

**Caso de Estudio (Informe Etapa 3): Microservicios de “EduTech Innovators SPA”**

**Alumnos:**

Franco Pisani

Vicente Riquelme

Ignacio López

**Docente:**

Eduardo Baeza

**Asignatura:**

Desarrollo Fullstack I

26-05-2025

**ÍNDICE**

**1 .Introducción**

**2. Diagrama de Arquitectura de Microservicios**

**3. Plan de Pruebas**

3.1. Pruebas Unitarias

3.2. Pruebas de Integracion

**4. Ejecucion de Pruebas**

**5. Git / Github**

5.1. Comandos utilizados

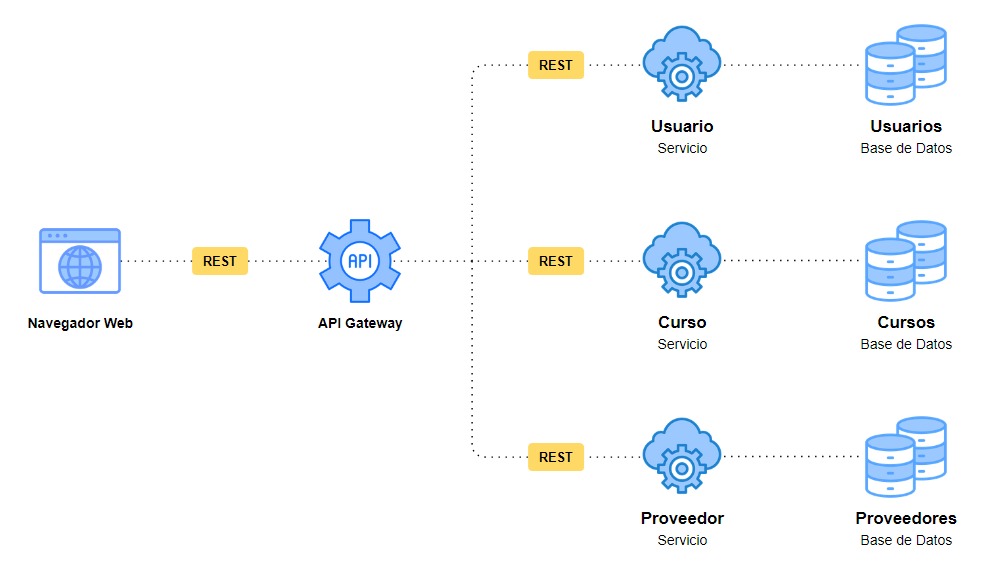
**6. Conclusión**

**1. INTRODUCCIÓN**

Durante el desarrollo de este proyecto, tuvimos la oportunidad de aplicar por testings a distintos componentes del backend, lo que representa algo muy positivo para las buenas prácticas de programacion. Este estuvo centrado en verificar el correcto funcionamiento de un RestController y un Service. A través de la implementación de pruebas unitarias y de integración, fue posible validar el comportamiento esperado de funcionalidades asociadas a tres clases principales: Usuario, Curso y Proveedor.

Esta experiencia permitió comprender de forma práctica la importancia del testing en el ciclo de desarrollo de software, no solo como una herramienta para detectar errores, sino también como un mecanismo para asegurar la calidad, estabilidad y mantenimiento del código. Además, se logró afianzar el uso de frameworks como JUnit y Mockito, aprendiendo a simular comportamientos y verificar respuestas de manera controlada. En definitiva, esta primera aproximación al testing marcó un avance significativo en el proceso de profesionalización del desarrollo, sentando bases sólidas para futuros proyectos.

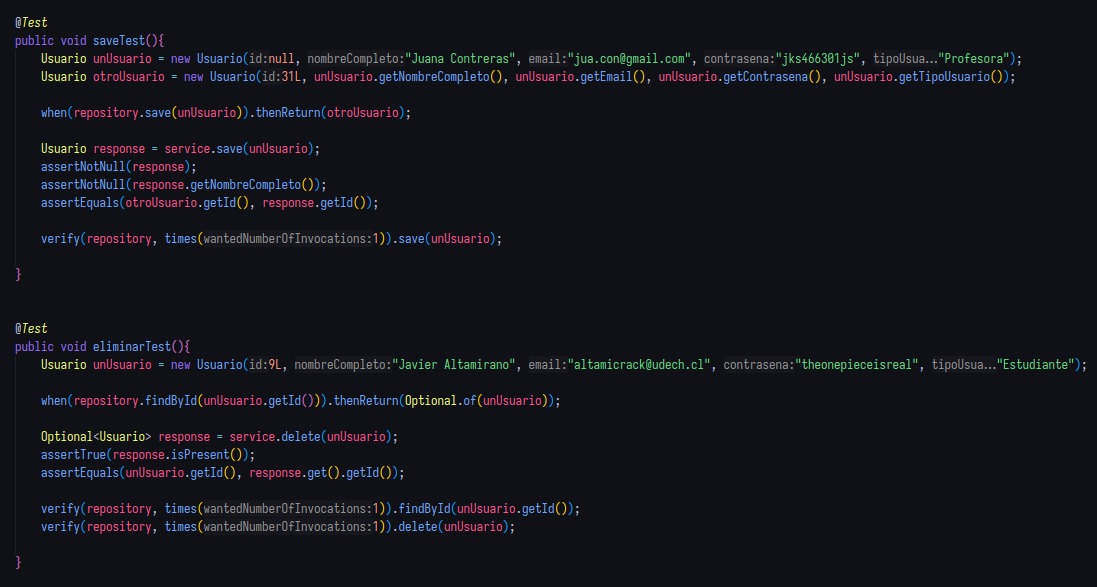
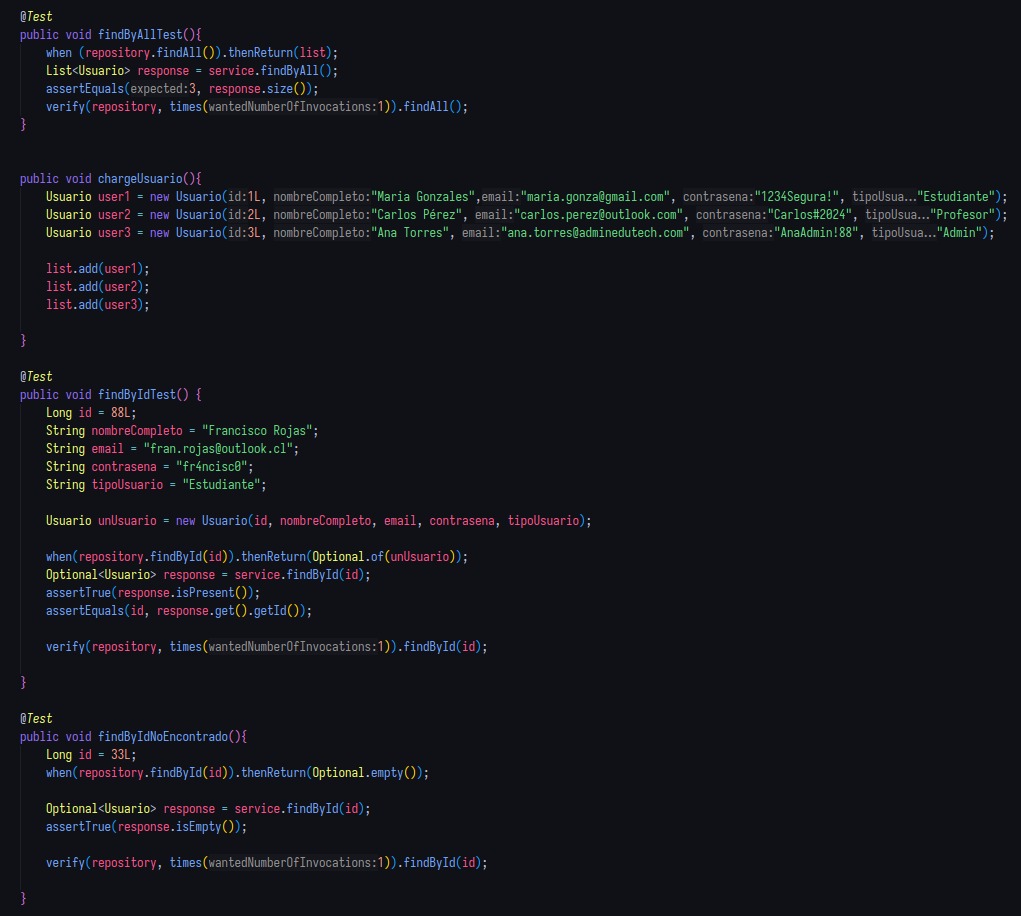
**2. DIAGRAMA DE ARQUITECTURA DE MICROSERVICIOS**

****

**3. PLAN DE PRUEBAS**

**3.1. PRUEBAS UNITARIAS**

**USUARIO**



**CURSO**

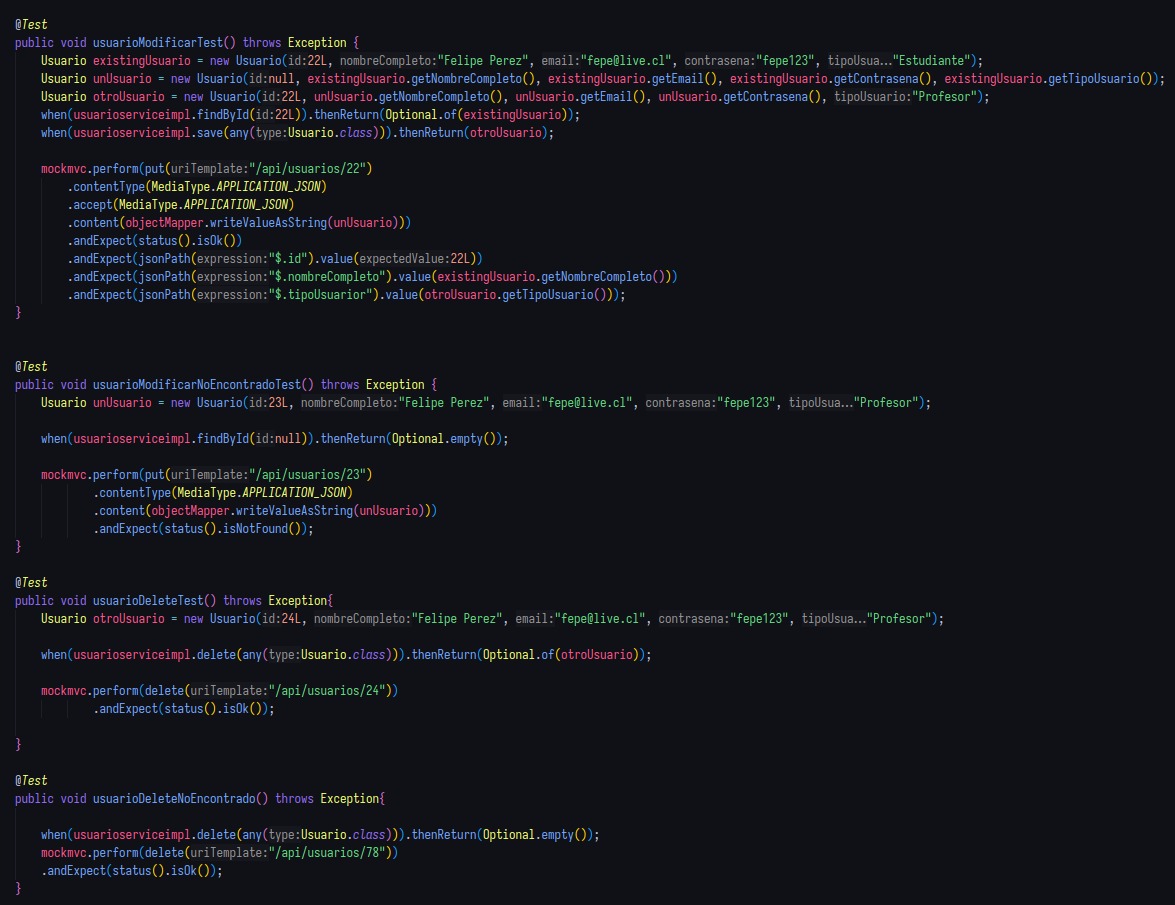
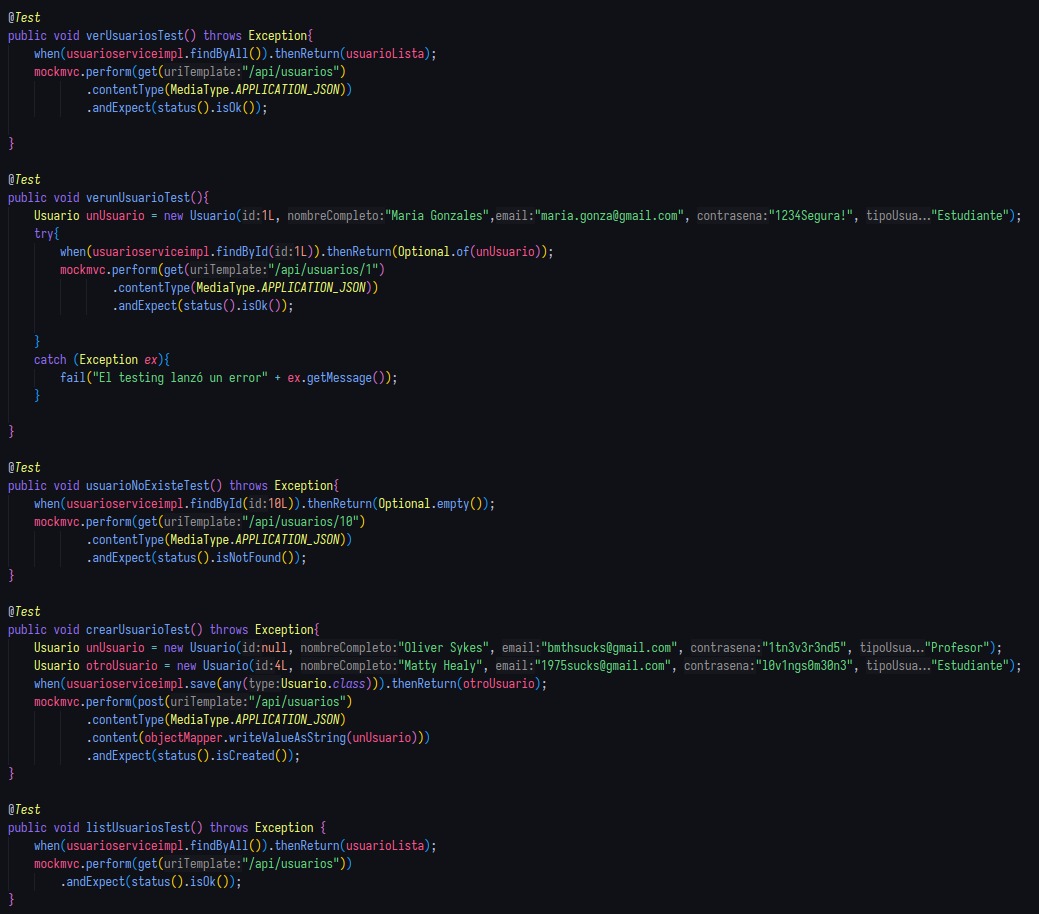


**PROVEEDOR**

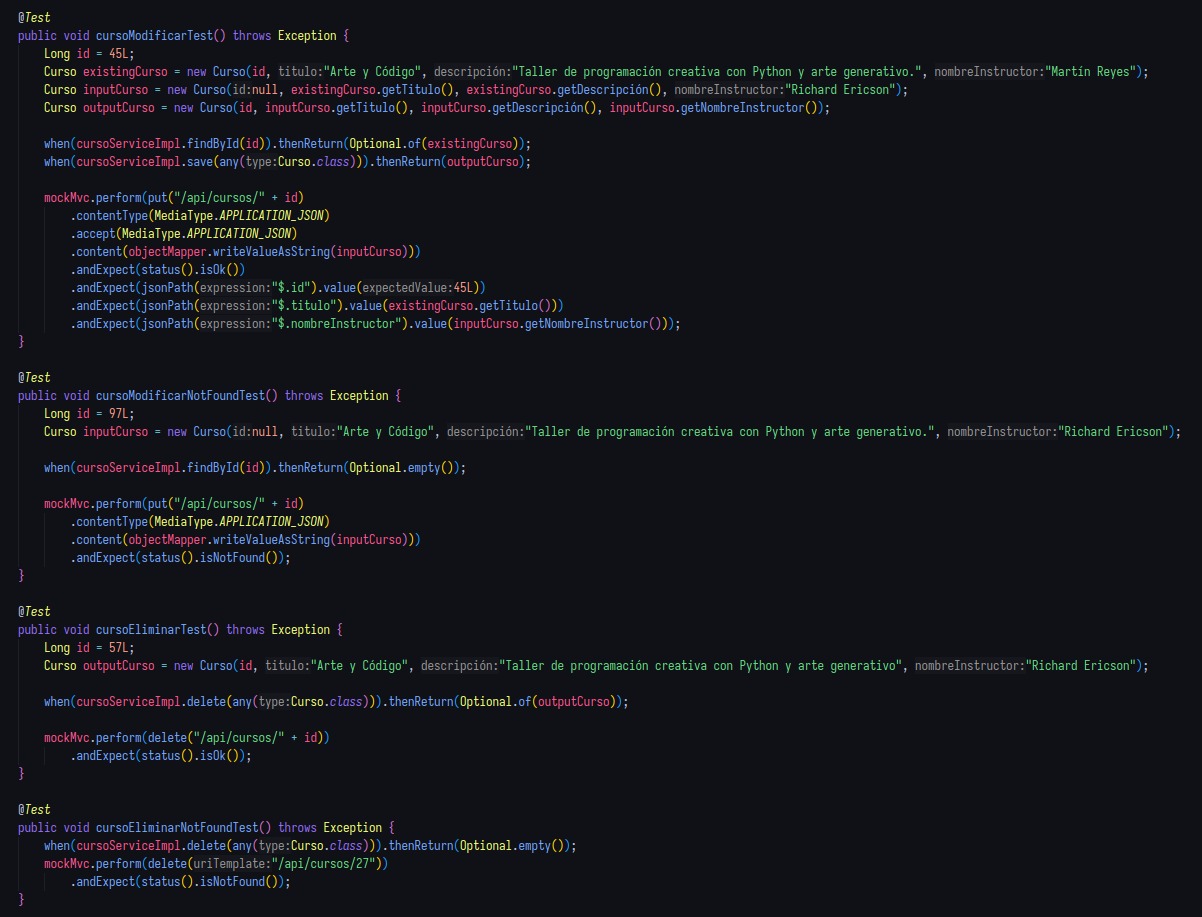


**3.2. PRUEBAS DE INTEGRACION**

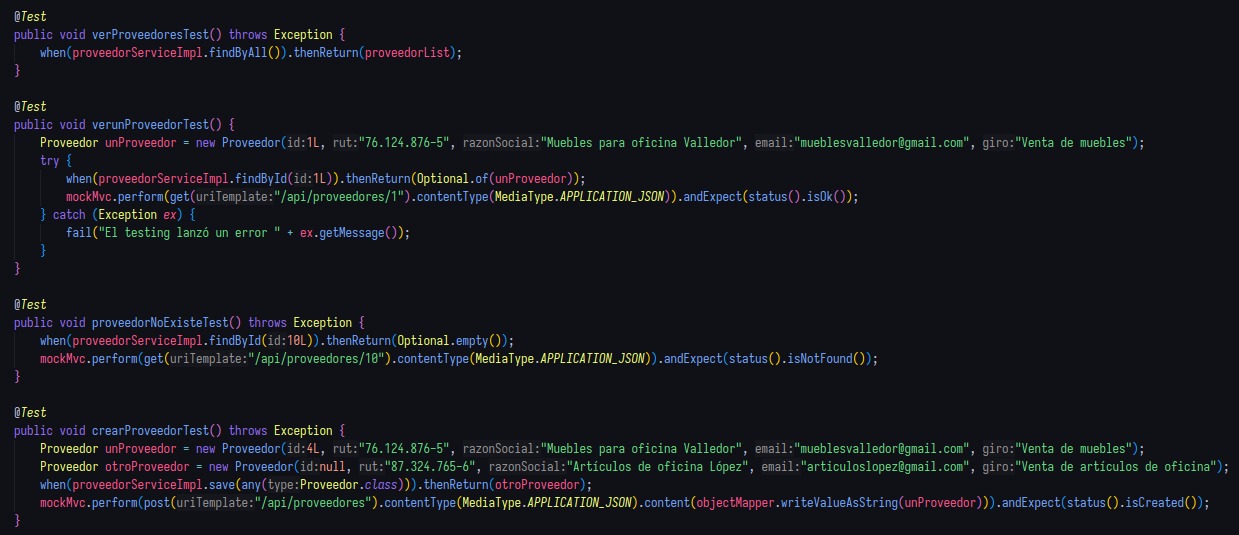
**USUARIO**



**CURSO**

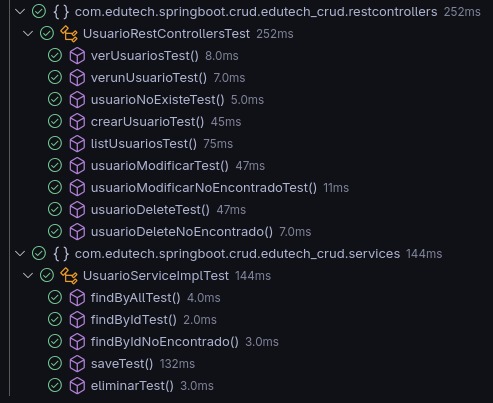


**PROVEEDOR**



**4. EJECUCION DE PRUEBAS**

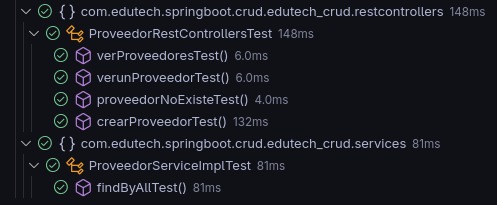
**USUARIO**



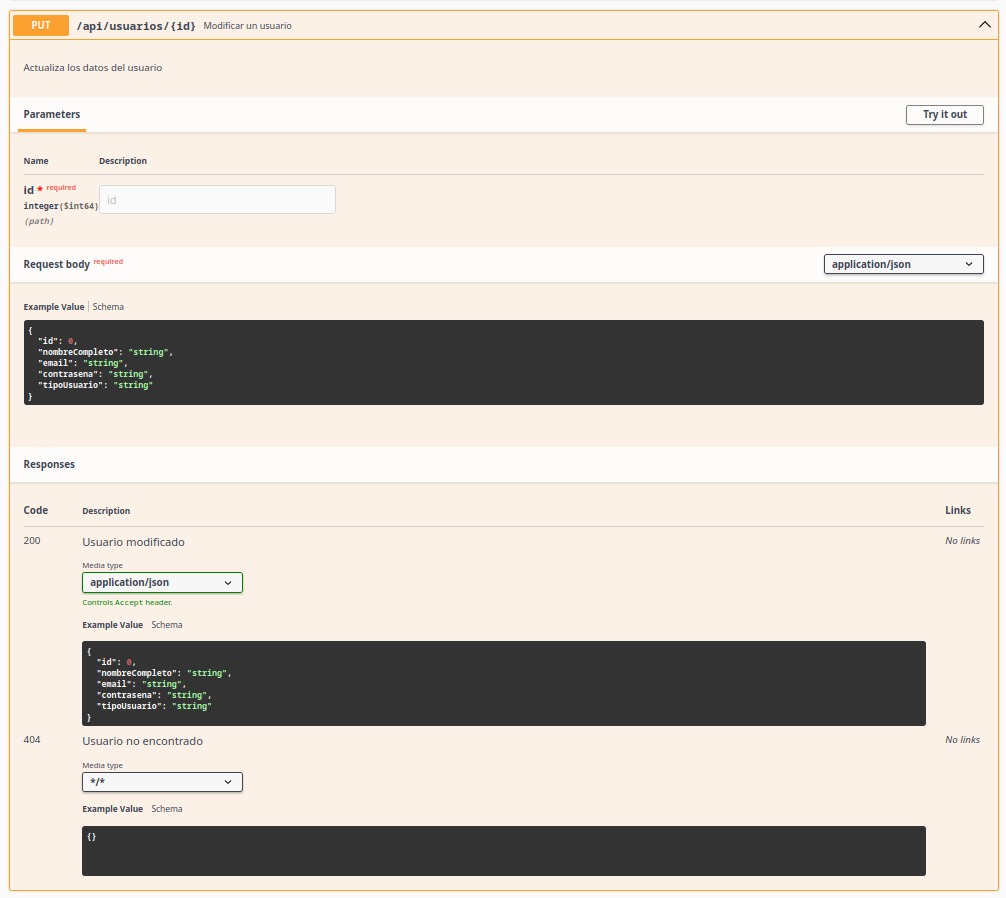
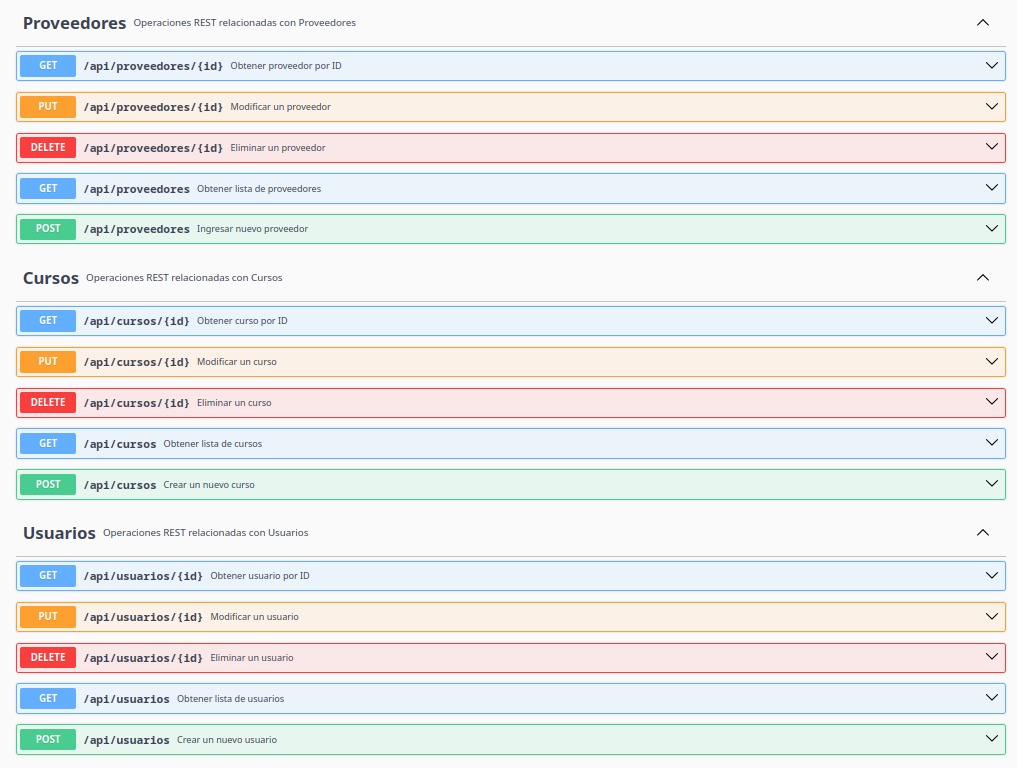
**CURSO**



**PROVEEDOR**



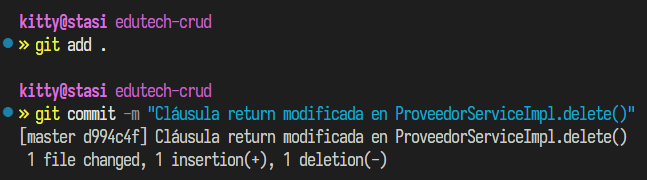
**OPEN API**



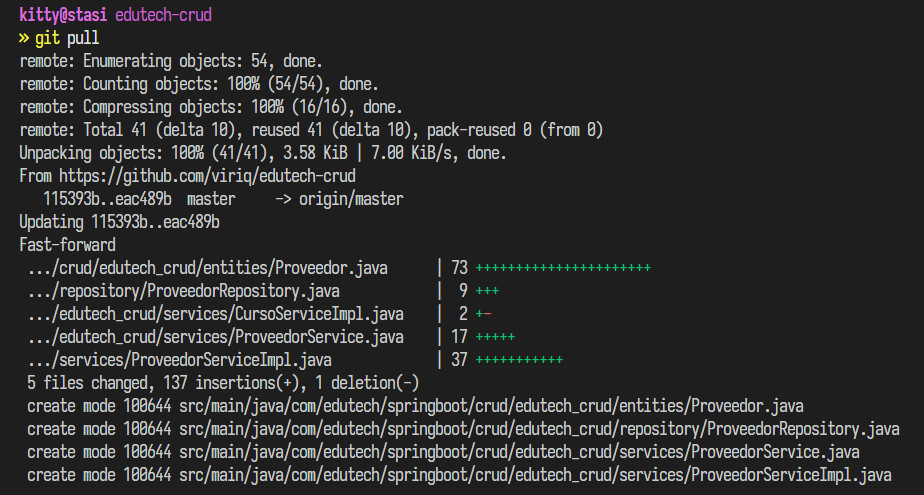
**5. GIT / GITHUB**

**5.1. Comandos utilizados**

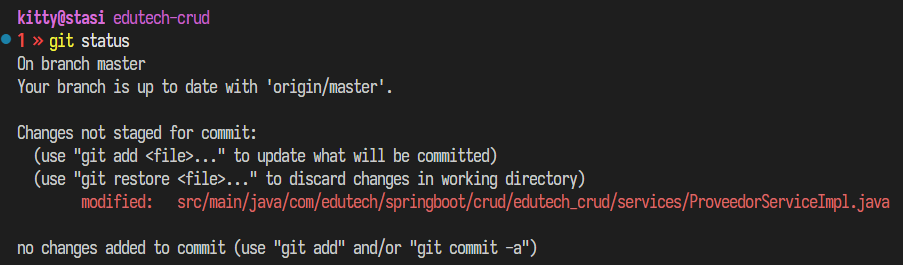
GIT ADD COMMIT



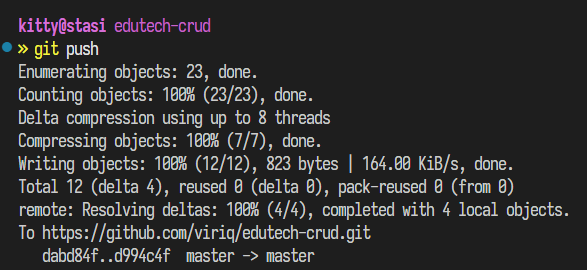
GIT PULL



GIT STATUS



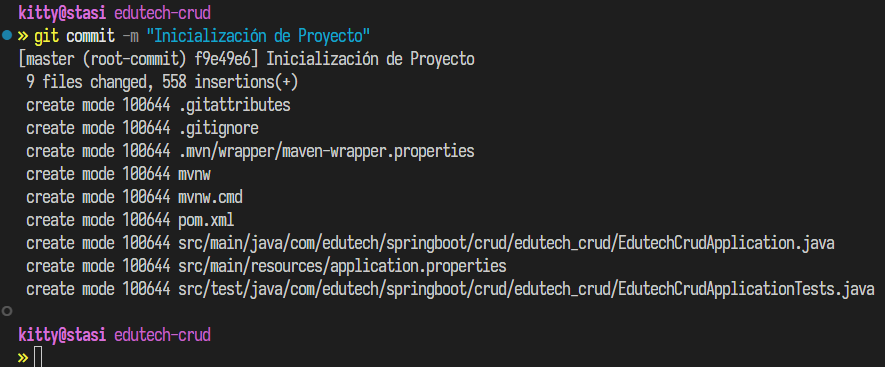
GIT PUSH



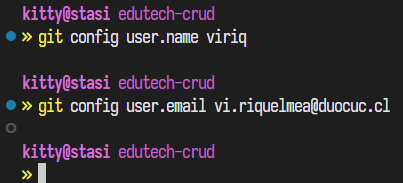
INIT GIT ADD



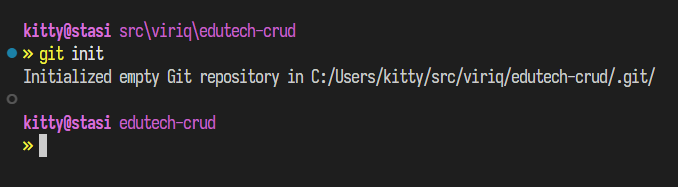
INIT GIT COMMIT



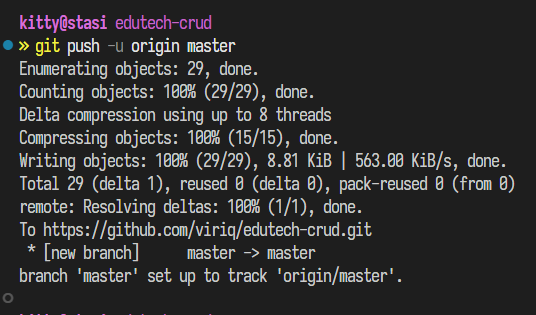
INIT GIT CONFIG



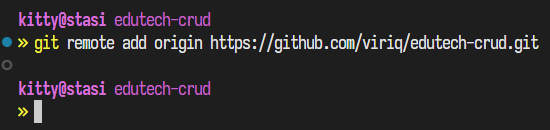
INIT GIT INIT



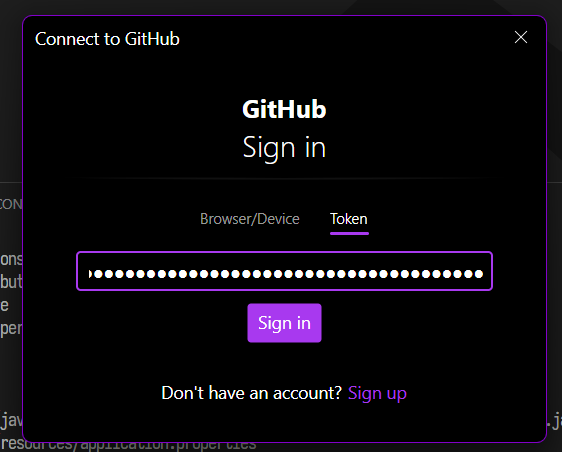
INIT GIT PUSH U



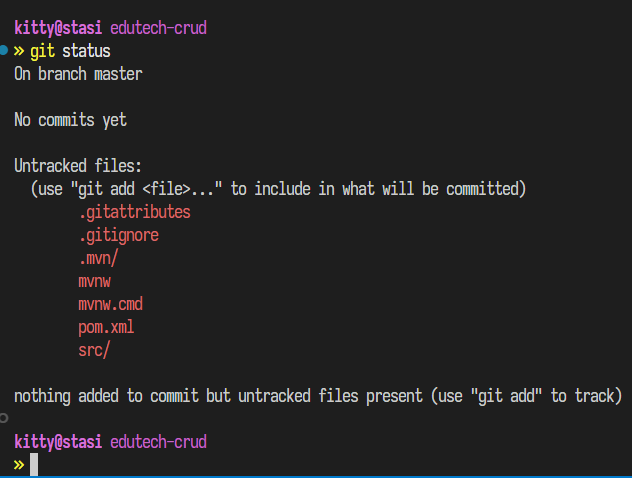
INIT GIT REMOTE ADD ORIGIN



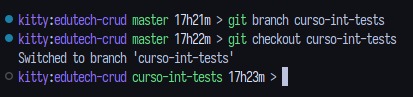
INIT GIT SIGNIN TOKEN



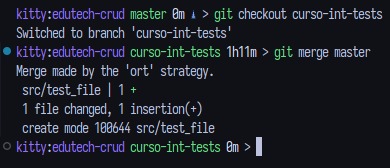
INIT GIT STATUS



GIT CHECKOUT Y GIT BRANCH



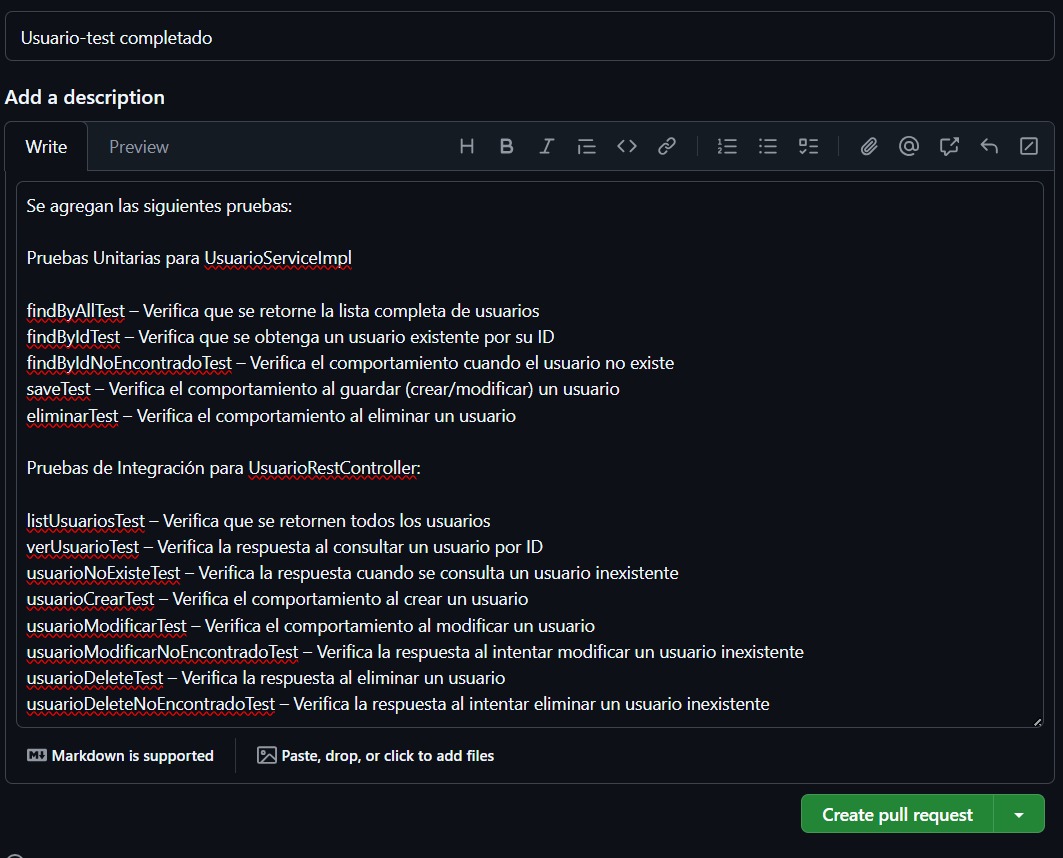
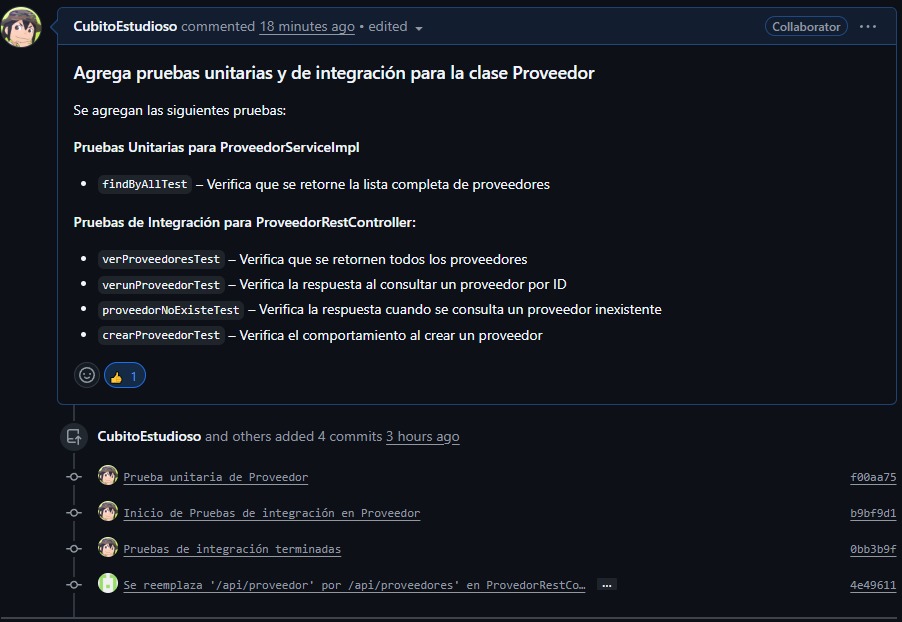
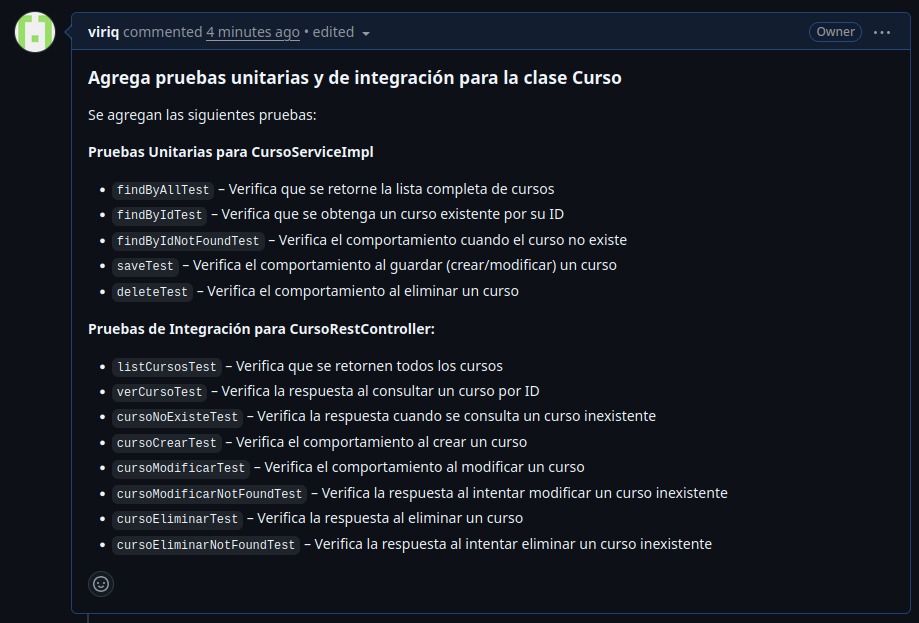
GIT MERGE



GIT RESET



PULL REQUESTS



**CONCLUSIÓN**

Para esta tercera fase del proyecto, la implementación del testing en este proyecto resultó ser una acción fundamental para asegurar la calidad y el funcionamiento correcto de los metodos desarrollados. Con testings aplicados en los distintos RestControllers y Services de las clases Usuario, Curso y Proveedor, se logra encontrar errores y se puede ver el correcto comportamiento del sistema de forma anticipada y oportuna.

Esta ultima práctica no solo mejoró la confiabilidad del código, sino que también ayudó al momento de mantener y hacer escalable el proyecto, en caso de futuras mejoras o adiciones.

En resumen, la incorporación de pruebas durante el desarrollo fortaleció significativamente la calidad del software entregado, evidenciando la importancia de integrar buenas prácticas de testing desde las primeras etapas del ciclo de vida del proyecto.