**5 LOGICA LOGIN USUARIO**

**5.1 INSTALACION DE DEPENDENCIAS**

Utilizaremos una tecnología llamada JWT(JSON WEB TOKEN), necesitaremos instalar una librería para poder utilizarla.

Esta librería nos permitirá generar tokens cuando el usuario se identifique en la web, y en ese token estará guardado la autentificación del usuario exceptuando la contraseña(x seguridad). Este token servirá para autentificarnos y esta viajando en cada una de las peticiones que hagamos en la web identificándonos.

El back que estamos programando, revisaremos en cada petición si ese token es correcto y por tanto ese usuario puede realizar peticiones o no.

Para instalar esta librería en Laravel modificaremos el archivo composer.json añadiendo la siguiente línea:



A continuación, no situaremos dentro de la carpeta del proyecto en la consola y ejecutaremos la siguiente instrucción: **composer update**

**5.2 SISTEMA AUTENTIFICACION**

Lo utilizaremos para dos cosas fundamentalmente:

1. Generar el token cuando se identifique un usuario y que este Token contenga toda la información de usuario
2. Devolver en base a un token toda la información de usuario.

Para organizar mejor el proyecto crearemos una carpeta nueva en app, la llamaremos Helpers y en ella guardaremos todos los archivos con utilidades que vayamos desarrollando, uno de ellos será el que cree los tokens como iremos viendo a continuación.

Crearemos la carpeta, ubicaremos el nameSpace, incorporaremos todos los archivos que vayamos a utilizar y lo mas importante para poder gastarla en Laravel, deberemos crear un provider ejecutando la siguiente instrucción en la consola dentro de la carpeta del proyecto: **php artisan make:provider jwtAuthServiceProvider**

Una vez tenemos el provider creado, deberemos registrar la ruta donde tenemos en este caso el jwt :

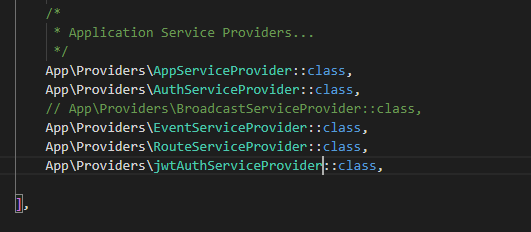
public function register()

    {

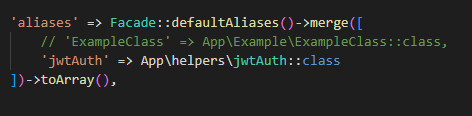
        require\_once app\_path().'/helpers/jwtAuth.php';

    }

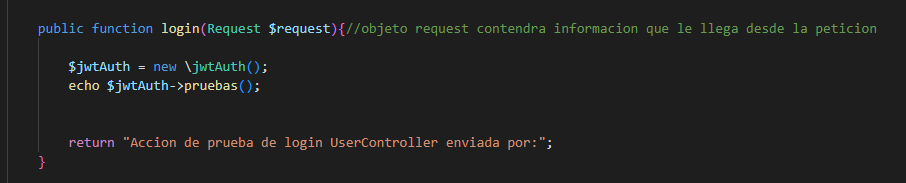
Y por último para poder gastarlo en el archivo config/app.php deberemos añadir la ruta del provider que acabamos de añadir:



Crearemos un alias:



Por último, vamos a comprobar que todo ha ido correctamente:



Lo que veremos es que se ejecuta el método pruebas y luego el return dado que hacemos primero el echo. Dado que hemos definido un alias, no será necesario incorporar la clase, es decir esto sobra:

//use App\helpers\jwtAuth; --> no sera necesario dado que tenemos el alias declarado

**5.3 LÓGICA METODO SIGNUP**