**DOCUMENTACIÓN PROYECTO -PABLO REBOLLO**

**Laravel 11.**

**PHP 8.3.12**

**BOOTSTRAP**

**LIBRERIAS UTILIZADAS**

Spatie Laravel Permission – Roles y permisos Laraavel – Al requerir de roles y permisos, he utilizado esta librería que es, en mi opinión la más estable para Laravel y la que mejor funciona en este tipo de sistemas.

Laravel Sanctum: Para protección mediante autenticación-token de las apis. Librería que viene ya por defecto en Laravel 11 y por ello la utilice y para demostrar el funcionamiento de la protección de rutas.

Laravel Carbon: Viene también directamente integrado con Laravel 11 y es una librería muy completa para dar formato y tratar todo tipo de fechas.

------------------------------------------------

Chart JS es la librería utilizada para las gráficas en Javascript. Utilicé esta porque era la que más conocía y es fácil de utilizar para los requerimientos solicitados.

**A TENER EN CUENTA:**

- Establecer en el .env las credenciales de la base de datos.

- En el .env queda definido cuantos documentos crea automáticamente el seeder al lanzarlo. Por defecto está puesto en 60.

- Los documentos se generan aleatoriamente entre 2 opciones de muestra (prueba1.pdf y prueba2.pdf) ubicados en la carpeta storage\app\documentos\_de\_prueba.

- Los documentos generados son almacenados en base de datos como base64 para no sobrecargar el proyecto de archivos. En un proyecto y con un despliegue en producción también podríamos tener un repositorio de Archivos y almacenar la ruta.

-Se generarán 3 usuarios (uno por cada role) para realizar la prueba con su password correspondiente (userAdmin=passAdmin123, userResponsable=passResponsable123, userAsignado=passAsignado123).

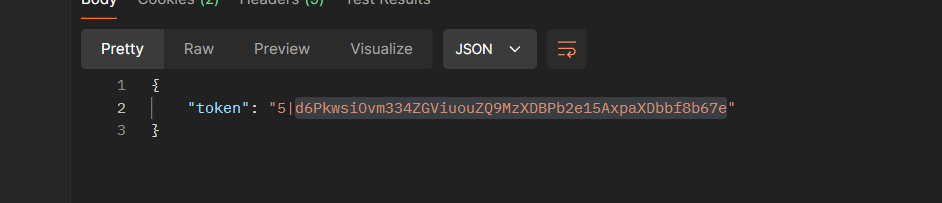
- Disponemos de 2 Endpoints. Uno de login por el que recogeremos el token y que recogerá los siguientes parámetros y los completaremos con los usuarios y contraseñas mencionados anteriormente. El endpoint será serivor(nombre\_proyecto)**/ api/login**

{

    "nick\_name":"userAdmin",

    "password": "passAdmin123"

}

Y recibiremos el token  


- El siguiente Endpoint será el que nos de la información de los documentos por Relevancia. Se llamará de la siguiente manera serivor(nombre\_proyecto)**/ api/** **getDocuments.**

Y le pasaremos por “Bearer” el token recibido anteriormente. Esperando recibir una respuesta similar a esta:



**ASPECTOS DE LA APLICACIÓN:**



