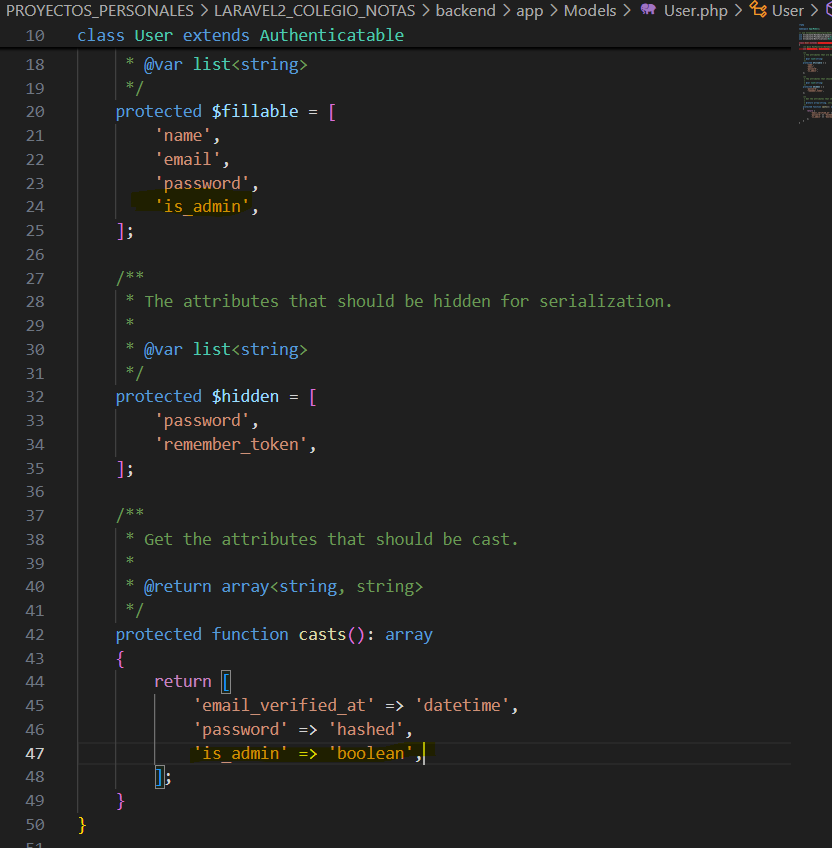
*composer create-project laravel/laravel toDoApp*

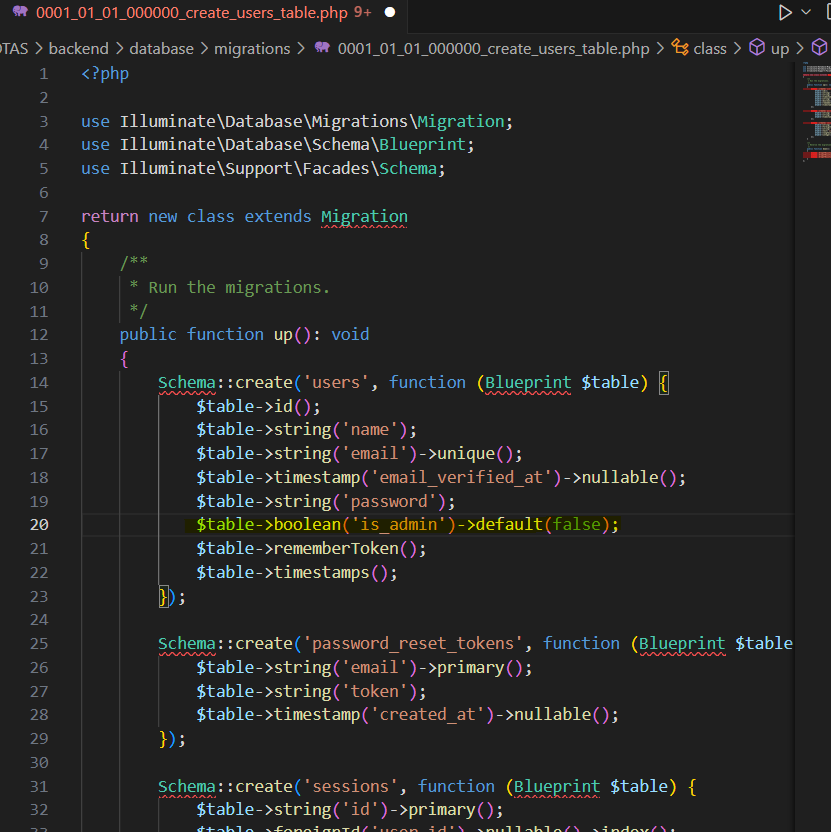
Crear base de datos en phpmyadmin

Modificar .env para que conecte a la base de datos

Modificar Models/User.php



Modificar Database/migrations/usermigrations



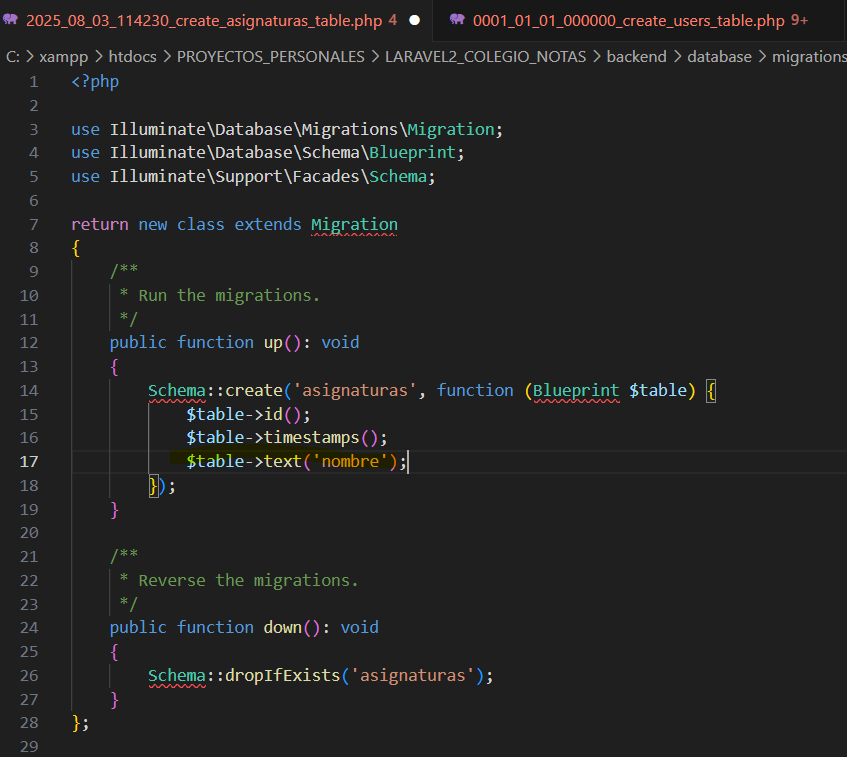
php artisan migrate

Relación muchos a muchos entre usuario y asignatura, por lo que necesitamos tabla intermedia

Esto crea el modelo y la migración de asignaturas

php artisan make:model Asignatura –m

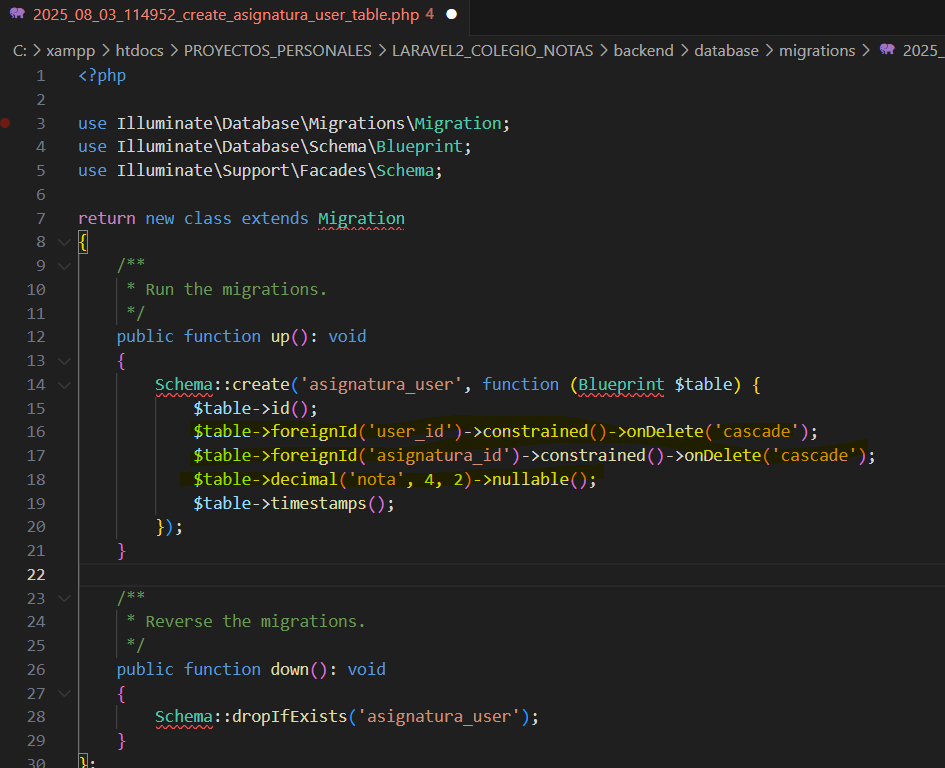
database/migrations/asignaturas



Y esto crea la tabla intermedia, que se llama asi por convención, y solamente se crea la migración, no va a tener modelo ni nada

php artisan make:migration create\_asignatura\_user\_table --create=asignatura\_user

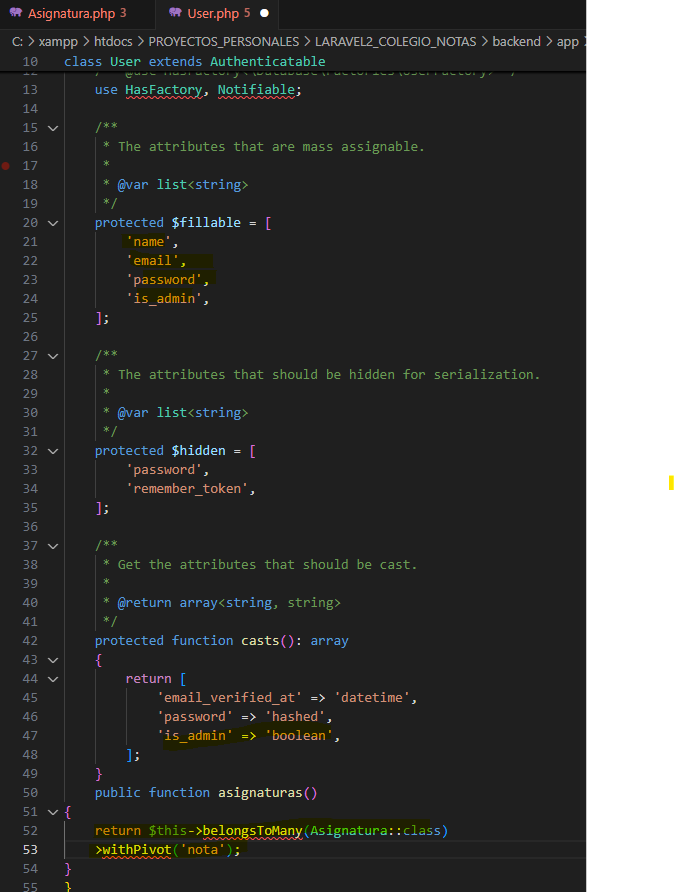
database/migrations/asignatura\_user\_table



(nota decimal, max 4 numeros, y detrás de la , 2 )

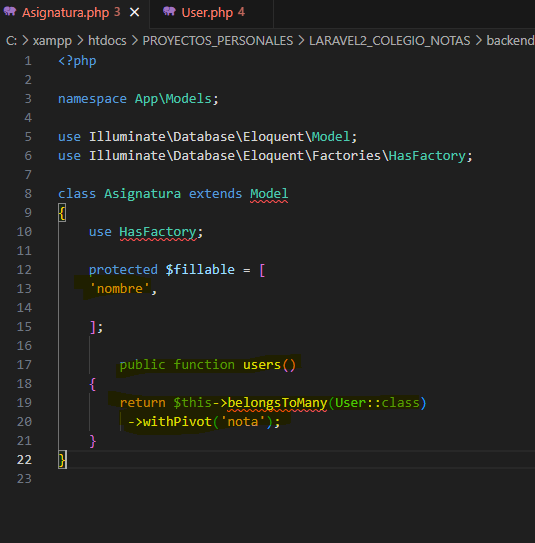
La nota hay que ponerla aquí, porque en usuario no se puede, y en asignatura tampoco (cada usuario tendra una nota distinta)

App/Models/User.php



Aqui indico que cuando pida el user, me devuelva, las asignaturas a las que pertenece, y el valor nota de la tabla pivot

App/Models/Asignatura.php



Ponemos que cuando traigamos asignatura, nos traiga con los users a los que pertenece, y además con el valor nota de la pivot table

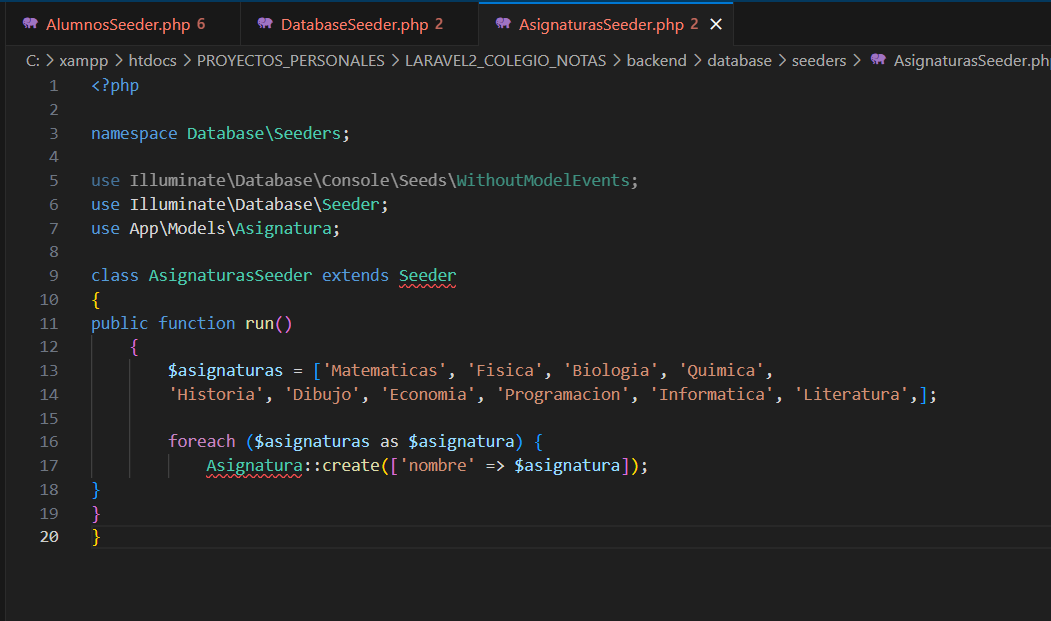
Php artisan migrate

Ahora seeders, para crear 10 asignaturas, 20 alumnos, y que cada alumno tenga 8 asignaturas random. Por último crearemos 1 usuario administrador, que no tendrá ninguna asignatura

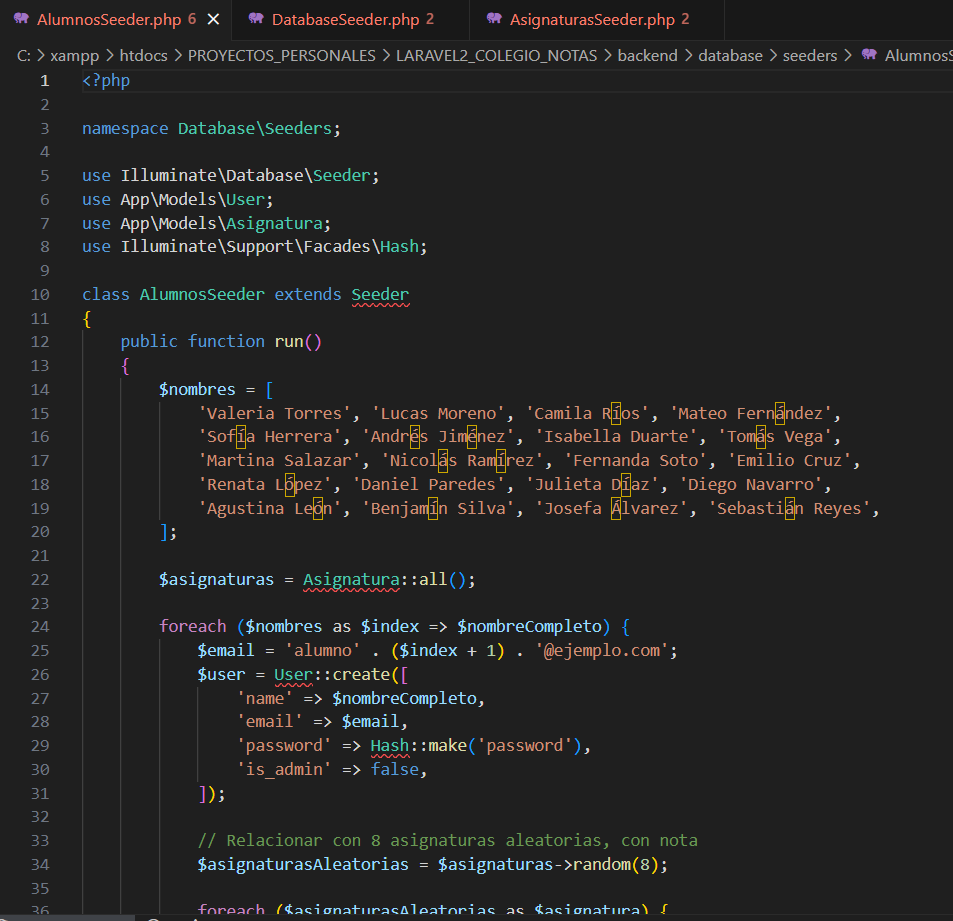
php artisan make:seeder AsignaturasSeeder

php artisan make:seeder AlumnosSeeder

database/seeders/AsignaturasSeeder.php

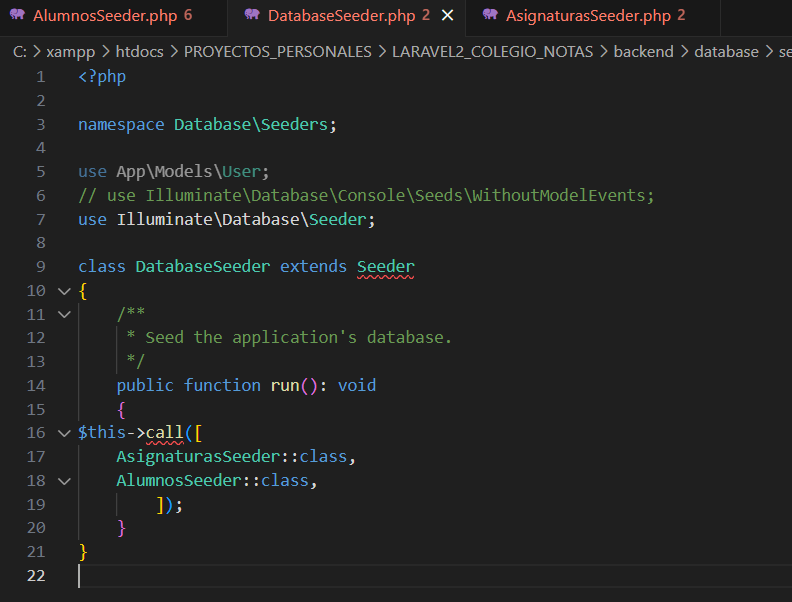


Database/seeders/AlumnosSeeder.php





Database/seeders/DatabaseSeeder.php



php artisan make:controller UserController –resource

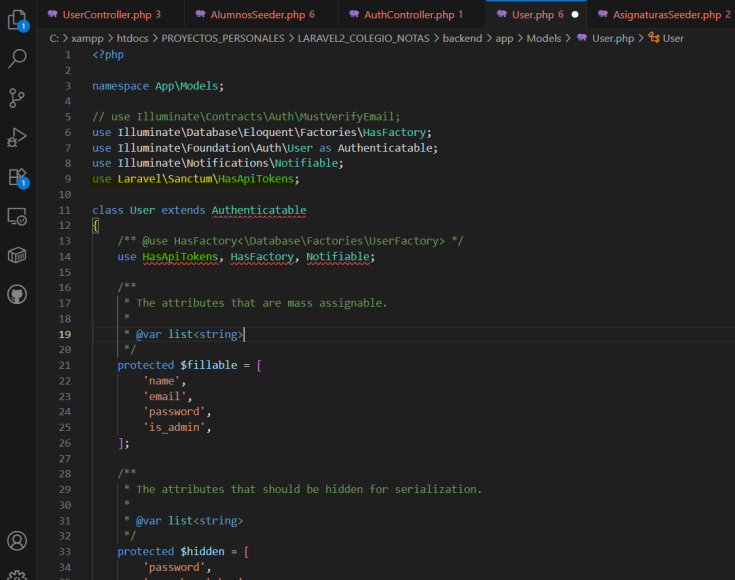
php artisan make:controller AuthController

composer require laravel/sanctum

php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"

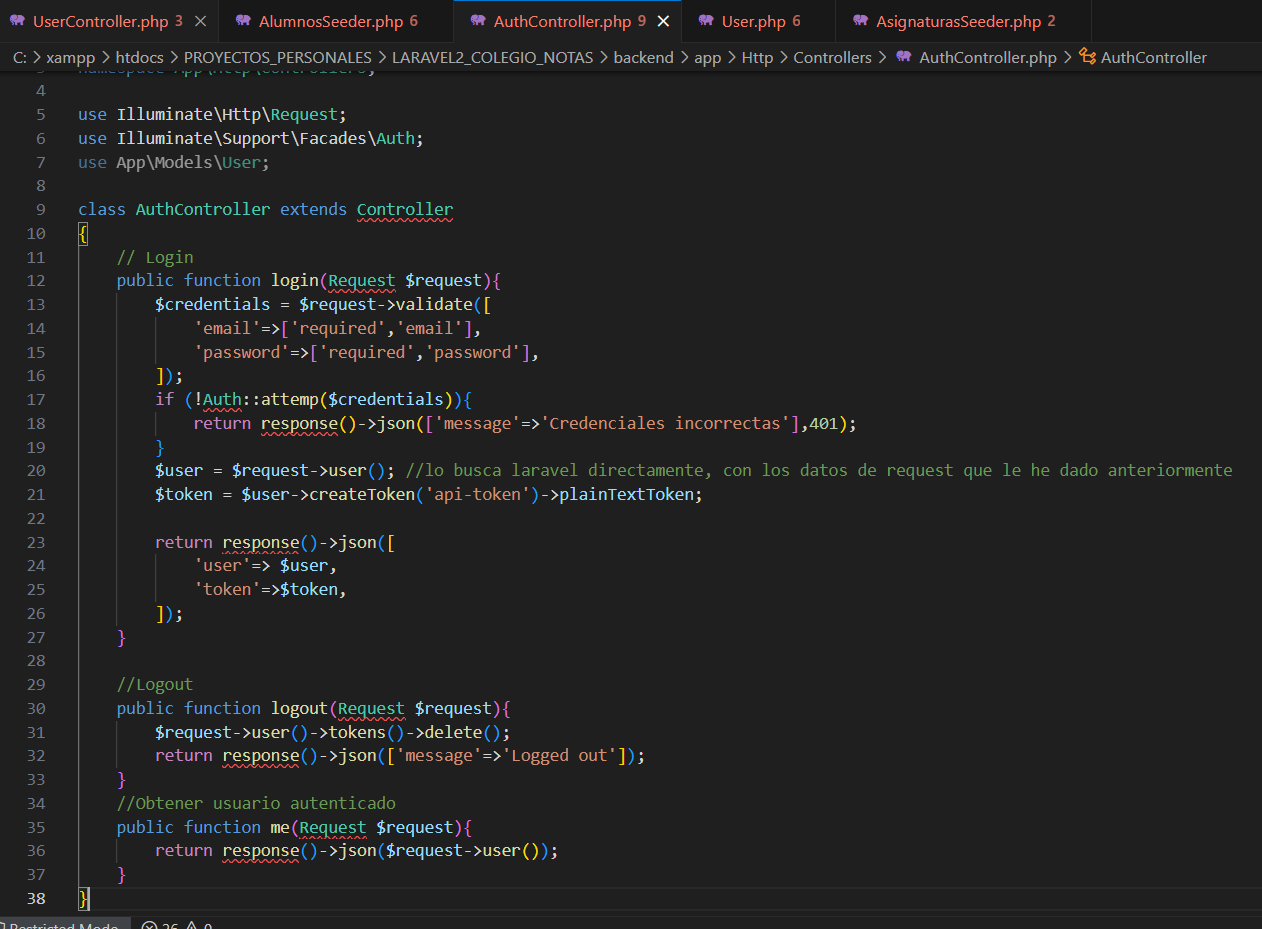
php artisan migrate

hago este cambio en el modelo de users

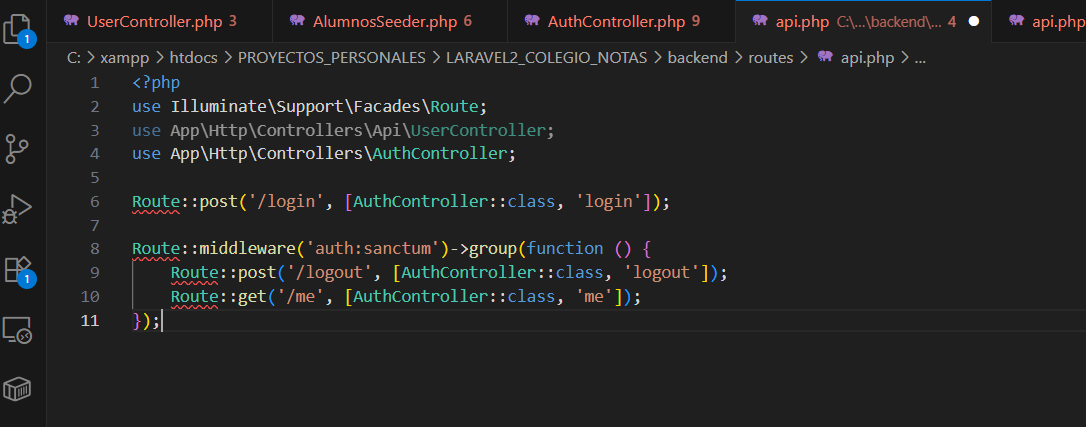


Y con esto ya podre proteger las rutas con sanctum, pero primero vamos a definir los métodos de login, logout en authcontroller

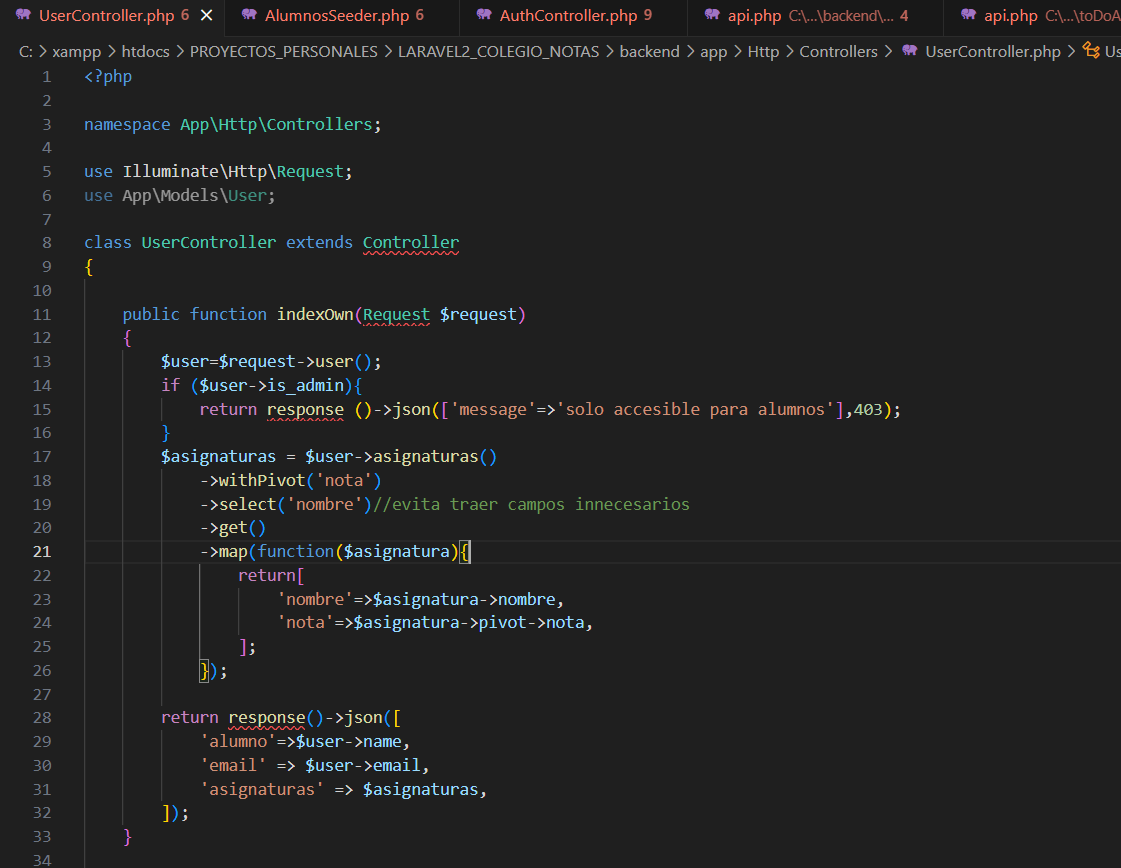
Authcontroller.php



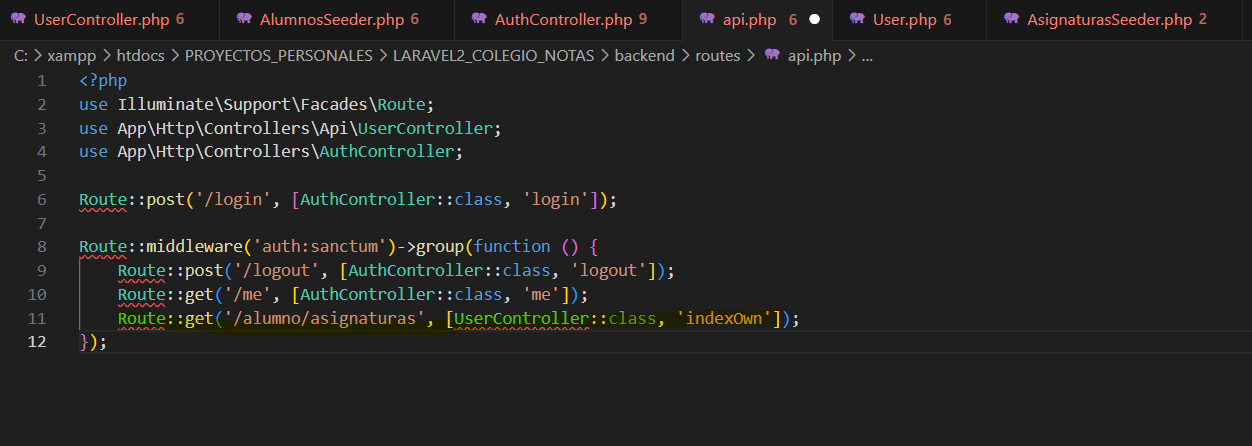
Y ahora en routes/api.php (lo acabo de crear, ponemos)



Ahora en Usercontroller.php, ponemos el método para que un alumno vea todas sus asignaturas

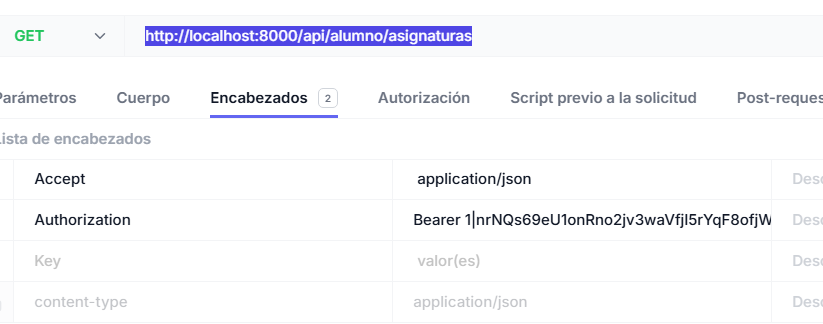


Y ponemos la ruta a este método en routes/api.php



Ahora, ya puedo hacer la prueba de pasar a login un email y su pasword, y que me devuelva el usuario y el token

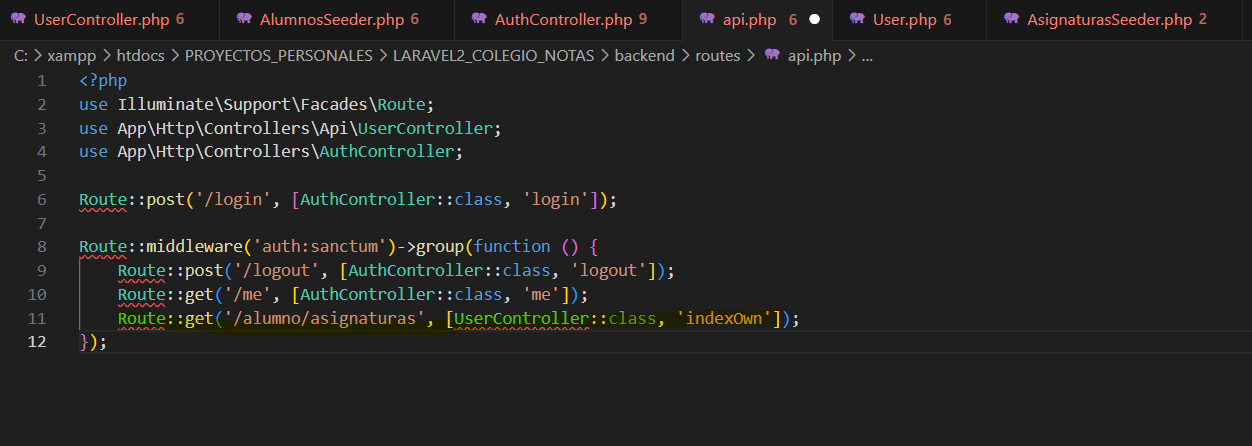
Y con este token hacer



Sin nada mas, sin cuerpo de la petición, solo con el token, porque esto va a pasar por el middleware de sanctum, y sanctum va a buscar el usuario que tiene este token, y va a pasar a el método indexOwn ese usuario como request, para que devuelva los datos del usuario que tiene la sesión iniciada, y no de otro usuario

Es decir, yo no elijo el usuario, y lo paso como un request al método indexOwn, lo coge sancutm, del token con el que tengo iniciada la sesión, que esta asociado a un usuario de cuando hice el login

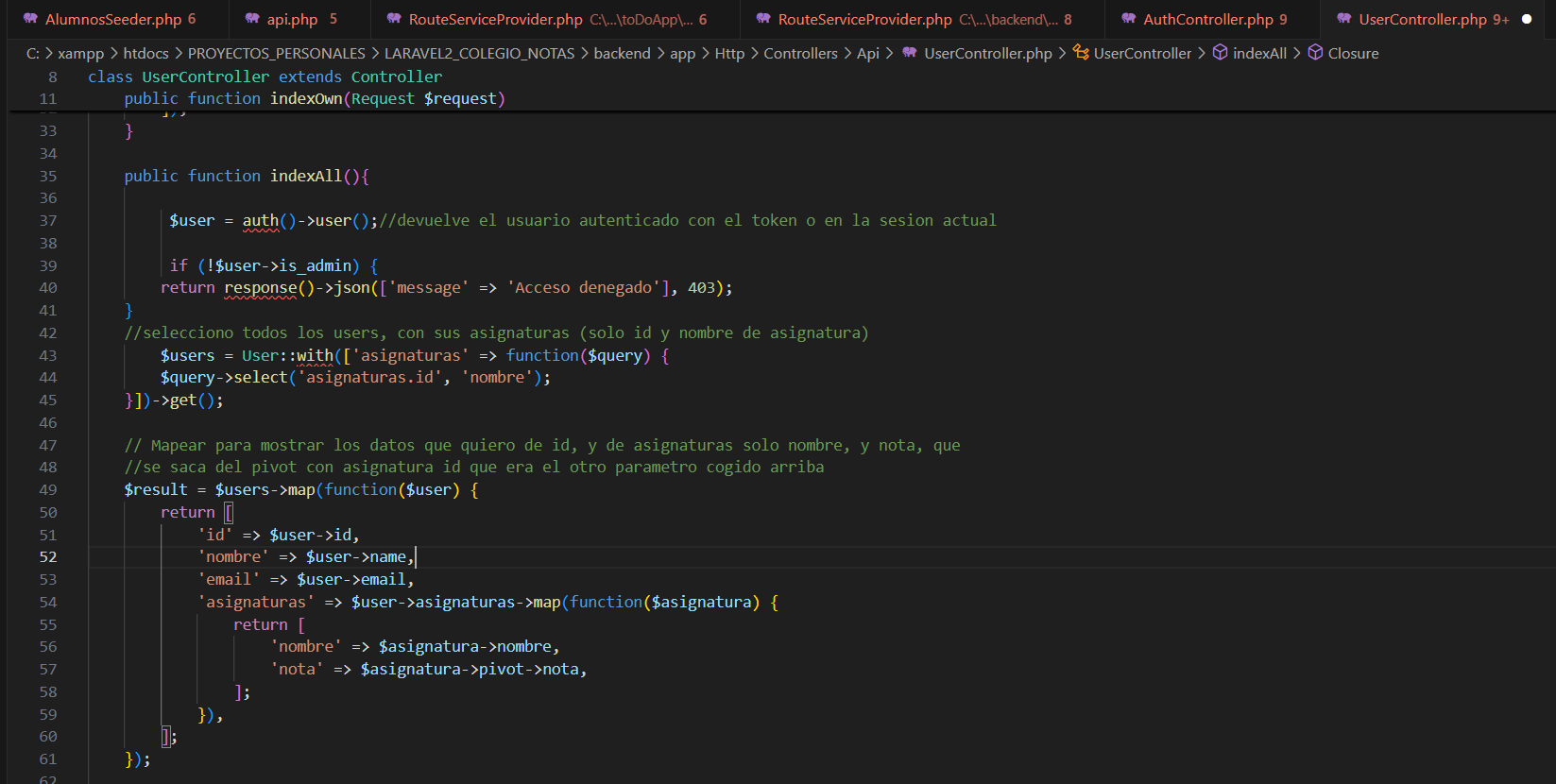
Asi un usuario no puede mirar lo de otro usuario



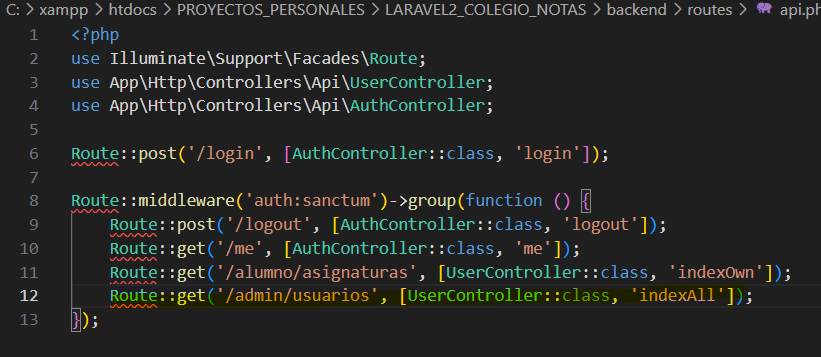
Y devuelve esto



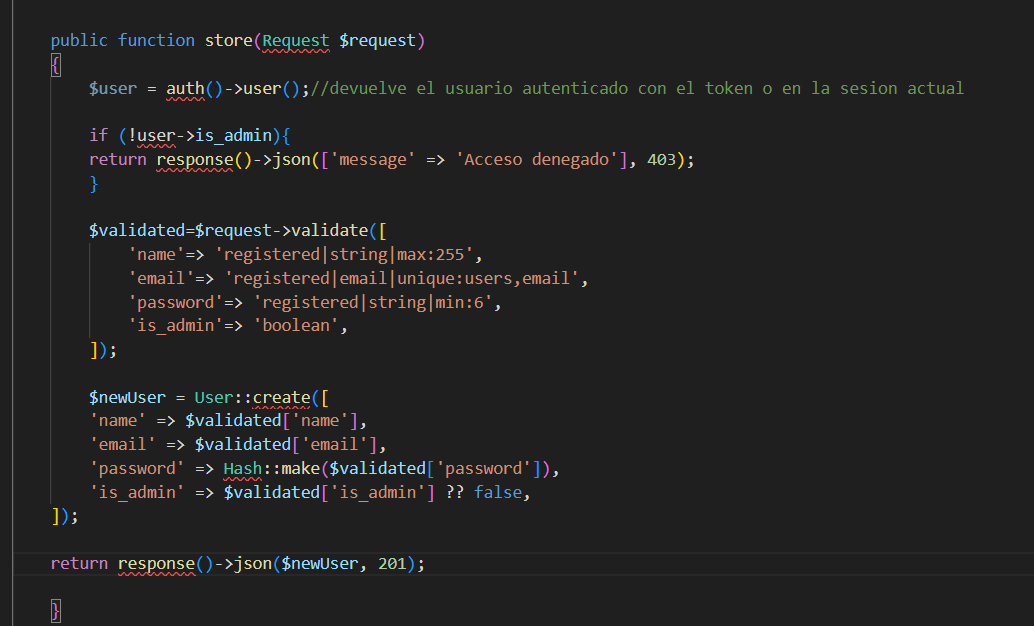
Ahora para los métodos de administrador (de momento mostrar todos los usuarios, hago en UserController esto

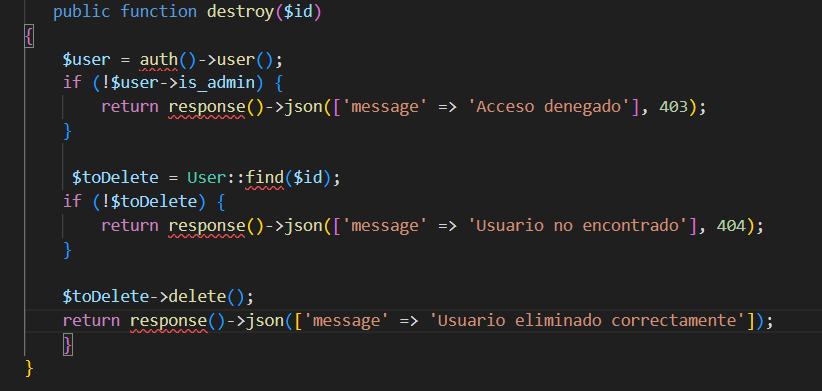


Y en routes/api.php hago

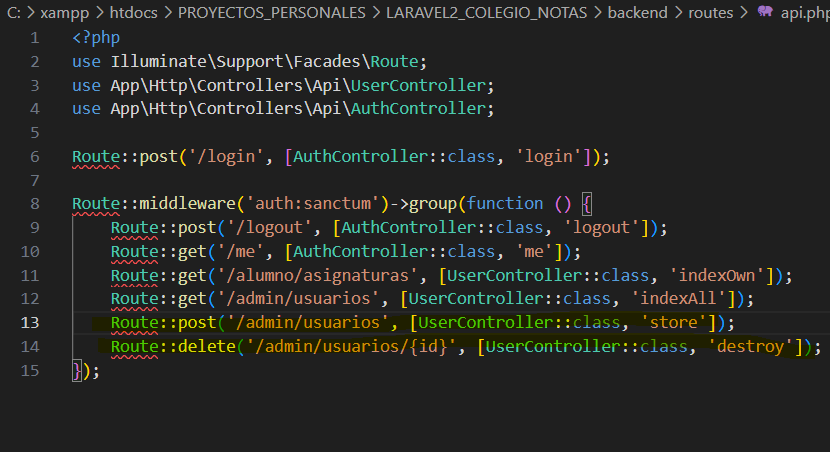


Ahora para almacenar nuevos usuarios (método store) y para borrarlos (método destroy), en UserController hacemos



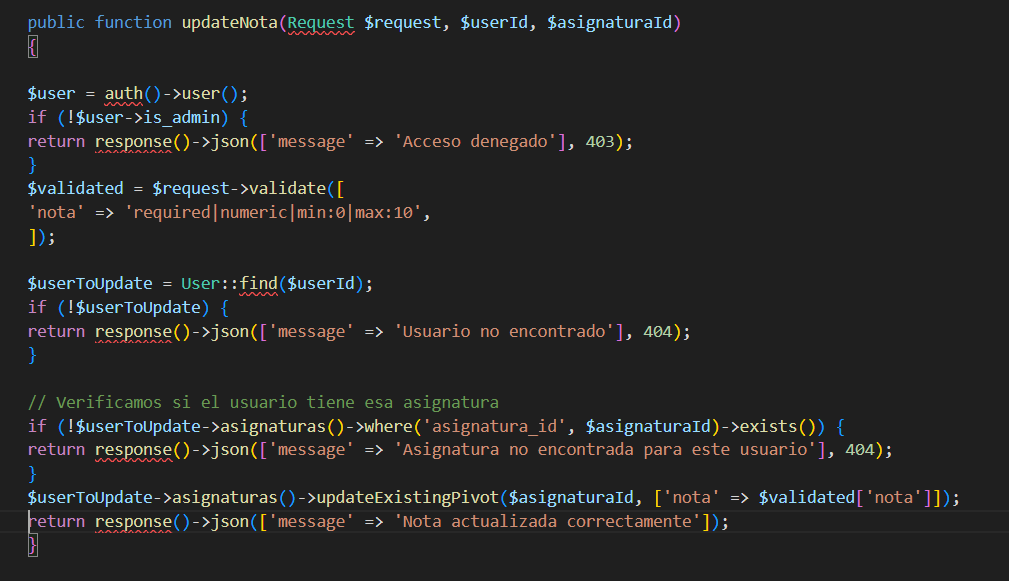


Ahora ponemos las rutas para crear y eliminar (en routes/api.php)

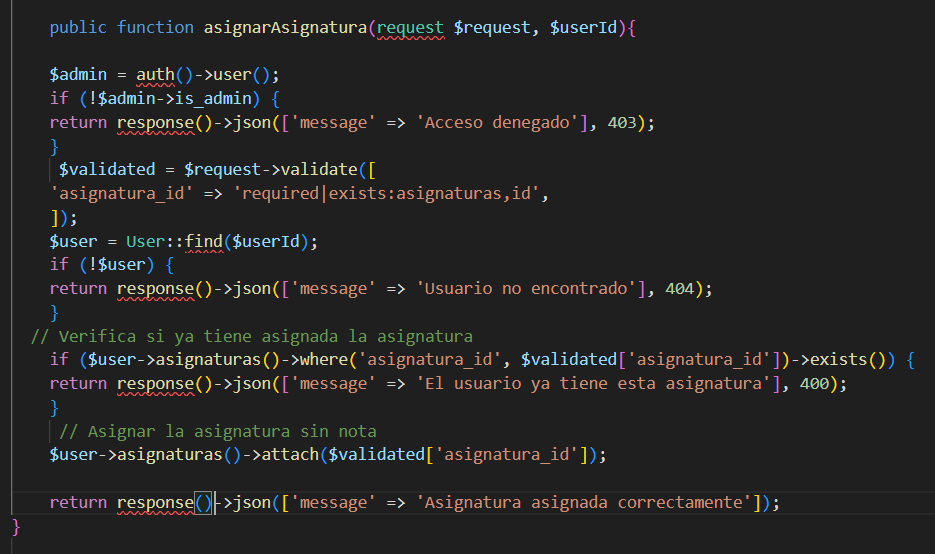


Y por ultimo en UserController los métodos para cambiar la nota, o añadir o quitar asignaturas

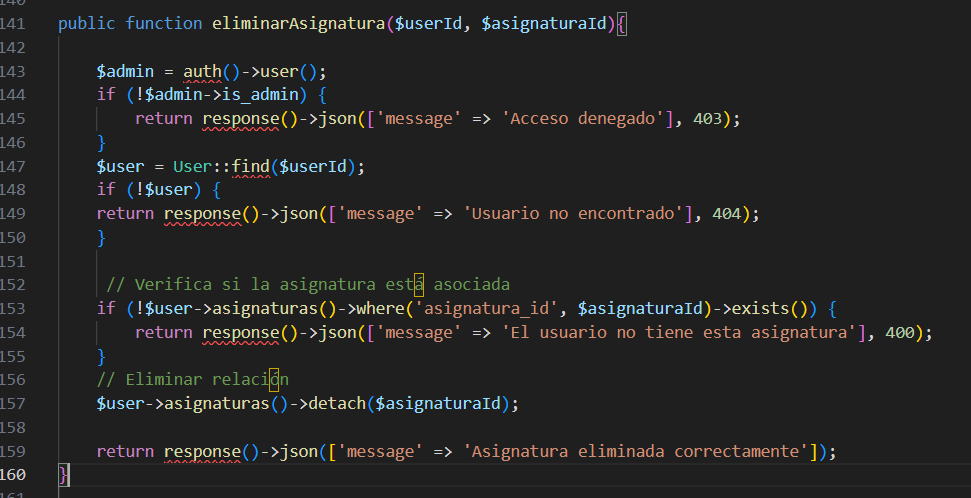
Cambiar la nota



Asignar nueva asignatura



Y eliminar asignatura



Y pongo todo esto en las rutas (fijarme que en la ruta se pone como parámetro userId, asignaturaId, todo lo que se pasa, menos el Request, que se pasa casi siempre, pero no sale aquí, ver en el front, como llamo a la api y como le paso el request, (supongo que como son post, en el cuerpo de la petición, pero ver si hay alguna diferencia en como se pasa el request y como se pasa por ejemplo userIde o asignaturaID)

