Informe EduTech Innovators SPA

Antonio Valdes | Desarrollo Fullstack

**Índice**

1. Introducción
2. Diagramas
3. Plan de pruebas

- Pruebas unitarias

- Pruebas de integración

1. Ejecución de pruebas  
   -Documentación de las pruebas ejecutadas
2. GitHub y los comandos utilizados
3. Conclusión

1.Introducción

EduTech Innovators SPA es una empresa dedicada a la educación online. Debido al crecimiento de la demanda y la necesidad de escalar su sistema, se optó por rediseñar la solución tecnológica basada en microservicios. Este informe presenta el diseño e implementación de tres microservicios principales: Usuarios, Cursos y Notificaciones, usando Spring Boot, MySQL y control de versiones con Git y GitHub.

2.Diagramas

Diagrama de arquitectura de microservicios:

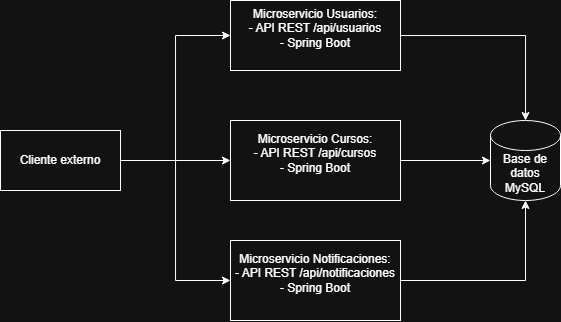


Diagrama de Casos de uso cambiado:

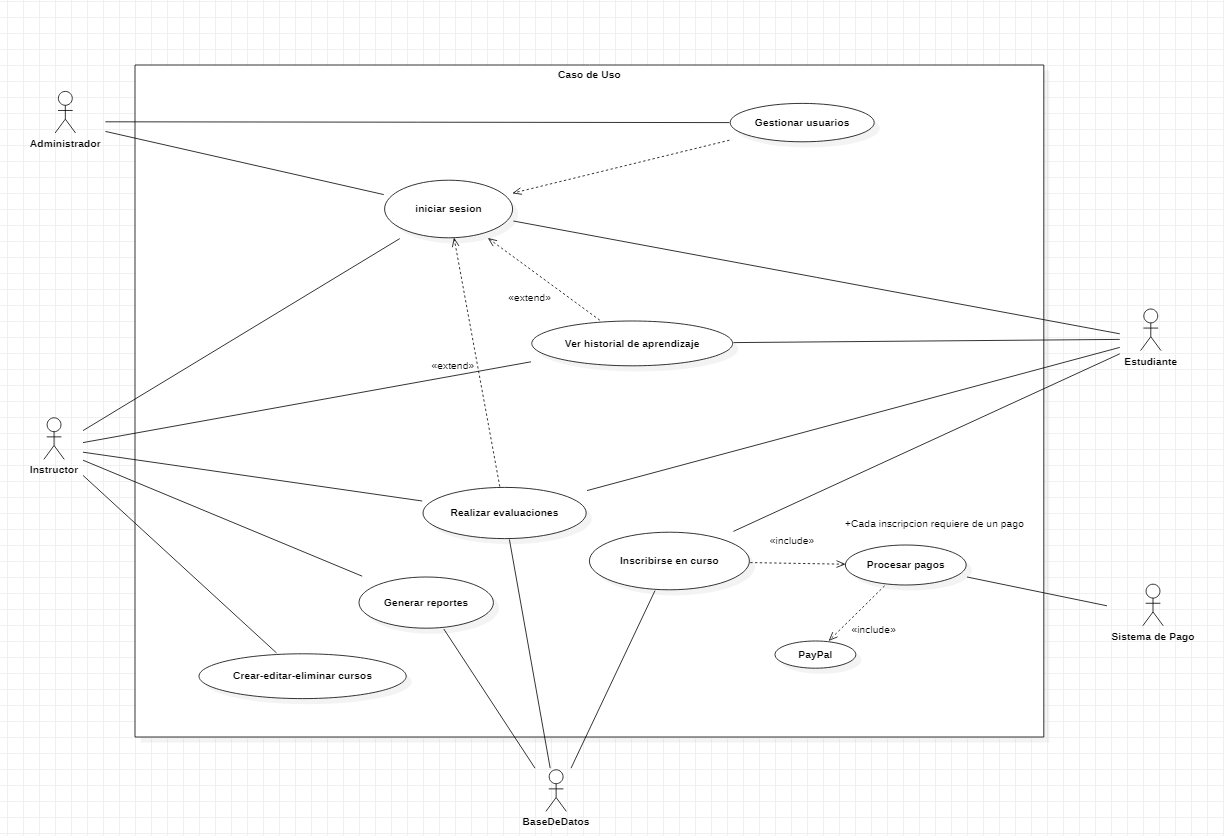


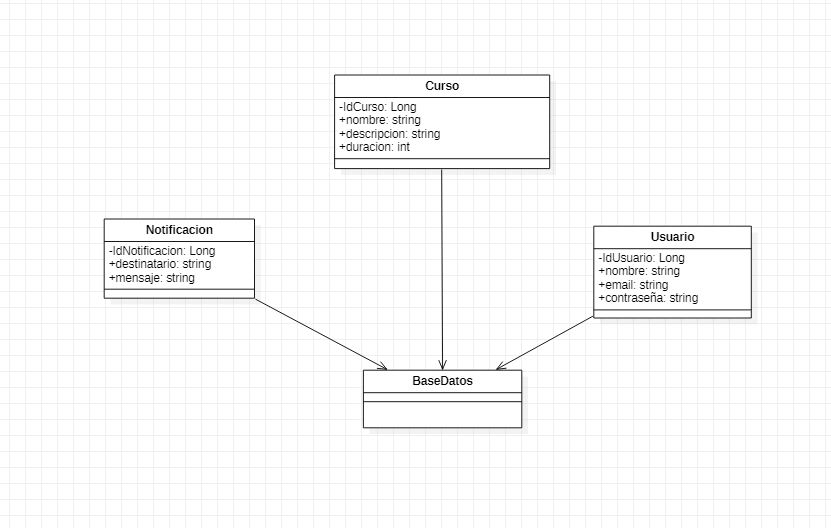
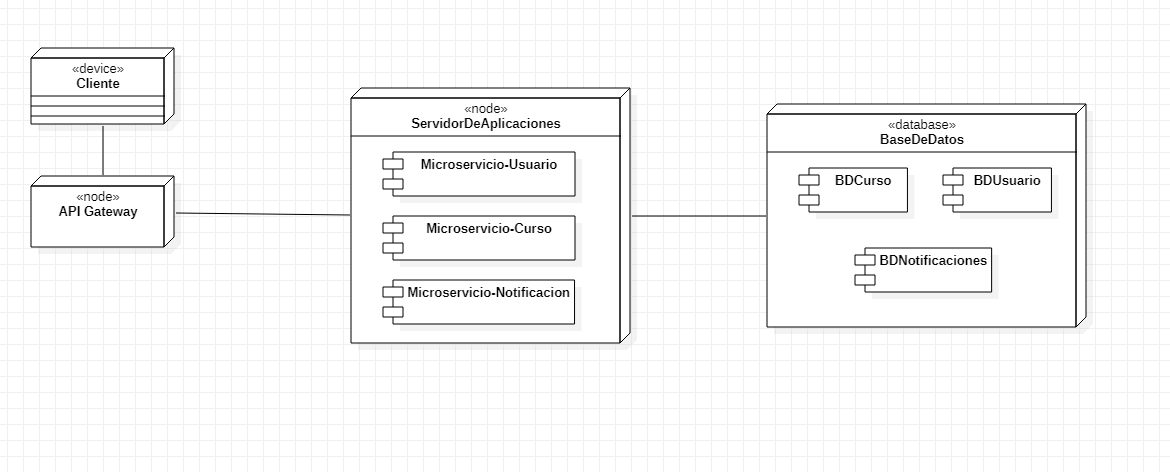
Diagrama de clases cambiado:  


Diagrama de despliegue cambiado:



3.plan de pruebas

-Pruebas Unitarias: Se utilizó JUnit para definir la estructura de los tests y Mockito para simular las dependencias como los repositorios. Esto permite probar la lógica del servicio sin necesidad de conectarse a una base de datos real

Imports:

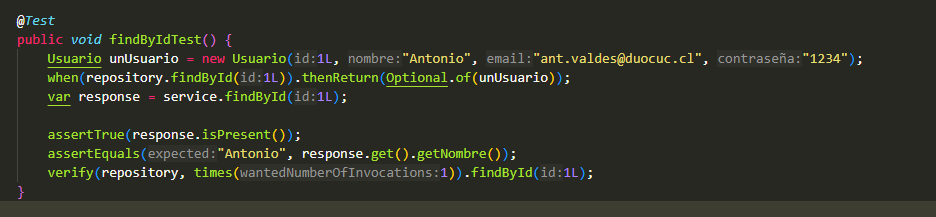


Métodos utilizados:

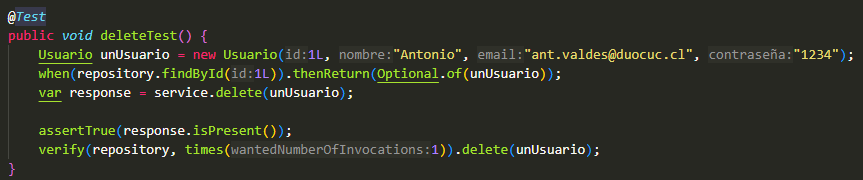
Método findByAllTest()



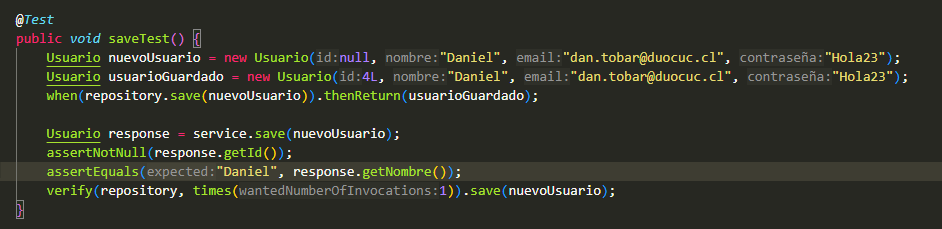
Método findByIDTest()



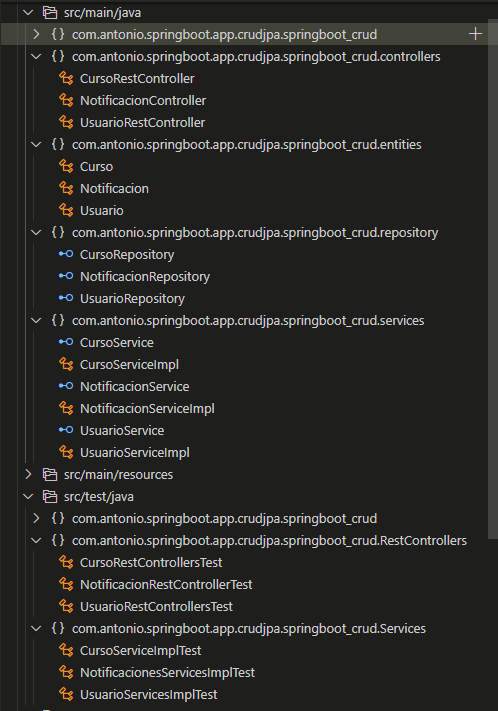
Método deleteTest()



Método saveTest()

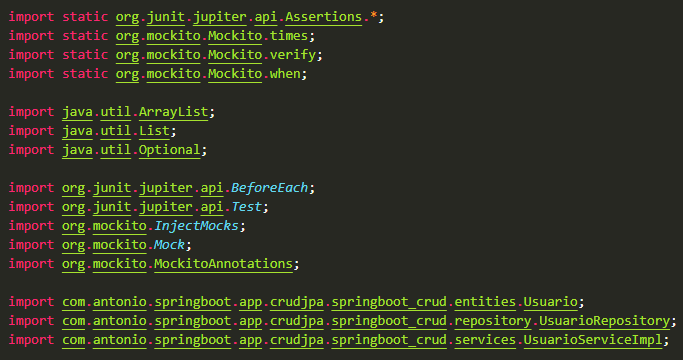


Packages y Clases:



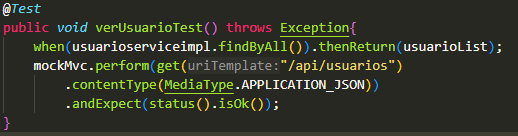
-Pruebas de Integración: MockMvc permite simular peticiones HTTP al controlador sin levantar un servidor completo, lo cual agiliza la verificación del comportamiento de los endpoints REST.

Imports:

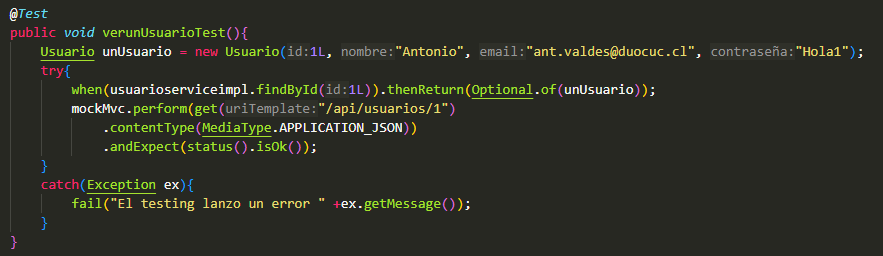


Métodos utilizados:

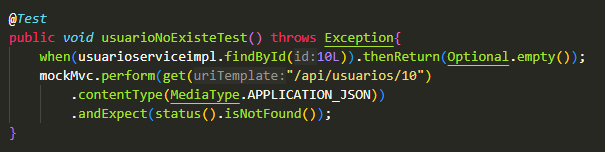
Método verUsuarioTest()



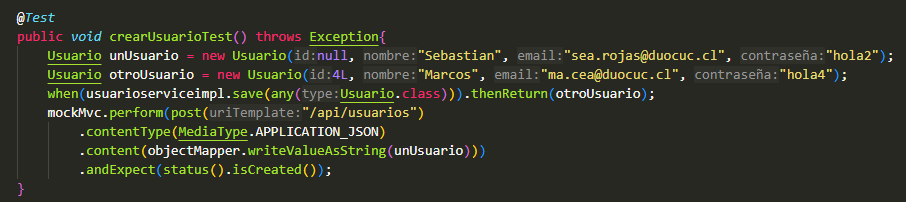
Método verunUsuarioTest()



Método usuarioNoExisteTest()



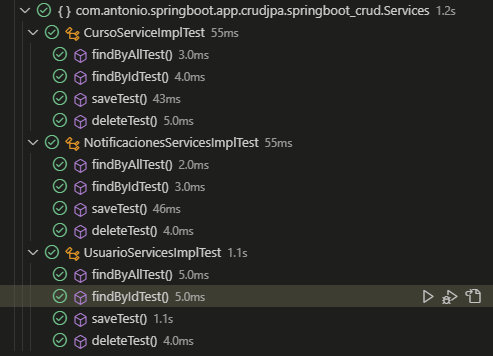
Método crearUsuarioTest()



4.Ejecución de pruebas

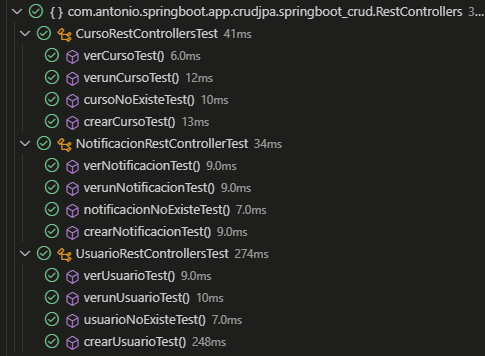
Se hicieron pruebas unitarias con JUnit y Mockito para comprobar que los métodos implementados funcionan bien. También se realizaron pruebas de integración usando @SpringBootTest y MockMvc para asegurar que los endpoints de los RestController respondan correctamente.

Pruebas Unitarias exitosas:



En esta imagen se observa la ejecución de las pruebas unitarias para los packages ServiceImplTest, donde se validan los métodos principales del servicio con resultados exitosos.

Pruebas de integración correctas:



Las pruebas de integración comprobaron que los endpoints de la API respondieran bien. Se usó MockMvc para simular peticiones y se revisaron las respuestas esperadas como 200 OK, 201 Created y 404 Not Found.

5..Git y GitHub

Comandos utilizados en git:

Git init:



Git config user.name:



Git config user.email:



Git config user.password:



Git add .:



Git commit -m:



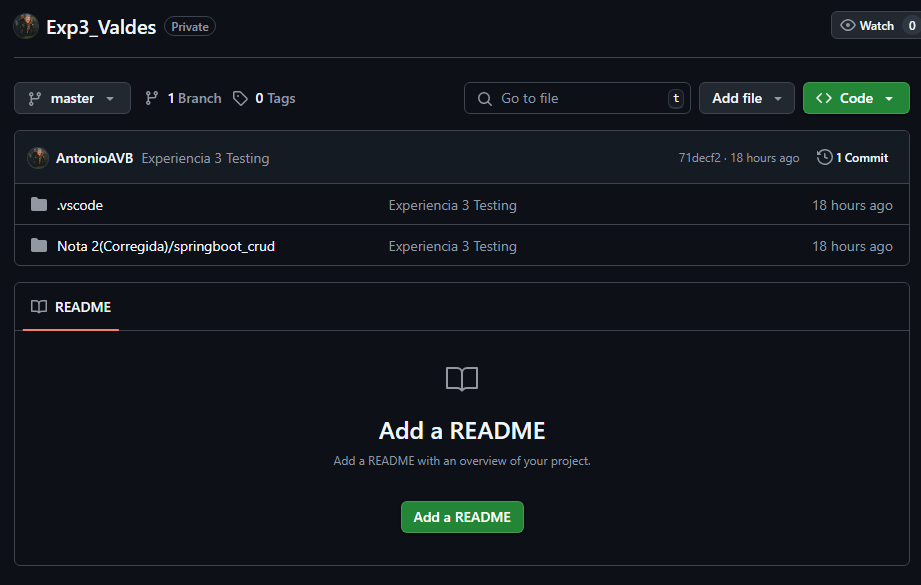
Git remote add origin:



Git push -u origin master:



Repositorio de GitHub:



6.Conclusión

En esta experiencia aprendimos a implementar pruebas unitarias e integración para asegurar que el sistema funcione correctamente. También usamos Swagger para documentar la API y facilitar su uso. Aplicar testing nos ayudó a detectar errores y mejorar la calidad del código. Fue una experiencia útil para prepararnos en el desarrollo de proyectos reales con buenas prácticas.