

Grupo 4 - Integrantes: Miguel Luna, Anthony Gutiérrez y Luis Armijos

1. Tema:

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

2. 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.

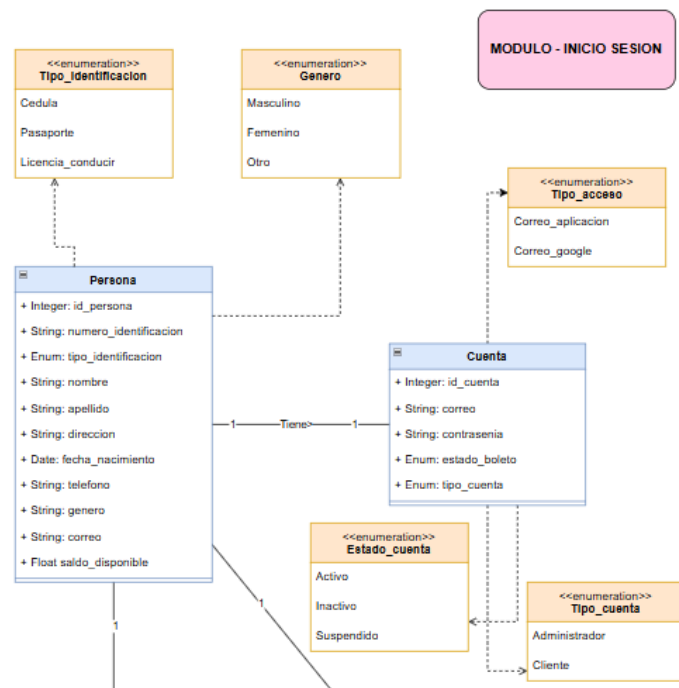
3. Desarrollo de la actividad

3.1. Resumen del microservicio creado

- **Nombre del microservicio, propósito y relación con el sistema principal.**

El microservicio de Inicio de Sesión tiene como propósito gestionar la autenticación de los usuarios en el sistema. En esta ampliación, se ha incorporado la posibilidad de acceder no solo mediante el correo electrónico registrado en la aplicación, sino también a través de una cuenta de Google. Este microservicio se integra directamente con el sistema principal, permitiendo validar credenciales y mantener la seguridad en el acceso a las demás funcionalidades de la plataforma.

- **Diagrama simple de su estructura.**



3.2. Integración mediante APIs

- **Descripción del método de comunicación; REST, API Gateway o endpoints compartidos.**

El microservicio de Inicio de Sesión se comunica con el API principal del sistema mediante peticiones HTTP POST y GET, gestionadas por los controladores internos y coordinadas a través de los servicios (Services) que actúan como intermediarios entre la capa lógica y la API.

En este proceso, el usuario puede autenticarse de dos formas:

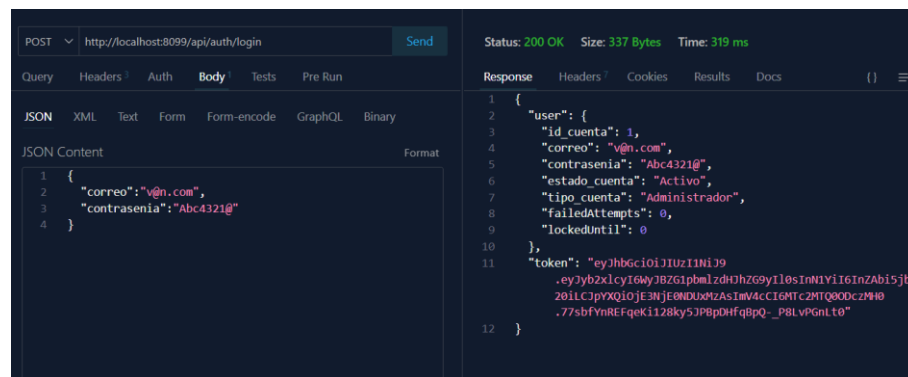
- Inicio de sesión tradicional: mediante el correo electrónico y contraseña registrados en la base de datos del sistema.
- Inicio de sesión con Google: a través de la integración con la API de autenticación de Google (OAuth 2.0).

Quando el usuario elige la opción de autenticarse con Google, la aplicación redirige la solicitud al endpoint de autorización de Google, donde el usuario concede los permisos necesarios una vez aprobada la autenticación, Google envía un token seguro (ID Token) al microservicio, el cual lo valida utilizando las librerías de verificación de Google si el token es válido, se extrae la información del usuario (nombre, correo electrónico y foto de perfil) y se envía una solicitud al servicio interno del sistema para comprobar si el usuario ya existe si no existe, se crea automáticamente un registro en la tabla Persona y se asocia a su respectiva Cuenta.

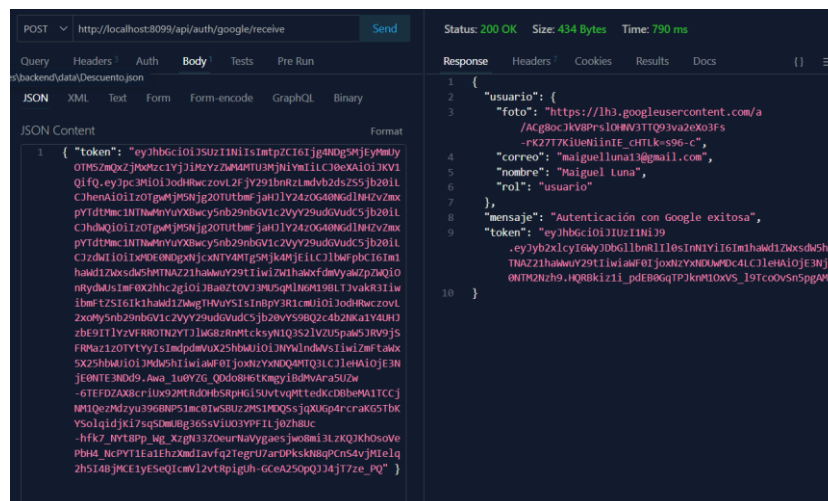
Además, las clases `Persona` y `Cuenta` se comunican entre sí tanto al crear una nueva cuenta como al iniciar sesión durante el registro, la clase `Persona` gestiona los datos personales del usuario (nombre, apellido, correo, etc.) y los envía al servicio correspondiente, que crea el registro en la base de datos simultáneamente, el servicio `CuentaService` crea una instancia de `Cuenta` vinculada a esa persona, asignándole un rol y credenciales de acceso.

- **Ejemplos de llamadas y respuestas; fragmentos Postman o Swagger.**

Ruta para la autenticación: <http://localhost:8099/api/auth/login>



Ruta para la autenticación con google: <http://localhost:8099/api/auth/login>

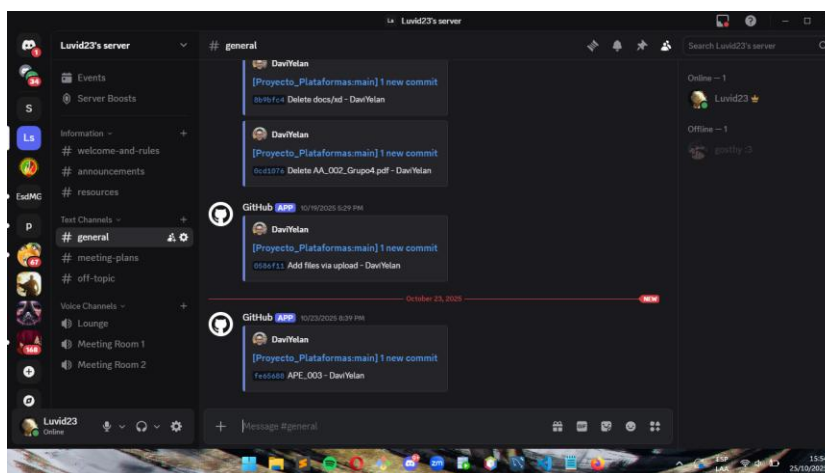


3.3. Herramientas colaborativas utilizadas

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

Para la coordinación del equipo, utilizamos GitHub como herramienta principal para el desarrollo colaborativo y la gestión del código fuente. A través de este repositorio compartimos los avances del proyecto y controlamos las versiones de cada componente. Además, el repositorio se encuentra integrado con Discord, lo que nos permite recibir notificaciones automáticas sobre los cambios realizados y mantener una comunicación constante. También empleamos reuniones periódicas en Discord para planificar tareas, revisar el progreso y coordinar las actividades de cada integrante del equipo.

- Evidenciar cómo estas herramientas contribuyeron a la planificación y seguimiento de tareas.



3.4. Buenas prácticas y aprendizajes

- Identificar dos buenas prácticas aplicadas durante el desarrollo.

Durante el desarrollo del proyecto se aplicaron diversas buenas prácticas, entre las cuales destacan las siguientes:

Uso de control de versiones con GitHub: permitió mantener un historial ordenado de los cambios realizados, facilitar la colaboración entre los integrantes de nuestro equipo y evitar conflictos en el código.

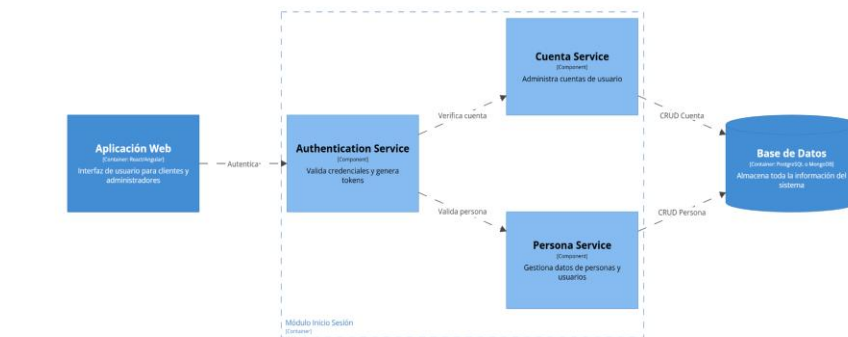
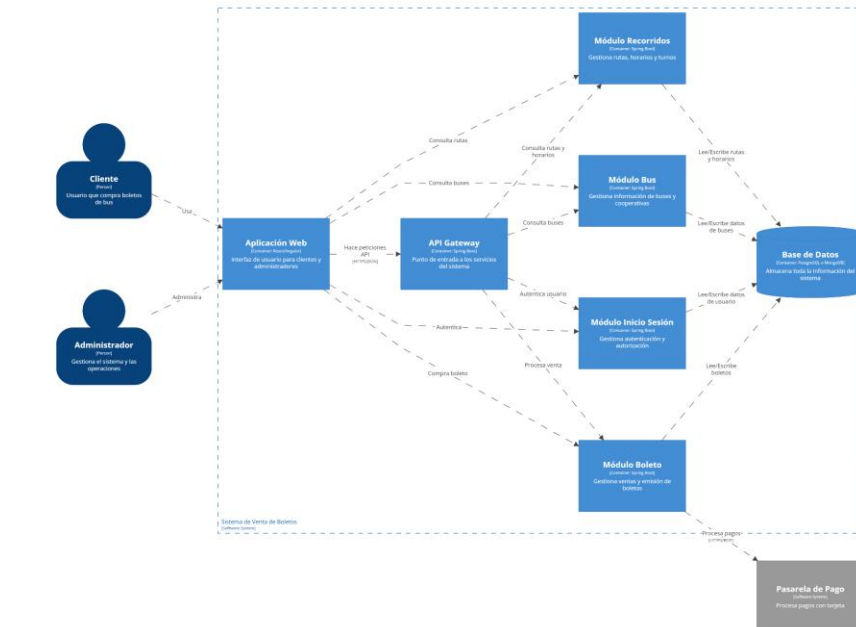
Aplicación de revisiones de código: antes de integrar nuevas funcionalidades al repositorio principal, se revisó el código para asegurar su calidad, legibilidad y cumplimiento de los estándares definidos por el equipo.

- Reflexión personal sobre cómo la modularización mejora la escalabilidad y mantenimiento del proyecto.

La modularización mejora significativamente la escalabilidad y el mantenimiento de un proyecto al dividir el sistema en componentes independientes con responsabilidades claramente definidas. Esta organización facilita la comprensión, desarrollo y prueba de cada módulo de forma aislada, permitiendo que los equipos trabajen de manera más eficiente y coordinada. Además, favorece la escalabilidad, ya que nuevas funcionalidades pueden incorporarse como módulos adicionales sin alterar la estructura existente, reduciendo riesgos y mejorando la adaptabilidad del sistema. En cuanto al mantenimiento, permite identificar y corregir errores con rapidez, así como actualizar o reemplazar

partes específicas sin afectar al conjunto. En síntesis, la modularización aporta orden, flexibilidad y sostenibilidad al desarrollo del software, garantizando su evolución continua con menor complejidad.

4. C4 actualizado



5. 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.