Actividad Autónoma Nro. 001

1. Datos Generales

Asignatura	Desarrollo Basado en Plataformas
Ciclo	5 A
Unidad	1
Nombre del Docente	Edison Leonardo Coronel Romero
Integrantes	Eberson Guayllas Wilmer Pardo Santiago Tamayo
Fecha	viernes 17 de octubre

2. Tema

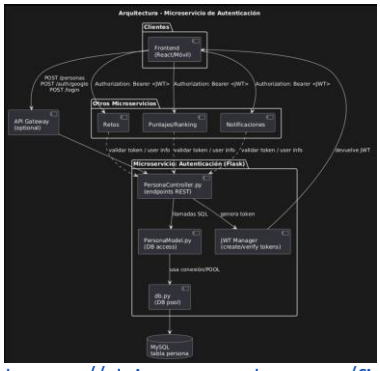
Documentación del desarrollo de microservicios y su integración mediante herramientas colaborativas y APIs.

3. Propósito

El estudiante consolida lo aprendido sobre la arquitectura de microservicios, su integración mediante API REST, y el uso de herramientas colaborativas para la gestión y documentación técnica del proyecto.
Como resultado, elaborará un documento técnico que refleje las decisiones de diseño, herramientas utilizadas y el impacto de los microservicios en la escalabilidad y mantenibilidad del sistema.

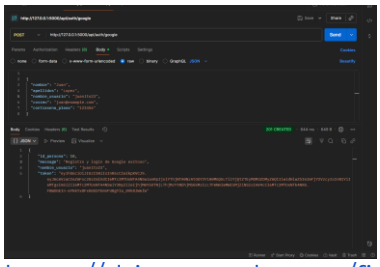
4. Desarrollo

1. *Resumen del microservicio creado*

Nombre del microservicio	Propósito y relación con el sistema principal	Diagrama de sus estructura
Autenticación de usuarios	El propósito de este servicio es otorgar a los usuarios un token por medio de JWT que valide que ya se han registrado en la aplicación y por tanto pueden acceder a las rutas con los decoradores designados,	 https://drive.google.com/file/d/1Zmt3RYYknLILgiSdE

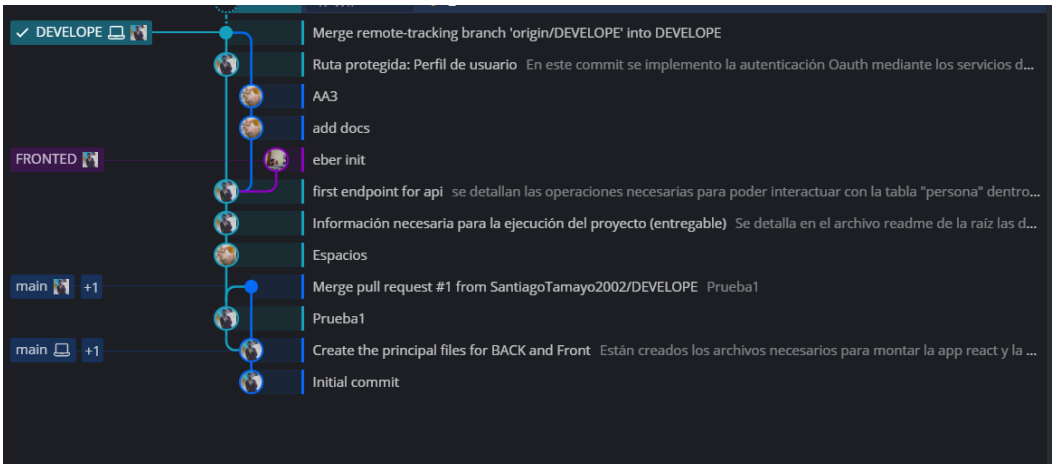
		otZvfSgx4wfR0x7/view?usp=drive_link

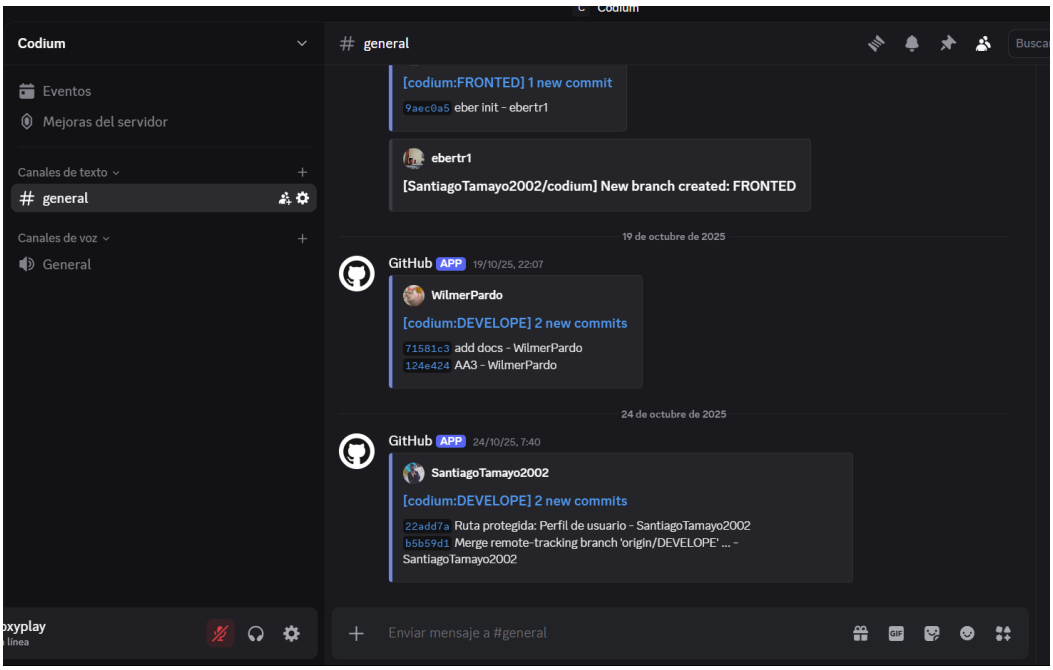
2. *Integración mediante APIs*

Nombre del microservicio	Descripción del método de comunicación	Ejemplos de llamadas y respuestas con Postman
Autenticación de usuarios	La comunicación que interviene en este microservicio es de tipo REST API síncrono, ya que estamos trabajando con arquitecturas cliente-servidor. Se envía una petición por parte del cliente haciendo uso del protocolo HTTP. Esta petición es un archivo que contiene la información de nombre, correo y contraseña del usuario. Cuando esta solicitud llega al endpoint. Localhost/api/auth/google . el procesador desempaqueta la información proveniente del cliente y valida si el usuario enviado existe en la base de datos, si es así genera un token con ID igual al usuario con ese ID en la base de datos, si no existe lo crea asignando todos los datos que llegaron por parte del cliente y de la misma manera le genera un token por medio de JWT, la respuesta que obtiene nuestro cliente es su ID con el que ha sido guardado, el token, y la respuesta HTTP 201	 https://drive.google.com/file/d/1Zmt3RYYknLILgiSdEotZvfSgx4wfR0x7/view?usp=sharing

3. *Herramientas colaborativas utilizadas*
    - Describir el uso de Trello, Taiga, GitHub Projects, Discord u otras herramientas para la coordinación del equipo.

      En las herramientas colaborativas que utilizamos tenemos varias. Por parte del desarrollo hemos utilizado el sistema de repositorios de GitHub y GitKraken para poder llevar un correcto manejo de las versiones de nuestro sistema y poder trabajar en múltiples funcionalidades del sistema al mismo tiempo sin que el flujo de trabajo de uno de los compañeros interfiera con el de otros. para poder tener una robustez de comunicación de los cambios que cada compañero hacía en el código conectamos la plataforma de Discord para estar al tanto de los commits que hacía cada uno de los miembros del equipo.





4. *Buenas prácticas y aprendizajes*

- Control de versiones y trabajo colaborativo en GitHub.- Se ha trabajado con ayuda de un flujo organizado con GitHub y Git Kraken, esta práctica permite tener un historial de versiones que facilita el desarrollo de cambios y ayuda a evitar conflictos mediante las diferentes ramas del proyecto. Gracias a ello, cada integrante puede desarrollar módulos independientes, sin interferir en el resto del trabajo, fomentando la autonomía de los avances.
- Comunicación constante mediante herramientas colaborativas.- El uso de aplicaciones como Discord ayuda al coordinamiento de esfuerzos, ya que es una herramienta esencial en la comunicación continua, a través de sus notificaciones automáticas sobre los commits y reuniones, el El equipo se puede sincronizar para decidir cambios y resolver incidencias. Esta práctica fortalece la eficiencia en la toma de decisiones.
- La modularización es la base de la arquitectura de los microservicios; al dividir un problema/aplicación en módulos autónomos, cada uno con su responsabilidad, se obtiene una estructura más flexible. En el presente proyecto, gracias a esa separación se puede desarrollar, probar y desplegar componentes independientes.

  El enfoque modular genera una práctica más ordenada y dirigida a la responsabilidad individual, esta arquitectura no solo facilita la escalabilidad, sino que también mejora el mantenimiento a largo plazo. Sin embargo, también impone retos, se debe comunicar los módulos, gestionar las dependencias y mantener la coherencia del sistema.

5. Preguntas de reflexión

1. *¿Qué ventajas observas en la integración mediante APIs REST respecto a un monolito tradicional?*

Las APIs REST permiten una comunicación escalable entre servicios independientes, facilitando la actualización de componentes sin afectar a todo el sistema. Además, promueve la reutilización de código y la integración con diversas tecnologías, optimizando la evolución del proyecto a largo plazo.

2. *¿Cómo aportan las herramientas colaborativas a la gestión técnica de los microservicios?*

Estas herramientas ayudan en la coordinación de tareas, control de versiones y comunicación entre desarrolladores; ayuda a facilitar la asignación de responsabilidad, otorga un seguimiento del progreso y gracias a ello se puede detectar y solucionar rápidamente algún problema.

3. *¿Qué riesgos existen al distribuir un sistema en varios microservicios y cómo pueden mitigarse?*

El principal riesgo es la complejidad de la comunicación entre servicios, asimismo gestionar las dependencias y posibles fallos de sincronización, también existe el aumento de costos en el despliegue y monitoreo, pero estos riesgos se resuelven mediante la aplicación de buenas prácticas como el uso de API gateways, pruebas automatizadas, contenedores estandarizados y una arquitectura bien documentada.

6. Bibliografía

[1] Newman, S. (2021). *Building Microservices.* O'Reilly Media.
[2] Richardson, C. (2022). *Microservices Patterns.* Manning Publications.
[3] OWASP Foundation (2023). *OWASP Secure Microservices Design.*
[4] Docker Inc. *Docker Documentation.*
[5] GitHub Docs. *Collaborative Development Workflows.*