

# **Informe N°2**

## **Integrantes:**

- **Maximiliano Hernández**
- **Renato Valenzuela**
- **Diego Zamora**

# Índice

<b>Introducción.....</b>	<b>2</b>
<b>Desarrollo.....</b>	<b>3</b>
Problema:.....	3
Análisis de requerimientos:.....	3
Requisitos funcionales.....	4
Requisitos no funcionales.....	4
Sistema actual:.....	4
Planificación de la migración:.....	7
Estructura del proyecto.....	7
Base de Datos.....	8
Implementación de los servicios.....	8
Git - GitHub.....	11
<b>Conclusión.....</b>	<b>18</b>

# Introducción

En el presente informe abordamos la problemática que enfrenta la empresa EduTech Innovators SPA, cuyo sistema monolítico ha comenzado a presentar deficiencias ante el creciente aumento de usuarios y la demanda de nuevas funcionalidades. Como equipo, nuestro objetivo es proponer una solución tecnológica que permita mejorar el rendimiento y la escalabilidad del sistema. A lo largo del informe, detallamos cómo diagnosticamos la situación actual y desarrollamos una propuesta concreta de mejora.

Para ello, realizamos un análisis exhaustivo de los requerimientos empresariales, distinguiendo entre requerimientos funcionales y no funcionales, además de llevar a cabo entrevistas con distintos tipos de usuarios con el fin de comprender sus necesidades y expectativas. A partir de este diagnóstico, identificamos las limitaciones del sistema actual y planteamos una solución basada en una arquitectura de microservicios, acompañada de nuevas funcionalidades orientadas a optimizar la experiencia del usuario y el rendimiento del sistema.

Este informe incluye la representación del sistema propuesto mediante diagramas de casos de uso, diagramas de clases y diagramas de despliegue, que permiten visualizar y comprender los distintos aspectos del diseño. Finalmente, se presenta un plan de migración gradual, pensado para asegurar una transición eficiente y de bajo riesgo desde el sistema actual hacia la nueva arquitectura.

# Desarrollo

## Problema:

El problema principal de EduTech Innovators SPA es en su sistema de software monolítico actual, que en un inicio podría haber servido con un número limitado de usuarios, pero ahora que la demanda de usuario a crecido exponencialmente, este software que en un inicio funcionaba bien ahora está sufriendo problemas de rendimiento como tiempos lentos de respuesta, transacciones no procesadas a tiempo o incluso caídas del sistema. Esto debido a lo complejo que es escalar un sistema de software monolítico.

## Análisis de requerimientos:

Entrevista al perfil de Administrador:

- Buenos días, usted como administrador del sistema dígame ¿qué es lo que requiere su software?
- Necesitamos gestionar usuarios para crear, actualizar, desactivar y eliminar cuentas, además de configurar permisos de acceso a diferentes módulos. También requerimos monitorizar el sistema, recibir alertas de fallos y evaluar su rendimiento, e implementar respaldos periódicos para garantizar la restauración de datos en caso de fallos.

Entrevista al perfil de Vendedor:

- Buenas tardes, como vendedor que es, quisiera saber los requisitos que necesita el software.
- Los requerimientos que necesitamos incluyen gestionar cursos, generar reportes, asignar instructores, evaluar y actualizar contenido, manejar evaluaciones, interactuar con estudiantes mediante foros y chats, y monitorear su progreso.

Entrevista al perfil de Cliente:

- Buenos días, usted como cliente me podría decir ¿que requerimientos o funciones le gustaría que el software implementara?
- Lo que necesito es registrarme en la plataforma para crear una cuenta y luego iniciar sesión con mis credenciales. Quiero poder explorar los cursos fácilmente, usando filtros y una barra de búsqueda para encontrar lo que necesito. Después de eso, me

gustaría inscribirme en los cursos que elija. También es importante poder consultar mi progreso en los cursos que estoy tomando. Necesitaría acceso a soporte para enviar cualquier consulta o problema mediante un formulario o chat en línea. Además, quiero dejar reseñas y calificaciones para compartir mi opinión sobre los cursos. Finalmente, me gustaría usar cupones y descuentos durante el proceso de inscripción, ingresando los códigos promocionales.

En cuanto a las necesidades del cliente, se organizó una lista de requerimientos, la cual se dividió en dos categorías: aquellos con interacción directa con el usuario (requisitos funcionales) y aquellos relacionados con el funcionamiento interno de la aplicación (requisitos no funcionales). A continuación, los ya mencionados:

### **Requisitos funcionales**

- Gestión de usuarios
- Configuración de permisos
- Respaldo y restauración de datos
- Monitorización del sistema

### **Requisitos no funcionales**

- Un sistema de microservicios
- Un motor de base de datos SQL
- Plan de migración
- Análisis de requerimientos
- Análisis del sistema actual
- Diseño de la nueva arquitectura.

## **Sistema actual:**

El sistema actual de EduTech Innovators SPA es un sistema monolítico, lo que significa que todas las funcionalidades están integradas en una única aplicación. Esto incluye la gestión de usuarios, la administración de cursos, la entrega de contenido educativo y la evaluación de estudiantes.

Como componentes posee una interfaz de usuario, manejo de inscripciones como a su vez el seguimiento del progreso una vez hecha la inscripción y un sistema de base de datos para almacenar la información de los usuarios, como así también, el contenido de los cursos.

## **Nueva arquitectura:**

Teniendo esto en cuenta como equipo haciendo análisis de los requerimientos, decidimos que la forma más eficiente de llevar las necesidades actuales de la empresa es cambiando su estructura de sistema monolítico a una arquitectura de microservicios, debido a que este sistema se basa en dividir una aplicación en pequeños servicios independientes que se

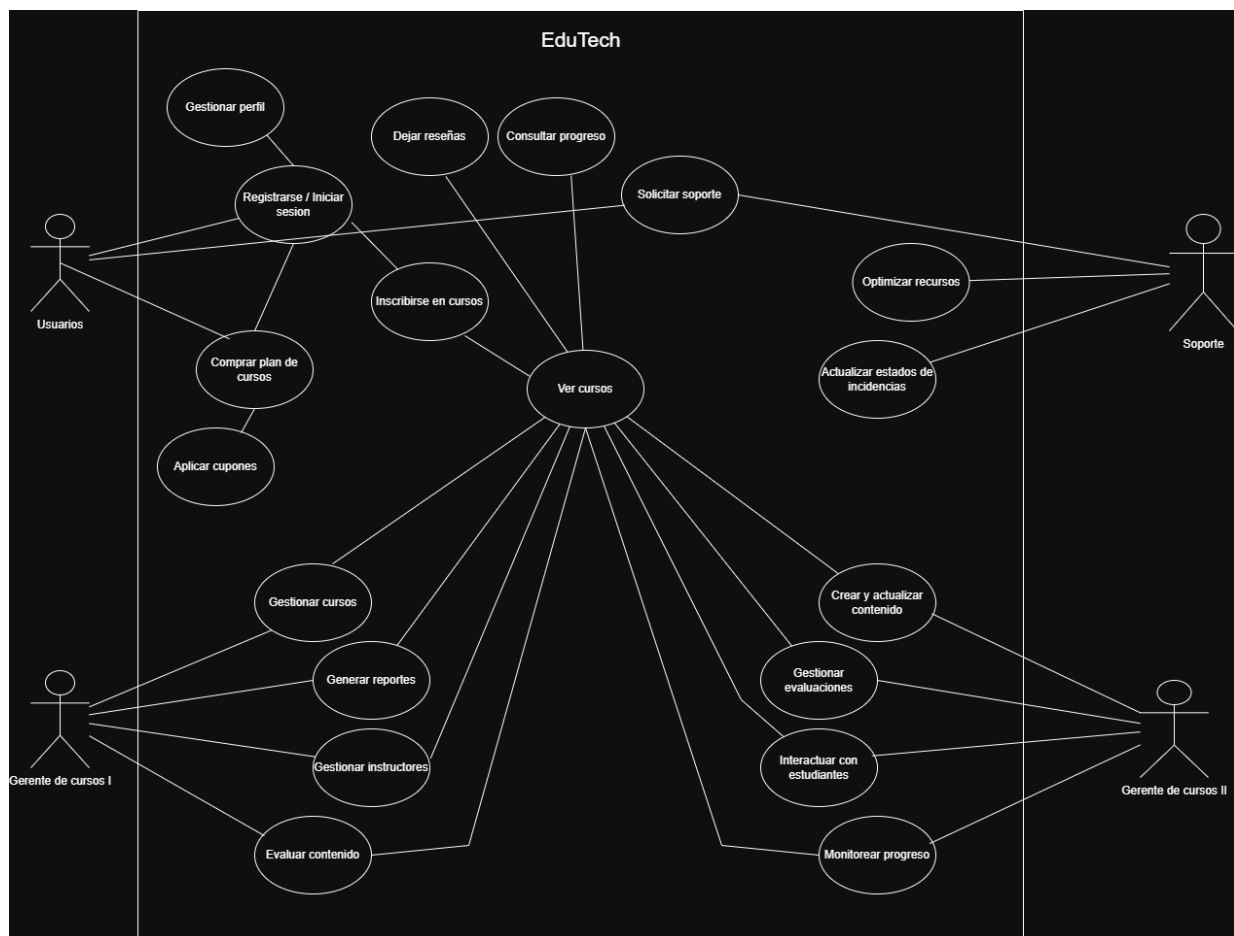
encargan de tareas específicas que se comunican entre sí mediante APIs o mensajes, cada microservicio es autónomo dando así una mayor flexibilidad y escalabilidad.

Esto es conveniente igual por el lado de tener una escalabilidad independiente, un desarrollo más ágil, la resiliencia que vendría siendo que cuando un microservicio falle, este no afectará a toda la aplicación, tendremos una flexibilidad tecnológica ya que los microservicios pueden ser desarrollados usando distintas tecnologías (como lenguajes de programación) y finalmente, tendrá un mantenimiento mucho más simple al ser su identificación y corrección de errores más sencilla.

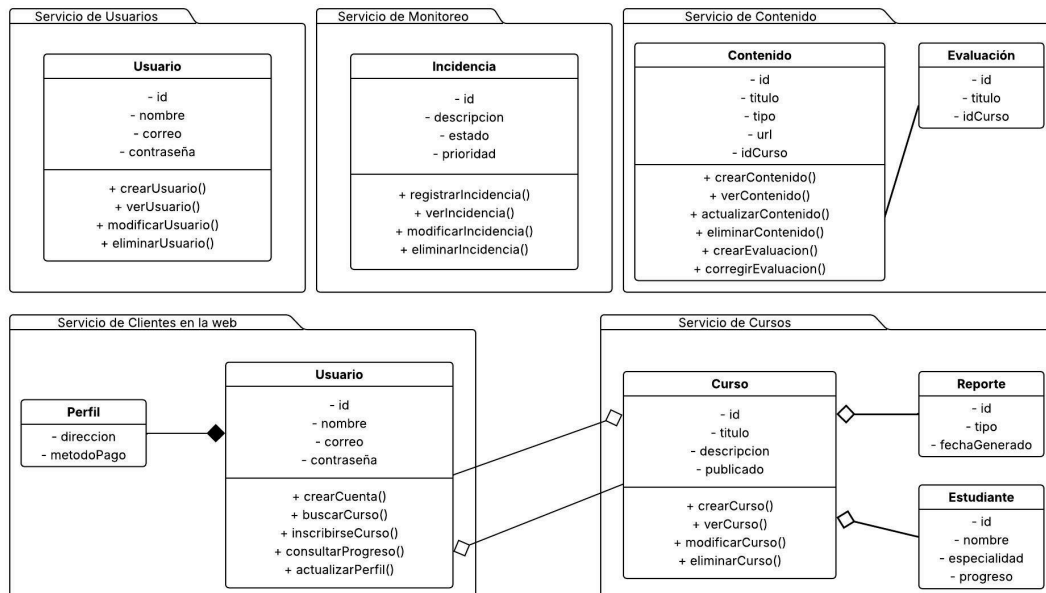
Esto resulta conveniente, ya que permite una escalabilidad independiente, facilita un desarrollo más ágil y aporta resiliencia: si un microservicio falla, no afectará al funcionamiento de toda la aplicación. Además, ofrece flexibilidad tecnológica, ya que cada microservicio puede desarrollarse con tecnologías distintas (como también diferentes lenguajes de programación). Por último, el mantenimiento se vuelve más sencillo, ya que la identificación y corrección de errores es más rápida y localizada.

Para complementar esta nueva arquitectura, a continuación se presentan tres diagramas que ilustran distintos aspectos clave del sistema.

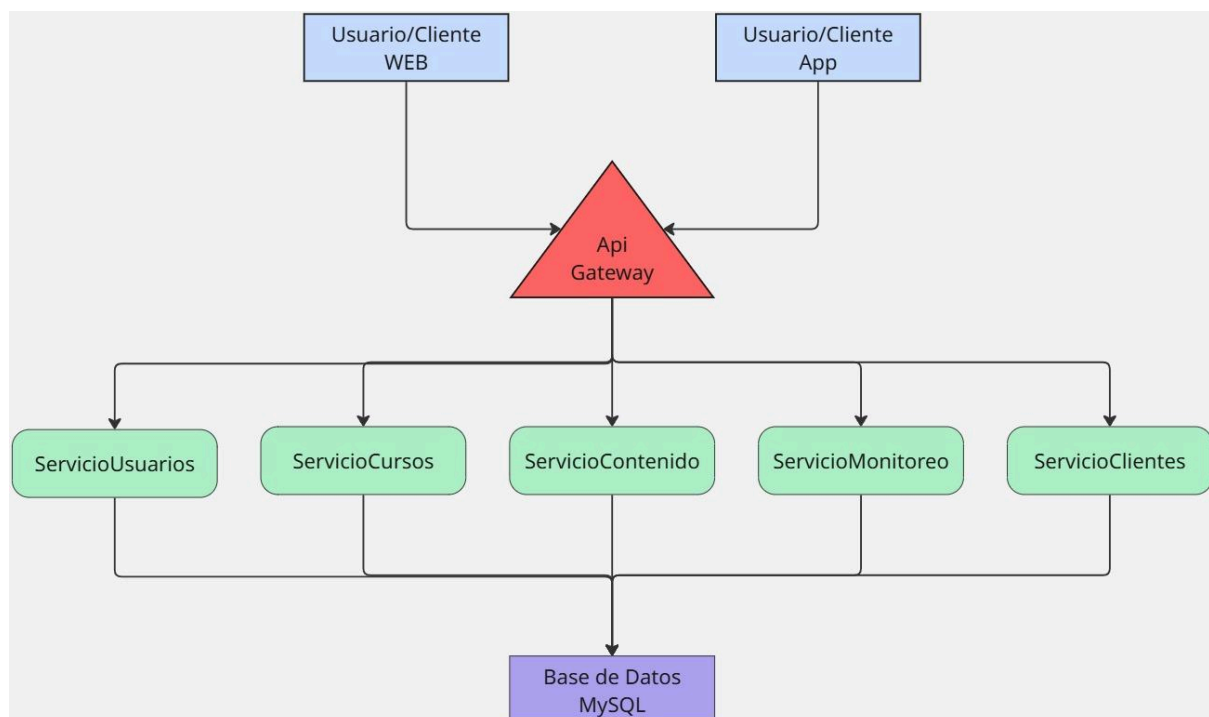
El *diagrama de casos de uso* muestra las principales interacciones entre los actores y las funcionalidades del sistema.



El *diagrama de clases* detalla la estructura lógica de los objetos y sus relaciones dentro de los microservicios.



Finalmente, el *diagrama de despliegue* representa la arquitectura física del sistema, incluyendo la distribución de los microservicios, sus bases de datos y las formas de comunicación entre ellos.



## Planificación de la migración:

Nuestra estrategia de migración no busca reescribir todo el sistema de una sola vez, sino extraer funcionalidades específicas para convertirlas en microservicios gradualmente. Este enfoque nos permite una transición más ordenada y reduce los riesgos asociados a cambios drásticos. Además, definiremos cómo se comunicarán entre sí los microservicios (por ejemplo, mediante APIs) y cada uno contará con su propia base de datos, lo que ayudará a evitar acoplamientos y a mantener su independencia.

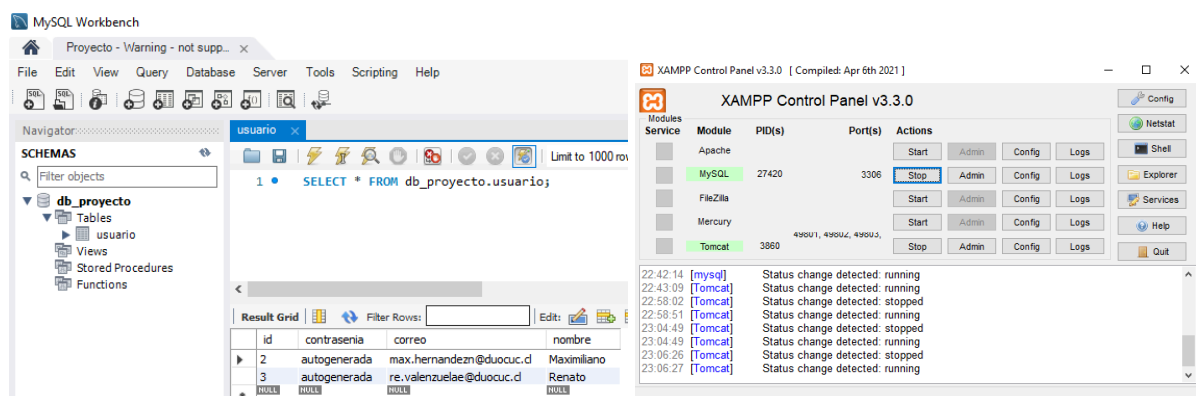
## Estructura del proyecto

Haciendo uso del editor de texto Visual Studio Code, creamos un proyecto con el framework de SpringBoot utilizando Maven dado que trabajamos con java, con este mismo instalamos cuatro dependencias: Spring Web, DevTools, Spring Data JPA y MySQL Driver.

Spring Web nos proporcionó lo necesario para trabajar con API RESTful, DevTools nos mejora la experiencia cuando estamos trabajando con la aplicación levantada (por ejemplo, actualizando en tiempo real cambios que se vayan haciendo en el código), JPA nos ayudó a la conexión con la base de datos, en específico el acceso, también otorgando herramientas para crear interfaces CRUD y por último MySQL Driver es el conector “oficial” entre Java/Springboot con bases de datos en MySQL y le permite a JPA hacer sus consultas en este mismo.

## Base de Datos

Se utilizó el software MySQL Workbench, en específico la versión 8.0.26 para hacer la conexión a la base de datos, pero antes, se hizo uso de el software XAMPP para levantar la conexión del editor de texto donde se trabaja el código a la de base de datos.



The image shows two screenshots side-by-side. The left screenshot is of MySQL Workbench, displaying a query result for the 'usuario' table in the 'db\_proyecto' database. The query is 'SELECT \* FROM db\_proyecto.usuario;'. The result grid shows 3 rows of data:

id	contrasenia	correo	nombre
2	autogenerada	max.hernandez@duocuc.d	Maximiliano
3	autogenerada	re.valenzuelae@duocuc.d	Renato

The right screenshot is of the XAMPP Control Panel v3.3.0. It shows a list of services and their status:

Service	Module	PID(s)	Port(s)	Actions
Apache				Start Admin Config Logs
MySQL		27420	3306	Stop Admin Config Logs
FileZilla				Start Admin Config Logs
Mercury			49001, 49002, 49003	Start Admin Config Logs
Tomcat		3860		Stop Admin Config Logs

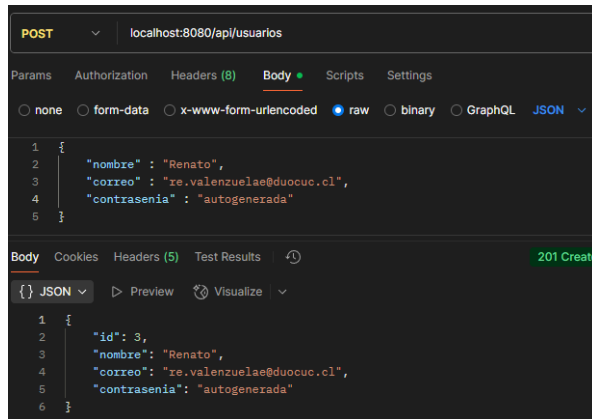
The bottom of the XAMPP Control Panel shows a log of status changes for the services.



# Implementación de los servicios

Se implementó el servicio de usuarios donde se creó la clase usuario en la cual podrá ingresar uno con su nombre, su correo y su contraseña.

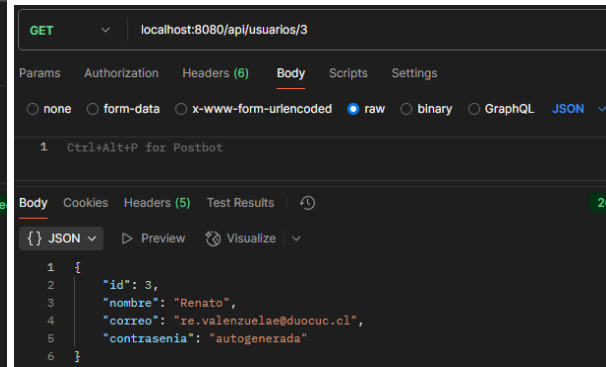
## Crear Usuario



```

POST localhost:8080/api/usuarios
Body
{
  "nombre": "Renato",
  "correo": "re.valenzuelae@duocuc.cl",
  "contrasenia": "autogenerada"
}
201 Create
  
```

## Ver Usuario

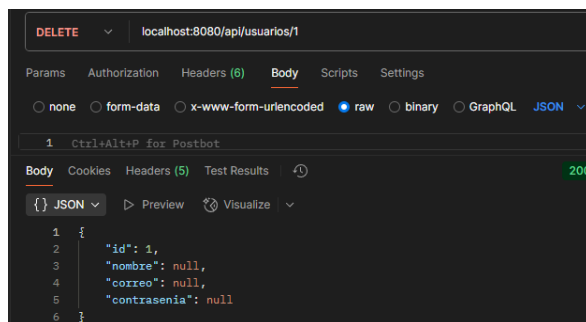


```

GET localhost:8080/api/usuarios/3
Body
{
  "id": 3,
  "nombre": "Renato",
  "correo": "re.valenzuelae@duocuc.cl",
  "contrasenia": "autogenerada"
}
  
```

	id	contrasenia	correo	nombre
▶	1	NULL	max.hernandezn@duocuc.d	Maximiliano
	2	H20466593-1	max.hernandezn@duocuc.d	Maximiliano
	3	autogenerada	re.valenzuelae@duocuc.d	Renato
*	NULL	NULL	NULL	NULL

## Eliminar Usuario

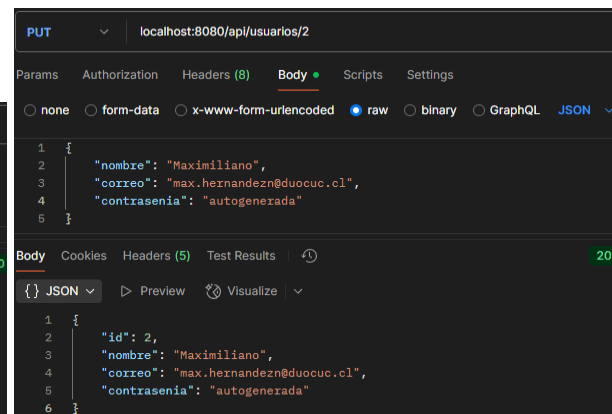


```

DELETE localhost:8080/api/usuarios/1
Body
{
  "id": 1,
  "nombre": null,
  "correo": null,
  "contrasenia": null
}
200
  
```

	id	contrasenia	correo	nombre
▶	2	H20466593-1	max.hernandezn@duocuc.d	Maximiliano
	3	autogenerada	re.valenzuelae@duocuc.d	Renato
*	NULL	NULL	NULL	NULL

## Modificar Usuario

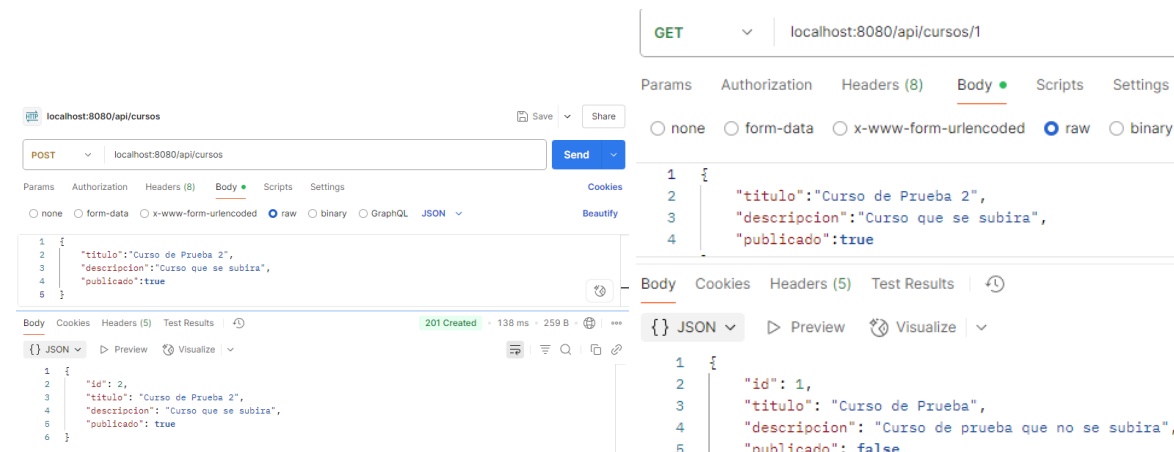


```

PUT localhost:8080/api/usuarios/2
Body
{
  "nombre": "Maximiliano",
  "correo": "max.hernandezn@duocuc.cl",
  "contrasenia": "autogenerada"
}
200
  
```

	id	contrasenia	correo	nombre
▶	2	autogenerada	max.hernandezn@duocuc.d	Maximiliano
	3	autogenerada	re.valenzuelae@duocuc.d	Renato
*	NULL	NULL	NULL	NULL

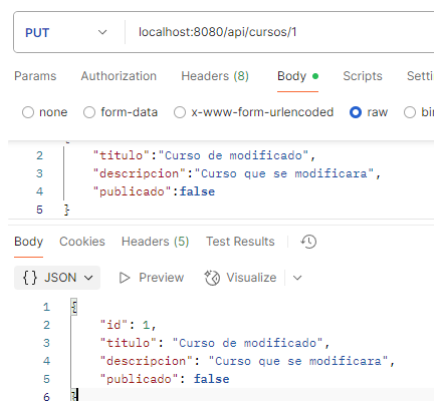
Agregando, también se agregó el servicio de cursos, el cual por ahora tiene la clase curso que nos permite ingresar los datos principales de este como el título, una descripción y si el curso está activo o no (si está publicado).



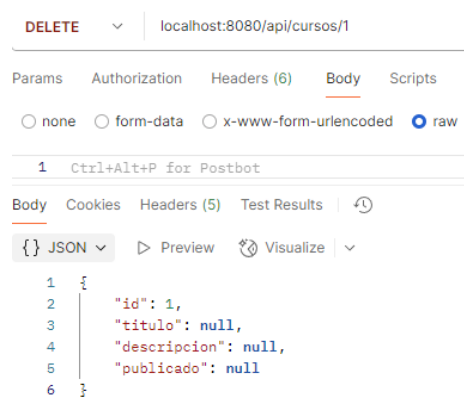
## Crear Cursos

## Ver cursos

id	descripcion	publicado	titulo
1	Curso que se modificara	0	Curso de modificado
2	Curso que se subira	1	Curso de Prueba 2
NULL	NULL	NULL	NULL



## Modificar curso

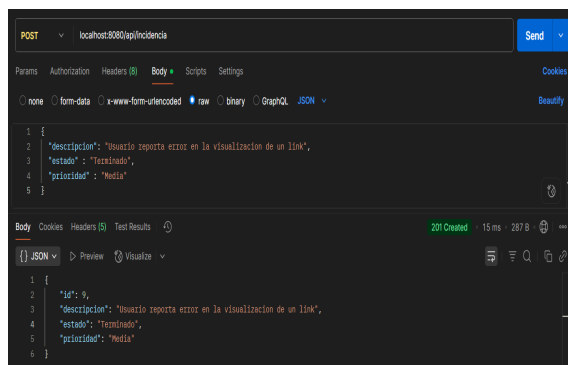


## Borrar Curso

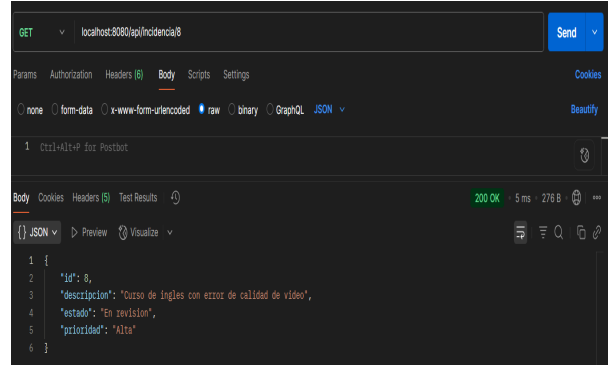
id	descripcion	publicado	titulo
1	Curso que se modificara	0	Curso de modificado
2	Curso que se subira	1	Curso de Prueba 2
NULL	NULL	NULL	NULL

id	descripcion	publicado	titulo
2	Curso que se subira	1	Curso de Prueba 2
NULL	NULL	NULL	NULL

Por último, se agregó el servicio de monitoreo donde se incluyó la clase principal de incidencia, estas tendrán una descripción de esta, su estado y su prioridad.

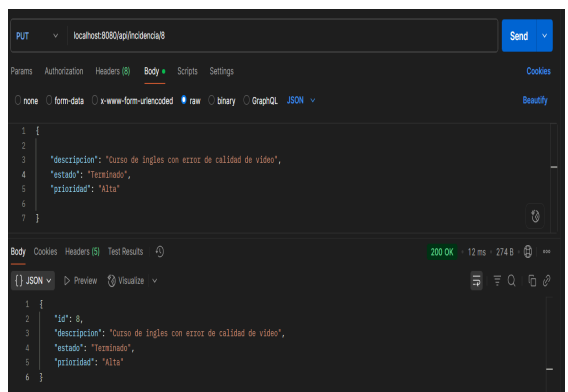


Crear Incidencia

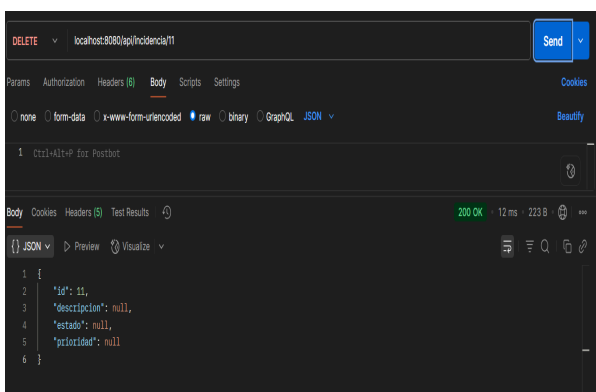


Ver Incidencia

id	descripcion	estado	prioridad
8	Curso de ingles con error de calidad de video	En revision	Alta
9	Usuario reporta error en la visualizacion de un link	Terminado	Media
10	Actualizar descripcion del curso	Pendiente	Baja
11	Actualizar lista de usuarios del curso	Terminada	Alta
NULL	NULL	NULL	NULL



Modificar Incidencia



Eliminar Incidencia

id	descripcion	estado	prioridad
8	Curso de ingles con error de calidad de video	Terminado	Alta
9	Usuario reporta error en la visualizacion de un link	Terminado	Media
10	Actualizar descripcion del curso	Pendiente	Baja
11	Actualizar lista de usuarios del curso	Terminada	Alta
NULL	NULL	NULL	NULL

id	descripcion	estado	prioridad
8	Curso de ingles con error de calidad de video	Terminado	Alta
9	Usuario reporta error en la visualizacion de un link	Terminado	Media
10	Actualizar descripcion del curso	Pendiente	Baja
NULL	NULL	NULL	NULL

## Git - GitHub

Para conectar el proyecto a GitHub, primero conectamos la carpeta donde tenemos el proyecto.

```
MINGW64:/c:/Users/renax/OneDrive/Documentos/ProyectoSemestral

renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git init
Initialized empty Git repository in C:/Users/renax/OneDrive/Documentos/ProyectoSemestral/.git/

renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$
```

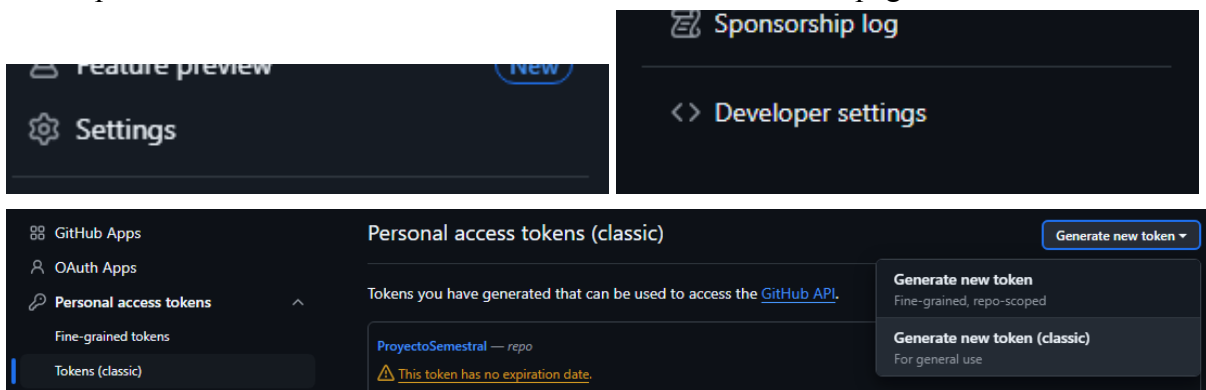
Luego, creamos las credenciales con el usuario dueño del repositorio.

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git config user.name RenatoV

renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git config user.email renax.valenzuela@duocuc.cl

renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git config user.password ghp_0x0t1p0a_m0G1n4d0mGAT7p0d0T7h0k1p0d0g1g0k
```

Donde para la contraseña se hizo uso de un Token creado desde la página de GitHub



Conectamos el repositorio de GitHub con el siguiente comando:

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git remote add origin https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
```

Esto hará que podamos trabajar el repositorio de forma remota desde nuestros ordenadores. Luego, añadimos nuestras carpetas donde estamos trabajando, ya que añadiremos todo sería con:

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git add .
warning: in the working copy of '.vscode/launch.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'springboot-backend/.gitattributes', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'springboot-backend/.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'springboot-backend/.mvn/wrapper/maven-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'springboot-backend/mvnw.cmd', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'springboot-backend/pom.xml', LF will be replaced by CRLF the next time Git touches it
```

Luego, subimos los archivos que añadimos a través de un commit:

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (master)
$ git commit -m "Subida de los archivos del proyecto"
[master (root-commit) 070e5d7] Subida de los archivos del proyecto
17 files changed, 796 insertions(+)
create mode 100644 .vscode/launch.json
create mode 100644 "Caso de estudio N°302\2602 EduTech Innovators SPA.pdf"
create mode 100644 Informe 1, Full Stack 1.pdf
create mode 100644 springboot-backend/.gitattributes
create mode 100644 springboot-backend/.gitignore
create mode 100644 springboot-backend/.mvn/wrapper/maven-wrapper.properties
create mode 100644 springboot-backend/mvnw
create mode 100644 springboot-backend/mvnw.cmd
create mode 100644 springboot-backend/pom.xml
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/SpringbootBackendApplication.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/controllers/UsuarioController.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/entities/Usuario.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/repository/UsuarioRepository.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/services/UsuarioService.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/bac
kend/springboot_backend/services/UsuarioServiceImpl.java
create mode 100644 springboot-backend/src/main/resources/application.properties
create mode 100644 springboot-backend/src/test/java/com/proyecto/springboot/bac
kend/springboot_backend/SpringbootBackendApplicationTests.java
```

y para subirlo, le hacemos un push:

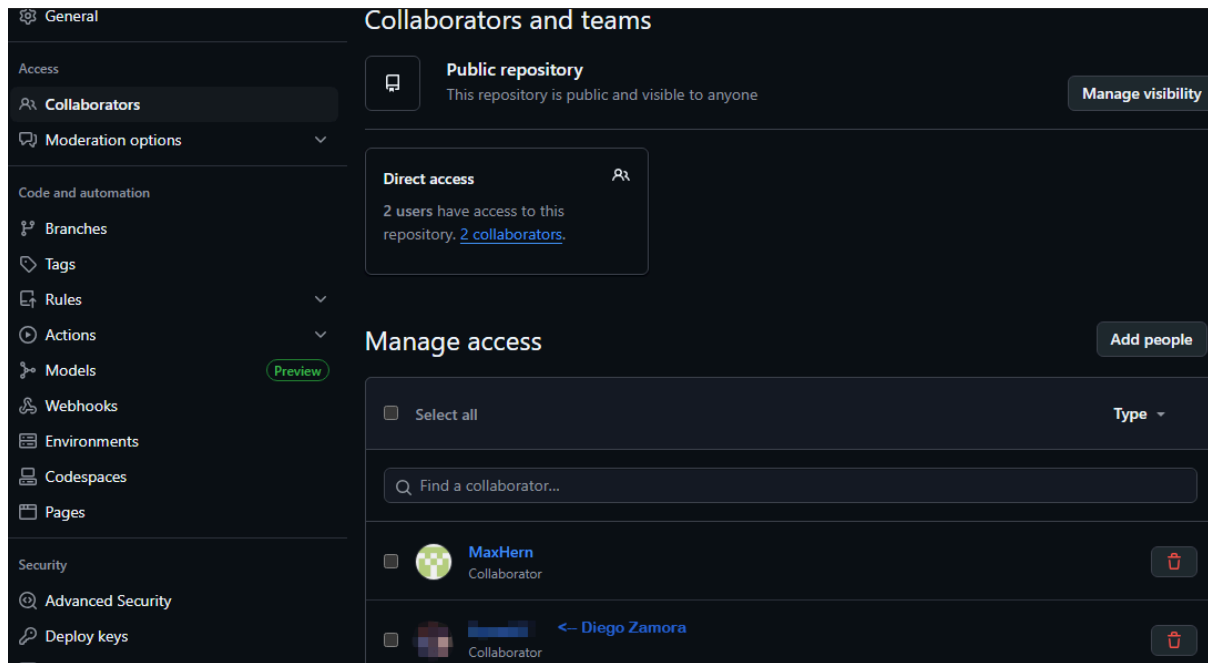
```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (main)
$ git push origin main
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 12 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (43/43), 11.96 KiB | 3.99 MiB/s, done.
Total 43 (delta 2), reused 34 (delta 2), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
2717ec0..4d77520 main -> main
```

Así se vería en GitHub:

RenatoValenzuela262		5 Commits
.vscode	Prueba commit	2 days ago
springboot-backend	Prueba commit	2 days ago
Caso de estudio N°2 EduTech Innovators SPA.p...	Entrega 1. Caso EduTech Innovators	last month
Informe 1, Full Stack 1.pdf	Entrega 1. Caso EduTech Innovators	last month

Aquí se subió un CRUD, para los otros dos se crearon Branches para trabajar los otros dos CRUDs por separado.

Antes de subir archivos a los branch creados, se añadieron como colaboradores al github los participantes restantes:



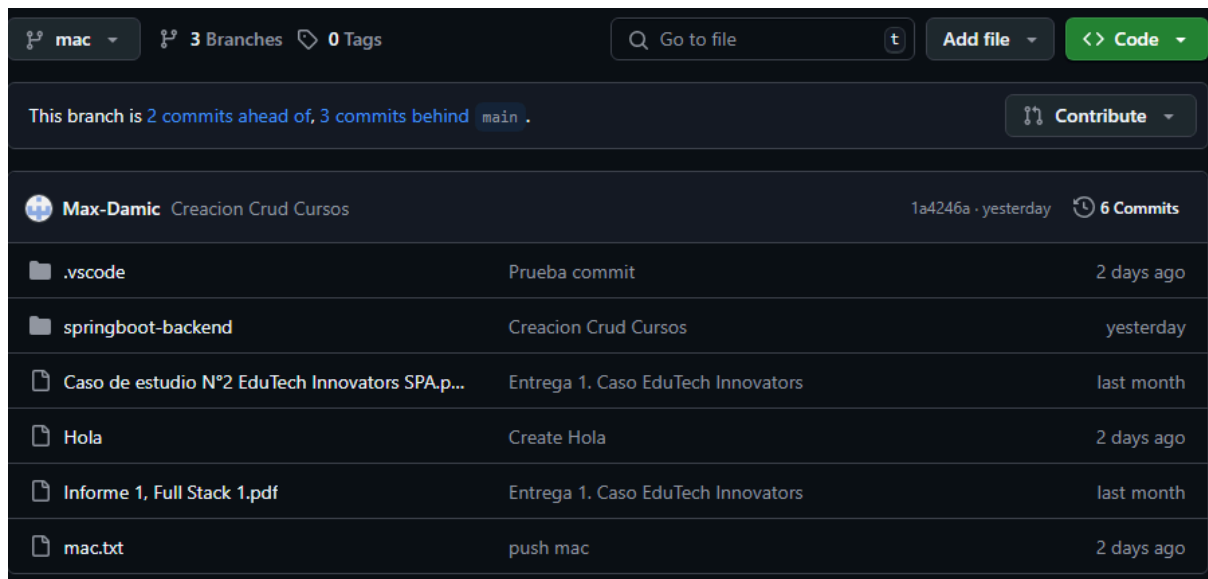
Luego de la creación de ambos branch de trabajo, la branch y el CRUD de cursos se subió al branch asignado a Maximiliano Hernández. El commit de este CRUD:

```
maxim@DESKTOP-8IGITE2 MINGW64 ~/OneDrive/Documentos/proyectoSemestral/Proyecto-Semestral (mac)
$ git add .

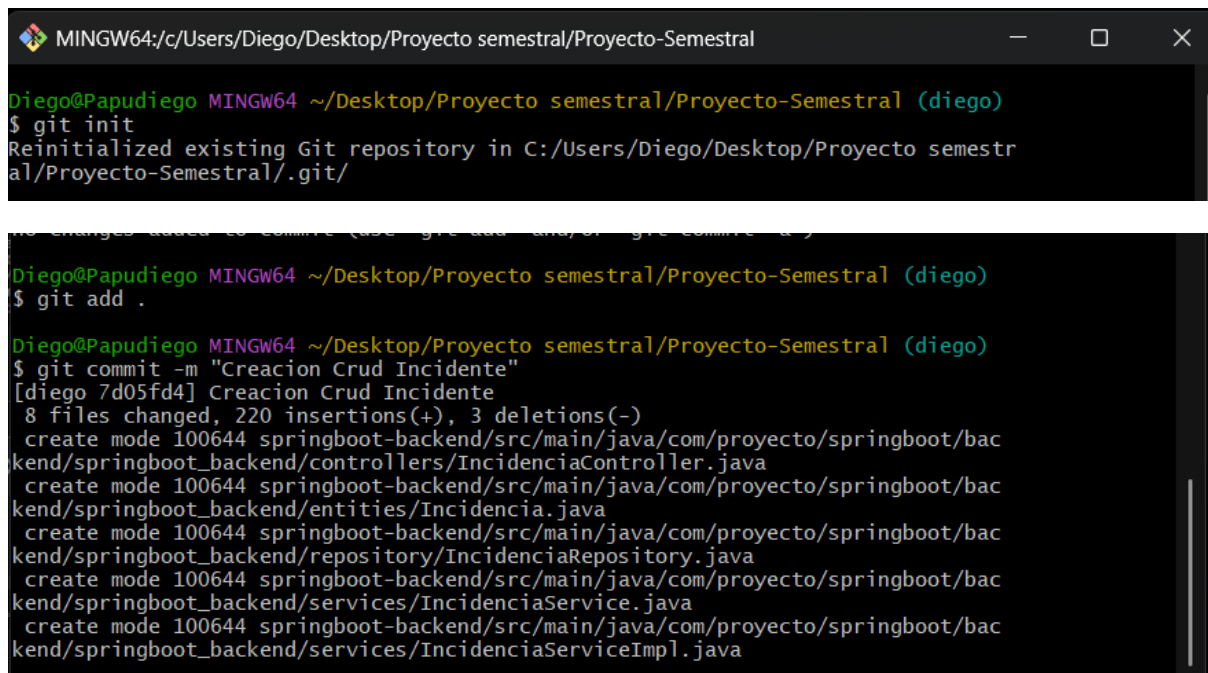
maxim@DESKTOP-8IGITE2 MINGW64 ~/OneDrive/Documentos/proyectoSemestral/Proyecto-Semestral (mac)
$ git commit -m "Creacion Crud Cursos"
[mac 1a4246a] Creacion Crud Cursos
 5 files changed, 216 insertions(+)
 create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/controllers/CursoController.java
 create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/entities/Curso.java
 create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/repositories/CursoRepository.java
 create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/CursoService.java
 create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/CursoServiceImpl.java

maxim@DESKTOP-8IGITE2 MINGW64 ~/OneDrive/Documentos/proyectoSemestral/Proyecto-Semestral (mac)
$ git push -u origin mac
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 6 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (20/20), 2.98 KiB | 610.00 KiB/s, done.
Total 20 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
 54d31c8..1a4246a mac -> mac
branch 'mac' set up to track 'origin/mac'.
```

Lo cual en su branch en GitHub se vería así:



Por último, la branch y el CRUD de incidencia se subieron al branch asignado a Diego Zamora. El commit de este CRUD:





```
Diego@Papudiego MINGW64 ~/Desktop/Proyecto semestral/Proyecto-Semestral (diego)
$ git push -u origin diego
Enumerating objects: 41, done.
Counting objects: 100% (41/41), done.
Delta compression using up to 22 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (24/24), 3.45 KiB | 220.00 KiB/s, done.
Total 24 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 4 local objects.
To https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
   e270032..7d05fd4  diego -> diego
branch 'diego' set up to track 'origin/diego'.

Diego@Papudiego MINGW64 ~/Desktop/Proyecto semestral/Proyecto-Semestral (diego)
$
```

Finalmente, trabajando en branches distintas nos fijamos que funcionaran de forma independiente y las fusionamos a la rama principal:

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (main)
$ git merge origin/mac main
Merge made by the 'ort' strategy.
 mac.txt | 1 +
.../controllers/CursoController.java | 74 +++++
.../backend/springboot_backend/entities/Curso.java | 63 +++++
.../repository/CursoRepository.java | 11 +++
.../springboot_backend/services/CursoService.java | 18 +++++
.../services/CursoServiceImpl.java | 50 +++++
6 files changed, 217 insertions(+)
create mode 100644 mac.txt
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/controllers/CursoController.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/entities/Curso.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/repository/CursoRepository.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/CursoService.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/CursoServiceImpl.java
```

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 274 bytes | 274.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
   4d77520..98c513c  main -> main
branch 'main' set up to track 'origin/main'.
```

```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (main)
$ git merge origin/diego main
Merge made by the 'ort' strategy.
 papuDiego.txt | 1 +
springboot-backend/pom.xml | 2 +-
.../controllers/IncidenciaController.java | 76 +++++
.../springboot_backend/entities/Incidencia.java | 65 +++++
.../repository/IncidenciaRepository.java | 9 +++
.../services/IncidenciaService.java | 21 +++++
.../services/IncidenciaServiceImpl.java | 46 +++++
.../services/UsuarioServiceImpl.java | 2 +-
.../src/main/resources/application.properties | 2 +-
9 files changed, 221 insertions(+), 3 deletions(-)
create mode 100644 papuDiego.txt
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/controllers/IncidenciaController.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/entities/Incidencia.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/repository/IncidenciaRepository.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/IncidenciaService.java
create mode 100644 springboot-backend/src/main/java/com/proyecto/springboot/backend/springboot_backend/services/IncidenciaServiceImpl.java
```



```
renax@DESKTOP-37USU06 MINGW64 ~/OneDrive/Documentos/ProyectoSemestral (main)
$ git push -u origin main
Enumerating objects: 43, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 1.41 KiB | 481.00 KiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/RenatoValenzuela262/Proyecto-Semestral.git
 98c513c..a9b398e  main -> main
branch 'main' set up to track 'origin/main'.
```

Lo cual, finalmente, reflejado en GitHub nos queda así:

<ul style="list-style-type: none"> <li>controllers</li> <li>entities</li> <li>repository</li> <li>services</li> </ul>	<ul style="list-style-type: none"> <li>db_proyecto             <ul style="list-style-type: none"> <li>Tables                 <ul style="list-style-type: none"> <li>curso</li> <li>incidencia</li> <li>usuario</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>CursoController.java</li> <li>IncidenciaController.java</li> <li>UsuarioController.java</li> </ul>	<ul style="list-style-type: none"> <li>Curso.java</li> <li>Incidencia.java</li> <li>Usuario.java</li> </ul>
<ul style="list-style-type: none"> <li>CursoRepository.java</li> <li>IncidenciaRepository.java</li> <li>UsuarioRepository.java</li> </ul>	<ul style="list-style-type: none"> <li>CursoService.java</li> <li>CursoServiceImpl.java</li> <li>IncidenciaService.java</li> <li>IncidenciaServiceImpl.java</li> <li>UsuarioService.java</li> <li>UsuarioServiceImpl.java</li> </ul>

# Conclusión

En conclusión, el análisis y rediseño del sistema de software para EduTech Innovators SPA representan un paso clave hacia la modernización tecnológica de la empresa. La adopción de una arquitectura basada en microservicios ofrece una solución robusta y escalable frente a las limitaciones del sistema monolítico actual, permitiendo afrontar con éxito el aumento sostenido de usuarios y la necesidad de mayor flexibilidad en el desarrollo de nuevas funcionalidades.

Gracias a la identificación detallada de los requerimientos funcionales y no funcionales, junto con las entrevistas realizadas a distintos perfiles de usuarios, fue posible comprender en profundidad las problemáticas existentes y diseñar una propuesta alineada con las necesidades reales de la organización. Este proceso fue acompañado por la elaboración de diagramas de casos de uso, clases y despliegue, herramientas clave para visualizar la solución planteada desde distintas perspectivas.

Finalmente, el plan de migración gradual propuesto asegura una transición controlada y eficiente hacia la nueva arquitectura, minimizando riesgos operativos. Esta transformación no solo mejorará el rendimiento y la experiencia del usuario, sino que también posicionará a EduTech Innovators SPA en una mejor situación para enfrentar los desafíos tecnológicos del futuro.