

Segundo Entrega



Universidad
del Cauca

Vigilada Mineducación

Ingeniería de Software II

Presentado por:

Jose David Arteaga Fernández

Juan Camilo Benavides Salazar

Juan Esteban Chavez Collazos

Jhoan Sebastian Garcia Camacho

Juan Diego Perez Martinez

Profesor:

Wilson Libardo Pantoja

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería de Sistemas

Popayán, Febrero de 2025

CONTENIDO

1. Requisitos funcionales.....	1
2. Tablero de tareas Sprint 3 (pantallazo de Trello o Jira).....	1
3. Escenario de calidad de escalabilidad.....	1
Contexto:.....	1
Estímulo:.....	2
Respuesta:.....	2
Microservicio síncrono:.....	2
Microservicios asíncronos:.....	2
Medición de Calidad:.....	3
Tiempo de Respuesta:.....	3
Escalabilidad Horizontal:.....	3
Disponibilidad:.....	3
Resultado Esperado:.....	3
4. Documentación de la arquitectura de software, usando el modelo C4 (versión microservicios).....	5
Diagrama de Contexto:.....	5
Diagrama de Contenedores:.....	6
Diagrama de componentes (MicroUsuario):.....	7
Diagrama de componentes (MicroEmpresa):.....	7
Diagrama de Clases MicroUsuario:.....	8
5. Diagrama de contextos de la solución(Bounded Context).....	9
6. URL del repositorio en GitHub.....	9
7. URL del video(nuevo).....	9

Resumen

El Sistema de Gestión de Proyectos Académicos para Empresas es una plataforma web diseñada para fortalecer la vinculación entre la universidad y el sector empresarial. Su objetivo es facilitar la gestión y asignación de proyectos empresariales a estudiantes de últimos semestres, permitiéndoles desarrollar soluciones de software reales como parte de sus prácticas profesionales, pasantías o proyectos académicos.

La plataforma permite a las empresas registrarse, publicar sus necesidades de software y seguir el estado de sus solicitudes. Por su parte, el coordinador del programa evalúa y asigna los proyectos a estudiantes, asegurando que estos se alineen con el currículo académico. Los estudiantes, a su vez, pueden postularse a proyectos de su interés.

Este sistema no solo optimiza el proceso de asignación de proyectos, sino que también garantiza que los estudiantes adquieran experiencia práctica en el desarrollo de soluciones tecnológicas con impacto real en el sector empresarial.

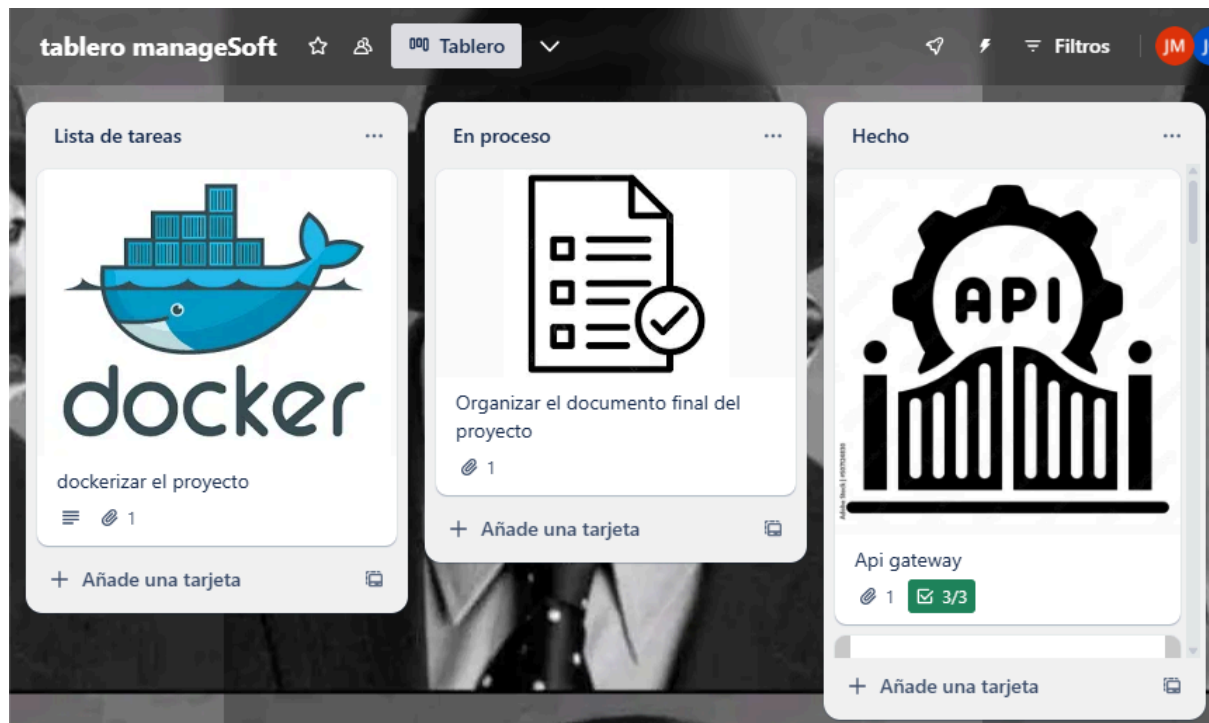
1. Requisitos funcionales

Los requisitos funcionales están descritos en la siguiente hoja de cálculo que muestra las épicas, las historias de usuario y los criterios de aceptación.

Link: [Requisitos Funcionales](#)

2. Tablero de tareas Sprint 3 (pantallazo de Trello o Jira)

Link: [Trello](#)



3. Escenario de calidad de escalabilidad

Contexto:

El Sistema de Gestión de Proyectos Académicos para Empresas ha migrado a una arquitectura de microservicios, debido a su creciente adopción por universidades y empresas. Este cambio se llevó a cabo para resolver los

problemas de escalabilidad derivados del aumento en la cantidad de usuarios concurrentes, especialmente durante períodos de alta demanda como el inicio del semestre académico o las convocatorias masivas de proyectos. La cantidad de usuarios, incluyendo estudiantes, coordinadores y empresas, genera picos de actividad que ponen presión sobre los servicios clave, como los servicios de Proyectos y Notificaciones. Sin una estrategia adecuada de escalabilidad, estos picos podrían afectar la experiencia del usuario, resultando en tiempos de respuesta más lentos o incluso en caídas del servicio.

Estímulo:

Durante los picos de actividad, como al inicio del semestre académico o durante la publicación de nuevas convocatorias por parte de empresas, el número de solicitudes aumenta drásticamente. Este incremento genera una alta demanda en los microservicios de Proyectos y Notificaciones, lo que podría provocar una sobrecarga y tiempos de espera prolongados si no se gestionan adecuadamente.

Respuesta:

Para resolver los problemas de escalabilidad, el sistema ha adoptado una arquitectura de microservicios tanto síncronos como asíncronos, permitiendo que los servicios se escalen de manera independiente según la carga:

Microservicio síncrono:

El microservicio que gestiona el registro de usuarios y el inicio de sesión se ha implementado como un servicio síncrono. Este microservicio es fundamental para que los usuarios puedan ingresar al sistema y acceder a sus funcionalidades. Está diseñado para escalar horizontalmente, añadiendo réplicas automáticamente según la carga de tráfico en los picos de acceso. Al ser un servicio síncrono, garantiza que las solicitudes de inicio de sesión sean atendidas rápidamente, proporcionando una experiencia fluida al usuario.

Microservicios asíncronos:

Los servicios de notificaciones y otros procesos secundarios, como las actualizaciones de estado de las postulaciones, se gestionan de manera asíncrona utilizando RabbitMQ. Este enfoque asegura que los picos de solicitudes de notificación no afecten el rendimiento de los servicios principales, procesando las tareas de manera en segundo plano sin impactar la experiencia del usuario.

La escalabilidad de cada microservicio se gestiona mediante la escalabilidad horizontal, lo que permite añadir instancias de los servicios según la carga, garantizando que el sistema pueda adaptarse dinámicamente a los cambios en la demanda sin comprometer la estabilidad del sistema.

Medición de Calidad:

La efectividad de la escalabilidad se medirá en función de los siguientes criterios:

Tiempo de Respuesta:

El sistema debe mantener un tiempo de respuesta promedio por debajo de los 2 segundos para el 95% de las solicitudes, incluso durante los picos de tráfico.

Escalabilidad Horizontal:

El sistema debe poder aumentar entre 3 y 10 réplicas de cada microservicio dentro de un lapso de 3 minutos cuando se detecte un aumento en la carga.

Disponibilidad:

El sistema debe mantener una disponibilidad del 99.9%, asegurando que los usuarios puedan acceder a los servicios sin interrupciones significativas, incluso en períodos de alta demanda.

Resultado Esperado:

Gracias a la migración a una arquitectura basada en microservicios síncronos y asíncronos, el sistema será capaz de manejar de manera eficiente los picos de tráfico sin degradar la experiencia del usuario. Los estudiantes podrán postularse a proyectos y consultar el estado de sus aplicaciones en tiempo real, mientras que las empresas podrán publicar nuevos proyectos sin inconvenientes. Las notificaciones se entregarán con una latencia mínima, incluso en los momentos de mayor demanda. Esta solución no solo garantiza una escalabilidad óptima, sino que también permite un crecimiento sostenible y eficiente del sistema con una mayor flexibilidad operativa

4. Documentación de la arquitectura de software, usando el modelo C4 (versión microservicios).

Diagrama de Contexto:

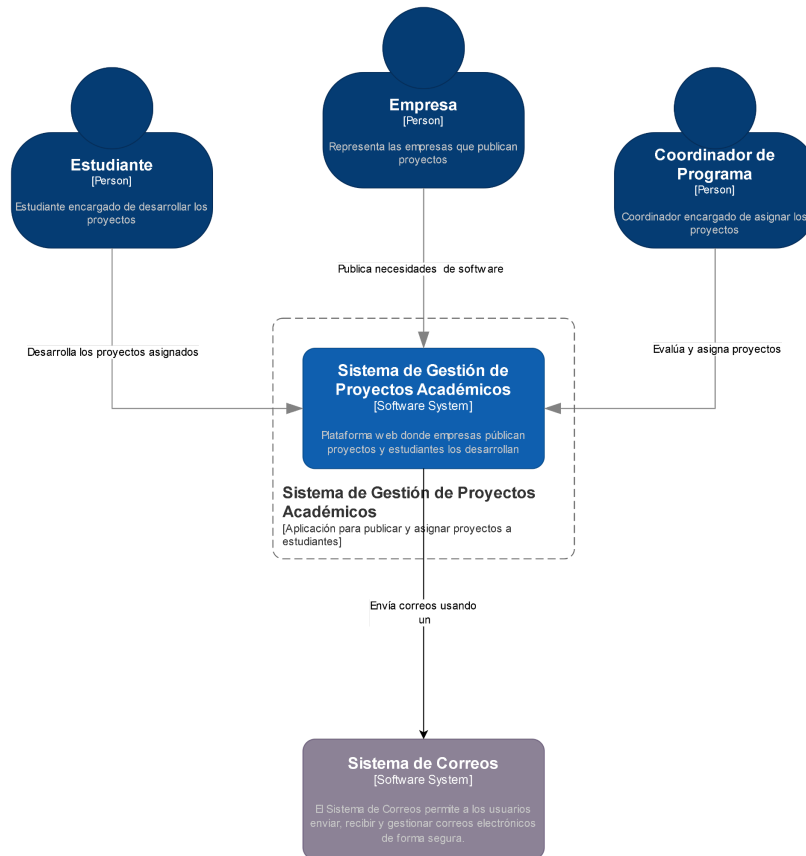


Diagrama de Contenedores:

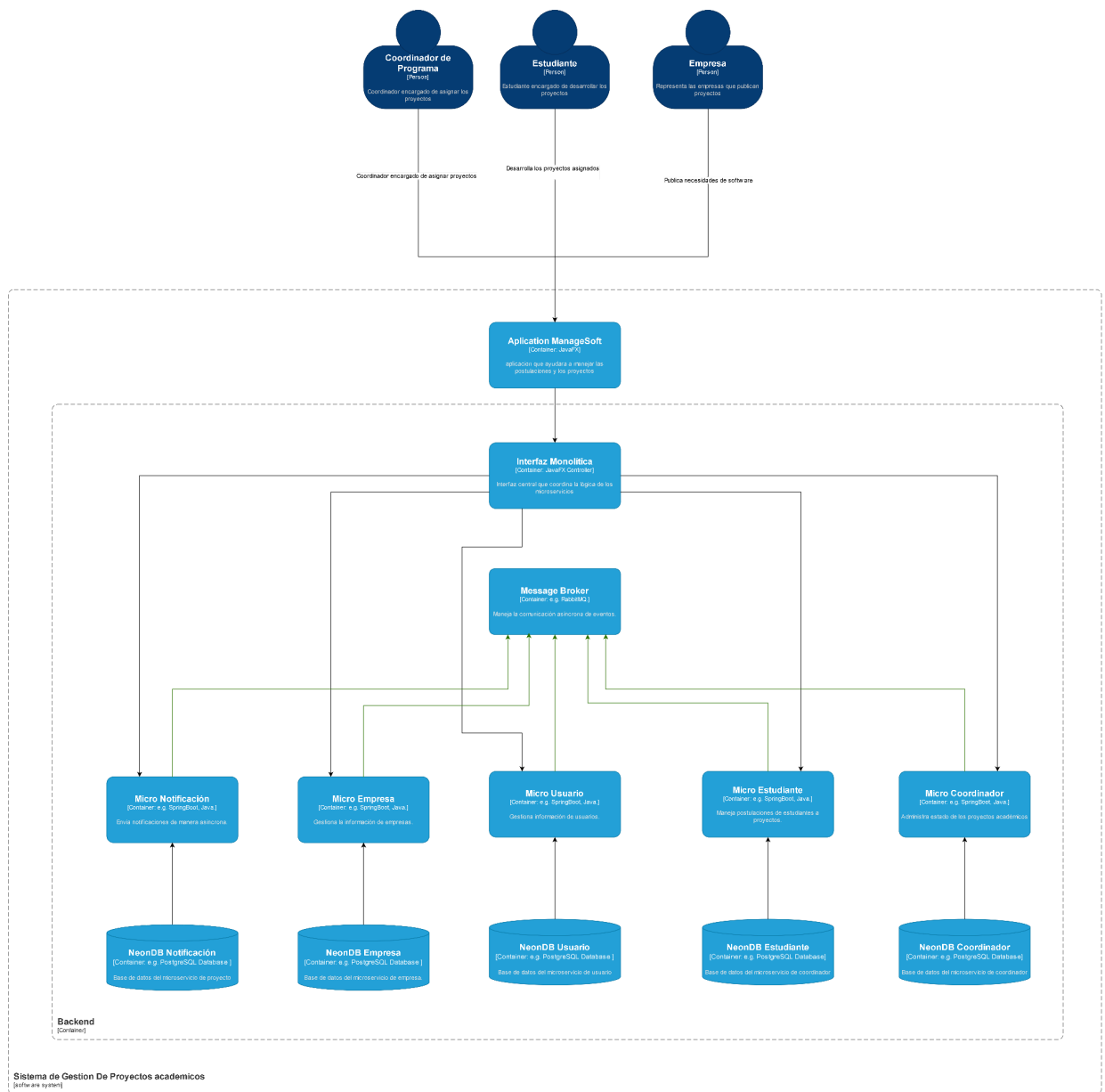


Diagrama de componentes (MicroUsuario):

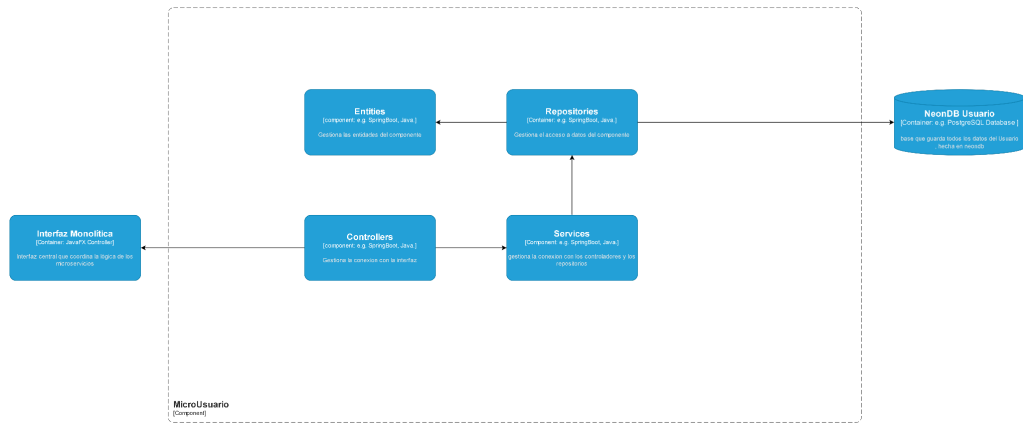


Diagrama de componentes (MicroEmpresa):

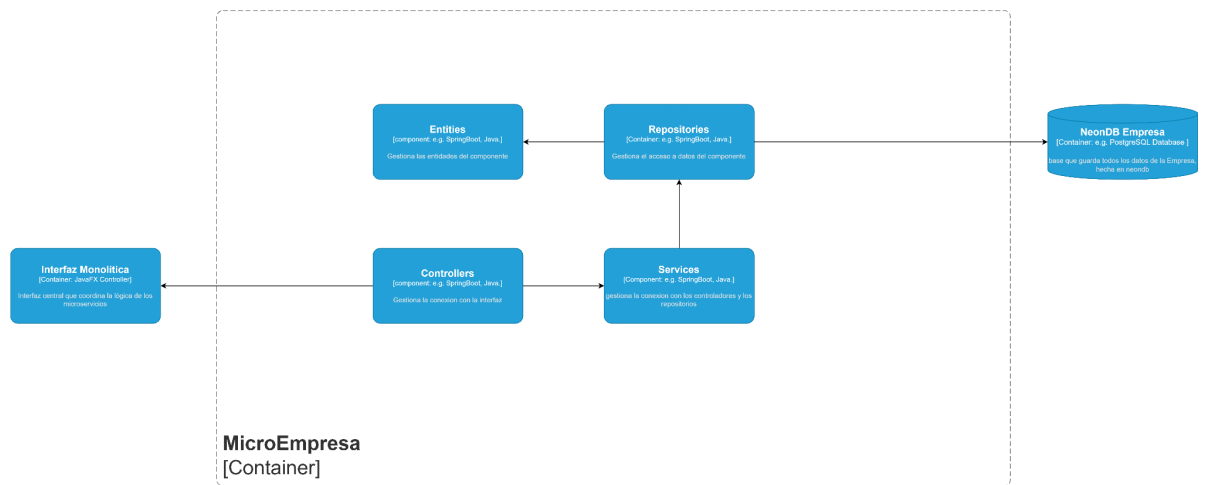
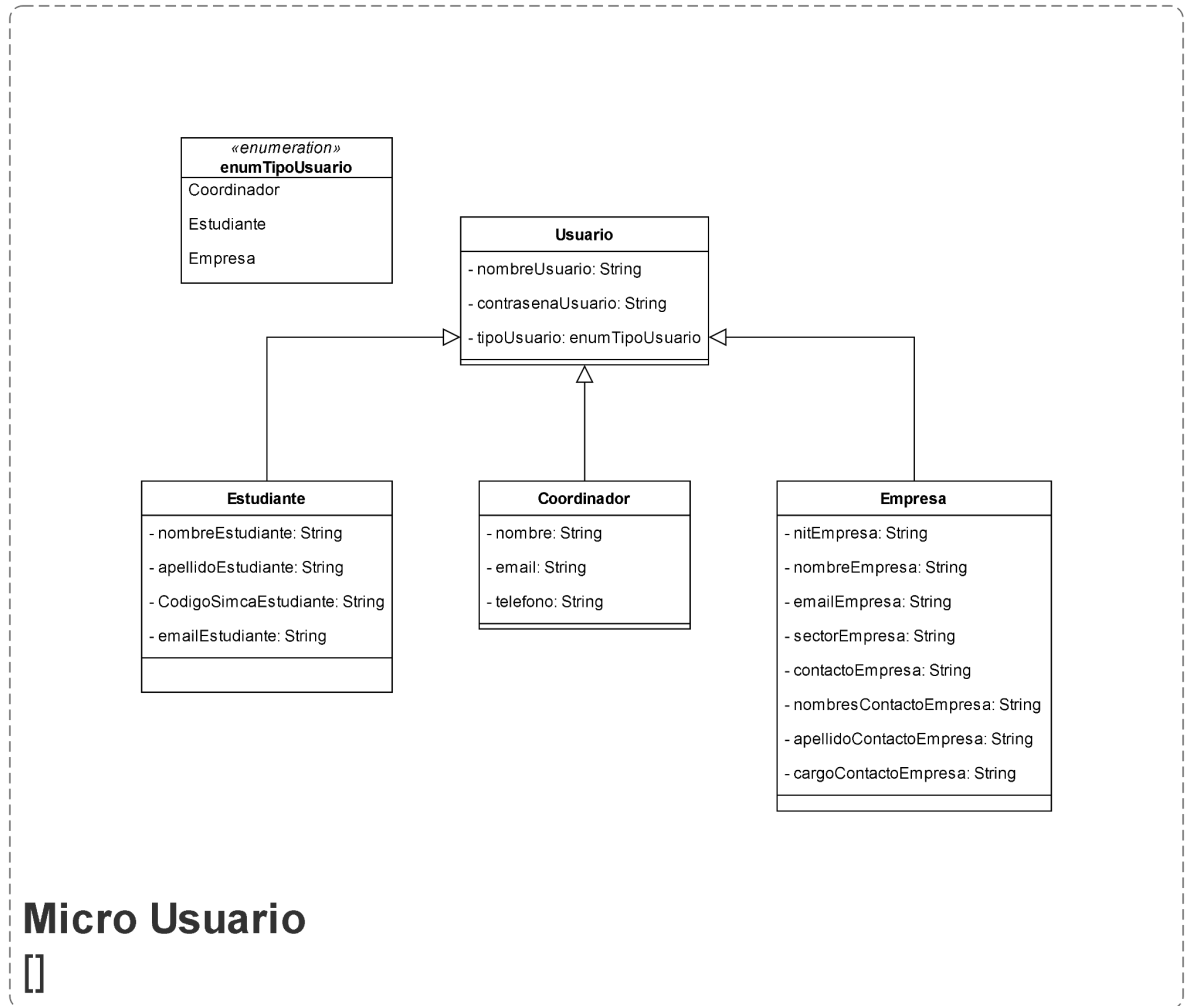
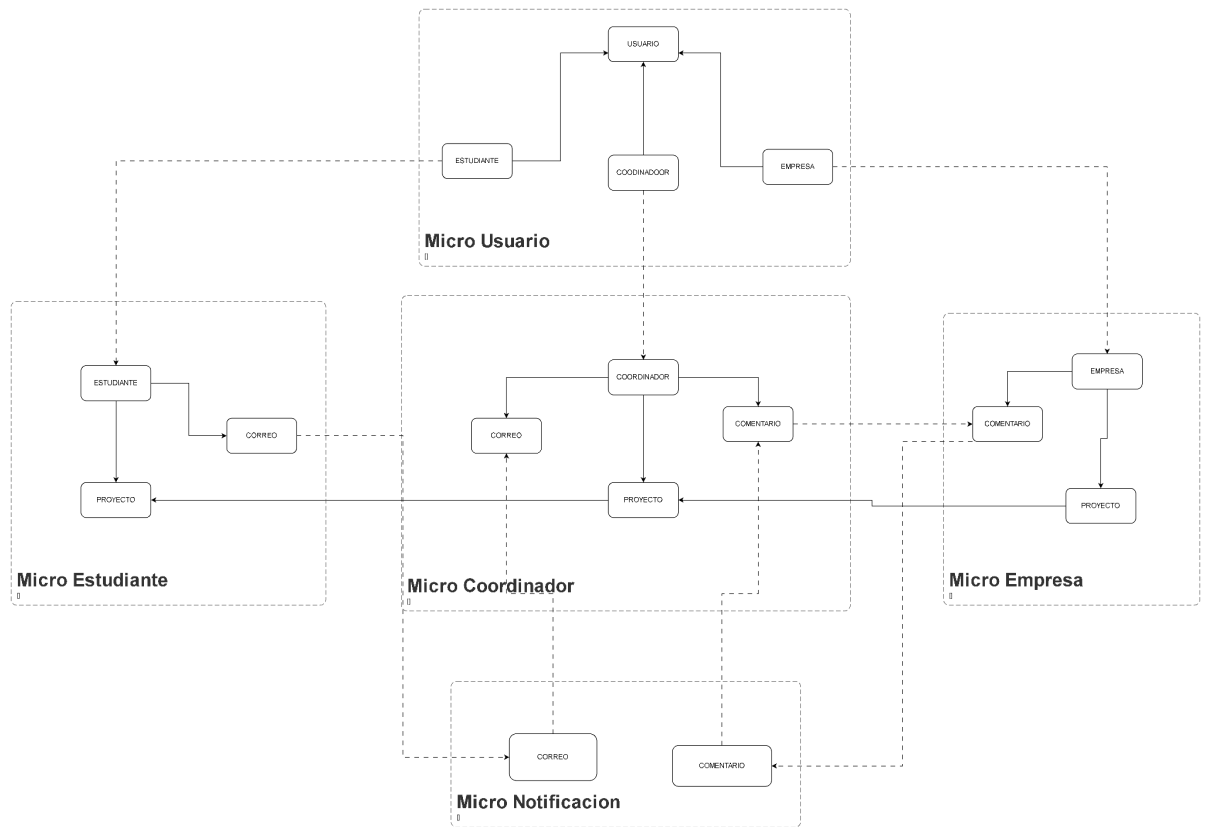


Diagrama de Clases MicroUsuario:



5. Diagrama de contextos de la solución(Bounded Context).



6. URL del repositorio en GitHub

Link: <https://github.com/StreetRogue/ManageSoft>

7. URL del video(nuevo)

Link: <https://youtu.be/zH-mTkClh1o>

Referencias

Anaya, R. (2006). Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software. *REVISTA Universidad EAFIT*, 42(141).

CASTAÑO, J. F. (2021). Métricas en la evaluación de la calidad del software: una revisión conceptual. *Computer and Electronic Sciences: Theory and Applications*, 2(2). <https://doi.org/10.17981/cesta.02.02.2021.03>

Gordón Graell, R. D. (2023). INGENIERÍA DE SOFTWARE: *Revista FAECO Sapiens*, 6(2). <https://doi.org/10.48204/j.faeco.v6n2.a4014>

Krishna Madasu, V., Venkata Swamy Naidu Venna, T., & Eltaeib, T. (2015). SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, 2(2).

Morales Vargas, A., & Codina, L. (2019). Atributos de calidad web para repositorios de datos de investigación en universidades. *Hipertext.Net*, 19. <https://doi.org/10.31009/hipertext.net.2019.i19.04>