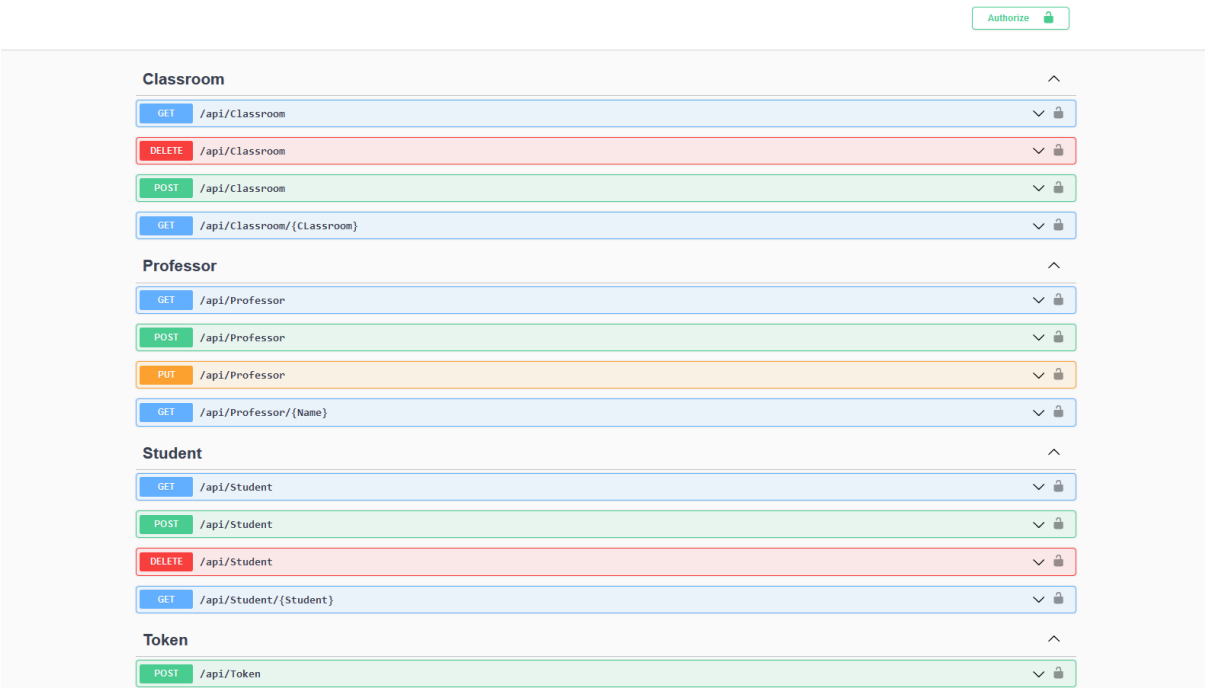


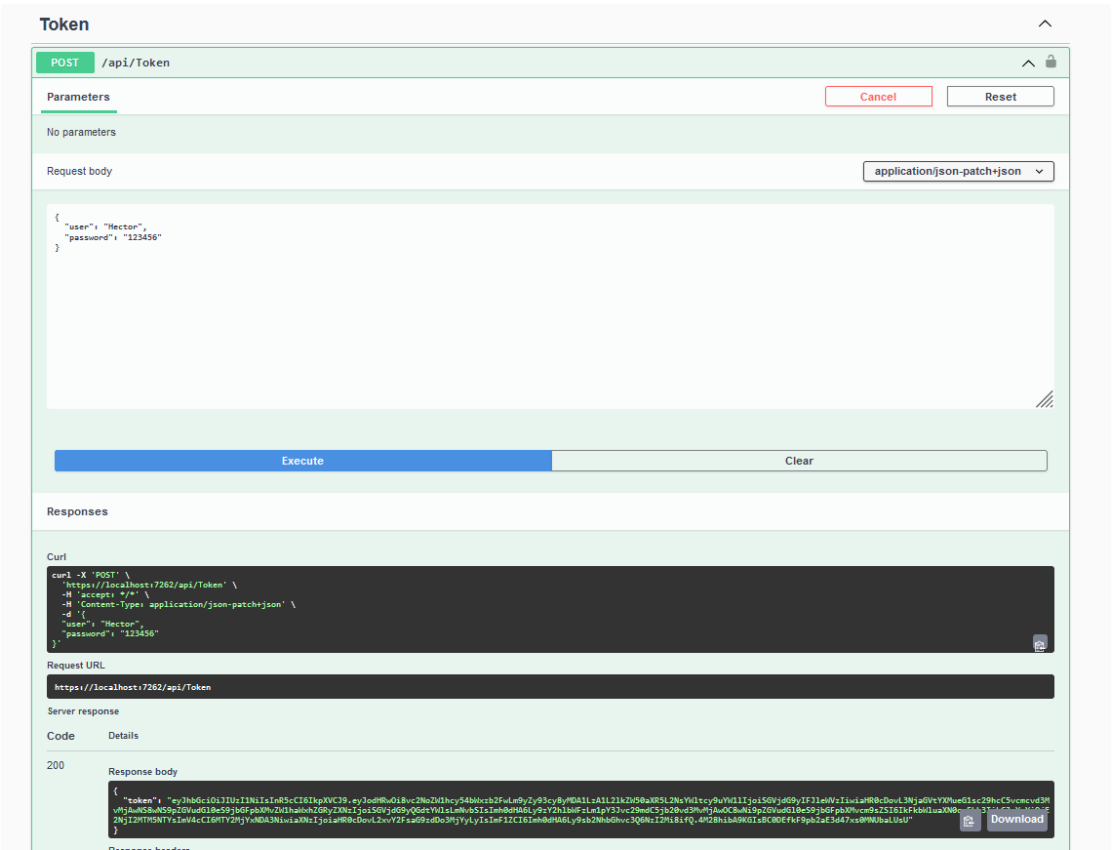
Documento técnico

Como Usarlo:

Al ejecutar el proyecto se abrirá el navegador donde se podrá encontrar con la IU de Swagger.



Para poder probar la funcionalidad de cada uno de los endpoints tendrá que ejecutar el endpoint de Token, el mismo solicitará un usuario y una contraseña, la introduce y devolverá un token.



apiagenda

vtGAS

https://localhost:7202/swagger/v1/swagger.json

Classroom

GET

/api/Classroom

DELETE

/api/Classroom

POST

/api/Classroom

GET

/api/Classroom/{Classroom}

Professor

GET

/api/Professor

POST

/api/Professor

Available authorizations

Bearer (apiKey)

JWT Authorization header using the Bearer scheme.
Enter 'Bearer'[space] and then your token in the text input below:
Example: "Bearer 12345abcde"
Name: Authorization
In: header
Value:

Bearer eyJhbGciOiJIUzI1NiIs

Authorize

Close

Authorize

Con estos pasos nuestro api estará disponible para utilizarse:

Classroom

GET /api/Classroom ⌵ 🔒

Parameters Cancel

No parameters

Execute
Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7262/api/Classroom' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50LmV4IiwiaWF0IjoiSjQyY0kyIFJleWZlcnkiOiAibG9vLnNjaW4tYXBwcm9keSc29kCnvczdDwrtJaW6MwAg92QWxkdldSPjGpfps<
  <
```

Request URL

```
https://localhost:7262/api/Classroom
```

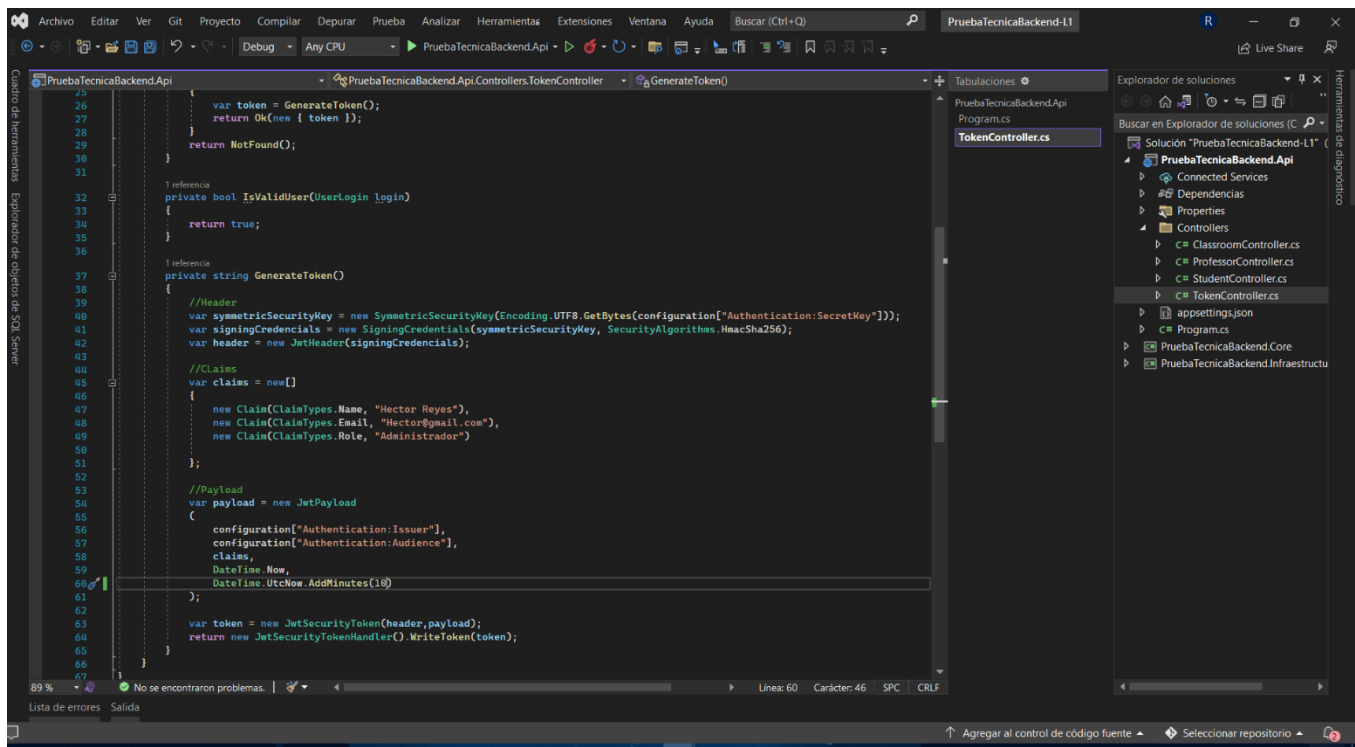
Code	Details
200	<p>Response body</p> <pre style="background-color: #2d3748; color: white; padding: 10px; border-radius: 5px;">{ [{ "classroomId": 1, "course": "Curso 1" }, { "classroomId": 2, "course": "Curso 1" }] }</pre> <p style="text-align: right;"> Download</p> <p>Response headers</p> <pre style="background-color: #2d3748; color: white; padding: 10px; border-radius: 5px;">content-length: 75 content-type: application/json; charset=utf-8 date: Thu, 08 Sep 2022 09:20:27 GMT server: Kestrel x-frame-options: deny</pre>

Responses

Code	Description

[Links](#)

El token tiene un tiempo de caducidad de 10 minutos, si le quiere poner un tiempo mayor, vaya al controlador TokenController y en el método GenerateToken donde dice AddMinutes cambie el 10 por el numero deseado.



Funcionalidades:

Classroom

Classroom			
GET	/api/Classroom		
DELETE	/api/Classroom		
POST	/api/Classroom		
GET	/api/Classroom/{Classroom}		

- Get: obtiene todos los cursos
- Delete: borrar cursos por Id
- Post: crea cursos
- Get: busca cursos por curso

Professor

Professor			
GET	/api/Professor		
POST	/api/Professor		
PUT	/api/Professor		
GET	/api/Professor/{Name}		

Get: obtiene todos los profesores.

Post: crea profesores. No se puede insertar un profesor en el mismo curso mas de una vez, no puede dar clases en el mismo turno, puede dar clases en diferentes aulas con turnos diferentes, los profesores pueden dar clases en el mismo turno, pero en cursos diferentes.

Put: elimina los usuarios actualizando su campo States a activo y su campo DateDelete a la fecha actual al momento de actualizarlo.

Get: busca profesores por nombre

Student

Student			^
GET	/api/Student	✓	🔒
POST	/api/Student	✓	🔒
DELETE	/api/Student	✓	🔒
GET	/api/Student/{Student}	✓	🔒

Get: obtiene todos los estudiantes

Post: crea estudiantes. No se pueden crear mas de 30 estudiantes por aula

Delete: borrar estudiantes por Id

Get: busca estudiantes por nombre

ScheduleWeek: *Este sería el CRUD del plus*

ScheduleWeek			^
GET	/api/ScheduleWeek	✓	🔒
POST	/api/ScheduleWeek	✓	🔒
DELETE	/api/ScheduleWeek	✓	🔒
PUT	/api/ScheduleWeek	✓	🔒
GET	/api/ScheduleWeek/GetScheduleWeekDayMonthYear	✓	🔒
GET	/api/ScheduleWeek/GetScheduleWeekMonthYear	✓	🔒
GET	/api/ScheduleWeek/GetScheduleWeek_WeekMonthYear	✓	🔒

Get: obtiene todos los horarios

Post: crea horarios

Delete: borrar horarios por Id

Put: Edita los horarios

Get: busca los horarios por día, mes y año

Get: busca los horarios por mes y año

Get: busca los horarios por semana, mes y año

Token

Token			^
POST	/api/Token	✓	🔒

Post: se inserta usuario y contraseña para obtener el token.

Herramientas Utilizadas

Visual Studio Community

SQL Server

Librerías Utilizadas:

AutoMapper: *lo utilizo para mapear las entidades en DTO, para copiar un objeto en otro.*

Newtonsoft.Json: *la utilizo para resolver o mas bien ignorar las referencias circulares.*

EntityFrameworkCore: *convierte estructuras de datos en clases para poder trabajar con esa información usando objetos.*