Guía de ayuda para instalación de laravel y algunas librerías más utilizadas durante algunos proyectos.

Instalación básica según sitio oficial: <https://laravel.com/>, para la versión de este proyecto utilizaremos PHP 7.2.5.

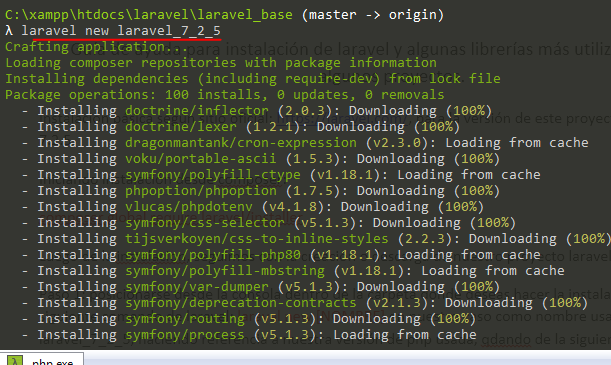
Iniciando instalación desde composer:

composer global require laravel/installer

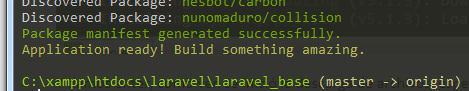
Luego de la instalacion, seguimos con la creación o descarga de nuestro proyecto laravel.

NOTA: Si solo descargo este proyecto, ejecute entonces composer install, los pasos a continuación explicados son los seguidos durante la creación de este proyecto

Paso 1: posicionarse desde la consola dentro de la carpeta donde deseas hacer la instalación y ejecutar el siguiente comando de laravel: laravel new [NOMBRE] en nuestro caso como nombre usaremos laravel\_7\_2\_5, haciendo referencia a nuestra versión de php usada, qdando de la siguiente manera:



Y damos enter, como ven se inició la descarga de archivos, esperamos a que termine.



Una vez terminado recibimos un mensaje parecido a este, y eso es todo ya tenemos la instalación base.

PASO 2: preparación para paquetes, archivos y librerías a utilizar.

Iniciamos con la creación del archivo .htaccess con este código tomado de la documentación oficial de laravel.

Options +FollowSymLinks -Indexes

RewriteEngine On

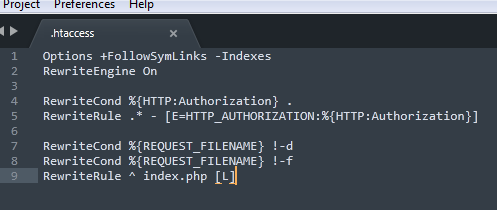
RewriteCond %{HTTP:Authorization} .

RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]

RewriteCond %{REQUEST\_FILENAME} !-d

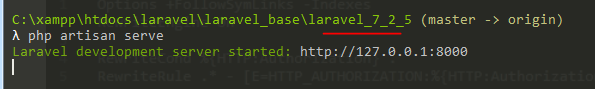
RewriteCond %{REQUEST\_FILENAME} !-f

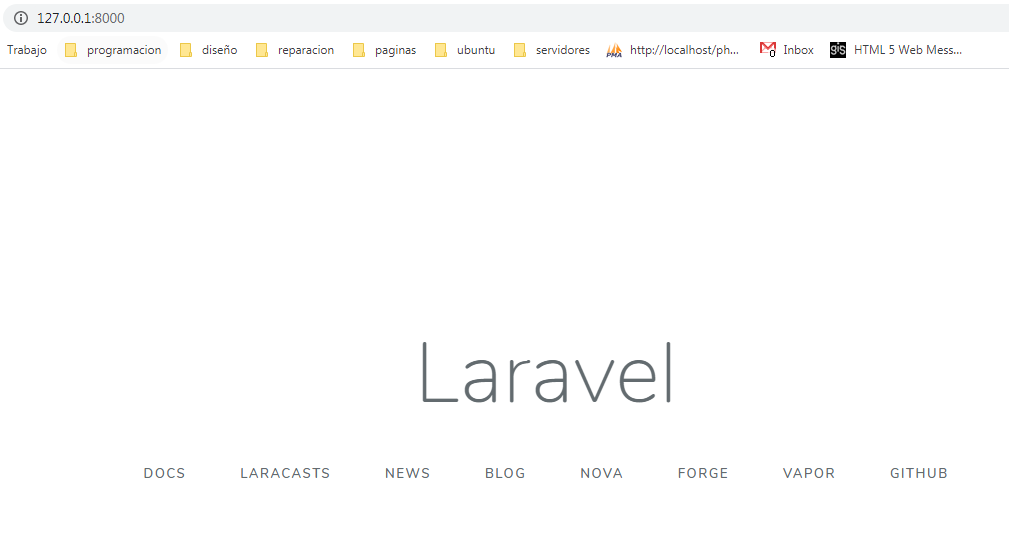
RewriteRule ^ index.php [L]



Y listo ya tenemos nuestro archivo htaccess.

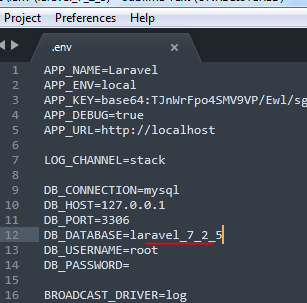
Probemos nuestra instalacion corriendo con: php artisan serve

Como ven se ejecutó desde dentro de nuestra carpeta, ahora corremos esa ruta en el navegador.



Como ven en la imagen ya está corriendo nuestro laravel, continuemos con sus paquetes o librerías adicionales.

Ahora conectamos la BD en nuestro archivo .env, para nuestro ejemplo hemos creado una BD con el mismo nombre del proyecto



Demás esta decir q se deben especificar user, pass y servidor

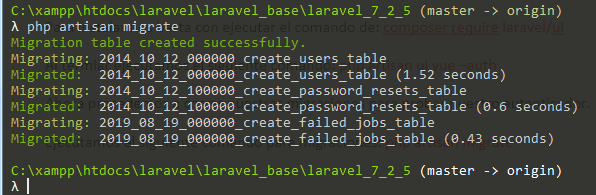
Instalando paquete de Autenticación según documentación: <https://laravel.com/docs/7.x/authentication#login-throttling>

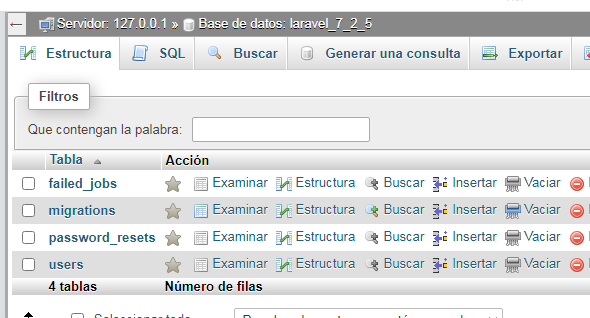
Para instalar nos basta con ejecutar el comando de: composer require laravel/ui

Al termiar ejecutamos el siguiente comando: php artisan ui vue –auth

Ahora procedemos a correr nuestras migraciones para probar nuestro autenticador.

Ejecutamos el siguiente comando para migraciones: php artisan migrate

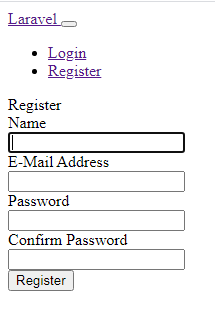
Podemos ver su ejecución, al terminar revisamos nuestro manejador de BD



Y podemos visualizar nuestras tablas

Ahora corremos nuestro proyecto nuevamente para revisar si ya tenemos nuestro autenticado corriendo 

Y si ejectivamente podemos verlo a nuestra derecha, registremos nuestro usuario de prueba.



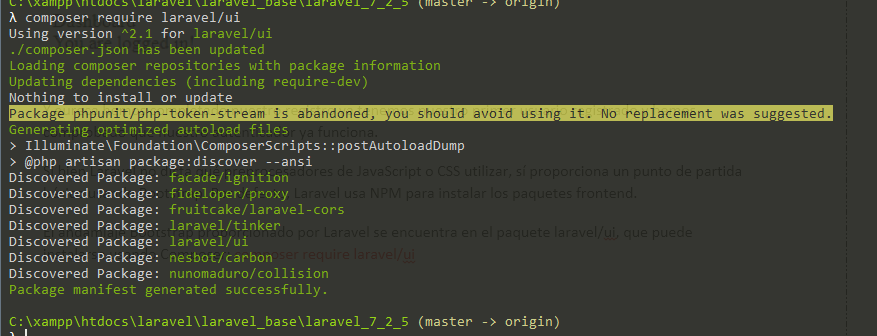
Obviamente sale sin estilos pues no hemos agregado ninguna librería css para darle un aspecto mas llamativo



Y eureca hemos completado nuestro registro ya tenemos nuestro primer usuario registrado y hemos comprobado que nuestro autenticador ya funciona.

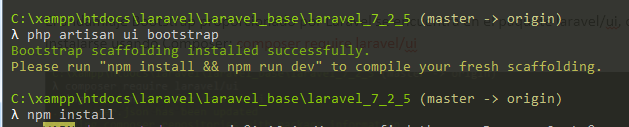
Si bien Laravel no dicta qué preprocesadores de JavaScript o CSS utilizar, sí proporciona un punto de partida básico usando Bootstrap. Por defecto, Laravel usa NPM para instalar los paquetes frontend.

El andamiaje Bootstrap proporcionado por Laravel se encuentra en el paquete laravel/ui, que puede instalarse usando Composer: composer require laravel/ui



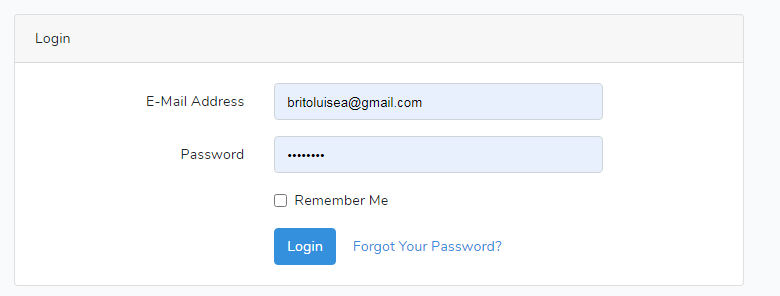
Una vez que se haya instalado el paquete laravel/ui, puedes instalar el andamiaje frontend utilizando el comando Artisan ui: php artisan ui Bootstrap

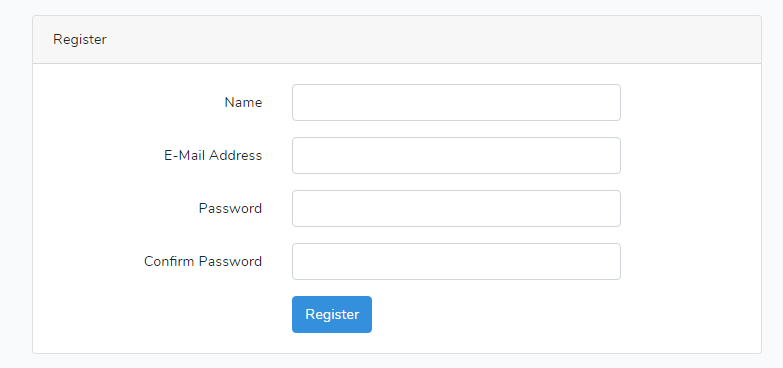
Instala las dependencias frontend de tu proyecto utilizando el administrador de paquetes de node (NPM): npm install

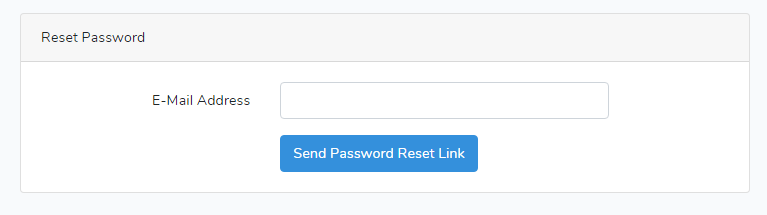


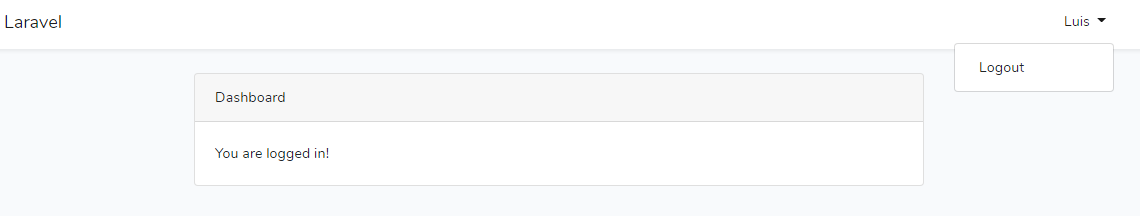
Una vez que las dependencias se han instalado usando npm install, puedes compilar tus archivos usando Laravel Mix. El comando npm run dev procesará las instrucciones en tu archivo webpack.mix.js

Y listo ya tenemos bootstrap en nuestro proyecto, corremos php artisan serve para verlo.









Y como ven ya tenemos nuestro diseño más atractivo que un principio