

Universidad de San Carlos de Guatemala USAC.

División de Ciencias de la Ingeniería.

Centro Universitario de Occidente CUNOC.

Análisis y diseño 1



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala
INGENIERIA
CUNOC



Proyecto 2

Manual Tecnico

Integrante:

David Enrique Lux Barrera

201931344

Quetzaltenango, 2024

Tecnologías Utilizadas en el proyecto

Frontend

- **Angular:** Framework para el desarrollo de interfaces de usuario dinámicas y responsivas.
- **Tailwind CSS:** Framework CSS para diseño rápido y moderno.

Backend

- **Spring Boot:** Framework para el desarrollo de aplicaciones Java basadas en microservicios.
- **Java:** Lenguaje de programación utilizado para la lógica del servidor.
- **Spring Security:** Implementación de autenticación y autorización para proteger el sistema.
- **Spring Web:** Módulo utilizado para desarrollar APIs RESTful.
- **Spring Data JPA:** Herramienta para la persistencia de datos utilizando JPA.
- **Spring Boot Actuator:** Módulo para monitoreo y métricas del sistema.
- **Java Mail Server:** Envío de correos electrónicos desde la aplicación.

Bases de Datos

- **MySQL:** Base de datos relacional principal utilizada en el proyecto.
- **PostgreSQL:** Opcional como alternativa o complemento para la gestión de datos.

Dependencias y Librerías

- **Lombok:** Reducción del código repetitivo mediante anotaciones para generar getters, setters, constructores y más.
- **Thymeleaf:** Motor de plantillas para la generación dinámica de vistas HTML.
- **MySQL Driver:** Controlador para conectar el proyecto con bases de datos MySQL.
- **PostgreSQL Driver:** Controlador para conectar el proyecto con bases de datos PostgreSQL.

API

- **REST:** Arquitectura utilizada para desarrollar servicios web que permiten la comunicación entre el cliente y el servidor.

Principales Tecnologías:



Lenguajes de programación y tecnologías

- **Backend**
 - **lenguaje de programación**
 - Java -v21
 -
 - **frameworks**
 - Spring boot
 - **entorno de desarrollo**
 - IDE: Intejili
 - SO: Linux (Ubuntu)
- **Frontend**
 - **lenguaje de programación**
 - typescript
 - **frameworks**
 - Angular
- **Bases de datos**
 - Mysql
- **Servicios en la Nube**



Despliegue:

Base de Datos

1. Requisitos Previos

- a. **Descargar los scripts DML y DDL:** Asegúrese de tener los archivos necesarios para la creación y manipulación de la base de datos.
- b. **Tener instalado MySQL:** Instale MySQL siguiendo las instrucciones correspondientes a su sistema operativo.
- c. **Ejecutar los scripts DML y DDL:** Una vez configurado MySQL, ejecute los scripts para crear y poblar la base de datos.

Instalación de MySQL

En Ubuntu

1. Actualizar el sistema

Abra una terminal y ejecute los siguientes comandos para actualizar los paquetes del sistema:

```
sudo apt update
```

2. Instalar MySQL

Instale el servidor MySQL con el siguiente comando:

```
sudo apt install mysql-server
```

3. Verificar el estado del servicio

Compruebe que MySQL esté en ejecución ejecutando:

```
sudo systemctl status mysql
```

4. Configurar MySQL

Ejecute el asistente de configuración para establecer una contraseña segura para el usuario **root** y realizar ajustes básicos:

```
sudo mysql_secure_installation
```

5. Acceder a MySQL

Use el siguiente comando para iniciar sesión en MySQL con el usuario **root**:

```
sudo mysql -u root -p
```

En Windows

1. Descargar MySQL Installer

Descargue el instalador desde el sitio oficial: [MySQL Installer](#).

2. Ejecutar el instalador

Inicie el instalador y seleccione la opción **Developer Default** para instalar las herramientas recomendadas. También puede personalizar la instalación si lo desea.

3. Configurar MySQL

Durante la instalación:

- Establezca una contraseña para el usuario **root**.
- Seleccione el modo de autenticación (se recomienda **Legacy Authentication** para compatibilidad).

4. Verificar el servicio

Asegúrese de que el servicio MySQL esté funcionando desde el panel de configuración o accediendo a los **Servicios** de Windows.

5. Acceso a MySQL

Abra **MySQL Workbench** o use la terminal de comandos para conectarse al servidor. Ingrese las credenciales configuradas previamente.

1. Backend:

a. Descargar el proyecto de backend en la siguiente url:

- i. <https://github.com/Elvis-Aguilar/appointmentManagement-API.git>

b. Tener Java JVM y java 21

c. Crear el archivo .env con el siguiente ejemplo:

```
DB_USER=root
DB_PASS=XXX
DB_HOST=localhost:3306
DB_DATABASE=appointment_DB
```

```
JWT_SECRET=21e5289c1cc84fafa5952558358b348ca7ab4306369cb20fd3cbbfbc04b69893

#var config clouinary
CLOUDINARY_CLOUD_NAME=ddkp3bobz
CLOUDINARY_API_KEY=722436462235999
CLOUDINARY_API_SECRET=6l_ZDaRdImmroKwQVlTiKe9X7k8

#CLOUDINARY_URL=cloudinary://722436462235999:6l_ZDaRdImmroKwQVlTiKe9X7k8@ddkp3bobz

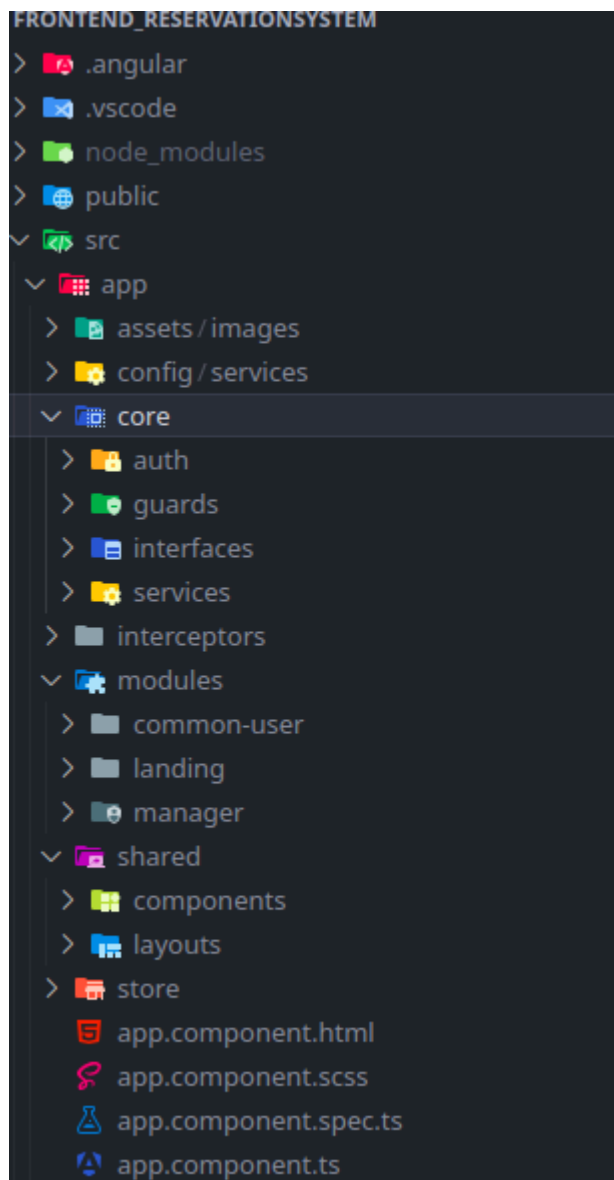
EMAIL_ADDRESS=testingcunoc@gmail.com
EMAIL_PASSWORD=mongldnuktzotkua
```

- d. Ejecutar el comando: **mvn clean**
- e. Levantar el proyecto **java -jar archivo.jar**

2. Frontend:

- a. Descargar el proyecto de backend en la siguiente url:
 - i. https://github.com/Elvis-Aguilar/frontend_ReservationSystem.git
- b. Instalar dependencias
 - i. `npm install`
- c. Levantar el proyecto
 - i. `ng serve`

Esquema general



Descripción de Carpetas y Archivos

1. **core/**: Contiene elementos globales utilizados en toda la aplicación.
2. **services/**: Servicios generales, por ejemplo, para autenticación y productos.
3. **guards/**: Guards que protegen rutas y controlan la navegación.
4. **interceptors/**: Interceptores HTTP para manejo de peticiones y respuestas.
5. **models/**: Modelos globales que definen la estructura de datos.
6. **shared/**: Contiene componentes, directivas y pipes que se utilizan en diferentes partes de la aplicación.
7. **components/**: Componentes reutilizables como encabezado y pie de página.

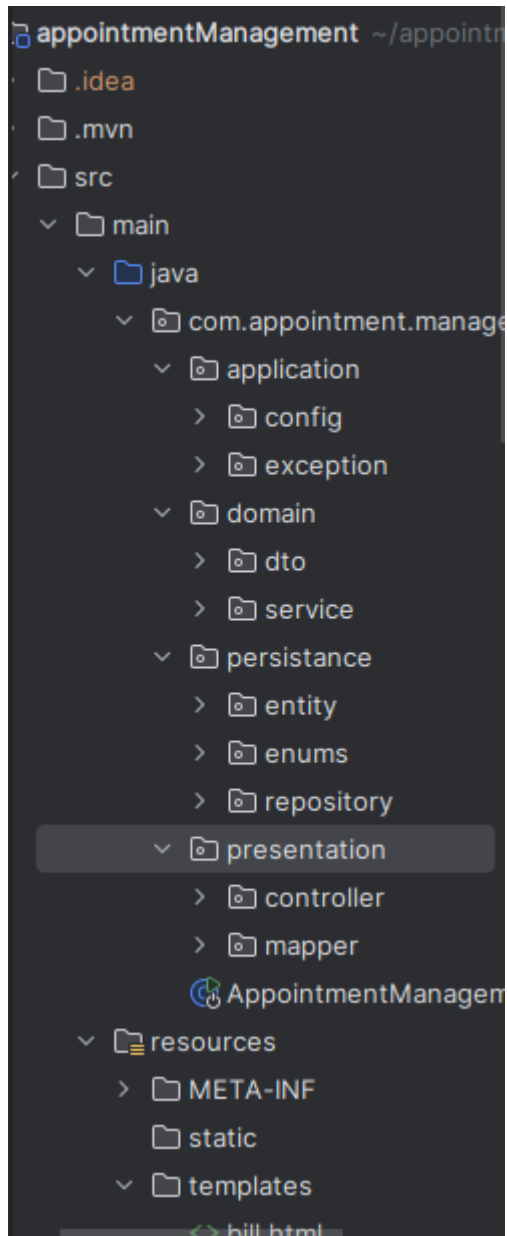
8. **features/**: Contiene módulos específicos para diferentes funcionalidades de la aplicación.
9. **session/**: Funcionalidades de autenticación de usuarios.
10. **products/**: Gestión de productos.
11. **cart/**: Manejo del carrito de compras.
12. **orders/**: Gestión de pedidos.
13. **Manager**: Manejo de Administrador
14. **common_user**: Manejo de usuario comun
15. **assets/**: Archivos estáticos y estilos globales.
16. **environments/**: Archivos de configuración para diferentes entornos (desarrollo, producción).
17. **index.html**: Archivo HTML principal que sirve como el contenedor para la aplicación Angular.
18. **main.ts**: Archivo de entrada principal para Angular que arranca la aplicación.
19. **styles.scss**: Archivo SCSS principal para los estilos globales

Estructura del Proyecto

La estructura del proyecto está organizada para soportar el desarrollo modular y facilitar el mantenimiento. La estructura del proyecto es:

Descripción de Carpetas y Archivos

1. **config/**: Contiene configuraciones de Spring Boot y Beans personalizados.
2. **controller/**: Contiene controladores REST que manejan las solicitudes HTTP.
3. **model/**: Contiene las entidades y modelos que representan las tablas en la base de datos.
4. **repository/**: Contiene interfaces de repositorio para la persistencia y recuperación de datos desde la base de datos.
5. **service/**: Contiene la lógica de negocio y los servicios que operan sobre los datos.
6. **util/**: Contiene utilidades y clases de soporte para el proyecto.



Despliegue:

Proceso de Despliegue

El proceso de despliegue para el backend implica:

1. **Construcción del Proyecto:** Ejecutar `mvn clean package` para crear el archivo JAR.
2. **Despliegue en Servidor:** Subir el archivo JAR al servidor y ejecutarlo con `java -jar archivo.jar`

Configuración del Servidor

Configura el servidor para ejecutar el archivo JAR y manejar las solicitudes HTTP. Asegúrate de que las variables de entorno y las configuraciones de red estén correctamente configuradas.

Mantenimiento y Actualización

- **Actualización de Dependencias**

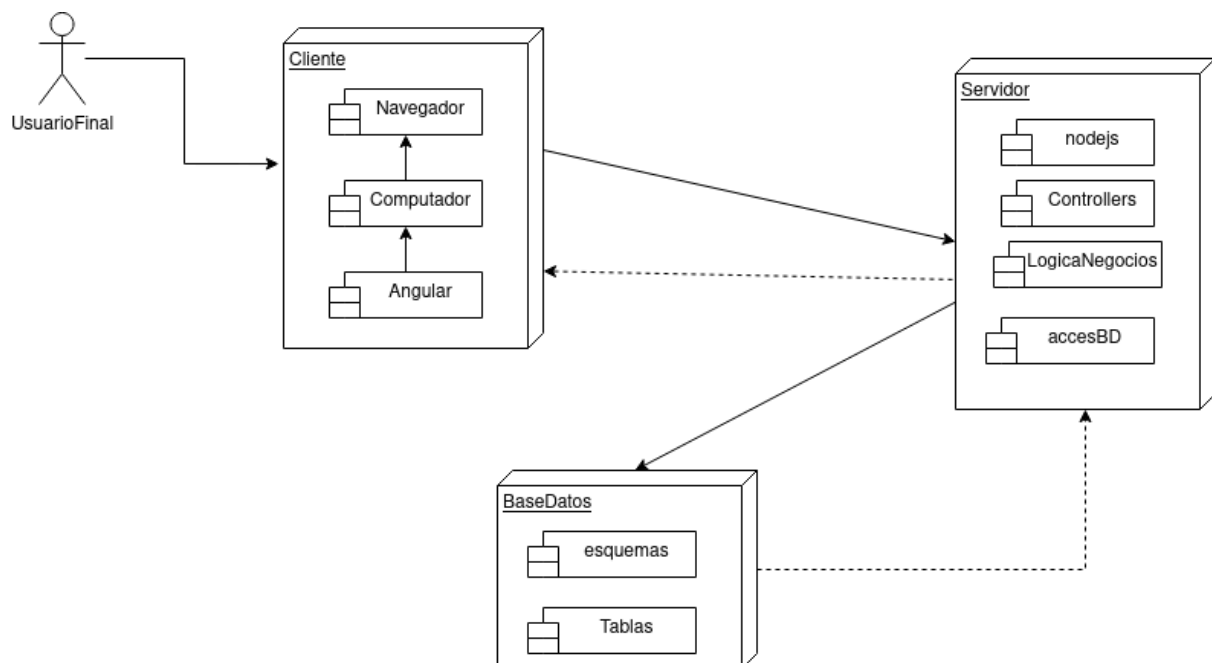
Revisa y actualiza regularmente las dependencias del proyecto para mantener la seguridad y la compatibilidad.

- **Mantenimiento del Código**

Sigue las mejores prácticas de desarrollo para mantener el código limpio y modular.

Realiza revisiones de código y refactoriza según sea necesario.

Secuencia



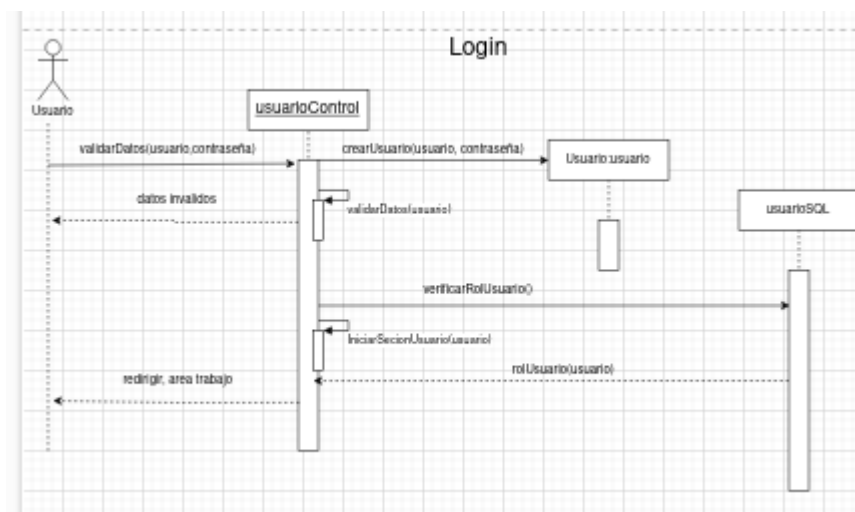
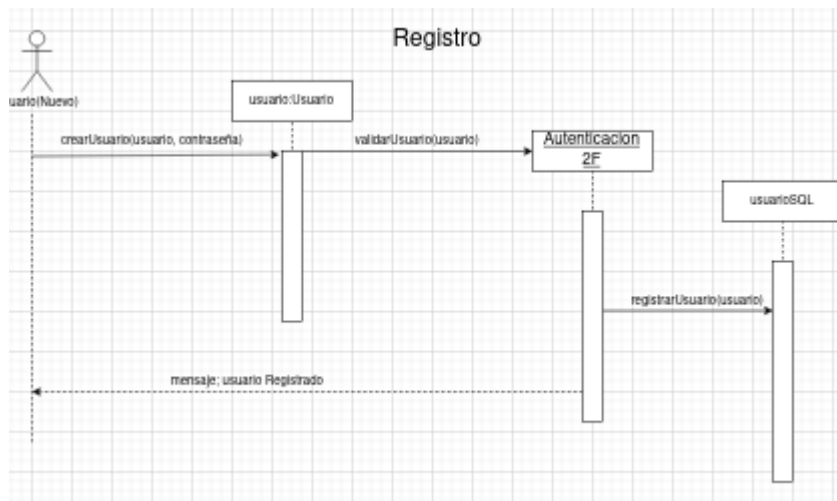
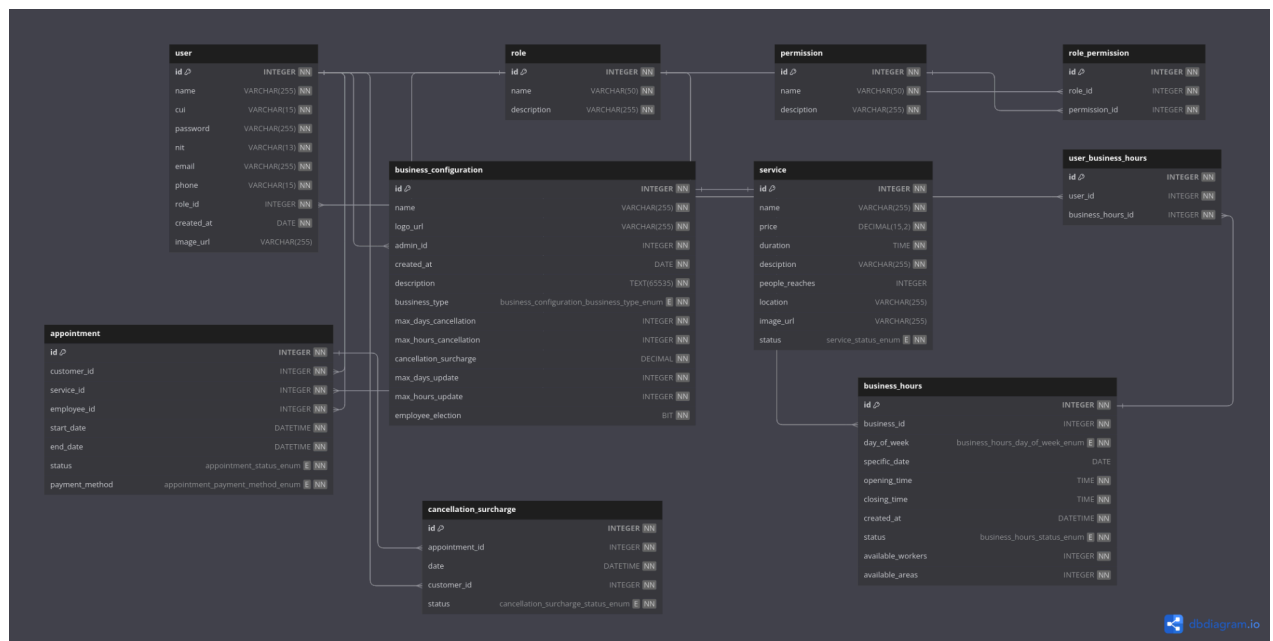
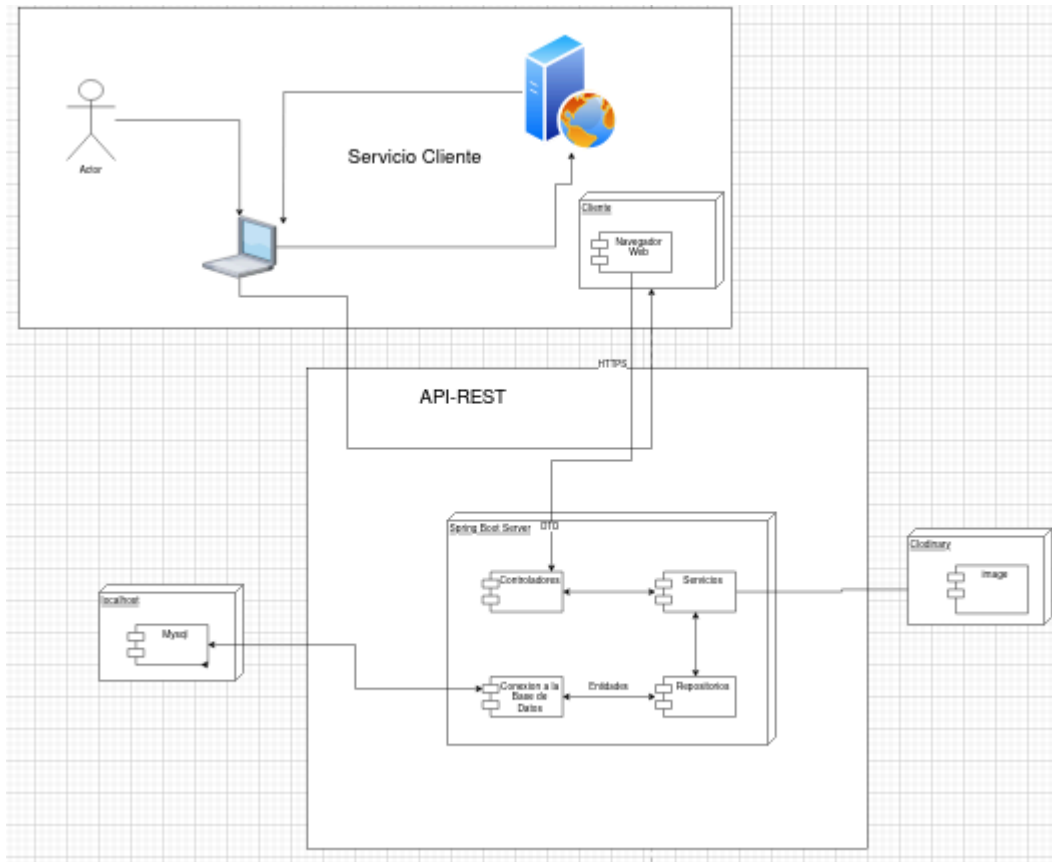


Diagrama de tablas



Arquitectura

Bajo nivel:



Alto nivel

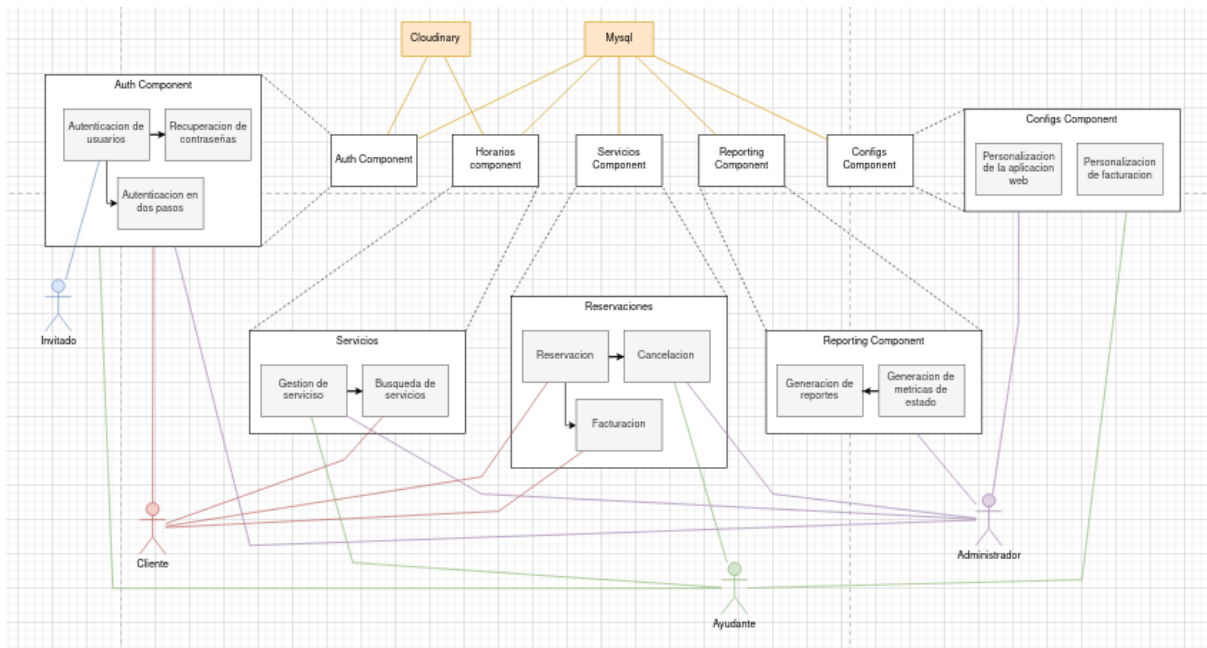
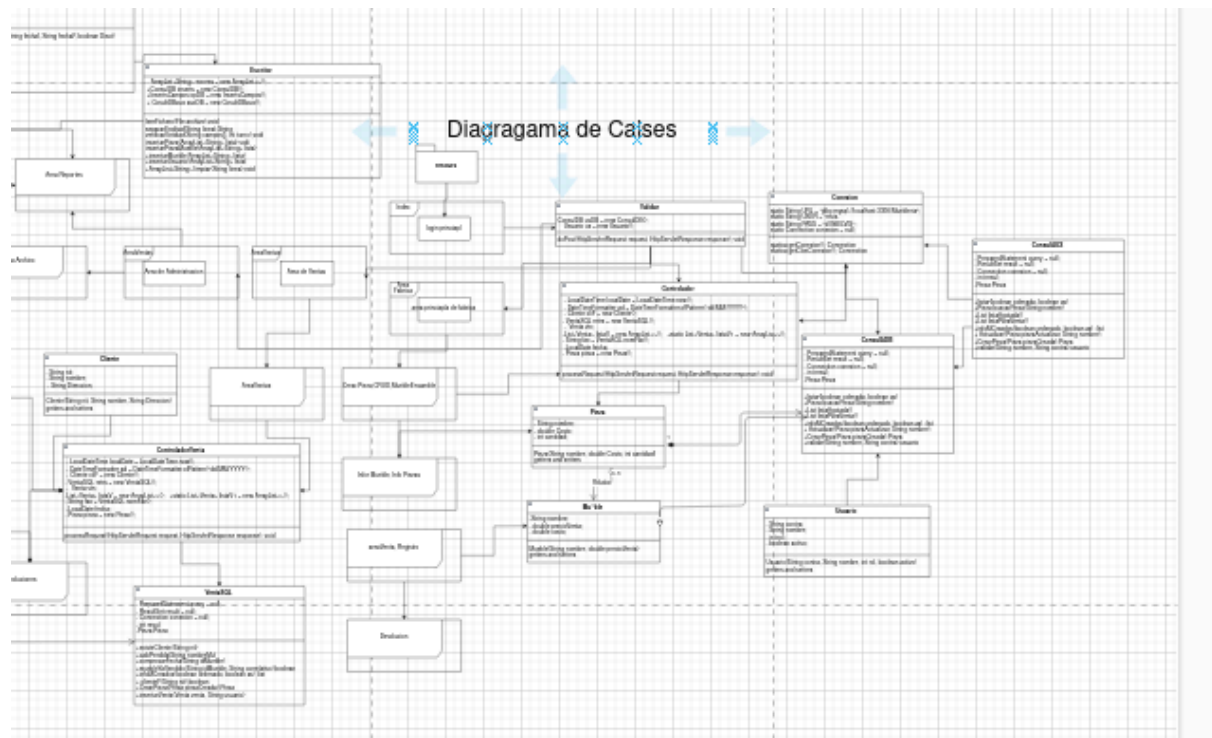


Diagrama de clases



Casos de Uso

Para: Nuevo Cliente

Número:	CU 001
Caso de uso:	Registrarse
Actores:	Cliente
Propósito:	Un nuevo cliente se registre en el sistema
Resumen:	El cliente debe de registrarse proporcionando datos como: Nombre, NIT, CUI, número de teléfono y correo electrónico. presionara el botón de registrarse, seguidamente se le enviará un código de verificación al correo e ingresa con éxito o será rechazado.
Tipo:	Primario y Esencial
Referencia:	CU 004

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas
Llena el formulario con todos los datos que se le solicite.	Todos los datos pre-validados correctamente (no se le muestra nada)	se le muestra una alerta en el campo que no llenó correctamente
Presiona el botón de registrarse	Se le muestra un alerta de todo OK y deberá ingresar el número de verificación que se le envió al correo.	El correo, o datos no son únicos en el sistema, validados en base de datos, alerta de ingrese datos correctos.
Ingresa el código de verificación	será redirigido al home o pantalla principal del sistema.	no lo dejara ingresar hasta que ingrese el código correctamente

Para: Cliente

Número:	CU 002
Caso de uso:	Inicio de sesión
Actores:	Cliente
Propósito:	Ingresar datos para iniciar sesión en el sistema

Resumen:	El cliente ingresa sus credenciales, Correo y contraseña , se le enviará un código de autenticación al correo (google authenticator) y deberá ingresarlo para validar identidad, seguidamente podrá ingresar al sistema.
Tipo:	Primario y Esencial
Referencia:	CU 004

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas
Llena el formulario con todos los datos que se le solicite.	Todos los datos pre-validados correctamente (no se le muestra nada)	se le muestra una alerta en el campo que no llenó correctamente
Presiona el botón de iniciar sesión	Se le envía código de autenticación.	Si el correo, o contraseña no son válidos se le mostrará la alerta correspondiente.
Ingresa el código de verificación	será redirigido al home o pantalla principal del sistema.	no lo dejara ingresar hasta que ingrese el código correctamente

Número:	CU 003
Caso de uso:	Recuperación de contraseña
Actores:	Cliente, ayudante, administrador
Propósito:	El cliente/ayudante/administrador recupere su contraseña
Resumen:	El cliente ingresa su correo , se le enviará un código de autenticación al correo (google authenticator) y deberá ingresarlo para validar identidad, seguidamente podrá ingresar al sistema directamente al formulario de ingreso de nueva contraseña.
Tipo:	Secundario
Referencia:	CU 002

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas

Llena el formulario con todos los datos que se le solicite.	Todos los datos pre-validados correctamente (no se le muestra nada)	se le muestra una alerta en el campo que no llenó correctamente
Presiona el botón de enviar código	Se le envía código de autenticación.	Si el correo, o contraseña no son válidos se le mostrará la alerta correspondiente.
Ingresa el código de verificación	será redirigido al formulario para ingresar la nueva contraseña.	no lo dejara ingresar hasta que ingrese el código correctamente
Llenará el formulario con la nueva contraseña	se validará y se le mostrará la alerta de ok!, se le redirigirá al inicio de sesión	Se le seguirá pidiendo la contraseña hasta ingresar una válida

Número:	CU 004
Caso de uso:	Vista inicial (Home) del cliente
Actores:	Cliente
Propósito:	El cliente podrá visualizar las opciones esenciales para realizar acciones en el sistema.
Resumen:	El cliente podrá ingresar al home, para poder visualizar las opciones principales para interactuar con el sistema, por ejemplo visualizar citas y agendar, cambiar cita, cancelar cita...
Tipo:	Primario
Referencia:	CU 005

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas
Visualizar todas las opciones disponibles en el sistema, así como una descripción.		
Presiona “ir a X opción”	será redirigido a la opción que haya elegido para poder interactuar más con el sistema	

Número:	CU 005
Caso de uso:	Visualización de horarios

Actores:	Cliente
Propósito:	El cliente podrá visualizar los horarios de atención y horarios disponibles
Resumen:	El cliente podrá ingresar a un apartado donde podrá visualizar un calendario con horarios de atención, días de atención, servicios/trabajadores/espacios disponibles (si está habilitado ver esa opción), podrá seleccionar el día en el calendario si está disponible, y ver más detalles de disponibilidad.
Tipo:	Primario y esencial
Referencia:	CU 006

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas
visualizará un calendario donde se le muestre los días hábiles de trabajo del negocio, así mismo podrá ver los horarios de atención.	Horarios de atención, días de atención.	
Selecciona un día habil	horarios disponibles para reservar. Tipos de servicios disponibles, duración de estos servicios. Lista de empleados para esos servicios (si está activado).	

Número:	CU 006
Caso de uso:	Agendar Cita
Actores:	Cliente
Propósito:	El cliente podrá agendar su cita al horario que mejor que convenga
Resumen:	Se le mostrará los horarios ya reservados, así mismo se dará la opción de seleccionar un horario disponible para reservarlo, tomando en cuenta que las horas aparecen según los servicios, es decir si hay servicios de 30 min, los horarios disponibles serán cada 30 min. al seleccionar este horario se validará para evitar mala concurrencia y se le mostrará una alerta de que ya está asignada en ese horario.

Tipo:	Primario y esencial
Referencia:	CU 007

Acción de Actores	Respuesta del sistema	
	Correctas	Erróneas/alternas
Al seleccionar una hora habilitada y disponible, deberá confirmar los siguientes aspectos, tipo de servicio, empleado que lo atiende (si está habilitada la opción), como lo pagará, tarjeta o efectivo.	Se validaron todos los datos y se registrará la cita, así mismo se le muestra una alerta de que ya ha sido agendada.	El espacio ya fue agendado por alguien más, justo antes que ese cliente.