



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



---

**UNIVERSIDAD TÉCNICA DE AMBATO**

**Facultad de Ingeniería en Sistemas, Electrónica e Industrial**

**Título:** Informe de la prueba práctica

**Carrera:** Software

**Unidad de Organización Curricular:** Profesional

**Nivel y Paralelo:** 6to Software “A”

**Alumnos Participantes:**

Carrasco Paredes Kevin Andres

Chimborazo Guaman William Andres

Quishpe Lopez Luis Alexander

Sailema Gavilanez Ismael Alexander

**Módulo y Docente:**

Aplicaciones Distribuidas

Ing. Caiza Caizabuano Jose Ruben

**Fecha:** 05/04/2025



---

**UNIVERSIDAD**  
**TÉCNICA DE AMBATO**



## **1. Objetivos**

### **1.1.Objetivo General**

Desarrollar y documentar un sistema distribuido para un centro hospitalario basado en una arquitectura de microservicios, garantizando escalabilidad, alta disponibilidad y confiabilidad en la gestión de datos administrativos y médicos.

### **1.2.Objetivos Especificos**

- Diseñar e implementar una arquitectura de microservicios que incluya módulos de Administración, Autenticación y Consultas Médicas, utilizando tecnologías como ASP.NET Core, gRPC y REST/HTTP para la comunicación, asegurando modularidad y escalabilidad.
- Configurar entornos de desarrollo y despliegue mediante contenedores Docker para los microservicios y bases de datos MariaDB, facilitando la portabilidad y la consistencia en diferentes entornos.
- Documentar detalladamente cada etapa del proceso de desarrollo, incluyendo diagramas de arquitectura, casos de uso, secuencias y configuraciones, para proporcionar una guía clara y reproducible del sistema implementado.

## **2. Introducción**

3. El presente informe documenta el proceso de desarrollo de un sistema distribuido para un centro hospitalario, diseñado para gestionar de manera eficiente las operaciones administrativas y médicas. La solución se basa en una arquitectura de microservicios que garantiza escalabilidad, alta disponibilidad y confiabilidad, utilizando tecnologías modernas como ASP.NET Core, React, gRPC y Docker. El proyecto abarca la creación de tres microservicios principales —Administración, Autenticación y Consultas Médicas— que interactúan a través de un API Gateway y se conectan a bases de datos MariaDB dockerizadas. Este informe detalla cada etapa del desarrollo, desde el diseño de la arquitectura hasta su implementación, incluyendo diagramas de casos de uso, secuencias y configuraciones, con el objetivo de proporcionar una visión integral del sistema y sus componentes.

## **4. Desarrollo**






**Información Azure devops**

**Información de la organización:**

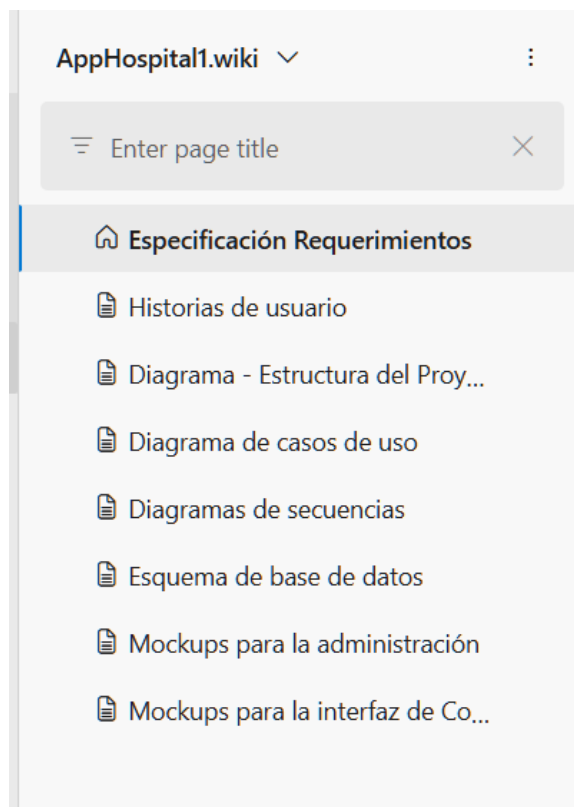


**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



Users			
All users			
Filter users		Access Level	Microsoft Entra User Type
Total 6		Summary	
Name	Access Level	Date Added	Last Accessed
 Caiza Caizabano Jose Ruben jr.caiza@uta.edu.ec	Stakeholder	16/4/2025	Never
 Carrasco Paredes Kevin Andres kcarrasco4232@uta.edu.ec	Basic	16/4/2025	28/4/2025
 Chimborazo Guaman William Andres wchimborazo2840@uta.edu.ec	Basic	21/6/2024	28/4/2025
 Quishpe Lopez Luis Alexander lquishpe5771@uta.edu.ec	Basic	16/4/2025	28/4/2025
 Sailema Gavilanez Ismael Alexander isailema4058@uta.edu.ec	Basic	15/4/2025	29/4/2025

### Estructura de wikis:



### Especificaciones de requerimientos:

#### Requerimientos Funcionales

Los requerimientos funcionales describen los comportamientos y capacidades específicas que el sistema debe proporcionar para satisfacer las necesidades de los usuarios, incluyendo administradores, personal médico y empleados hospitalarios.

#### 1. API de Administración

- Operaciones CRUD para Entidades:



- Crear, leer, actualizar y eliminar registros de **Centros Médicos, Médicos, Empleados y Especialidades**.
- Garantizar que las operaciones interactúen con la base de datos central (Quito) y las bases de datos locales de los hospitales (Guayaquil, Cuenca).
- **Gestión Centralizada de Datos:**
  - Permitir a los administradores gestionar registros globales (empleados, médicos, especialidades) almacenados en la base de datos central.
  - Proporcionar endpoints para obtener datos agregados de todos los centros médicos para reportes.

## 2. API de Consultas Médicas

- **Gestión de Consultas:**
  - Permitir a los hospitales crear, leer, actualizar y eliminar registros de **consultas médicas**, almacenados en sus bases de datos locales.
  - Soportar consultas del historial de consultas por médico o paciente dentro de un centro médico específico.

## 3. Interfaz de Usuario (UI) de Administración

- **Gestión de Entidades:**
  - Proporcionar una interfaz web (desarrollada en React) para que los administradores gestionen centros médicos, médicos, empleados y especialidades.
  - Soportar la creación, edición y eliminación de registros con validación de integridad de datos.
- **Estadísticas y Reportes:**
  - Mostrar estadísticas como resúmenes de nómina y reportes de consultas.
  - Permitir filtrado y ordenamiento de datos para reportes.

## 4. Interfaz de Usuario (UI) para Hospitales

- **Gestión de Consultas:**
  - Proporcionar una interfaz web (desarrollada en React) para que los médicos gestionen consultas, incluyendo crear, actualizar y visualizar registros.
  - Mostrar información relevante sobre especialidades para los médicos.
- **Autenticación de Usuarios:**
  - Implementar funcionalidad de inicio de sesión para restringir el acceso a usuarios autorizados (médicos, personal hospitalario).



- Asegurar que los usuarios solo accedan a los datos relevantes de su centro médico.

## **5. Arquitectura de Microservicios**

- **Servicios Desacoplados:**
  - Implementar microservicios separados para la administración (gestión de entidades globales) y consultas médicas (gestión de consultas locales).
  - Asegurar que cada microservicio pueda operar y escalar de forma independiente.
- **Comunicación entre Servicios:**
  - Permitir la comunicación entre microservicios mediante APIs RESTful para operaciones síncronas.
  - Soportar mensajería asíncrona (por ejemplo, con RabbitMQ o Kafka) para tareas como sincronización de datos o notificaciones.

## **6. Operaciones de Base de Datos**

- **Gestión de Bases de Datos Distribuidas:**
  - Soportar una base de datos central (MariaDB en Quito) para entidades globales (centros, médicos, empleados, especialidades).
  - Mantener bases de datos locales (MariaDB en Guayaquil y Cuenca) para registros de consultas específicas de cada hospital.
- **Replicación:**
  - Implementar replicación unidireccional desde la base de datos central a las bases locales para entidades globales.
  - Asegurar que los datos de consultas permanezcan locales en cada hospital, pero puedan ser consultados de forma centralizada para reportes.

## **7. Reportes y Análisis**

- **Reportes Centralizados:**
  - Habilitar a los administradores para generar reportes entre hospitales, agregando datos de consultas de todos los centros.

## **8. Despliegue**

- **Despliegue en la Nube:**
  - Desplegar microservicios backend y aplicaciones frontend en una plataforma en la nube (Azure App Services o similar).
  - Asegurar que las aplicaciones sean accesibles desde diferentes ubicaciones con balanceo de carga adecuado (usando Azure Load Balancer).

## **Requerimientos No Funcionales**



Los requerimientos no funcionales especifican los atributos de calidad y restricciones del sistema, garantizando rendimiento, seguridad, escalabilidad y mantenibilidad.

### **1. Confiabilidad**

- **Replicación de Datos:**
  - Garantizar una consistencia en la replicación de datos entre la base central y las bases locales.
  - Manejar fallos de replicación con mecanismos de reintentos y registro de errores.
  - Usar un esquema de maestro esclavo para obtener un modelo de base de datos que permita la replicación de datos.

### **3. Seguridad**

- **Autenticación y Autorización:**
  - Implementar autenticación segura de usuarios usando protocolos estándar (JWT).
  - Restringir el acceso a datos sensibles según los roles de usuario (por ejemplo, los médicos solo acceden a los datos de su centro).

### **4. Usabilidad**

- **Interfaces Intuitivas:**
  - Asegurar que las interfaces de administración y hospital sean fáciles de usar, con navegación clara y mínima capacitación requerida.
  - Proporcionar tooltips o documentación de ayuda para operaciones complejas (por ejemplo, generación de reportes).

### **5. Mantenibilidad**

- **Diseño Modular:**
  - Estructurar los microservicios para permitir actualizaciones independientes sin afectar otros servicios.
  - Usar APIs versionadas para soportar compatibilidad hacia atrás.
- **Documentación:**
  - Proporcionar documentación completa de la API usando Swagger para todos los endpoints.
  - Incluir documentación técnica detallada en Azure DevOps sobre arquitectura, configuraciones de bases de datos y procedimientos de despliegue.

### **6. Interoperabilidad**

- **Estándares de API:**



- Usar APIs RESTful con JSON para la comunicación entre microservicios y aplicaciones frontend.
- Asegurar compatibilidad con clientes HTTP estándar para integraciones externas.
- **Compatibilidad de Bases de Datos:**
  - Soportar MariaDB para bases de datos distribuidas, con configuraciones adaptables a otras bases SQL (por ejemplo, MySQL, PostgreSQL) si es necesario.

## 7. Despliegue e Infraestructura

- **Compatibilidad con la Nube:**
  - Desplegar todos los componentes (VMs, microservicios, frontend) en Azure, aprovechando Azure VMs, App Services y Load Balancer.
  - Asegurar que la infraestructura pueda replicarse en otros proveedores de nube (por ejemplo, AWS) con cambios mínimos.
- **Integración CI/CD:**
  - Usar Azure DevOps para integración y despliegue continuo, automatizando compilaciones, pruebas y despliegues.
  - Almacenar toda la documentación y el código fuente en repositorios de Azure DevOps.

## 8. Pruebas

- **Pruebas de Integración:**
  - Realizar pruebas de integración para verificar que los microservicios y la replicación de bases de datos funcionen correctamente.
  - Probar la sincronización de datos entre hospitales y los reportes centralizados.

## 9. Documentación y Entrega

- **Documentación Completa:**
  - Entregar documentación técnica que cubra diagramas de arquitectura, esquemas de bases de datos, interacciones entre microservicios y pasos de despliegue.
- **Control de Versiones:**
  - Mantener todo el código y la documentación en repositorios de Azure DevOps con mensajes de commit claros e historial de versiones.

## Historias de usuario:



### **Historias de Usuario para el Sistema de Gestión Hospitalaria**

A continuación, se presentan las historias de usuario basadas en los requerimientos del proyecto descrito en el documento proporcionado. Estas historias están diseñadas para ser incluidas en la wiki de Azure DevOps y reflejan las necesidades de los usuarios (administradores, médicos y personal hospitalario) en el contexto de un sistema distribuido de microservicios para la gestión hospitalaria.

#### **Historias de Usuario**

##### **1. Gestión de Centros Médicos por el Administrador**

Como administrador del sistema hospitalario, quiero gestionar los registros de los centros médicos, para mantener actualizada la información de todos los centros en la base de datos central.

- **Prioridad:** Alta

##### **2. Gestión de Médicos por el Administrador**

Como administrador del sistema hospitalario, quiero gestionar los registros de los médicos, para asegurar que la información de los médicos esté actualizada y disponible en todos los centros.

- **Prioridad:** Alta

##### **3. Gestión de Empleados por el Administrador**

Como administrador del sistema hospitalario, quiero gestionar los registros de los empleados, para mantener un control actualizado del personal en todos los centros médicos.

- **Prioridad:** Alta

##### **4. Gestión de Especialidades por el Administrador**

Como administrador del sistema hospitalario, quiero gestionar las especialidades médicas, para asegurar que los médicos puedan ser asignados correctamente a sus áreas de expertise.

- **Prioridad:** Media

##### **5. Registro de Consultas Médicas por Médicos**

Como médico en un centro médico, quiero registrar consultas médicas, para mantener un historial de las atenciones brindadas a los pacientes en mi centro.

- **Prioridad:** Alta

##### **6. Actualización y Eliminación de Consultas Médicas**

Como médico en un centro médico, quiero actualizar o eliminar consultas médicas, para corregir errores o eliminar registros obsoletos en mi centro





- **Prioridad:** Alta

#### **7. Visualización de Consultas por Médico**

Como médico en un centro médico,  
quiero ver un historial de mis consultas,  
para revisar las atenciones previas realizadas a los pacientes.

- **Prioridad:** Alta

#### **8. Generación de Reportes Centralizados por Administrador**

Como administrador del sistema hospitalario,  
quiero generar reportes centralizados de consultas,  
para obtener una visión general de la actividad en todos los centros médicos.

- **Prioridad:** Alta

#### **9. Autenticación de Usuarios**

Como usuario del sistema (médico o administrador),  
quiero autenticarme en la aplicación,  
para acceder solo a la información relevante de mi rol y centro médico.

- **Prioridad:** Alta

#### **10. Visualización de Estadísticas Administrativas**

Como administrador del sistema hospitalario,  
quiero ver estadísticas sobre el sistema,  
para tomar decisiones informadas sobre la gestión de centros y personal.

- **Prioridad:** Media

#### **11. Acceso a Información de Especialidades por Médicos**

Como médico en un centro médico,  
quiero ver información sobre especialidades,  
para entender las capacidades médicas disponibles en mi centro.

- **Prioridad:** Media

#### **13. Despliegue Seguro en la Nube**

Como administrador del sistema hospitalario,  
quiero que la aplicación esté desplegada en la nube,  
para garantizar su disponibilidad y accesibilidad desde cualquier ubicación.

- **Prioridad:** Alta

#### **14. Replicación de Datos entre Bases**

Como administrador del sistema hospitalario,  
quiero que los datos globales se repliquen entre bases de datos,  
para garantizar que todos los centros médicos tengan acceso a información actualizada.

- **Prioridad:** Alta

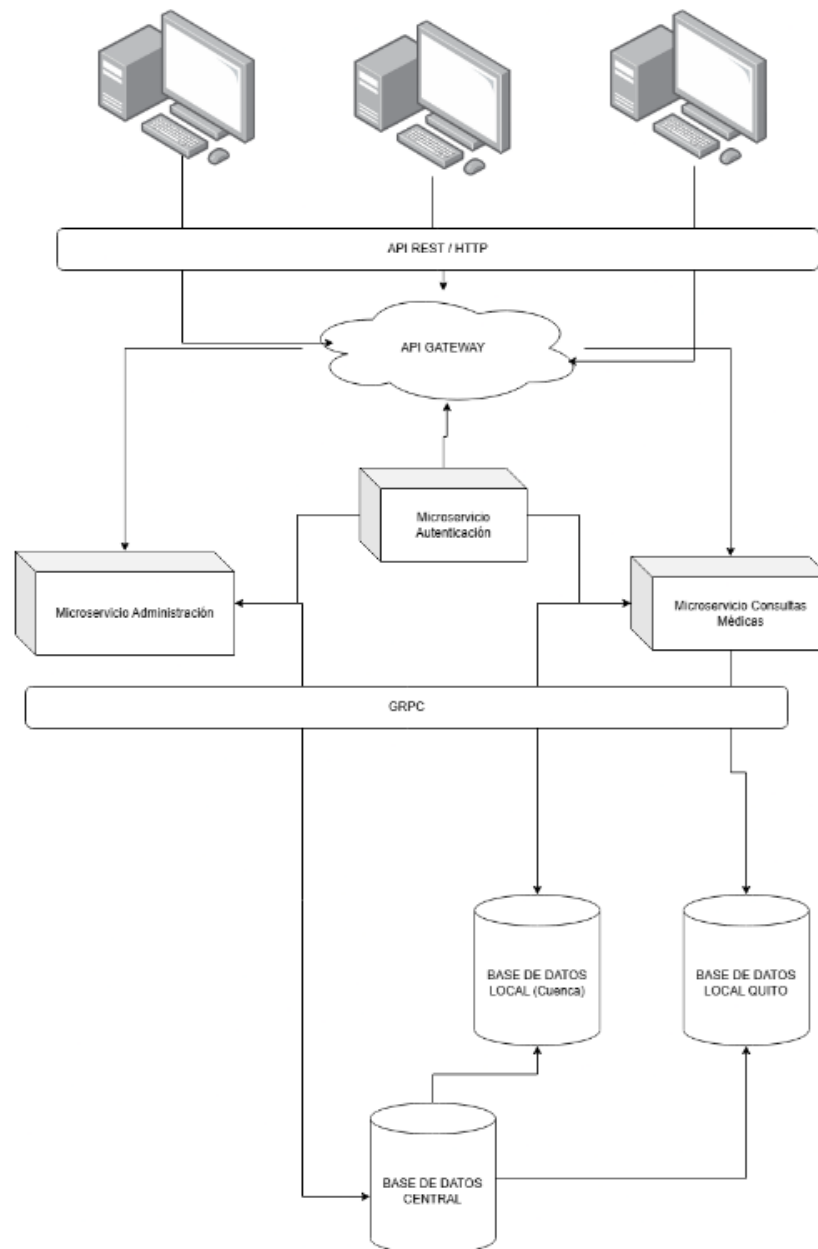
#### **15. Documentación de la API**



Como desarrollador del sistema,  
quiero una documentación clara de la API,  
para facilitar la integración y el mantenimiento del sistema.

- **Prioridad:** Media

### Diagrama de estructura del proyecto



### Diagrama de Arquitectura

El diagrama proporcionado ilustra la estructura de la aplicación:

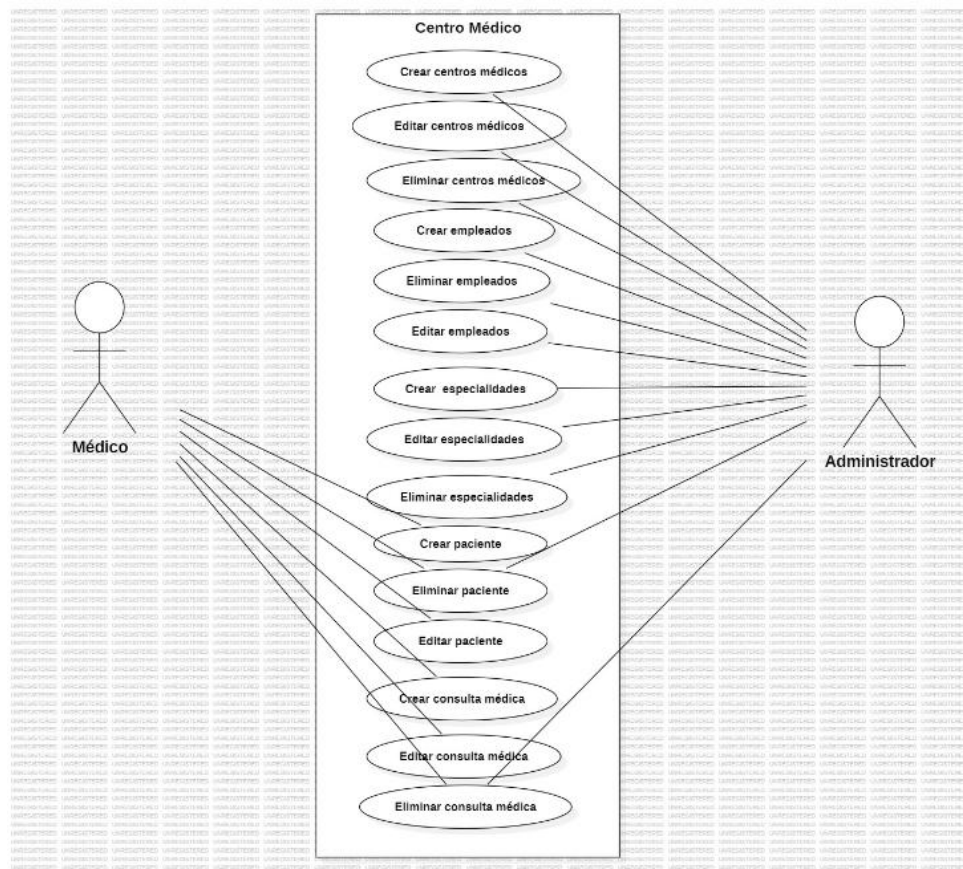


**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



- Los clientes (front-end) se conectan al API Gateway mediante REST/HTTP esto debido a la facilidad y compatibilidad que tiene este protocolo con los distintos navegadores.
- El API Gateway distribuye las solicitudes a los microservicios (Administración, Autenticación, Consultas Médicas).
- Los microservicios interactúan entre sí y con las bases de datos (Central, Local Cuenca, Local Quito) usando gRPC.

### Diagrama de casos de uso



### Descripción de Casos de Uso del Sistema del Centro Médico

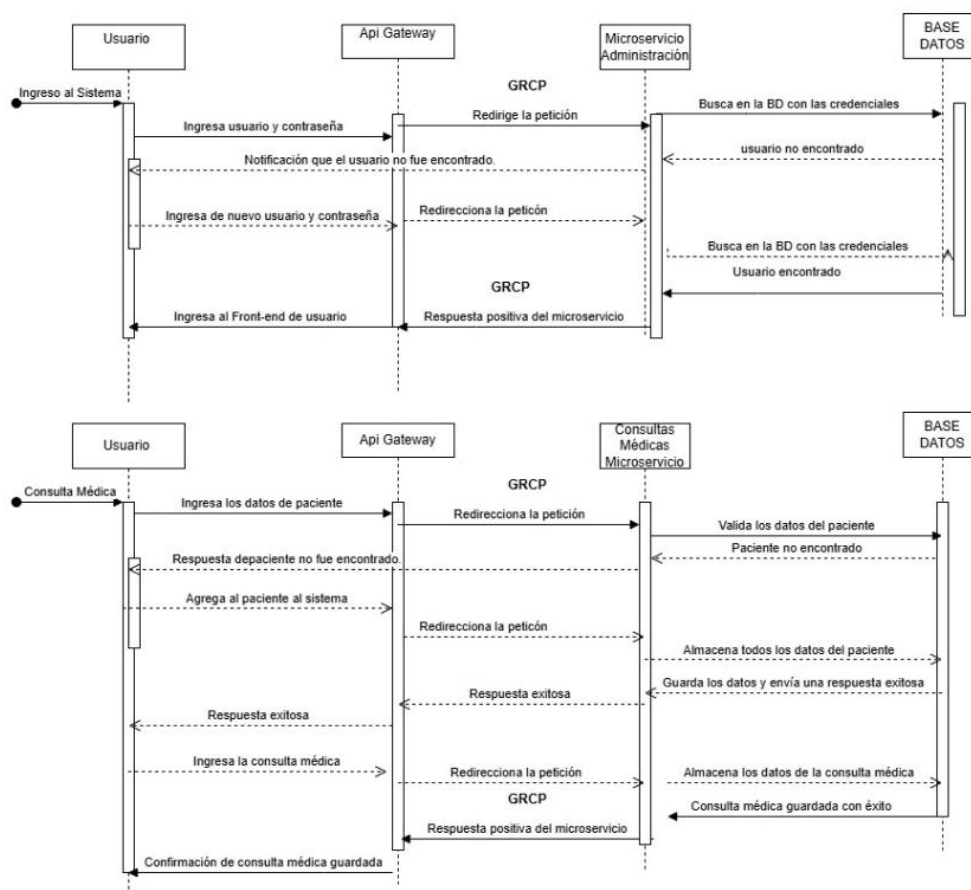
El diagrama de casos de uso del sistema del Centro Médico identifica las interacciones de dos actores principales, el Médico y el Administrador, con las funcionalidades del sistema, organizadas en torno a la gestión de centros médicos, empleados, especialidades, pacientes y consultas médicas.

- **Actores:**
  - **Médico:** Responsable de gestionar consultas médicas y acceder a información de pacientes.
  - **Administrador:** Encargado de la gestión administrativa del centro médico, incluyendo centros, empleados y especialidades.
- **Casos de Uso:**



- **Gestión de Centros Médicos:** El Administrador puede crear, editar y eliminar centros médicos, permitiendo la administración de las ubicaciones donde opera el sistema (por ejemplo, "Central" en Quito).
- **Gestión de Empleados:** El Administrador tiene la capacidad de crear, editar y eliminar empleados, lo que incluye la asignación de roles como "Administrador" o "Doctor" y la asociación con un centro médico.
- **Gestión de Especialidades:** El Administrador puede crear, editar y eliminar especialidades médicas (como "Sin Especialidad"), facilitando la categorización de los médicos según su área de experiencia.
- **Gestión de Pacientes:** El Administrador puede crear, editar y eliminar pacientes, asegurando que la base de datos del sistema mantenga información actualizada de los usuarios atendidos.
- **Gestión de Consultas Médicas:** El Médico puede crear, editar y eliminar consultas médicas, lo que permite registrar y gestionar las interacciones clínicas con los pacientes, mientras que el Administrador también puede supervisar estas acciones.

### Diagramas de secuencia

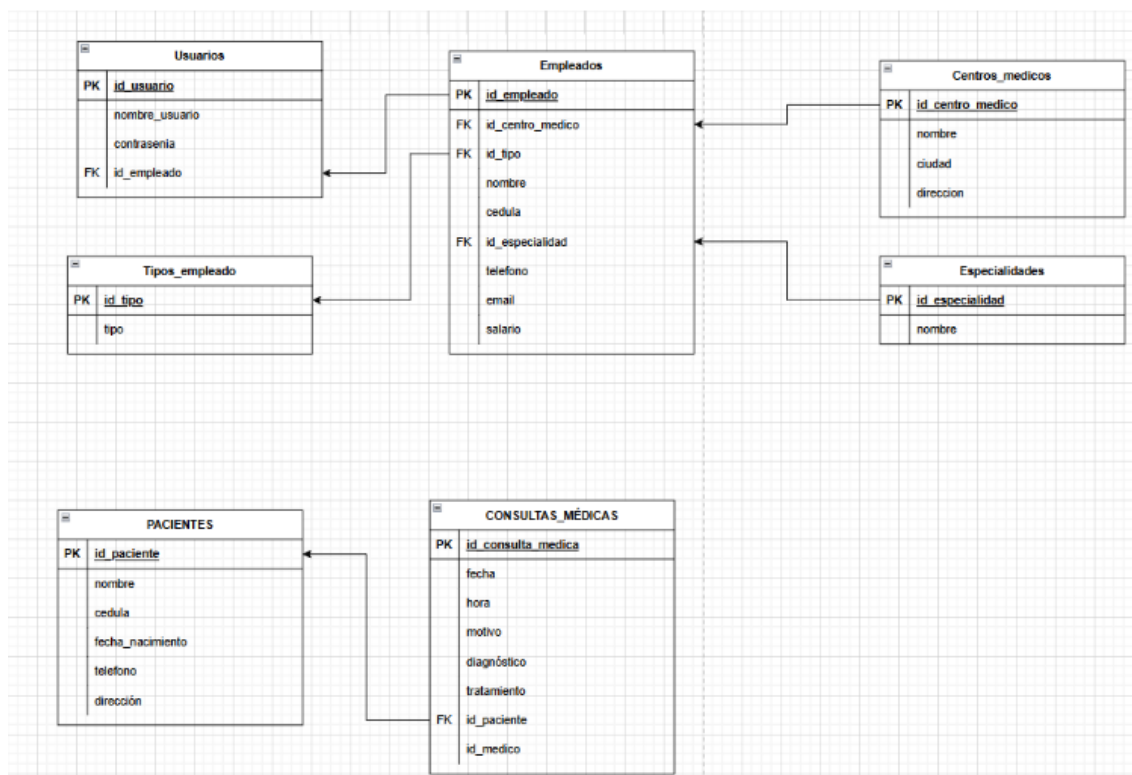




Los diagramas de secuencia ilustran las interacciones entre los componentes del sistema (Usuario, API Gateway, Microservicios y Base de Datos) para dos escenarios principales: el ingreso al sistema (Microservicio de Administración) y la gestión de una consulta médica (Microservicio de Consultas Médicas), utilizando comunicación gRPC.

- **Ingreso al Sistema (Microservicio de Administración):**  
El usuario ingresa sus credenciales (usuario y contraseña) a través del front-end, enviando una solicitud al API Gateway. Este redirige la petición al Microservicio de Administración mediante gRPC. El microservicio busca las credenciales en la base de datos; si el usuario no es encontrado, retorna una respuesta negativa al API Gateway, que notifica al usuario. Si las credenciales son válidas, el microservicio retorna una respuesta positiva, y el API Gateway permite el ingreso del usuario al sistema.
- **Consulta Médica (Microservicio de Consultas Médicas):**  
El usuario, a través del front-end, realiza acciones relacionadas con una consulta médica. Estas incluyen: ingresar datos del paciente, agregar el paciente al sistema, ingresar la consulta médica y confirmar que se guarde. Cada acción se envía al API Gateway, que redirige la petición al Microservicio de Consultas Médicas vía gRPC. El microservicio valida los datos del paciente con la base de datos; si no es encontrado, retorna un error. Si es válido, almacena los datos del paciente y la consulta médica, retornando una respuesta exitosa al API Gateway, que confirma al usuario el éxito de la operación.

### Diagrama de base de datos





**UNIVERSIDAD TÉCNICA DE AMBATO**  
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL  
CARRERA DE SOFTWARE  
CICLO ACADÉMICO: MARZO – JULIO 2025



## Mockups de la aplicación - Administración

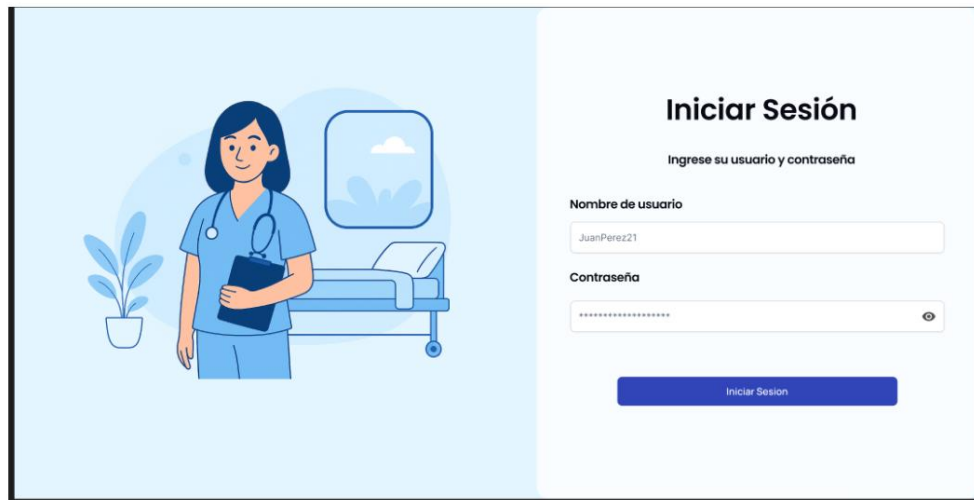
### Paleta de colores

#### Paleta de colores

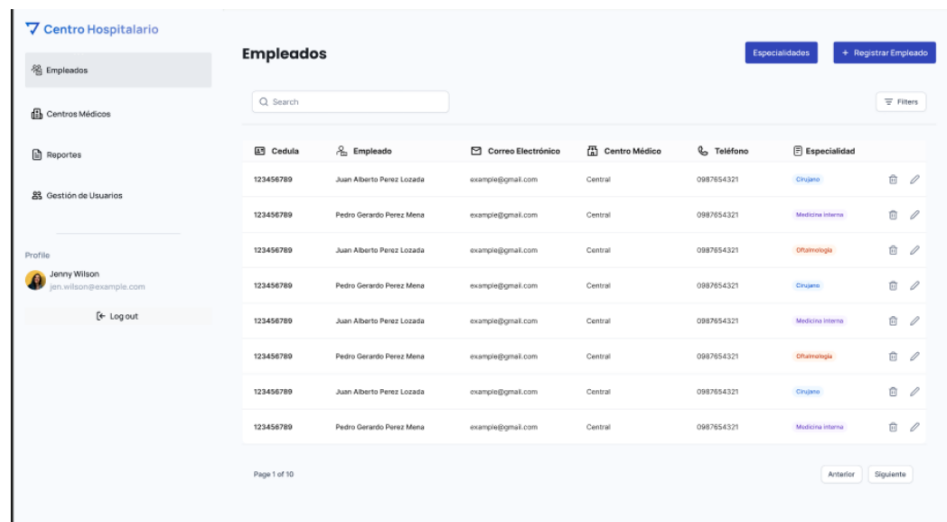


Color de fondo principal: #F8FAFD  
Color principal componentes: #3145B9  
Color secundario (hover): #0067D2  
Color principal para textos: #1E1E1E  
Color secundario para textos: #667085

### Mockup Login



### Mockup Empleados







**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



**\*\*Mockup Especialidades**

Centro Hospitalario

Empleados

Centros Médicos

Reportes

Gestión de Usuarios

Profile

Jenny Wilson

jen.wilson@example.com

Log out

Empleados

Search

Filters

Cedula

Empleados

123456789

Juan Albe

123456789

Pedro Ger

123456789

Juan Albe

123456789

Pedro Ger

123456789

Juan Albe

123456789

Pedro Ger

123456789

Juan Albe

123456789

Pedro Ger

123456789

Juan Albe

123456789

Pedro Ger

123456789

Pedro Gerardo Perez Mora

ejemplo@gmail.com

Central

Page 1 of 10

Anterior

Siguiente

Especialidades

Nombre

Quiropráctico

Cirujano

Odontólogo

Anestesiólogo

+ Añadir

Telefono

0987654321

0987654321

0987654321

0987654321

0987654321

0987654321

0987654321

0987654321

0987654321

0987654321

Especialidad

Ortopedia

Medicina Interna

Oftalmología

Cardiología

Medicina Interna

Oftalmología

Ortopedia

Medicina Interna

Oftalmología

Medicina Interna

Anterior

Siguiente

**Mockup Centros Médicos**

Centro Hospitalario

Empleados

Centros Médicos

Reportes

Gestión de Usuarios

Profile

Jenny Wilson

jen.wilson@example.com

Log out

Centros Médicos

Search

Filters

Nombre

Centro Hospitalario Ejemplo 1

Ciudad

Guayaquil

Dirección

Av. Los cacaos y las Moras

Cantidad Empleados

15

Centro Hospitalario Ejemplo 1

Cuenca

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Cuenca

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Guayaquil

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Cuenca

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Guayaquil

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Cuenca

Av. Los cacaos y las Moras

15

Centro Hospitalario Ejemplo 1

Cuenca

Av. Los cacaos y las Moras

15

Page 1 of 10

Anterior

Siguiente

**Mockup de reportes**





# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

### CARRERA DE SOFTWARE

#### CICLO ACADÉMICO: MARZO – JULIO 2025



#### Mockup de Gestión de usuarios

Centro Hospitalario

Empleados

Centros Médicos

Reportes

Gestión de Usuarios

Perfil

Jenny Wilson

jen.wilson@example.com

Log out

Gestión de usuarios

Cédula

Nombre Usuario

Empleado

Correo Electrónico

Centro Médico

Teléfono

123456789

Juan2022

Juan Alberto Perez Lozada

example@gmail.com

Central

0987654321

123456789

Leo455

Pedro Gerardo Perez Mena

example@gmail.com

Central

0987654321

123456789

Juan2022

Juan Alberto Perez Lozada

example@gmail.com

Central

0987654321

123456789

Leo455

Pedro Gerardo Perez Mena

example@gmail.com

Central

0987654321

123456789

Juan2022

Juan Alberto Perez Lozada

example@gmail.com

Central

0987654321

123456789

Leo455

Pedro Gerardo Perez Mena

example@gmail.com

Central

0987654321

123456789

Juan2022

Juan Alberto Perez Lozada

example@gmail.com

Central

0987654321

123456789

Leo455

Pedro Gerardo Perez Mena

example@gmail.com

Central

0987654321

Page 1 of 10

Anterior

Siguiente

#### Mockups de la aplicación – Consultas Médicas

##### Interfaz Consultas Médicas.

Nombre Centro

Consultas Médicas

Pacientes

Perfil

Jenny Wilson

jen.wilson@example.com

Log out

Consultas Médicas

Fecha

Cédula Paciente

Paciente

Teléfono Paciente

05/20/2025

1804925485

Juan Alberto Perez Lozada

0123456789

05/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

05/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

05/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

04/20/2025

1804925485

Luis Fernando Alfonso Lara

0123456789

04/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

04/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

04/20/2025

1804925485

Pedro Gerardo Perez Mena

0123456789

Page 1 of 10

Anterior

Siguiente

##### Modal del detalle de la cita médica.

Nombre Centro

Consultas Médicas

Pacientes

Perfil

Jenny Wilson

jen.wilson@example.com

Log out

Consultas

Fecha

Cédula

Nombre

Edad

Teléfono

05/20/2025

1804905692

Juan Pérez

26 años

0123456789

Información Paciente

Cédula: 1804905692

Teléfono: 0123456789

Nombre: Juan Pérez

Edad: 26 años

Dirección: Av. Centro America y Saltos

Información Consulta Médica

Motivo:

Diagnostico

Diagnostico

Page 1 of 10

Anterior

Siguiente





**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



**Interfaz Pacientes.**

Nombre Centro

Consultas Médicas

Pacientes

Profile  
Jenny Wilson  
jenny.wilson@example.com  
Oftalmólogo  
Logout

Pacientes

Filters

Cédula	Paciente	Edad	Teléfono	Dirección	
1804925485	Juan Alberto Perez Lozada	30	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	25	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	30	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	25	0123456789	Av. Los Andes Y Pichincha	
1804925485	Luis Fernando Alfonso Lara	34	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	25	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	29	0123456789	Av. Los Andes Y Pichincha	
1804925485	Pedro Gerardo Perez Mena	55	0123456789	Av. Los Andes Y Pichincha	

Page 1 of 10 Anterior Siguiente

**Asignación de tareas a miembros del equipo**

	Order	ID	Title	Assigned To	State	Tags
	1	58	Desarrollo del modulo de consultas médicas	Sailema Gavilan...	Done	
	2	40	Creación de la interfaz de Centros médicos	Sailema Gavilan...	Done	
	3	43	Interfaz de login	Sailema Gavilan...	Done	
	4	50	Integración del Api con la interfaz del centros médicos	Sailema Gavilan...	Done	
	5	51	Creación de api gateway	Carrasco Pared...	Done	
	6	36	Añadir Las historias de usuario en wiki	Sailema Gavilan...	Done	
	7	44	Creación de la plantilla principal y menu lateral	Sailema Gavilan...	Done	
	8	52	Integración de JWT con el login del frontEnd	Sailema Gavilan...	Done	
	9	39	Creación de la interfaz de empleados	Chimborazo Gu...	Done	
	10	53	Integracion del api con empleados en frontend	Chimborazo Gu...	Done	
	11	54	Integración apigateway de administración con los centros ...	Sailema Gavilan...	Done	
	12	49	Añadir los mockups de administración a la wiki	Sailema Gavilan...	Done	
	13	41	Creación de la interfaz de reportes	Chimborazo Gu...	Done	
	14	42	Creación de la interfaz de gestión de usuarios	Quishpe Lopez ...	Done	
	15	66	Desarrollo del módulo de pacientes en frontend	Quishpe Lopez ...	Done	
	16	33	Creación de la base de datos física en MariaDB	Carrasco Pared...	Done	
+	17	32	Creación de las máquinas virtuales en Azure	Carrasco Pared...	Done	
	18	30	Implementación de JWT en la API de administración	Carrasco Pared...	Done	
	19	28	>  Creación del APIREST de consultas médicas (Consultas médi...	Quishpe Lopez ...	Done	
	20	25	>  Creación del APIREST de consultas médicas (Pacientes)	Chimborazo Gu...	Done	
	21	21	>  Creación del APIREST de administración	Carrasco Pared...	Done	
	22	20	Creación del repositorio del proyecto en Github	Chimborazo Gu...	Done	
	23	19	Planificación de arquitecturas	Quishpe Lopez ...	Done	
	24	18	Diseño del esquema de la base de datos	Carrasco Pared...	Done	
	25	17	Investiaación de tecnologías para la base de datos	Sailema Gavilan...	Done	

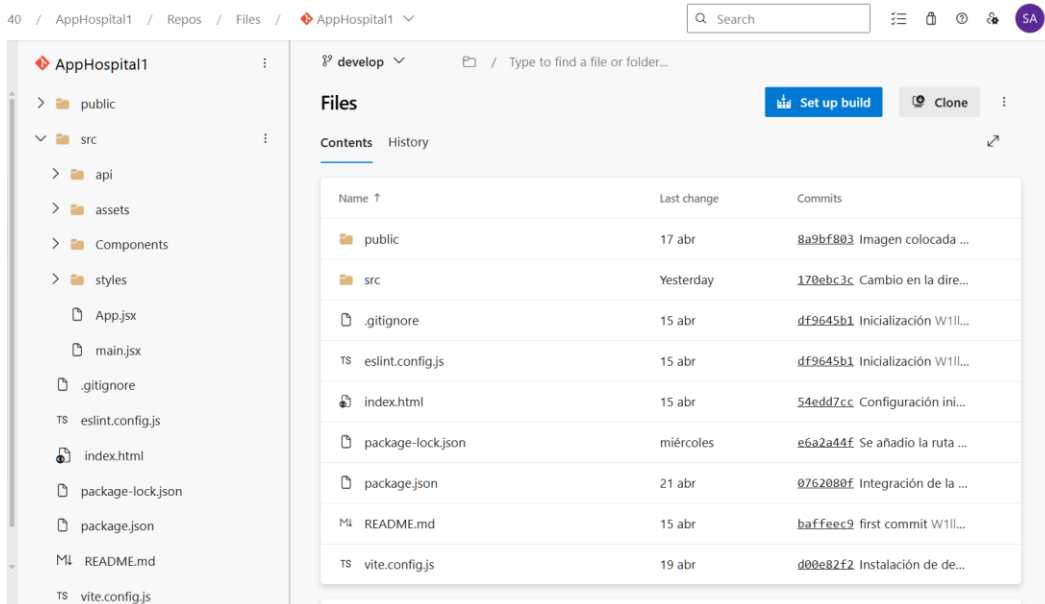


**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**  
**CARRERA DE SOFTWARE**  
**CICLO ACADÉMICO: MARZO – JULIO 2025**



26	57	Desarrollo del método de editar usuario en administración.	Quishpe Lopez ... Done
27	56	Desarrollo y control para la interfaz de centros médicos	Sailema Gavilan... Done
28	16	Subir los archivos de inicialización de los proyectos	Chimborazo Gu... Done
29	35	Añadir informe de requerimientos en wiki	Sailema Gavilan... Done
30	37	Añadir el diagrama del sistema a las wikis	Sailema Gavilan... Done
31	55	Desarrollo de mockups para la interfaz de consultas médicas	Sailema Gavilan... Done

## Repositorio publicado:



## 5. Conclusiones

- La implementación de microservicios (Administración, Autenticación y Consultas Médicas) con comunicación vía gRPC y REST/HTTP cumple con los requisitos de escalabilidad y modularidad, permitiendo que cada componente opere de forma independiente y se adapte a futuras necesidades.
- El uso de Docker Compose para gestionar los contenedores de los microservicios y bases de datos MariaDB asegura un entorno de desarrollo consistente y portátil, aunque se identifica la necesidad de implementar replicación de bases de datos para mejorar la disponibilidad.
- La documentación generada, que incluye diagramas de arquitectura, casos de uso, secuencias y configuraciones de Docker, proporciona un registro completo del proceso de desarrollo, facilitando el mantenimiento y la posible extensión del sistema en el futuro.