], function() {

/\*AÑADE AQUI LAS RUTAS QUE QUIERAS PROTEGER CON JWT\*/

});

