UNIVERSIDAD NACIONAL AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES

SEGUNDO EXAMEN PARCIAL SI-2

Sistema de gestión de historia clínica electrónicas

GRUPO: 19

Integrantes:

- Aliaga Valencia Jorge Arturo 218166141
- Manzoni Bravo Fabiana 221180478

Materia: Sistemas de Información II

Docente: Ing. Angelica Garzon

Fecha: 17/11/2024

CONTENIDO

1. FUNDAMENTOS TEÓRICOS..... 7
1.1 Introducción..... 7
1.2 Objetivo General..... 8
1.3 Objetivos Específicos..... 8
1.4 Alcance..... 8
2. MARCO TEÓRICO REFERENCIAL Y METODOLÓGICO..... 10
2.1. Marco Referencial..... 10
2.2. Marco del Trabajo Agil SCRUM..... 11
2.2.1. Definicion de SCRUM..... 11
2.2.2. Roles en SCRUM..... 11
2.2.3. Eventos y Artefactos de SCRUM..... 11
2.2.4. Aplicacion de SCRUM..... 12
3. HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO..... 13
2.1. Lenguaje de Programación..... 13
2.2. Frameworks y entornos de ejecución..... 13
2.3. Sistema de gestor de base de datos..... 14
2.4. Lenguaje de modelado de software..... 14
2.5. Herramienta de diseño y modelado..... 15
2.6. Entorno de Desarrollo..... 16
2.7. Infraestructura de software (IaaS)..... 16
2.7.1 Servicio en la nube (Cloud Computing)..... 16
2.7.2 SAAS (Software as a Service, SaaS)..... 16
2.8. Herramientas Colaborativas para Seguimientos de Proyectos..... 16
2.9. Sistema de control de Versiones de código..... 17
2.10. Herramientas de Gestión de Código en la Nube..... 17
4. REQUERIMIENTOS..... 17
4.1 Propósitos..... 17
4.2 Ámbito del Sistema..... 17
1. Ámbito Clínico..... 17
2. Ámbito Administrativo..... 18
3. Ámbito del Asegurado (Paciente)..... 18
4. Ámbito de Seguridad y Gestión de Usuarios..... 18
5. Ámbito de Reportes y Análisis..... 18
4.3 Definiciones, Acrónimos y Abreviaturas..... 19
4.4 Funciones del Producto..... 19
4.5 Product Backlog..... 20

4.6 Requisitos Funcionales..... 25
4.7 Requisitos no Funcionales..... 26
4.8 Lista de casos de uso (Web y Móvil)..... 26
4.9 Paquetes y casos de Uso..... 29
4.10 Planificación Sprint (Diagrama de Gantt)..... 31
5. PROCESO DE DESARROLLO DE SOFTWARE..... 41
Sprint 1..... 41
1. Sprint Planning..... 41
1.1. Objetivos del Sprint..... 41
1.1.1 Casos de uso..... 41
1.2. Historias de usuario..... 41
CU1: Gestión de Usuarios..... 41
CU2: Gestión de Roles..... 42
CU3: Gestión de Permisos..... 43
CU4: Gestionar Información de Pacientes..... 44
CU5: Iniciar Sesión (WEB)..... 45
CU5: Iniciar Sesión (MOVIL)..... 46
CU6: Cerrar Sesión..... 47
CU7: Recuperar Contraseña..... 48
1.3. Contexto del sistema..... 48
1.4. Sprint Backlog..... 48
1.5. Equipo SCRUM..... 50
2. Proceso/patrón de desarrollo por Historia de Usuario..... 52
2.1. Diseño..... 52
2.1.1. Diseñar de la arquitectura..... 52
a) Diagrama del Contexto del Sistema..... 52
b) Diagrama de Contenedores..... 53
c) Diagrama de Componentes..... 54
d) Diagrama de Código..... 54
2.1.2. Diseño de datos..... 54
a) Diagrama de clases..... 54
b) Script..... 55
c) Mapeo..... 56
2.1.3. Diseño de la lógica de negocio..... 57
a) Diagrama de Secuencia y Diseño de Clase..... 57
b) Diagrama de Estado..... 62
c) Diagrama de Tiempo..... 62
d) Diagrama de Navegación..... 66

2.2 Implementación..... 67
2.2.1. Componentes y artefactos generados..... 67
2.2.1.1 Diagrama de componentes.....67
2.3. Pruebas..... 67
2.3.1. Plan de pruebas (criterios de aceptación).....68
2.3.2. Reporte de prueba..... 71
3. Daily Scrum..... 76
4. Sprint Review..... 79
5. Sprint Retrospective..... 80
6. Burndown y BurnUp (Grafica de tareas y Datos de tareas).....81
7. Grafica de esfuerzo y Datos de esfuerzo.....82
8. Scrum TaskBoard (Backlog, to do, doing, done)..... 84
Sprint 2..... 85
1. Sprint Planning..... 85
1.1. Objetivos del Sprint.....85
1.1.1 Casos de uso..... 85
1.2. Historias de usuario..... 85
CU8: Gestión de Especialidades Médicas..... 85
CU9: Gestión de Médicos..... 86
CU10: Gestionar Horarios de Atención de Médicos..... 87
CU11: Gestionar Reserva..... 88
CU12: Gestionar Consultas Médicas..... 89
CU13: Solicitar Permiso..... 90
CU14: Gestionar Postergación..... 91
CU15: Gestionar Receta Médica..... 92
CU16: Programar Laboratorio.....93
CU17: Gestionar Laboratorio..... 94
CU18: Gestionar Medicamentos..... 95
CU19: Visualizar Historia Clínica..... 96
CU25: Enviar Recordatorio de Cita al Paciente..... 98
1.3. Contexto del sistema..... 98
1.4. Sprint Backlog..... 99
1.5. Equipo SCRUM..... 102
2. Proceso/patrón de desarrollo por Historia de Usuario..... 105
2.1. Diseño..... 105
2.1.1. Diseño de la arquitectura..... 105
a) Diagrama del Contexto del Sistema..... 105
b) Diagrama de Contenedores..... 106

c) Diagrama de Componentes..... 106
d) Diagrama de Código..... 107
2.1.2. Diseño de datos..... 108
a) Diagrama de clases..... 108
b) Script..... 108
c) Mapeo..... 113
2.1.3. Diseño de la lógica de negocio.....115
a) Diagrama de Secuencia y Diseño de Clase..... 115
b) Diagrama de Estado..... 128
c) Diagrama de Tiempo..... 129
d) Diagrama de Navegación..... 137
2.2 Implementación..... 138
2.2.1. Componentes y artefactos generados..... 138
2.2.1.1 Diagrama de componentes.....138
2.3. Pruebas..... 138
2.3.1. Plan de pruebas (criterios de aceptación).....138
2.3.2. Reporte de prueba..... 142
CU8: Gestión de Especialidades Médicas.....142
CU9: Gestión de Médicos..... 142
CU10: Gestionar Horarios de Atención de Médicos..... 143
CU11: Gestionar Reserva..... 144
CU12: Gestionar Consultas Médicas.....145
CU13: Solicitar Permiso.....145
CU14: Gestionar Postergación..... 146
CU15: Gestionar Receta Médica..... 146
CU16: Programar Laboratorio..... 146
CU17: Gestionar Laboratorio..... 147
CU18: Gestionar Medicamentos.....147
CU19: Visualizar Historia Clínica..... 148
CU23: Enviar Recordatorio de Cita al Paciente.....149
3. Daily Scrum..... 150
4. Sprint Review..... 154
5. Sprint Retrospective..... 157
6. Burndown y BurnUp (Grafica de tareas y Datos de tareas).....159
7. Grafica de esfuerzo y Datos de esfuerzo.....160
8. Scrum TaskBoard (Backlog, to do, doing, done)..... 161
Sprint 3..... 163
1. Sprint Planning.....163

1.1. Objetivos del Sprint....163
1.1.1 Casos de uso....163
1.2. Historias de usuario....163
CU20: Gestionar Reportes Personalizados....163
CU21: Visualizar Bitácora....164
CU22: Evaluación de Médicos....165
CU24: Exportar historia clínica....166
CU25: Gestión de Pagos de Seguros....167
CU26: Gestión de Método de Pago....168
CU27: Visualizar Historial de Pagos....169
1.3. Contexto del sistema....170
1.4. Sprint Backlog....170
1.5. Equipo SCRUM....172
2. Proceso/patrón de desarrollo por Historia de Usuario....174
2.1. Diseño....174
2.1.1. Diseñar de la arquitectura....174
a) Diagrama del Contexto del Sistema....174
b) Diagrama de Contenedores....175
c) Diagrama de Componentes....175
d) Diagrama de Código....176
2.1.2. Diseño de datos....177
a) Diagrama de clases....177
b) Script....177
c) Mapeo....182
2.1.3. Diseño de la lógica de negocio....184
a) Diagrama de Secuencia y Diseño de Clase....184
b) Diagrama de Estado....191
c) Diagrama de Tiempo....192
d) Diagrama de Navegación....196
2.2 Implementación....198
2.2.1. Componentes y artefactos generados....198
2.2.1.1 Diagrama de componentes....198
2.3. Pruebas....198
2.3.1. Plan de pruebas (criterios de aceptación)....198
2.3.2. Reporte de prueba....200
CU20: Gestionar Reportes Personalizados....200
CU21: Visualizar Bitácora....201
CU22: Evaluación de Médicos....201
CU24: Exportar historia clínica....202
CU25: Gestión de Pagos de Seguros....202
CU26: Gestión de Método de Pago....203

CU27: Visualizar Historial de Pagos..... 203
3. Daily Scrum.....204
4. Sprint Review.....207
5. Sprint Retrospective.....208
6. Burndown y BurnUp (Grafica de tareas y Datos de tareas).....210
7. Gráfica de esfuerzo y Datos de esfuerzo.....211
8. Scrum TaskBoard (Backlog, to do, doing, done).....212
CONCLUSIONES..... 214
BIBLIOGRAFIA..... 214
SOFTWARE (Avance en cada presentación).....214
Sprint 1: Fundamentos del Sistema.....214
Sprint 2: Funcionalidades Médicas..... 215
Sprint 3: Reportes y Funcionalidades Avanzadas..... 215
PEQUEÑO MANUAL..... 216
ANEXOS.....226

1. FUNDAMENTOS TEÓRICOS

1.1 Introducción

El Seguro de Salud Vida Sana (SSVS) requiere un sistema de gestión de historias clínicas electrónicas que permita optimizar el proceso de atención médica, garantizando la seguridad y trazabilidad de la información de los pacientes. El sistema debe gestionar eficientemente la solicitud de fichas de atención para especialidades médicas, la actualización de las historias clínicas electrónicas por parte de los médicos, y la restricción de acceso a la información clínica según el rol de los usuarios. Esto implica el desarrollo de un sistema integral que cubra tanto las necesidades administrativas como operativas, asegurando la correcta gestión de consultas, horarios de atención y usuarios.

El presente proyecto tiene como finalidad desarrollar una solución tecnológica que gestione de manera integral las historias clínicas electrónicas de los pacientes afiliados al SSVS. Los pacientes podrán solicitar fichas de atención para una especialidad médica específica, y los médicos podrán actualizar sus historias clínicas tras cada consulta. Esta actualización incluirá la carga de exámenes, informes médicos, imágenes de diagnóstico, recetas y tratamientos. Todo esto debe realizarse bajo un sistema de permisos y roles que garantice la seguridad de los datos y restrinja el acceso y la modificación de la información clínica a los actores autorizados.

Una de las particularidades del sistema es que los médicos solo tendrán acceso a las historias clínicas de los pacientes que atendieron en un día específico, sin la posibilidad de modificar registros previos, lo que garantiza la trazabilidad de la información. Asimismo, los pacientes tendrán acceso a su historia clínica para consulta, pero no podrán realizar modificaciones sobre la misma. El sistema deberá ser capaz de generar reportes detallados de atención médica, facilitando la toma de decisiones tanto para la administración del SSVS como para el propio paciente.

Además de las funcionalidades médicas, la solución debe incluir la gestión de especialidades, médicos y horarios de atención, asegurando que el flujo de atención sea ágil y organizado. También debe estar diseñada para ser accesible tanto en su versión web como móvil, con interfaces amigables que faciliten el uso del sistema por parte de los usuarios, independientemente de sus conocimientos tecnológicos.

En resumen, este proyecto tiene como objetivo no solo mejorar la calidad y eficiencia de la atención médica a través de la digitalización, sino también garantizar la seguridad y privacidad de los datos médicos de los pacientes, cumpliendo con los estándares exigidos para la gestión de información clínica.

1.2 Objetivo General

Desarrollar un sistema de gestión de historias clínicas electrónicas para el Seguro de Salud Vida Sana (SSVS), que permita gestionar las consultas médicas, actualizar las historias clínicas electrónicas y administrar las solicitudes de fichas de atención, con roles de usuario bien definidos (Administrador, Médico y Paciente), garantizando la integridad y seguridad de los datos médicos.

1.3 Objetivos Específicos

- **Desarrollar la gestión de especialidades médicas** para que los pacientes puedan solicitar atención en diferentes áreas de la salud, según sus necesidades.
- **Implementar la gestión de médicos**, incluyendo la creación, modificación y eliminación de información sobre médicos, así como la asignación de horarios de atención.
- **Crear un sistema de gestión de fichas de atención** que permita a los pacientes solicitar fichas para un día específico, limitando el número de fichas por día y especialidad.
- **Implementar la gestión de la historia clínica electrónica**, permitiendo que los médicos actualicen la historia clínica de los pacientes de manera segura, sin modificar información previamente registrada.
- **Desarrollar la gestión de usuarios**, estableciendo roles de acceso para Administrador, Médico y Paciente, y restringiendo acciones según el rol asignado.
- **Generar reportes personalizados** que permitan al administrador visualizar la atención de pacientes por diferentes criterios como fechas, médico, especialidad, y paciente.
- **Permitir la consulta de la historia clínica por parte del paciente**, garantizando que pueda acceder a sus datos clínicos, pero sin la posibilidad de modificarlos.
- **Desarrollar una interfaz amigable** tanto para la web como para la aplicación móvil, priorizando la facilidad de uso y la accesibilidad de las funciones esenciales del sistema.

1.4 Alcance

Módulo de Usuarios

- **RF1.** Gestión de Usuarios:
    - Permitir el CRUD de usuarios (crear, leer, actualizar y eliminar) para la administración general.
- **RF2.** Administración de Roles y Permisos:
    - CRUD de roles específicos como Administrador, Médico y Paciente.
    - Configuración de permisos específicos para cada rol para garantizar un acceso controlado a las funcionalidades.
- **RF3.** Autenticación y Autorización:
    - Control de acceso basado en roles mediante autenticación con Spring Security.

    - Manejo de sesiones y tokens para garantizar la seguridad en el acceso a la plataforma.

Módulo de Atención Médica

- **RF4.** Gestión de Especialidades Médicas:
    - CRUD de especialidades médicas, permitiendo agregar, editar y eliminar especialidades según los servicios disponibles.
- **RF5.** Gestión de Médicos:
    - CRUD de médicos con asignación de especialidades.
    - Configuración de horarios de atención específicos para cada médico.
- **RF6.** Gestión de Horarios de Atención:
    - CRUD de horarios para cada médico, definiendo la disponibilidad en días y horas específicos.
- **RF7.** Obtención de Fichas de Atención:
    - Permitir a los pacientes reservar una ficha de atención por día, asociada a un médico y horario específico.
    - Limitación de disponibilidad de fichas por médico para evitar sobrecarga.
- **RF8.** Gestión de Atención de Consulta:
    - Registro de la consulta médica actual en la historia clínica del paciente.
    - Incluye datos de diagnóstico, tratamientos y notas, con restricciones para no modificar registros de consultas previas.

Módulo de Historia Clínica

- **RF9.** Visualización de Historia Clínica:
    - Los médicos pueden ver el historial completo de cada paciente en formato de solo lectura para consultas anteriores.
    - Los pacientes pueden visualizar su propio historial en formato de solo lectura.
- **RF10.** Registro de Exámenes y Documentos:
    - Permitir que el médico cargue y visualice exámenes (como sangre, azúcar, orina) y documentos de imagen (como ecografías, radiografías) asociados a la consulta actual.

Módulo de Reportes

- **RF11.** Reportes para Administración:
    - Generación de reportes sobre la atención a pacientes, con criterios de filtrado como periodo de fechas, paciente, médico o especialidad.
- **RF12.** Exportación de Reportes en PDF/Excel:
    - Exportación de los reportes generados en formatos PDF y Excel para facilitar el análisis y la documentación.

2. MARCO TEÓRICO REFERENCIAL Y METODOLÓGICO

2.1. Marco Referencial

Historia Clínica Electrónica (HCE)
Una HCE es el registro digital que contiene toda la información relevante sobre la salud de un paciente a lo largo del tiempo. Este tipo de sistema digital permite la gestión de datos clínicos, mejorando el acceso, la actualización y el almacenamiento de la información médica. La HCE incluye desde los antecedentes médicos hasta los diagnósticos, tratamientos, resultados de exámenes y otras interacciones con el sistema de salud.

Ficha Médica
Es un documento utilizado para registrar la solicitud de atención médica de un paciente en una consulta o especialidad determinada. Las fichas médicas son esenciales para coordinar la atención médica, evitando la sobrecarga de consultas y asignando horarios específicos a los pacientes y médicos.

Sistemas de Gestión Documental
Son plataformas tecnológicas que permiten almacenar, organizar y recuperar documentos de manera eficiente. En el contexto médico, un sistema de gestión documental permite el almacenamiento de informes, análisis clínicos y exámenes médicos (como ecografías o radiografías), asegurando la disponibilidad y accesibilidad de estos documentos en el momento que el profesional médico lo requiera.

Autenticación y Control de Acceso
En cualquier sistema digital de información sensible, como las HCE, es fundamental la implementación de medidas de seguridad que controlen quién accede a qué información. La autenticación es el proceso de verificar la identidad de los usuarios, mientras que el control de acceso garantiza que solo los usuarios autorizados (según su rol) puedan ver o modificar la información relevante.

Trazabilidad en Sistemas Médicos
La trazabilidad es la capacidad de seguir el rastro de todas las acciones realizadas dentro del sistema de HCE. Esto es crucial para garantizar que toda modificación en la historia clínica de un paciente sea transparente y rastreable, permitiendo identificar quién realizó cada actualización y cuándo se llevó a cabo.

2.2. Marco del Trabajo Agil SCRUM

2.2.1. Definicion de SCRUM

SCRUM es un marco de trabajo ágil utilizado para gestionar proyectos complejos, especialmente en el ámbito del desarrollo de software. Se caracteriza por ser iterativo e incremental, lo que significa que el trabajo se realiza en ciclos cortos llamados Sprints, permitiendo revisiones frecuentes y ajustes en el proyecto. Este enfoque facilita la adaptación a cambios en los requisitos y asegura que el equipo se mantenga alineado con las expectativas del cliente.

Los principios básicos de SCRUM incluyen:

- Transparencia: Todos los aspectos del proceso deben ser visibles para todos los involucrados.
- Inspección: Revisiones frecuentes del progreso permiten detectar desviaciones y hacer correcciones rápidamente.
- Adaptación: Basado en la inspección, el proceso debe ser ajustado para mejorar continuamente.

2.2.2. Roles en SCRUM

SCRUM define tres roles clave que son esenciales para el éxito del proyecto:

- Product Owner: Es el responsable de definir los requisitos y priorizar las tareas en el Product Backlog. Representa los intereses del cliente y asegura que el equipo trabaje en lo que más valor aporta al proyecto.
- Scrum Master: Actúa como facilitador del equipo, eliminando impedimentos y asegurando que se sigan las prácticas ágiles. El Scrum Master no dirige al equipo, sino que lo ayuda a autogestionarse y a mejorar continuamente.
- Development Team: Es el equipo de trabajo que se encarga de desarrollar el producto. Es multifuncional y autónomo, lo que significa que puede hacer todo lo necesario para entregar un incremento de producto funcional al final de cada Sprint.

2.2.3. Eventos y Artefactos de SCRUM

SCRUM utiliza varios eventos y artefactos que estructuran el trabajo y aseguran una entrega constante de valor:

- Sprint Planning: Reunión al inicio de cada Sprint donde se planifica el trabajo que se va a realizar. Se define el Sprint Goal y se seleccionan las tareas del Product Backlog que se van a completar.
- Daily Standup (Daily Scrum): Reunión diaria breve donde el equipo sincroniza sus actividades y ajusta su plan para el día.
- Sprint Review: Al final del Sprint, el equipo presenta lo que ha completado, y se recoge feedback del Product Owner y otros interesados.
- Sprint Retrospective: Reunión para reflexionar sobre lo que fue bien, lo que no fue tan bien, y cómo mejorar en el próximo Sprint.

Artefactos clave:

- Product Backlog: Lista priorizada de todas las tareas y características que se deben implementar. Es dinámico y se actualiza continuamente.
- Sprint Backlog: Lista de tareas seleccionadas del Product Backlog que se completarán en un Sprint específico.
- Incremento: Es el resultado final de un Sprint, que debe ser un producto funcional y potencialmente entregable.

2.2.4. Aplicacion de SCRUM

La metodología SCRUM se aplicará de la siguiente manera:

- Estructura de los Sprints: Organización del trabajo en Sprints cortos, donde al final de cada Sprint se revise el progreso y se ajusten las prioridades según sea necesario.
- Gestión del Product Backlog: Debido al enfoque, el Product Backlog incluirá tareas relacionadas con la investigación, desarrollo de la aplicación, integración con otros sistemas y pruebas.
- Roles en el equipo: Les roles de Product Owner, Scrum Master y Development Team se definen claramente, aunque un mismo miembro pueda desempeñar más de un rol en distintos sprints.
- Eventos SCRUM: Realización de reuniones diarias para mantener la sincronización del equipo y al final de cada Sprint, se realiza una revisión y retrospectiva para mejorar continuamente.

3. HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO

2.1. Lenguaje de Programación

Web: Java y JavaScript son los lenguajes principales utilizados para desarrollar la aplicación web. Con herramientas como Spring boot y Angular, se podrá construir tanto la interfaz de usuario (frontend) como la lógica del servidor (backend).

Móvil: Se utilizara Dart como lenguaje de desarrollo para la aplicación web.

2.2. Frameworks y entornos de ejecución

Frameworks:

Angular: Angular será utilizado para desarrollar la interfaz web del sistema. Este framework de desarrollo frontend, basado en TypeScript, permite la creación de aplicaciones web dinámicas y modulares. Proporciona una estructura robusta para el desarrollo de interfaces de usuario, con un enfoque en la creación de componentes reutilizables y una eficiente gestión del estado.

Spring Boot: Spring Boot será el framework principal para el desarrollo del backend. Es un framework basado en Java que facilita la creación de aplicaciones web y APIs RESTful. Spring Boot ofrece una configuración simplificada y es altamente escalable, lo que lo convierte en una excelente opción para manejar la lógica de negocio del sistema, las interacciones con la base de datos y la seguridad.

Seguridad

Spring Security: Este módulo de Spring Boot se utilizará para gestionar la autenticación y autorización en el sistema. Implementará un sistema de control de acceso basado en roles (Administrador, Médico y Paciente) y garantizará que solo los usuarios con los permisos adecuados puedan acceder a la información sensible del sistema. Se implementará seguridad con tokens JWT (JSON Web Tokens) para manejar la autenticación de forma eficiente y segura.

Framework para Aplicación Móvil

Flutter: Flutter será utilizado para desarrollar la aplicación móvil dirigida a los asegurados (pacientes). Este framework, creado por Google, permite el desarrollo de aplicaciones móviles nativas para Android e iOS utilizando un único código base, lo que facilita el despliegue multiplataforma. Los pacientes podrán utilizar la aplicación móvil para solicitar fichas de atención, consultar sus historias clínicas y realizar otras operaciones.

Entornos de Ejecución

Backend

Java JDK (Java Development Kit): El entorno de ejecución para el backend será el JDK de Java, que permite compilar y ejecutar el código Java. Es esencial para ejecutar el framework Spring Boot y gestionar la lógica de negocio del sistema.

Frontend

Node.js: Node.js será el entorno de ejecución utilizado para el desarrollo con Angular. Permite ejecutar el código de JavaScript y TypeScript en el servidor, así como gestionar las dependencias del proyecto frontend.

NPM (Node Package Manager): NPM se utilizará para la gestión de las dependencias de Angular. Facilitará la instalación de paquetes y herramientas necesarias para el desarrollo del frontend, como bibliotecas de interfaz de usuario y herramientas de construcción.

2.3. Sistema de gestor de base de datos

PostgreSQL
Un sistema de gestión de bases de datos relacional y de código abierto que se destaca por su robustez, escalabilidad y soporte para consultas complejas, transacciones ACID, y extensibilidad. PostgreSQL es ideal para aplicaciones que requieren un alto nivel de integridad de datos y flexibilidad.

2.4. Lenguaje de modelado de software

Lenguaje de Modelado Unificado (UML): UML es un lenguaje de modelado que se utiliza para especificar, visualizar y documentar los componentes de un sistema de software. Los diagramas UML pueden representar la arquitectura del software, los procesos, las interacciones entre componentes, y más.

Diagrama C4: Un modelo simplificado que proporciona una vista de alto nivel de un sistema de software. Se enfoca en describir la arquitectura de un sistema utilizando cuatro niveles de abstracción: Contexto, Contenedores, Componentes y Código.

- **Diagrama de Contexto**

Propósito: Ofrecer una visión general del sistema en relación con su entorno externo.

Contenido: Muestra el sistema como una caja negra y sus interacciones con los actores externos (usuarios, otros sistemas, etc.). Indica cómo el sistema se comunica y se integra con los elementos externos.

- **Diagrama de Contenedores**

Propósito: Desglosar el sistema en contenedores de alto nivel que representan aplicaciones o servicios que ejecutan el sistema.

Contenido: Ilustra los contenedores (como aplicaciones web, aplicaciones móviles, servicios backend, bases de datos) y cómo interactúan entre sí y con los actores externos. Cada contenedor tiene una responsabilidad específica dentro del sistema.

- **Diagrama de Componentes**

Propósito: Detallar la estructura interna de cada contenedor.

Contenido: Desglosa los contenedores en componentes más pequeños, que pueden ser clases, módulos, o servicios dentro de un contenedor. Muestra cómo estos componentes interactúan entre sí.

- **Diagrama de Código**

Propósito: Proporcionar una vista detallada del código fuente y su organización.
Contenido: Ofrece una representación detallada de cómo se implementan los componentes a nivel de código. Puede incluir diagramas de clases o diagramas de paquetes.

2.5. Herramienta de diseño y modelado

Draw.io: Una herramienta de diagramación en línea que permite crear diagramas UML, diagramas C4, diagramas de flujo, mapas mentales, y otros tipos de diagramas técnicos y de negocios. Es muy utilizada en el diseño y modelado de sistemas de software.

2.6. Entorno de Desarrollo

Visual Studio Code (VS Code) : Un editor de código fuente altamente personalizable y de código abierto desarrollado por Microsoft. Es ligero pero potente, con soporte para la depuración, control de versiones, y una amplia gama de extensiones que mejoran su funcionalidad.

IntelliJ IDEA: Es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad y edición comercial.

2.7. Infraestructura de software (IaaS)

2.7.1 Servicio en la nube (Cloud Computing)

Railway: Es una plataforma de nube que facilita el despliegue y gestión de aplicaciones web y bases de datos. Ofrece una infraestructura como servicio (IaaS), permitiendo a los desarrolladores implementar proyectos de manera rápida y sencilla, sin necesidad de preocuparse por la configuración del servidor o la infraestructura subyacente. Railway soporta múltiples lenguajes de programación y servicios de bases de datos, con una interfaz intuitiva y automatización de flujos de trabajo, lo que la convierte en una opción popular para proyectos de desarrollo ágil y despliegues continuos.

2.7.2 SAAS (Software as a Service, SaaS)

GitHub: Una plataforma de alojamiento para el control de versiones utilizando Git. GitHub facilita la colaboración en proyectos de software mediante la administración del código fuente, el seguimiento de cambios y la integración continua. Es ampliamente utilizado por desarrolladores para gestionar y compartir código.

2.8. Herramientas Colaborativas para Seguimientos de Proyectos

Jira: Jira es una herramienta de software de gestión de proyectos desarrollada por Atlassian, utilizada principalmente para seguimiento de tareas, errores y gestión de proyectos ágiles como Scrum y Kanban. Permite a los equipos organizar, asignar y monitorear tareas en tiempo real mediante flujos de trabajo personalizables. También ofrece integración con otras herramientas como GitHub, Bitbucket y Slack, además de proporcionar informes y métricas detalladas para evaluar el progreso y el rendimiento del equipo.

2.9. Sistema de control de Versiones de código

Git: Un sistema de control de versiones distribuido que permite a los desarrolladores rastrear cambios en el código fuente, colaborar en equipo y revertir cambios si es necesario. Git es la base de plataformas como GitHub y es esencial para la gestión del código en proyectos de software.

2.10. Herramientas de Gestión de Código en la Nube

GitHub: Además de ser una plataforma para control de versiones, GitHub ofrece herramientas para la integración continua, la entrega continua (CI/CD), y la colaboración en equipo. Los repositorios en GitHub pueden ser públicos o privados, y permiten gestionar todo el ciclo de vida del desarrollo de software.

4. REQUERIMIENTOS

4.1 Propósitos

El propósito del Sistema de Gestión de Historias Clínicas Electrónicas del Seguro de Salud Vida Sana (SSVS) es optimizar y modernizar los procesos de atención médica mediante la digitalización y centralización de la información clínica de los pacientes. A través de la implementación de un sistema integral, se busca mejorar la eficiencia en la gestión de fichas de atención, facilitar el acceso seguro a las historias clínicas electrónicas por parte de los médicos, y ofrecer a los asegurados una plataforma accesible para consultar su información médica. Además, el sistema permitirá al personal administrativo generar reportes detallados que contribuyan a la toma de decisiones estratégicas, garantizando un servicio de salud más ágil, seguro y de calidad para todos los usuarios.

4.2 Ámbito del Sistema

El **Sistema de Gestión de Historias Clínicas Electrónicas del Seguro de Salud Vida Sana (SSVS)** abarca diferentes ámbitos que determinan las áreas funcionales y los usuarios involucrados en su operación. Los ámbitos del sistema permiten una clara delimitación de las funcionalidades y responsabilidades de cada actor, asegurando una correcta implementación y uso del sistema. A continuación, se describen los principales ámbitos:

1. Ámbito Clínico

Este ámbito está centrado en los médicos y su interacción con las historias clínicas electrónicas. Los médicos podrán acceder y actualizar la información médica de los pacientes durante las consultas, incluyendo diagnósticos, tratamientos, y exámenes. Sin embargo, solo podrán editar los datos de la consulta actual, sin la posibilidad de modificar o eliminar información previamente registrada. Este ámbito también incluye la generación de recetas, informes médicos y la consulta de exámenes complementarios como radiografías o análisis clínicos.

2. Ámbito Administrativo

El personal administrativo del SSVS utiliza el sistema para gestionar la operativa relacionada con los asegurados y los médicos. Esto incluye la gestión de las fichas de atención (creación de horarios y cupos por especialidad), la asignación de médicos a horarios de atención, la gestión de especialidades médicas y la generación de reportes. El ámbito administrativo se enfoca en la supervisión y control del flujo de pacientes y consultas, facilitando la planificación de recursos médicos y humanos.

3. Ámbito del Asegurado (Paciente)

Los asegurados tienen acceso a una interfaz que les permite interactuar con el sistema de manera directa. A través de la aplicación móvil, los pacientes pueden reservar fichas de atención para una especialidad médica, verificar su historial de citas y consultar su historia clínica sin la posibilidad de editarla. Este ámbito garantiza que los pacientes puedan gestionar sus citas y revisar su información médica de manera segura y accesible desde cualquier lugar.

4. Ámbito de Seguridad y Gestión de Usuarios

El sistema incorpora un ámbito dedicado a la gestión de usuarios, roles y permisos, administrado por los superusuarios o administradores. Este ámbito se encarga de definir los accesos y permisos que cada rol tiene dentro del sistema (Administrador, Médico, Asegurado), garantizando que cada actor solo pueda acceder a las funcionalidades correspondientes a su rol. Además, el ámbito de seguridad asegura que todas las interacciones con el sistema se realicen de forma segura mediante autenticación y autorización basada en tokens y cifrado de contraseñas.

5. Ámbito de Reportes y Análisis

Este ámbito permite la generación de reportes detallados sobre la atención a los pacientes, basados en diferentes criterios como fechas, especialidades, médicos, y asegurados. Los reportes pueden ser utilizados por el personal administrativo para tomar decisiones informadas y realizar evaluaciones sobre la eficiencia del servicio

médico. La información generada en este ámbito es clave para la planificación de recursos y la mejora continua del servicio.

4.3 Definiciones, Acrónimos y Abreviaturas

- **Sprint:** Período de tiempo definido, generalmente de 2 a 4 semanas, durante el cual el equipo Scrum trabaja para completar un conjunto específico de tareas del Product Backlog.
- **CU:** Casos de uso.
- **HU:** Historias de usuario.
- **RF:** Requisitos funcionales.
- **RNF:** Requisitos no funcionales.
- **SP:** Sprint.
- **DB:** Base de datos.

4.4 Funciones del Producto

El Sistema de Gestión de Historias Clínicas Electrónicas (HCE) para el **Seguro de Salud Vida Sana (SSVS)** ofrece las siguientes funcionalidades clave:

1. **Gestión de Usuarios (CRUD)**
    - Permite la creación, modificación, eliminación y visualización de usuarios con diferentes roles, como Administrador, Médico y Paciente.
    - Control de acceso basado en roles, garantizando que cada usuario solo acceda a las funciones permitidas según su rol.
2. **Gestión de Roles (CRUD)**
    - Creación y administración de roles, con la capacidad de asignar diferentes permisos a cada rol.
    - Modificación y eliminación de roles, garantizando que cada rol tenga los permisos adecuados para las funciones específicas del sistema.
3. **Gestión de Permisos**
    - Definición y asignación de permisos para los roles del sistema.
    - Control de acceso detallado para asegurar que solo los usuarios autorizados puedan realizar ciertas acciones, como la actualización de historias clínicas o la gestión de usuarios.
4. **Gestión de Pacientes (CRUD)**
    - Los médicos pueden gestionar la información clínica de los pacientes, incluyendo la actualización de la historia clínica durante las consultas.
    - Los pacientes pueden consultar su historia clínica, pero no pueden modificarla.
5. **Solicitud y Gestión de Fichas de Atención**

- Los pacientes pueden solicitar fichas de atención online para un día y horario específico, dentro de la disponibilidad de los médicos.
- El sistema controla que los pacientes no puedan solicitar más de una ficha por día.

6. **Gestión de Historias Clínicas**
   - Los médicos pueden visualizar y actualizar la historia clínica de los pacientes durante la consulta actual, sin posibilidad de modificar información previamente registrada.
   - El sistema permite la inclusión de exámenes, diagnósticos, tratamientos y recetas.
7. **Inicio y Cierre de Sesión**
   - Autenticación de usuarios mediante credenciales, con acceso restringido según el rol asignado.
   - Los usuarios pueden cerrar sesión de forma segura, invalidando su token de acceso.
8. **Recuperación de Contraseña**
   - Los usuarios pueden solicitar la recuperación de su contraseña mediante correo electrónico, y restablecerla a través de un enlace seguro.
9. **Generación de Reportes**
   - Los administradores pueden generar reportes basados en criterios como fechas, pacientes, médicos y especialidades, permitiendo un análisis detallado de las atenciones realizadas.
10. **Seguridad y Control de Acceso**

- Implementación de autenticación segura con JWT (JSON Web Tokens) para garantizar la integridad y seguridad de los datos.
- Control de acceso a las funcionalidades del sistema según los roles y permisos asignados.

4.5 Product Backlog

Nro	*Codigo*	*Descripción*	*Tiempo*	*Prioridad*
1	SP1-1	Crear un perfil que explique la finalidad del proyecto	2 hr	Alta
2	SP1-2	Explicar al equipo la definición del problema	5 hr	Alta
3	SP1-3	Asignación de roles	1 hr	Media
4	SP1-4	Capacitación de los miembros en las herramientas a usar	2 hr	Alta

5	SP1-5	Preparación del entorno de desarrollo de programación	4 hr	Media
6	SP1-6	Presentar un prototipo de una solución al problema	6 hr	Media
7	SP1-7	Encontrar casos de uso funcionales	2 hr	Alta
8	SP1-8	Realizar el modelado de la base de datos	8 hr	Alta
9	SP1-9	Diseño de la interfaz login	1 hr	Alta
10	SP1-10	Implementación del login	2 hr	Alta
11	SP1-11	Realizar las pruebas necesarias para el login	2 hr	Baja
12	SP1-12	Diseñar interfaz para Gestión de usuarios	3 hr	Alta
13	SP1-13	Implementar gestión de usuario	4 hr	Alta
14	SP1-14	Realizar pruebas necesarias para gestión de usuario	1 hr	Baja
15	SP1-15	Diseñar interfaz para gestionar rol	3 hr	Alta
16	SP1-16	Implementar la gestión de rol	8 hr	Alta
17	SP1-17	Realizar pruebas para gestión de rol	8 hr	Baja
18	SP1-18	Diseñar interfaz para asignar permisos al rol	3 hr	Alta
19	SP1-19	Implementar la asignación de permisos	8 hr	Alta
20	SP1-20	Diseñar interfaz para Autenticación y Autorización	6 hr	Alta
21	SP1-21	Implementar Autenticación y Autorización	8 hr	Alta
22	SP1-22	Realizar pruebas de Autenticación y Autorización	3 hr	Baja
23	SP1-23	Diseñar interfaz para Gestión Información de Pacientes	6 hr	Alta
24	SP1-24	Implementar Gestión Información de Pacientes	8 hr	Alta
25	SP1-25	Realizar pruebas de Gestión Información de Pacientes	2 hr	Media
26	SP1-26	Diseñar interfaz para Recuperación Contraseña	2 hr	Alta
27	SP1-27	Implementar Recuperación Contraseña	8 hr	Alta

28	SP1-28	Realizar pruebas de Recuperación Contraseña	3 hr	Baja
29	SP2-1	Diseñar interfaz para Gestion de Especialidades Médicas	3 hr	Alta
30	SP2-2	Implementar Gestion de Especialidades Médicas	7 hr	Alta
31	SP2-3	Realizar pruebas de Gestion de Especialidades Médicas	2 hr	Media
32	SP2-4	Diseñar interfaz para Gestion de Médicos	4 hr	Alta
33	SP2-5	Implementar Gestion de Médicos	9 hr	Alta
34	SP2-6	Realizar pruebas de Gestion de Médicos	2 hr	Media
35	SP2-7	Diseñar interfaz para Gestionar Horarios de Atención de Médicos	3 hr	Alta
36	SP2-8	Implementar Gestionar Horarios de Atención de Médicos	8 hr	Alta
37	SP2-9	Realizar pruebas de Gestionar Horarios de Atención de Médicos	2 hr	Media
38	SP2-10	Diseñar interfaz para Gestionar Reserva	4 hr	Alta
39	SP2-11	Implementar Gestionar Reserva	9 hr	Alta
40	SP2-12	Realizar pruebas de Gestionar Reserva	2 hr	Media
41	SP2-13	Diseñar interfaz para Gestionar Consultas Médicas	2 hr	Alta
42	SP2-14	Implementar Gestionar Consultas Médicas	6 hr	Alta

43	SP2-15	Realizar pruebas de Gestionar Consultas Médicas	2 hr	Media
44	SP2-16	Diseñar interfaz para Solicitar Permiso	3 hr	Media
45	SP2-17	Implementar Solicitar Permiso	6 hr	Media
46	SP2-18	Realizar pruebas de Solicitar Permiso	2 hr	Media
47	SP2-19	Diseñar interfaz para Gestionar Postergación	3 hr	Alta
48	SP2-20	Implementar Gestionar Postergación	7 hr	Alta
49	SP2-21	Realizar pruebas de Gestionar Postergación	2 hr	Media
50	SP2-22	Diseñar interfaz para Gestionar Receta Médica	4 hr	Alta
51	SP2-23	Implementar Gestionar Receta Médica	9 hr	Alta
52	SP2-24	Realizar pruebas de Gestionar Receta Médica	2 hr	Media
53	SP2-25	Diseñar interfaz para Programar Laboratorio	3 hr	Alta
54	SP2-26	Implementar Programar Laboratorio	8 hr	Alta
55	SP2-27	Realizar pruebas de Programar Laboratorio	2 hr	Media
56	SP2-28	Diseñar interfaz para Gestionar Laboratorio	4 hr	Alta
57	SP2-29	Implementar Gestionar Laboratorio	9 hr	Alta
58	SP2-30	Realizar pruebas de Gestionar Laboratorio	2 hr	Media
59	SP2-31	Diseñar interfaz para Gestionar Medicamentos	2 hr	Alta

60	SP2-32	Implementar Gestionar Medicamentos	6 hr	Alta
61	SP2-33	Realizar pruebas de Gestionar Medicamentos	2 hr	Media
62	SP2-34	Diseñar interfaz para Visualizar Historia Clínica	3 hr	Media
63	SP2-35	Implementar Visualizar Historia Clínica	5 hr	Alta
64	SP2-36	Realizar pruebas de Visualizar Historia Clínica	2 hr	Media
65	SP2-37	Diseñar interfaz para Enviar Recordatorio de Cita al Paciente	5 hr	Alta
66	SP2-38	Implementar Enviar Recordatorio de Cita al Paciente	3 hr	Alta
67	SP2-39	Realizar pruebas de Enviar Recordatorio de Cita al Paciente	1 hr	Media
68	SP3-1	Diseñar interfaz para Gestionar Reportes Personalizados	4 hr	Alta
69	SP3-2	Implementar la Gestionar Reportes Personalizados	8 hr	Alta
70	SP3-3	Realizar pruebas de Gestionar Reportes Personalizados	3 hr	Alta
71	SP3-4	Diseñar interfaz para Visualizar Bitacora	1 hr	Alta
72	SP3-5	Implementar Visualizar Bitacora	4 hr	Alta
73	SP3-6	Realizar pruebas de Visualizar Bitácora	1 hr	Alta
74	SP3-7	Diseñar interfaz para Evaluación Medicos	2 hr	Alta
75	SP3-8	Implementar Evaluación Medicos	4 hr	Alta
76	SP3-9	Realizar pruebas de Evaluación Medicos	2 hr	Alta
77	SP3-10	Diseñar interfaz para Exportar historia clínica	2 hr	Alta
78	SP3-11	Implementar Exportar historia clínica	4 hr	Alta

79	SP3-12	Realizar pruebas de Exportar historia clínica	2 hr	Alta
80	SP3-13	Diseñar interfaz para Gestión de Pagos de Seguros	2 hr	Alta
81	SP3-14	Implementar Gestión de Pagos de Seguros	4 hr	Alta
82	SP3-15	Realizar pruebas de Gestión de Pagos de Seguros	2 hr	Alta
83	SP3-16	Diseñar interfaz para Gestión de Método de Pago	2 hr	Alta
84	SP3-17	Implementar Gestión de Método de Pago	4 hr	Alta
85	SP3-18	Realizar pruebas de Gestión de Método de Pago	2 hr	Alta
86	SP3-19	Diseñar interfaz para visualizar Historial de Pagos	2 hr	Alta
87	SP3-20	Implementar Visualizar Historial de Pagos	4 hr	Alta
88	SP3-21	Realizar pruebas de visualizar Historial de Pagos	2 hr	Alta

4.6 Requisitos Funcionales

Requisitos Funcionales
RF1. Gestión de Usuarios
RF2. Administración de Roles y Permisos
RF3. Autenticación y Autorización
RF4. Gestión de Especialidades Médicas
RF5. Gestión de Médicos
RF6. Gestión de Horarios de Atención
RF7. Obtención de Fichas de Atención
RF8. Gestión de Atención de Consulta
RF9. Visualización de Historia Clínica

RF10. Registro de Exámenes y Documentos
RF11. Reportes para Administración
RF12. Exportación de Reportes en PDF/Excel

4.7 Requisitos no Funcionales

Requisitos No Funcionales
RNF1. Cambio de contraseña
RNF2. Encriptación de contraseña
RNF3. Mantenimiento del sistema
RNF4. Integridad de datos
RNF5. Portabilidad de distintas plataformas

4.8 Lista de casos de uso (Web y Móvil)

Lista de Casos de Uso (Web)				
Modulo	**Nro**	**Descripción**	**Estado**	**Sprint**
M. Usuarios	**CU1**	Gestión de Usuarios	Terminado	1
M. Usuarios	**CU2**	Gestión de Roles	Terminado	1
M. Usuarios	**CU3**	Gestión de Permisos	Terminado	1
M. Usuarios	**CU4**	Gestionar Información de Pacientes	Terminado	1
M. Usuarios	**CU5**	Iniciar Sesión	Terminado	1
M.	**CU6**	Cerrar Sesión	Terminado	1

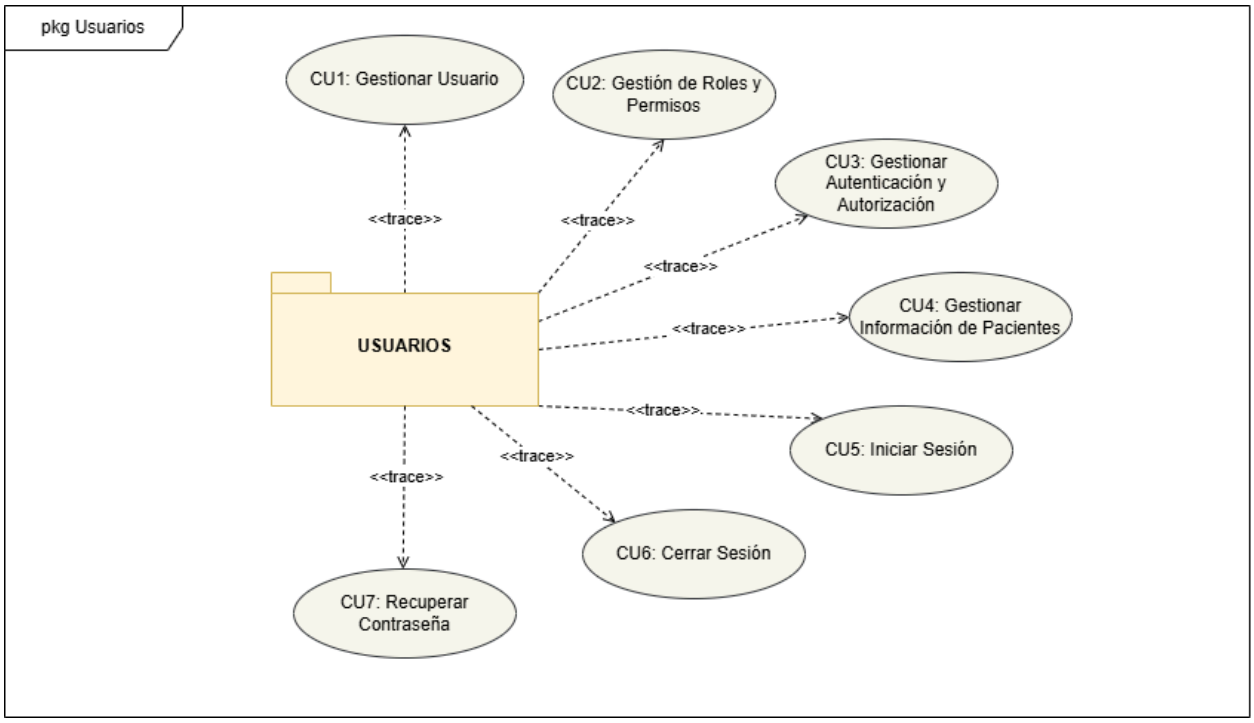
Usuarios				
M. Usuarios	**CU7**	Recuperar Contraseña	Terminado	1
M. Atencion Medica	**CU8**	Gestión de Especialidades Médicas	Terminado	2
M. Atencion Medica	**CU9**	Gestión de Médicos	Terminado	2
M. Atencion Medica	**CU10**	Gestionar Horarios de Atención de Médicos	Terminado	2
M. Atencion Medica	**CU11**	Gestionar Reserva	Terminado	2
M. Atencion Medica	**CU12**	Gestionar Consultas Médicas	Terminado	2
M. Atencion Medica	**CU13**	Solicitar Permiso	Terminado	2
M. Atencion Medica	**CU14**	Gestionar Postergación	Terminado	2
M. Atencion Medica	**CU15**	Gestionar Receta Médica	Terminado	2
M. Atencion Medica	**CU16**	Programar Laboratorio	Terminado	2
M. Atencion Medica	**CU17**	Gestionar Laboratorio	Terminado	2
M. Atencion Medica	**CU18**	Gestionar Medicamentos	Terminado	2

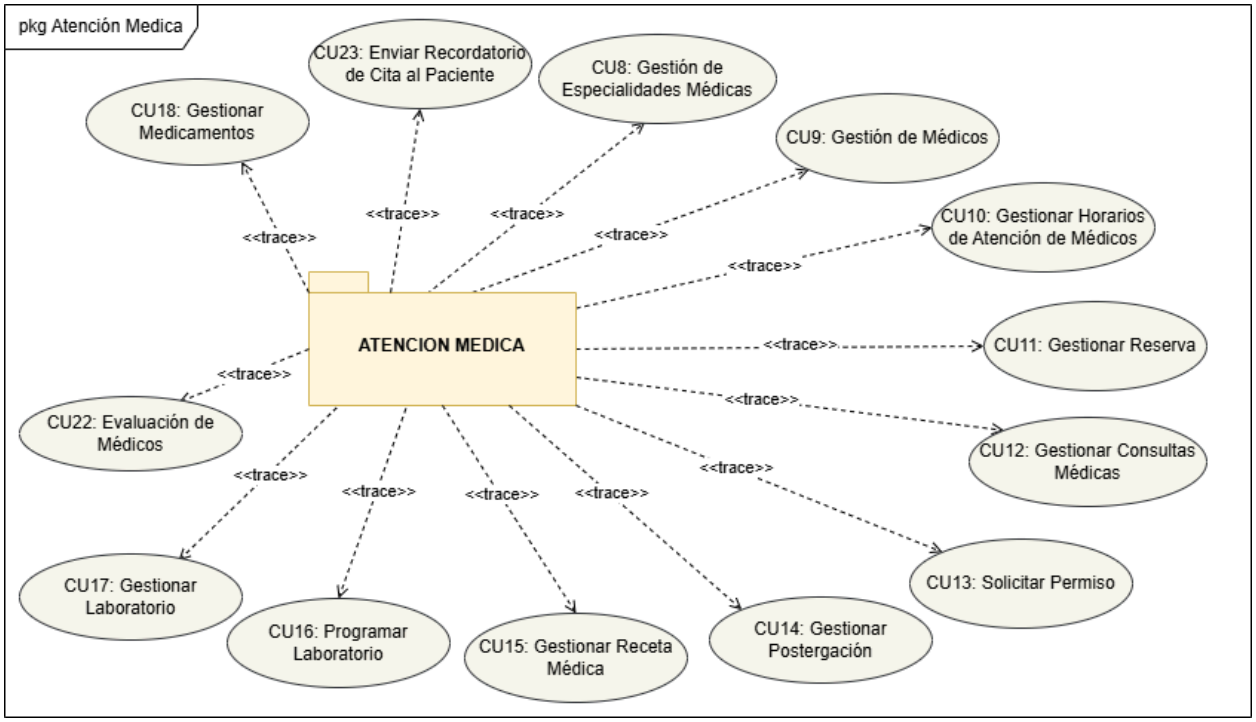
M. Historia Clinica	**CU19**	Visualizar Historia Clínica	Terminado	2
M. Reportes	**CU20**	Gestionar Reportes Personalizados	Terminado	3
M. Reportes	**CU21**	Visualizar Bitacora	Terminado	3
M. Atencion Medica	**CU22**	Evaluación de Médicos	Terminado	3
M. Historia Clinica	**CU24**	Exportar historia clínica	Terminado	3
M. Pagos	**CU25**	Gestión de Pagos de Seguros	Terminado	3
M. Pagos	**CU26**	Gestión de Método de Pago	Terminado	3
M. Pagos	**CU27**	Visualizar Historial de Pagos	Terminado	3

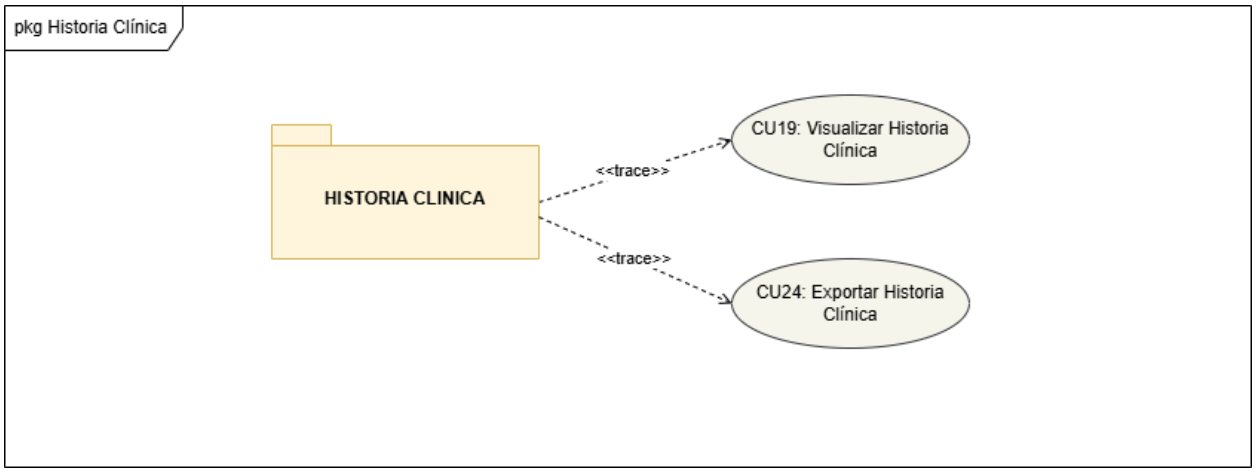
Lista de Casos de Uso (Móvil)				
Modulo	**Nro**	**Descripción**	**Estado**	**Sprint**
M. Usuarios	**CU5**	Iniciar Sesión	Terminado	1
M. Usuarios	**CU6**	Cerrar Sesión	Terminado	1
M. Usuarios	**CU7**	Recuperar Contraseña	Terminado	1
M. Atencion Medica	**CU11**	Gestionar Reserva	Terminado	2
M. Atencion Medica	**CU15**	Gestionar Receta Médica	Terminado	2
M.	**CU19**	Visualizar Historia Clínica	Terminado	2

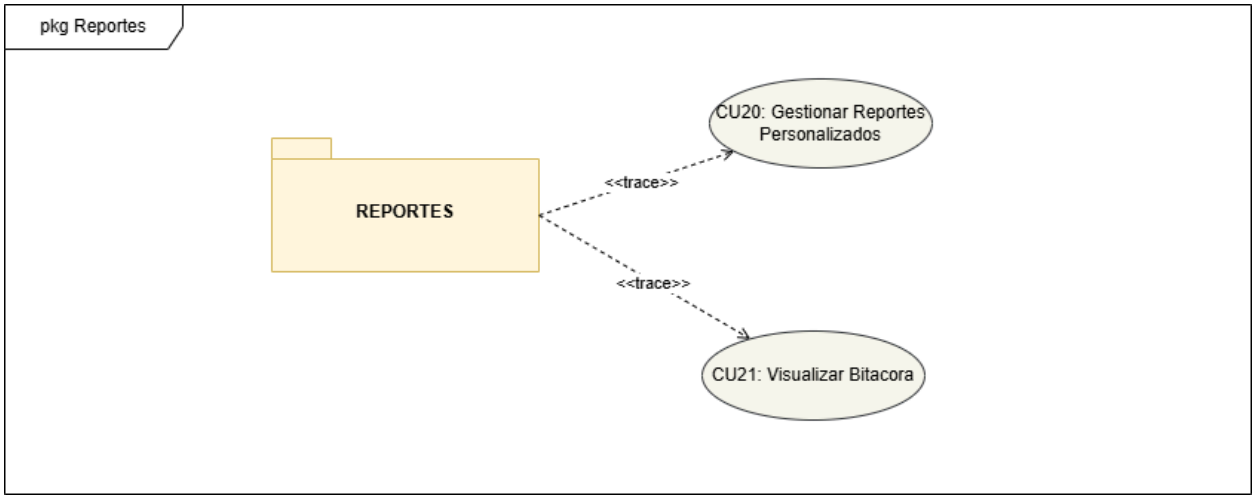
Historia Clinica				
M. Atencion Medica	**CU23**	Enviar Recordatorio de Cita al Paciente	Terminado	2
M. Atencion Medica	**CU22**	Evaluación de Médicos	Terminado	3
M. Atencion Medica	**CU24**	Exportar historia clínica	Terminado	3
M. Pagos	**CU27**	Visualizar Historial de Pagos	Terminado	3

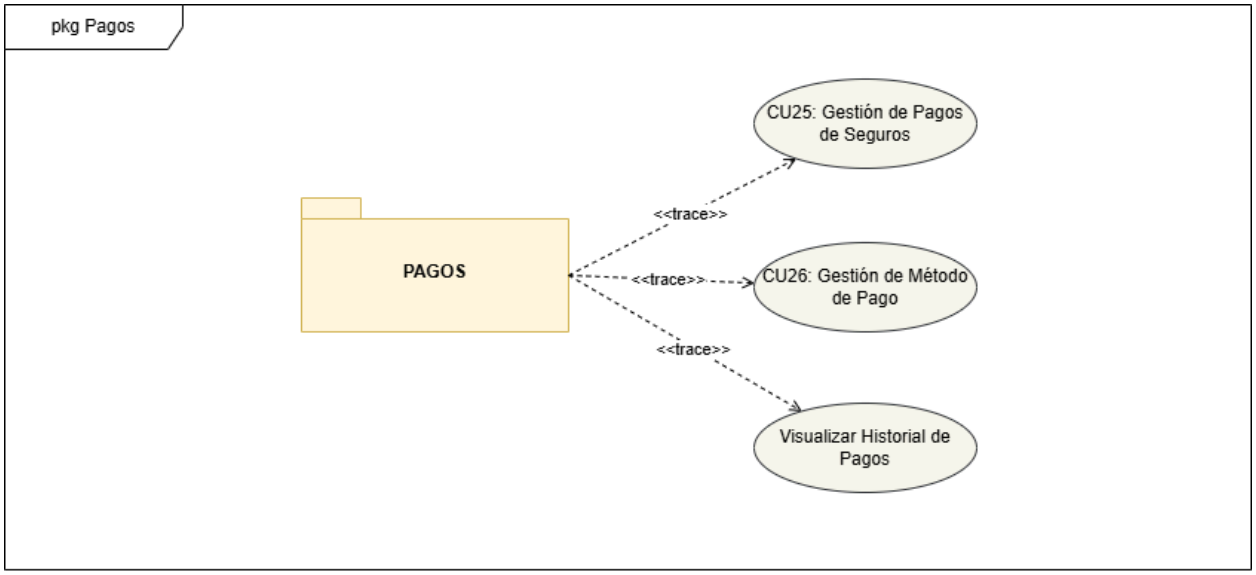
4.9 Paquetes y casos de Uso




pkg Atención Medica
CU23: Enviar Recordatorio de Cita al Paciente
CU8: Gestión de Especialidades Médicas
CU18: Gestionar Medicamentos
CU9: Gestión de Médicos
CU10: Gestionar Horarios de Atención de Médicos
ATENCION MEDICA
CU11: Gestionar Reserva
CU22: Evaluación de Médicos
CU12: Gestionar Consultas Médicas
CU17: Gestionar Laboratorio
CU13: Solicitar Permiso
CU16: Programar Laboratorio
CU15: Gestionar Receta Médica
CU14: Gestionar Postergación
<<trace>>


pkg Historia Clínica
HISTORIA CLINICA
CU19: Visualizar Historia Clínica
CU24: Exportar Historia Clínica
<<trace>>


pkg Reportes
REPORTES
CU20: Gestionar Reportes Personalizados
CU21: Visualizar Bitacora
<<trace>>



4.10 Planificación Sprint (Diagrama de Gantt)

Codigo	Tarea	21/10/2024 - 27/10/2024	27/10/2024 - 07/11/2024	07/11/2024 - 17/11/2024	
SP1-1	Crear un perfil que explique la finalidad del proyecto				
SP1-2	Explicar al equipo la definición del problema				
SP1-3	Asignación de roles				
SP1-4	Capacitación de los miembros en las herramientas a usar				
SP1-5	Preparación del entorno de desarrollo de programación				
SP1-6	Presentar un prototipo de una				

SPR

	solución al problema				**I N T 1**
SP1-7	Encontrar casos de uso funcionales				
SP1-8	Realizar el modelado de la base de datos				
SP1-9	Diseño de la interfaz login				
SP1-10	Implementación del login				
SP1-11	Realizar las pruebas necesarias para el login				
SP1-12	Diseñar interfaz para Gestión de usuarios				
SP1-13	Implementar gestión de usuario				
SP1-14	Realizar pruebas necesarias para gestión de usuario				
SP1-15	Diseñar interfaz para gestionar rol				
SP1-16	Implementar la gestión de rol				
SP1-17	Realizar pruebas para gestión de rol				
SP1-18	Diseñar interfaz para asignar permisos al rol				

SP1-19	Implementar la asignación de permisos				
SP1-20	Diseñar interfaz para Autenticación y Autorización				
SP1-21	Implementar Autenticación y Autorización				
SP1-22	Realizar pruebas de Autenticación y Autorización				
SP1-23	Diseñar interfaz para Gestión Información de Pacientes				
SP1-24	Implementar Gestión Información de Pacientes				
SP1-25	Realizar pruebas de Gestión Información de Pacientes				
SP1-26	Diseñar interfaz para Recuperación Contraseña				
SP1-27	Implementar Recuperación Contraseña				
SP1-28	Realizar pruebas de Recuperación Contraseña				
SP2-1	Diseñar interfaz para Gestión de				

	Especialidades Médicas				
SP2-2	Implementar Gestion de Especialidades Médicas				
SP2-3	Realizar pruebas de Gestion de Especialidades Médicas				
SP2-4	Diseñar interfaz para Gestión de Médicos				
SP2-5	Implementar Gestión de Médicos				
SP2-6	Realizar pruebas de Gestion de Médicos				
SP2-7	Diseñar interfaz para Gestionar Horarios de Atención de Médicos				**SPRINT 2**
SP2-8	Implementar Gestionar Horarios de Atención de Médicos				

SP2-9	Realizar pruebas de Gestionar Horarios de Atención de Médicos				
SP2-10	Diseñar interfaz para Gestionar Reserva				
SP2-11	Implementar Gestionar Reserva				
SP2-12	Realizar pruebas de Gestionar Reserva				
SP2-13	Diseñar interfaz para Gestionar Consultas Médicas				
SP2-14	Implementar Gestionar Consultas Médicas				
SP2-15	Realizar pruebas de Gestionar Consultas Médicas				
SP2-16	Diseñar interfaz para Solicitar Permiso				

SP2-17	Implementar Solicitar Permiso				
SP2-18	Realizar pruebas de Solicitar Permiso				
SP2-19	Diseñar interfaz para Gestionar Postergación				
SP2-20	Implementar Gestionar Postergación				
SP2-21	Realizar pruebas de Gestionar Postergación				
SP2-22	Diseñar interfaz para Gestionar Receta Médica				
SP2-23	Implementar Gestionar Receta Médica				
SP2-24	Realizar pruebas de Gestionar Receta Médica				
SP2-25	Diseñar interfaz para Programar Laboratorio				

SP2-26	Implementar Programar Laboratorio				
SP2-27	Realizar pruebas de Programar Laboratorio				
SP2-28	Diseñar interfaz para Gestionar Laboratorio				
SP2-29	Implementar Gestionar Laboratorio				
SP2-30	Realizar pruebas de Gestionar Laboratorio				
SP2-31	Diseñar interfaz para Gestionar Medicamentos				
SP2-32	Implementar Gestionar Medicamentos				
SP2-33	Realizar pruebas de Gestionar Medicamentos				
SP2-34	Diseñar interfaz para Visualizar Historia Clínica				

SP2-35	Implementar Visualizar Historia Clínica				
SP2-36	Realizar pruebas de Visualizar Historia Clínica				
SP2-37	Diseñar interfaz para Enviar Recordatorio de Cita al Paciente				
SP2-38	Implementar Enviar Recordatorio de Cita al Paciente				
SP2-39	Realizar pruebas de Enviar Recordatorio de Cita al Paciente				
SP3-1	Diseñar interfaz para Gestionar Reportes Personalizados				**SPRINT 3**
SP3-2	Implementar la Gestionar Reportes Personalizados				
SP3-3	Realizar pruebas de Gestionar Reportes Personalizados				

SP3-4	Diseñar interfaz para Visualizar Bitácora				
SP3-5	Implementar Visualizar Bitacora				
SP3-6	Realizar pruebas de Visualizar Bitácora				
SP3-7	Diseñar interfaz para Evaluación Medicos				
SP3-8	Implementar Evaluación Medicos				
SP3-9	Realizar pruebas de Evaluación Medicos				
SP3-10	Diseñar interfaz para Exportar historia clínica				
SP3-11	Implementar Exportar historia clínica				
SP3-12	Realizar pruebas de Exportar historia clínica				
SP3-13	Diseñar interfaz para Gestión de Pagos de Seguros				
SP3-14	Implementar Gestión de Pagos de Seguros				

SP3-15	Realizar pruebas de Gestión de Pagos de Seguros				
SP3-16	Diseñar interfaz para Gestión de Método de Pago				
SP3-17	Implementar Gestión de Método de Pago				
SP3-18	Realizar pruebas de Gestión de Método de Pago				
SP3-19	Diseñar interfaz para visualizar Historial de Pagos				
SP3-20	Implementar Visualizar Historial de Pagos				
SP3-21	Realizar pruebas de visualizar Historial de Pagos				

5. PROCESO DE DESARROLLO DE SOFTWARE

Sprint 1

1. Sprint Planning

1.1. Objetivos del Sprint

El objetivo de este sprint es implementar las funcionalidades esenciales para la gestión de usuarios y control de acceso en el sistema de gestión de historias clínicas electrónicas del Seguro de Salud Vida Sana (SSVS). Esto permitirá garantizar la seguridad de la información clínica de los pacientes y una gestión eficiente de los roles y permisos del sistema. Específicamente, se buscará desarrollar la infraestructura de autenticación y autorización, así como las operaciones clave relacionadas con la gestión de usuarios, pacientes y la consulta médica.

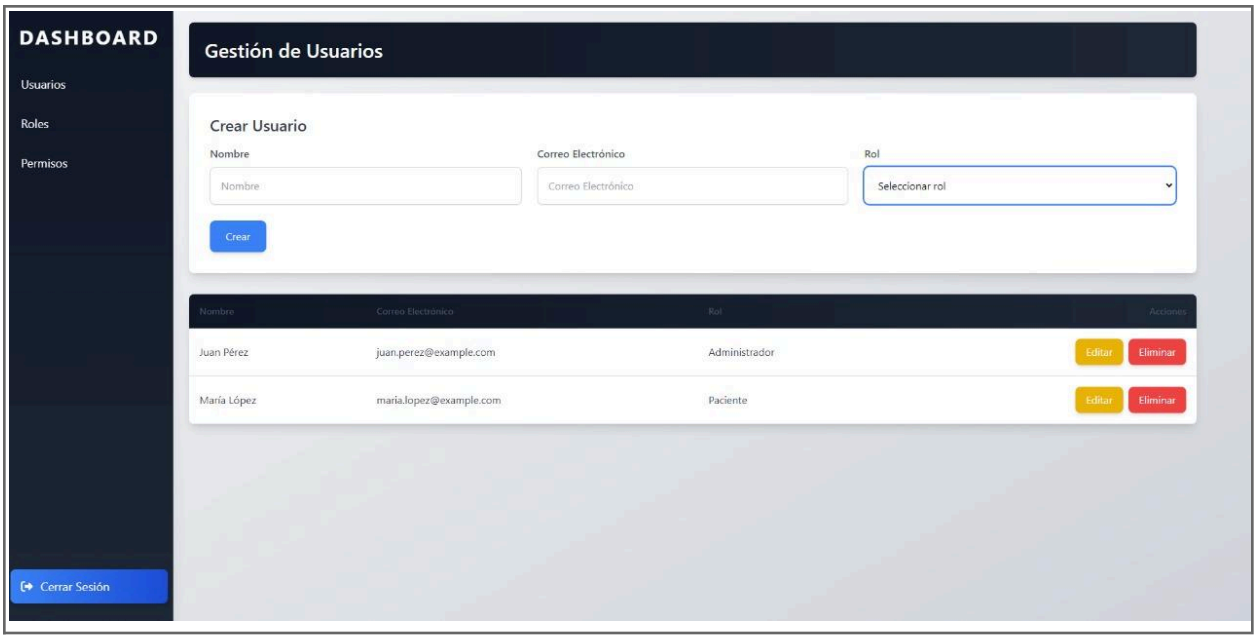
1.1.1 Casos de uso

- CU1: Gestión de Usuarios
- CU2: Gestión de Roles
- CU3: Gestión de Permisos
- CU4: Gestionar Información de Pacientes
- CU5: Iniciar Sesión
- CU6: Cerrar Sesión
- CU7: Recuperar Contraseña

1.2. Historias de usuario

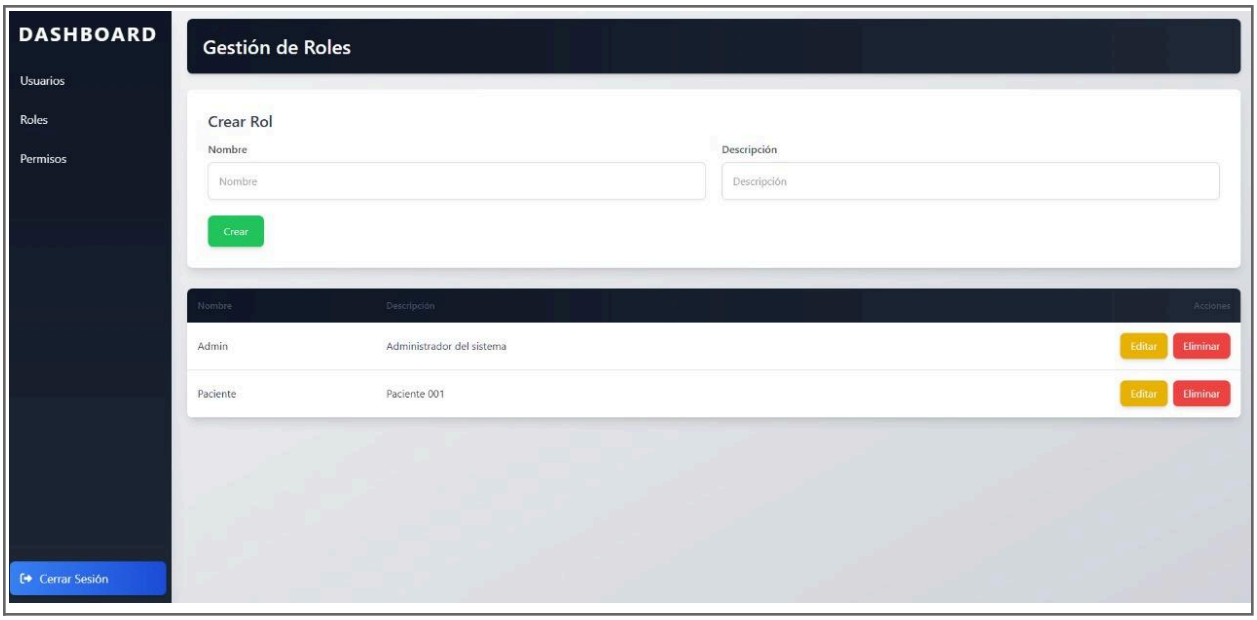
CU1: Gestión de Usuarios

Gestión de Usuarios		
ID: 1	**Descripción: Registrar usuario en el sistema**	
Prioridad: Alta		**Estimación PHU: 6hr**
Criterios de aceptación: ● Los administradores deben crear, consultar, modificar y eliminar usuarios en el sistema.		
Desarrollador a cargo: Fabiana Manzoni Bravo		



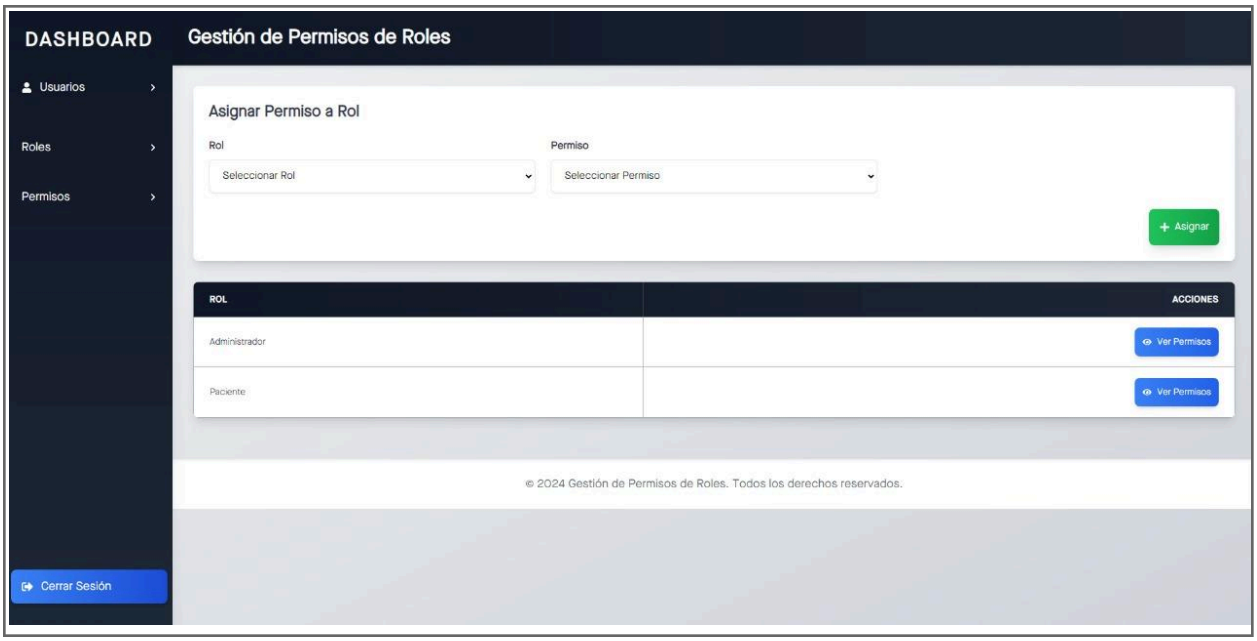
CU2: Gestión de Roles

Gestión de Roles	
ID: 2	**Descripción: Gestionar roles de usuarios en el sistema**
Prioridad: Alta	**Estimación PHU: 5hr**
Criterios de aceptación: ● Los administradores deben crear, consultar, modificar y eliminar roles de usuarios.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



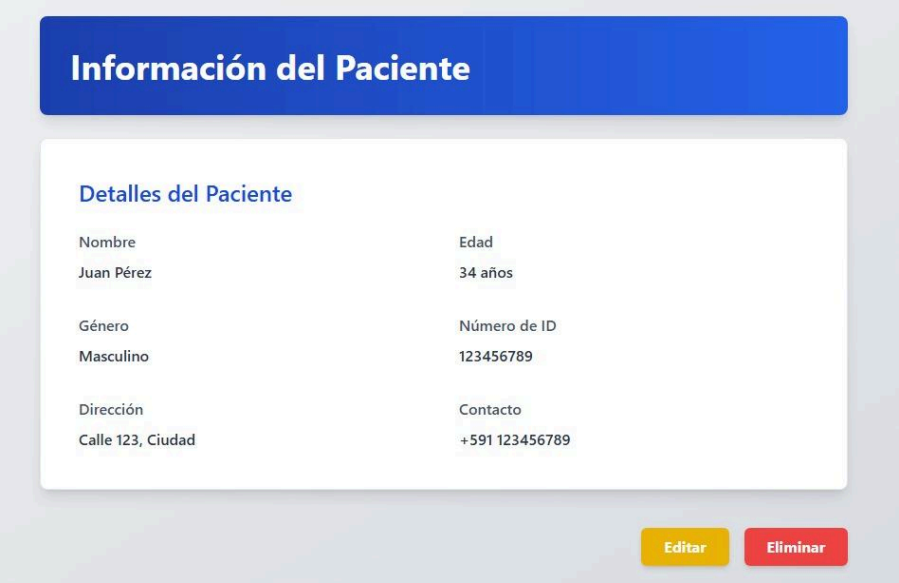
CU3: Gestión de Permisos

Gestión de Permisos	
ID: 3	**Descripción: Gestionar permisos de roles en el sistema**
Prioridad: Alta	**Estimación PHU: 5hr**
Criterios de aceptación: ● Los administradores deben crear, consultar, modificar y eliminar permisos con roles en específicos.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



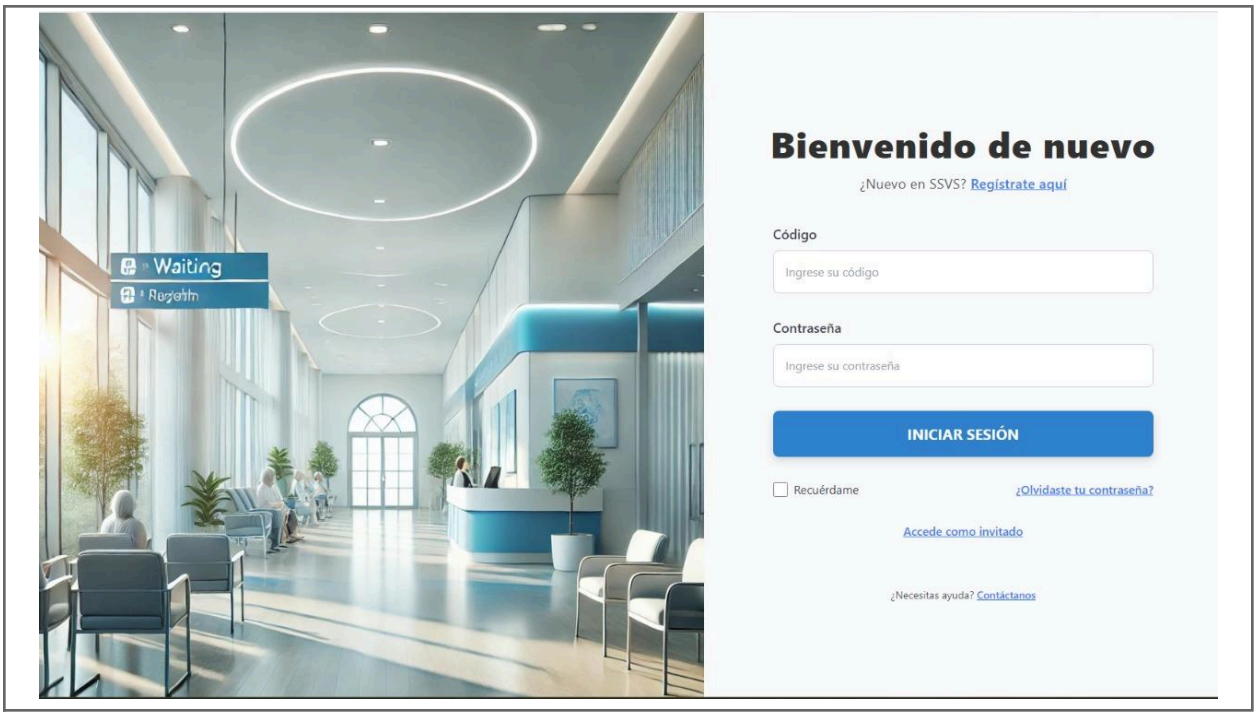
CU4: Gestionar Información de Pacientes

Gestionar Información de Pacientes	
ID: 4	**Descripción: Gestionar Información de Paciente en el sistema**
Prioridad: Alta	**Estimación PHU: 6hr**
Criterios de aceptación: ● Registrar Paciente: Permitir el registro de usuarios con el rol "Paciente" y sus datos personales específicos. ● Consultar Información de Paciente: Visualizar y administrar la información de los pacientes. ● Modificar Información de Paciente: Permitir al administrador actualizar ciertos detalles personales de los pacientes.	
Desarrollador a cargo: Jorge Arturo Aliaga Valencia	



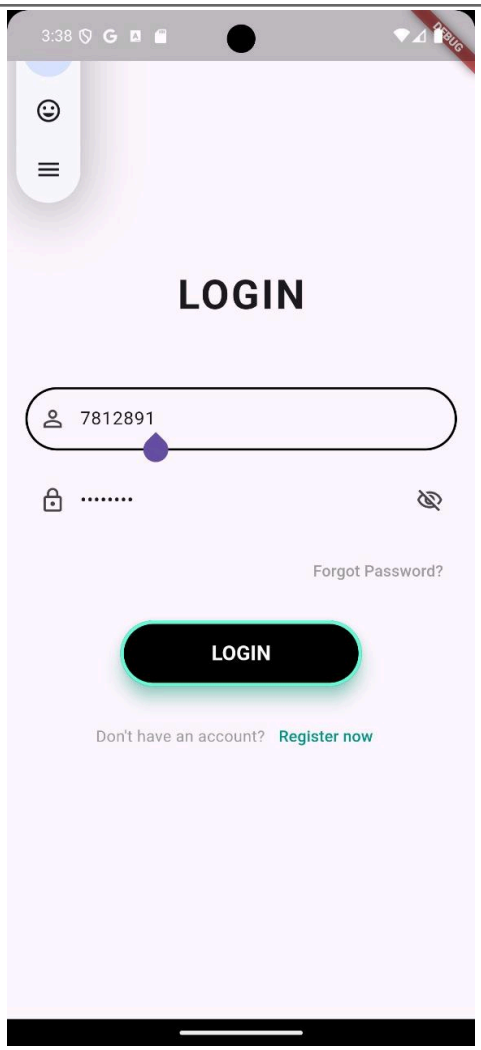
CU5: Iniciar Sesión (WEB)

Iniciar Sesión	
ID: 5	**Descripción: Inicio de sesión en el sistema**
Prioridad: Alta	**Estimación PHU: 5hr**
Criterios de aceptación: ● Permitir inicio de sesión seguro en el sistema.	
Desarrollador a cargo: Jorge Arturo Aliaga Valencia	



CU5: Iniciar Sesión (MOVIL)

Iniciar Sesión	
ID: 5	**Descripción: Inicio de sesión en el sistema**
Prioridad: Alta	**Estimación PHU: 5hr**
Criterios de aceptación: ● Permitir inicio de sesión seguro en el sistema.	
Desarrollador a cargo: Fabiana Manzoni Bravo	

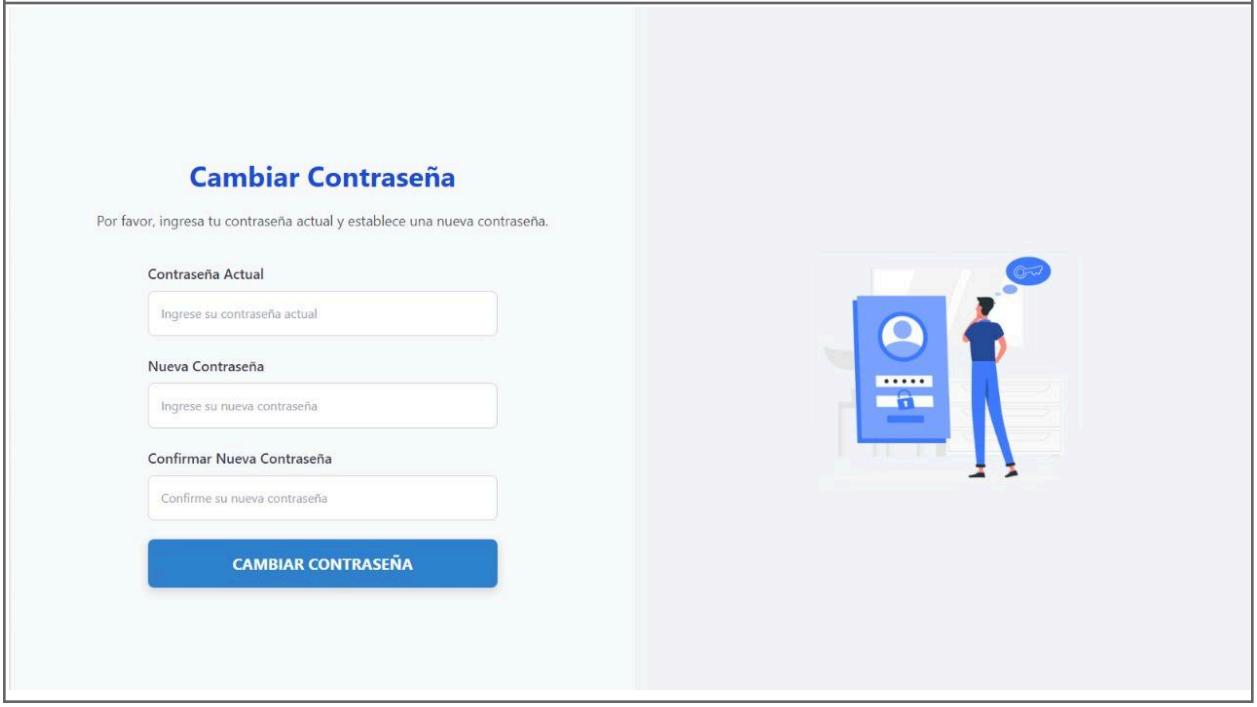


CU6: Cerrar Sesión

Cerrar Sesión	
ID: 6	**Descripción: Cierre de sesión en el sistema**
Prioridad: Alta	**Estimación PHU: 2hr**
Criterios de aceptación: ● Permitir Cierre de sesión seguro en el sistema.	
Desarrollador a cargo: Jorge Arturo Aliaga Valencia	

CU7: Recuperar Contraseña

Recuperación de Contraseña	
ID: 7	**Descripción: Recuperar Contraseña en el sistema**
Prioridad: Alta	**Estimación PHU: 7hr**
Criterios de aceptación: ● Permitir recuperación y restablecimiento de contraseñas.	
Desarrollador a cargo: Jorge Arturo Aliaga Valencia	



1.3. Contexto del sistema

En este sprint, se enfocará en establecer la base esencial del sistema. El objetivo principal de esta fase es implementar la gestión de usuarios, roles y permisos, así como la configuración de la autenticación, autorización, la gestión de información de pacientes y la funcionalidad de recuperación de contraseñas.

1.4. Sprint Backlog

Sprint Backlog	
Numero de Sprint: 1	**Tiempo programado: 1 semana**

Objetivo: El objetivo del sprint es desarrollar e implementar las funcionalidades centrales del Sistema de Gestión de Historias Clínicas Electrónicas (HCE) para el Seguro de Salud Vida Sana (SSVS), enfocándose en la gestión de usuarios, roles, permisos y la recuperación de contraseñas. Además, se incluirán funcionalidades críticas relacionadas con la gestión de pacientes.					
Product Owner: No establecido					
Fecha de inicio: 21/10/2024			**Fecha de finalización: 27/10/2024**		
Id	**Tarea**	**Tipo**	**Estimación (hr)**	**Responsable**	**Estado**
1	Implementación del frontend para iniciar y cerrar sesion.	Desarrollo	5	Jorge Arturo Aliaga Valencia	Terminado
2	Implementacion del backend para iniciar y cerrar sesion.	Desarrollo	6	Jorge Arturo Aliaga Valencia	Terminado
3	Implementación movil para iniciar y cerrar sesion.	Desarrollo	4	Fabiana Manzoni Bravo	Terminado
4	Implementación del frontend para la gestión de usuarios.	Desarrollo	7	Fabiana Manzoni Bravo	Terminado
5	Implementacion del backend para la gestión de usuarios.	Desarrollo	8	Fabiana Manzoni Bravo	Terminado
6	Implementacion del frontend para la gestión de pacientes.	Desarrollo	5	Jorge Arturo Aliaga Valencia	Terminado
7	Implementacion del backend para la gestión de pacientes.	Desarrollo	6	Jorge Arturo Aliaga Valencia	Terminado
8	Implementacion del frontend para la gestión de roles.	Desarrollo	6	Fabiana Manzoni Bravo	Terminado
9	Implementacion del backend para la gestión de roles.	Desarrollo	7	Fabiana Manzoni Bravo	Terminado
10	Implementacion del frontend para la	Desarrollo	4	Fabiana Manzoni Bravo	Terminado

	asignación de permisos.				
11	Implementacion del backend para la asignación de permisos.	Desarrollo	5	Fabiana Manzoni Bravo	Terminado
12	Implementacion del backend para Recuperar Contraseña.	Desarrollo	6	Jorge Arturo Aliaga Valencia	Terminado
13	Implementacion del frontend para Recuperar Contraseña.	Desarrollo	5	Jorge Arturo Aliaga Valencia	Terminado

1.5. Equipo SCRUM

Integrantes	*Rol*	*Caso de Uso*	*Tareas*	*Nota*
Jorge Arturo Aliaga Valencia	Developer	**WEB:** **CU4:** Gestionar Información de Pacientes **CU5:** Iniciar Sesión **CU6:** Cerrar Sesión **CU7:** Recuperar Contraseña	**Tarea 1:** Implementación del frontend para iniciar y cerrar sesion. **Tarea 2:** Implementacion del backend para iniciar y cerrar sesion. **Tarea 6:** Implementación del frontend para la gestión de pacientes. **Tarea 7:** Implementacion del backend para la gestión de pacientes. **Tarea12**: Implementacion del backend para Recuperar Contraseña. **Tarea 13:** Implementacion del frontend para Recuperar Contraseña.	95
Fabiana Manzoni Bravo	Developer	**WEB:** **CU1:** Gestión de Usuarios	**Tarea 3:**	95

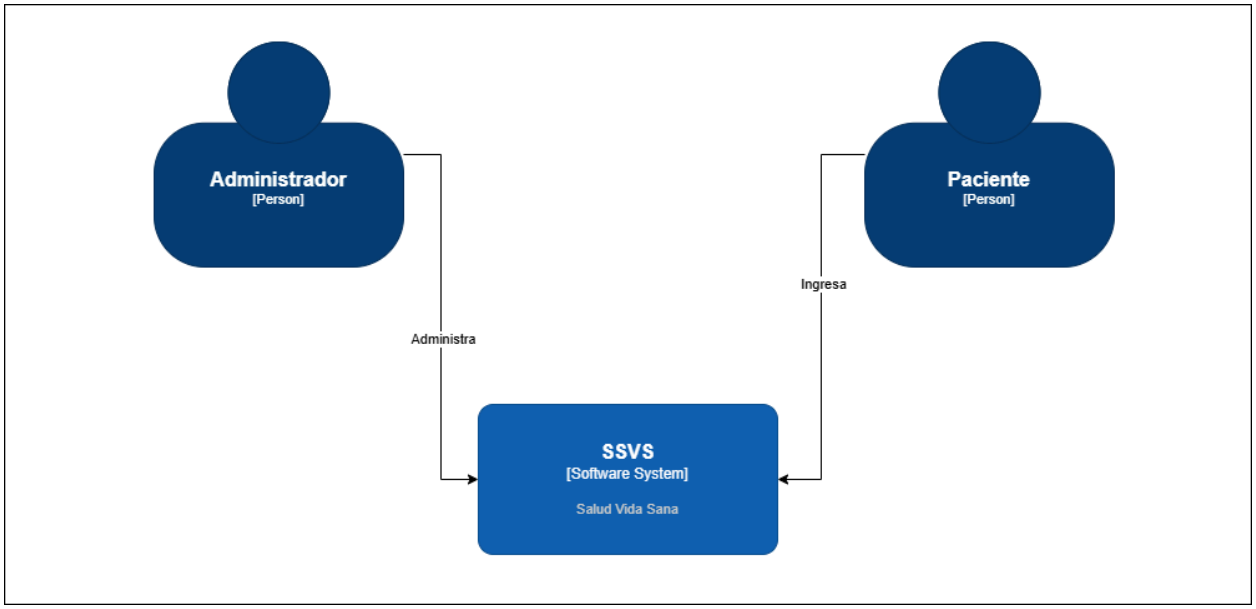
		CU2: Gestión de Roles **CU3:** Gestión de Permisos **MOVIL:** **CU5:** Iniciar Sesión **CU6:** Cerrar Sesión	Implementación movil para iniciar y cerrar sesion. **Tarea 4:** Implementación del frontend para la gestión de usuarios. **Tarea 5:** Implementacion del backend para la gestión de usuarios. **Tarea 8:** Implementacion del frontend para la gestión de roles. **Tarea 9:** Implementación del backend para la gestión de roles. **Tarea 10:** Implementacion del frontend para la asignación de permisos. **Tarea 11:** Implementacion del backend para la asignación de permisos.	

2. Proceso/patrón de desarrollo por Historia de Usuario

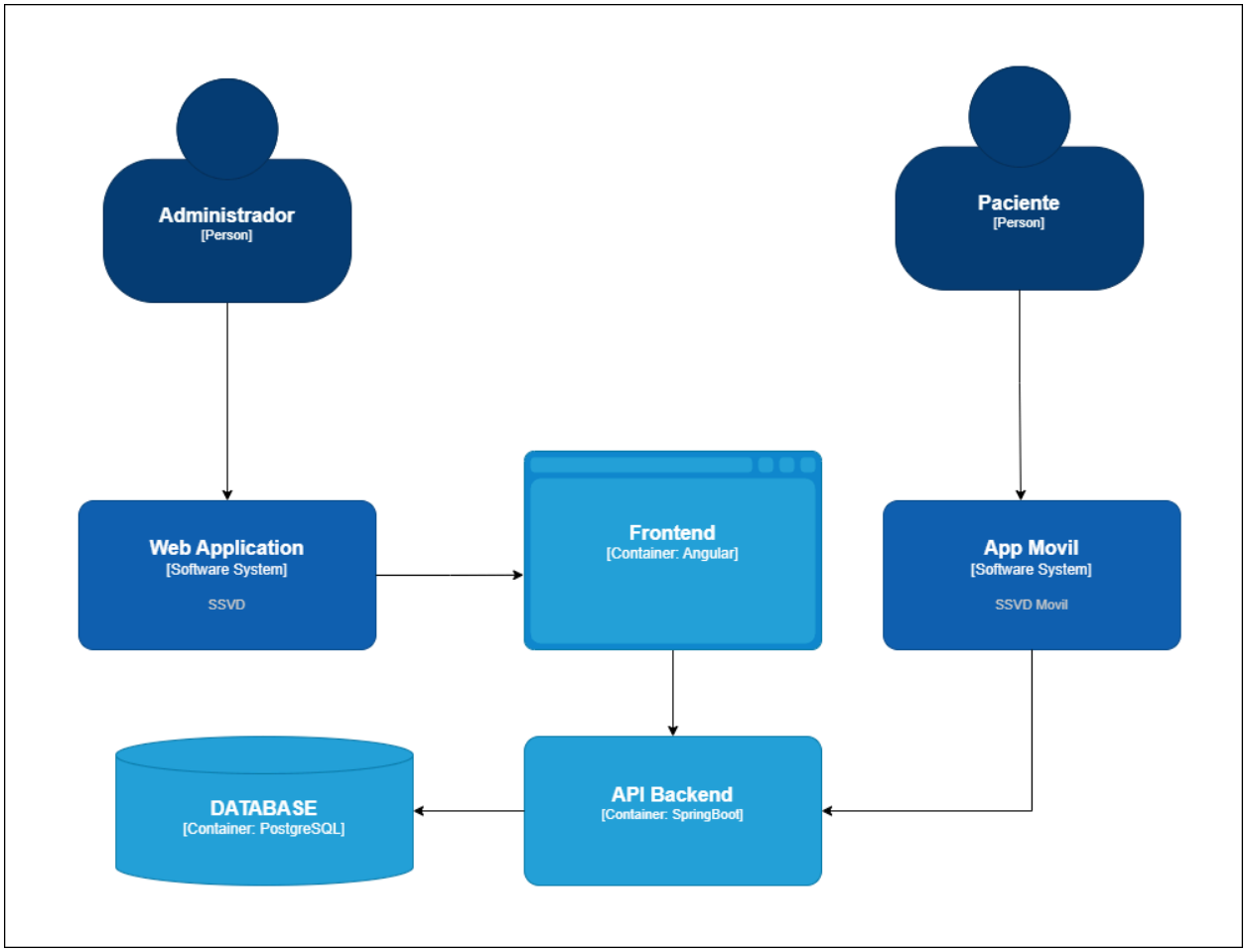
2.1. Diseño

2.1.1. Diseñar de la arquitectura

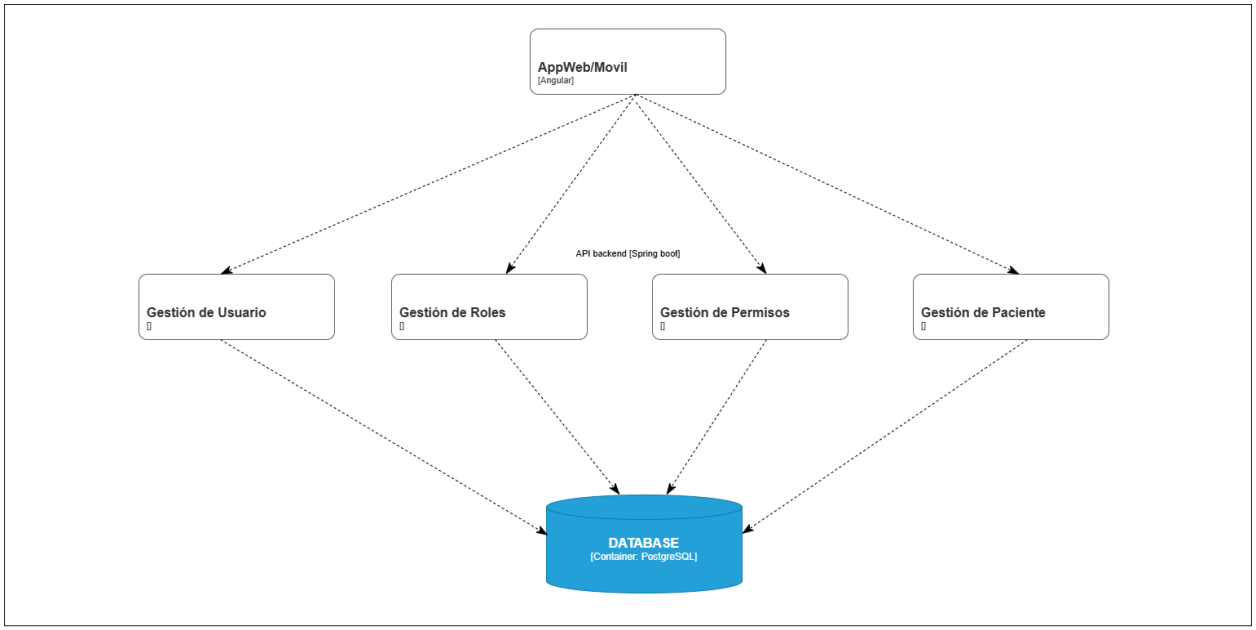
a) Diagrama del Contexto del Sistema



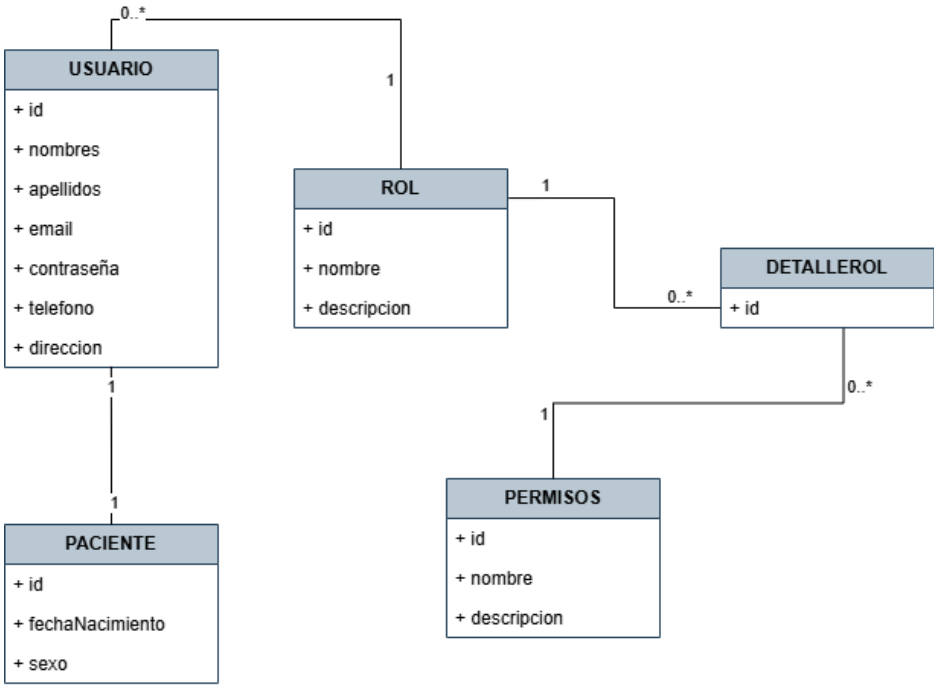
b) Diagrama de Contenedores



c) Diagrama de Componentes

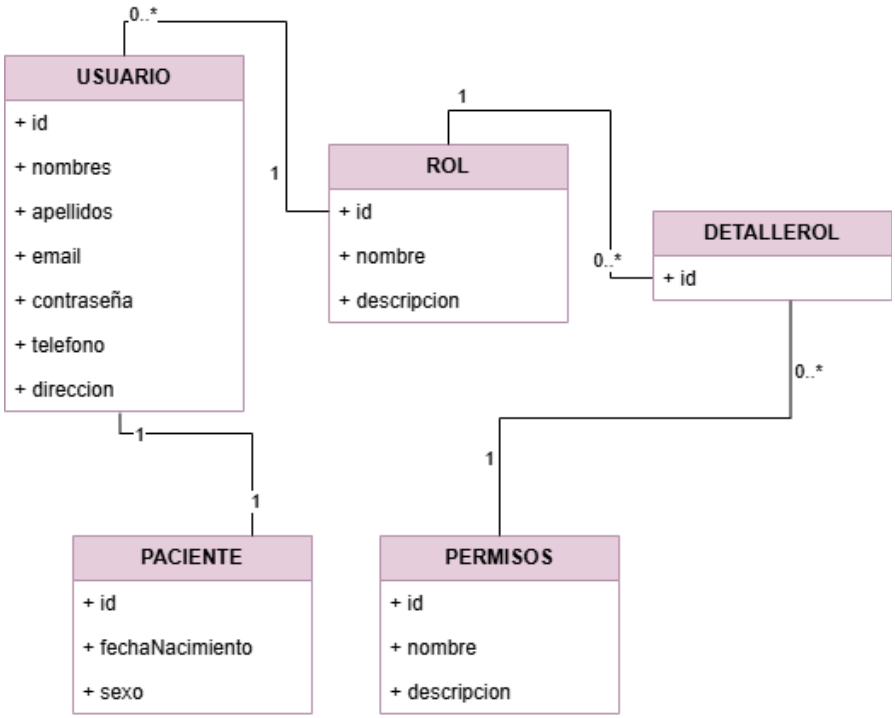


d) Diagrama de Código



2.1.2. Diseño de datos

a) Diagrama de clases



b) Script

```
CREATE DATABASE sistema_historia_clinica;

\c sistema_historia_clinica;

CREATE TABLE Roles (
   rol_id SERIAL PRIMARY KEY,
   nombre VARCHAR(50) UNIQUE NOT NULL,
   descripcion TEXT
);

CREATE TABLE Permisos (
   permiso_id SERIAL PRIMARY KEY,
   nombre VARCHAR(50) UNIQUE NOT NULL,
   descripcion TEXT
);

CREATE TABLE Roles_Permisos (
   rol_id INT REFERENCES Roles(rol_id) ON DELETE CASCADE,
   permiso_id INT REFERENCES Permisos(permiso_id) ON DELETE CASCADE,
   PRIMARY KEY (rol_id, permiso_id)
```

```
);

CREATE TABLE Usuarios (
    usuario_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    contraseña TEXT NOT NULL,
    telefono VARCHAR(20),
    direccion TEXT,
    rol_id INT REFERENCES Roles(rol_id) ON DELETE SET NULL
);

CREATE TABLE Pacientes (
    paciente_id SERIAL PRIMARY KEY,
    usuario_id INT REFERENCES Usuarios(usuario_id) ON DELETE CASCADE,
    fecha_nacimiento DATE,
    genero VARCHAR(10),
    direccion TEXT,
    CONSTRAINT fk_rol_paciente CHECK ((SELECT rol_id FROM Usuarios WHERE
usuario_id = usuario_id) = (SELECT rol_id FROM Roles WHERE nombre = 'Paciente'))
);
```

c) Mapeo

USUARIO

id pk	**nombres**	**apellidos**	**email**	**contraseña**	**telefono**	**direccion**	**id_rol fk**

ROL

id pk	**nombre**	**descripción**

PERMISO

id pk	**nombre**	**descripción**

DETALLEROL

id pk	id_rol fk	id_permiso fk

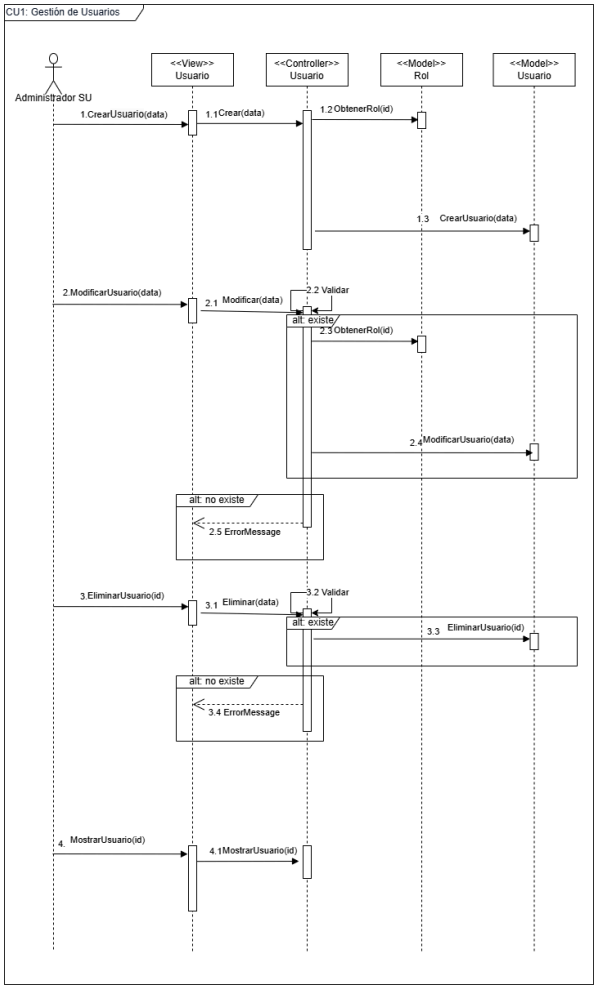
PACIENTE

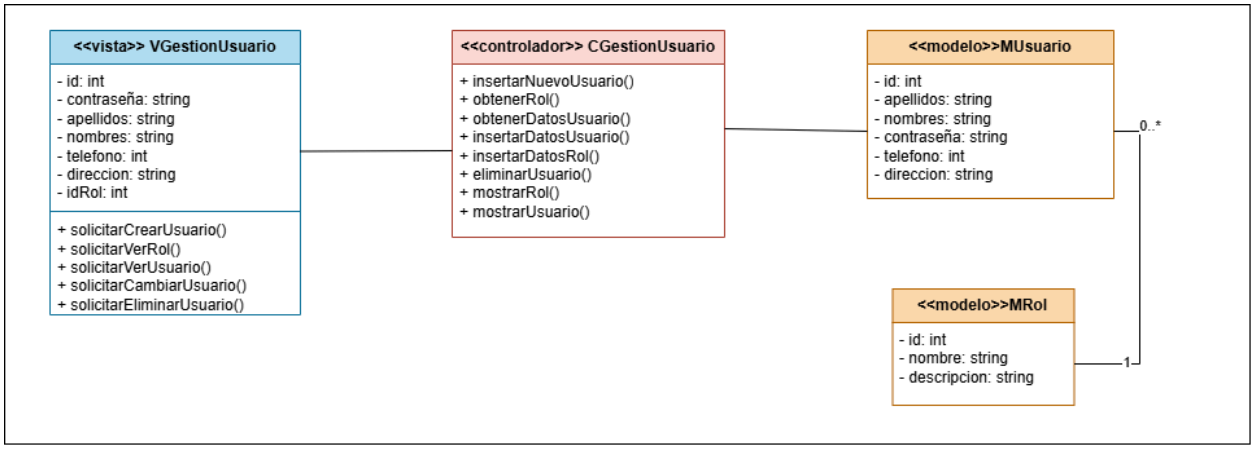
id pk	fechaNacimiento	sexo	id_usuario fk

2.1.3. Diseño de la lógica de negocio

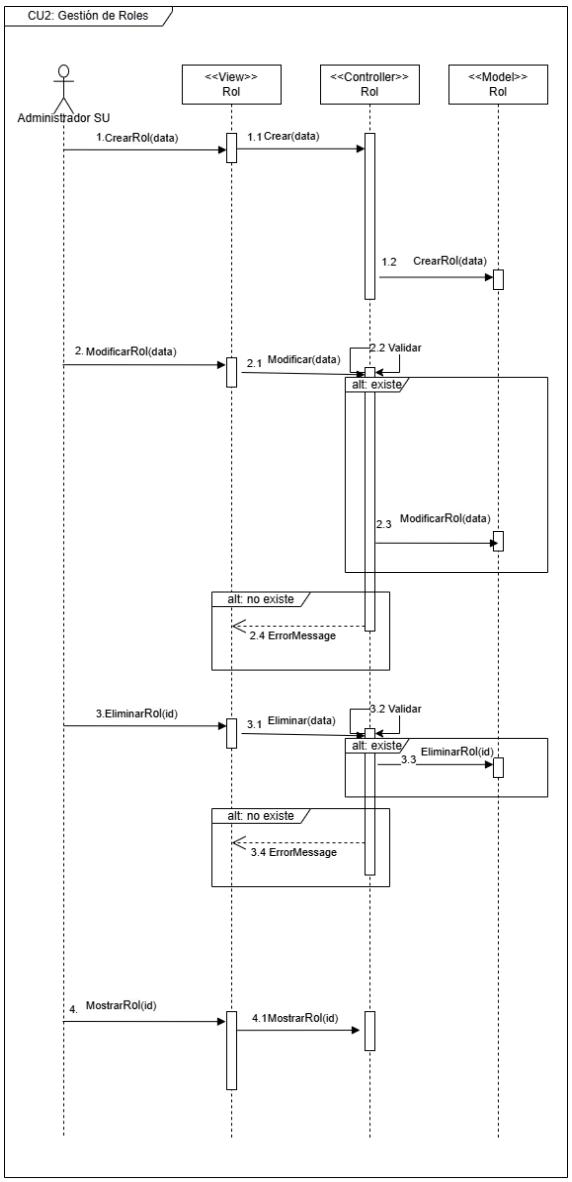
a) Diagrama de Secuencia y Diseño de Clase

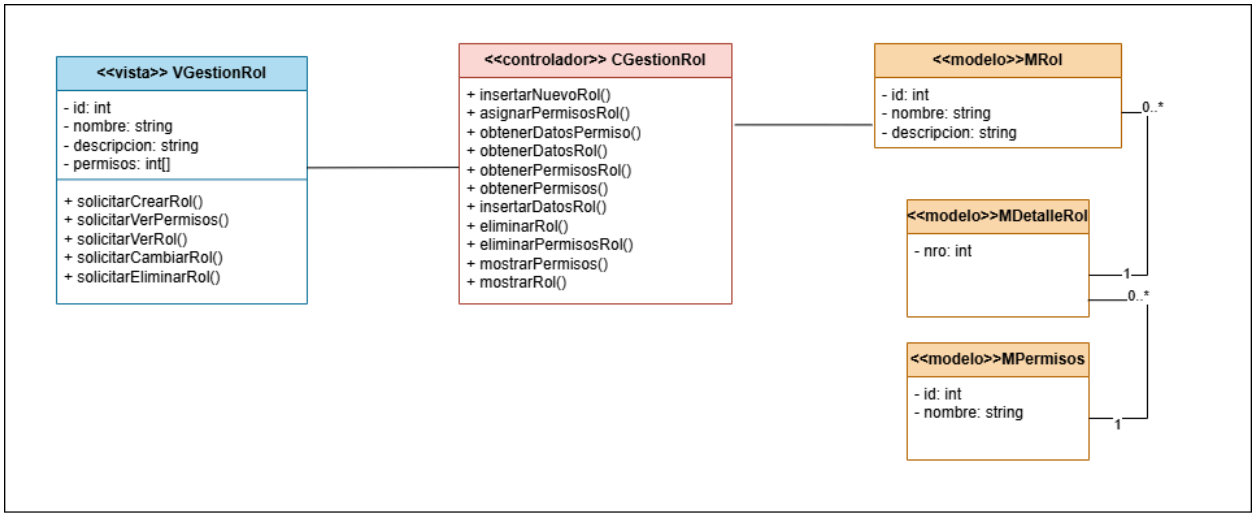
CU1: Gestión de Usuarios



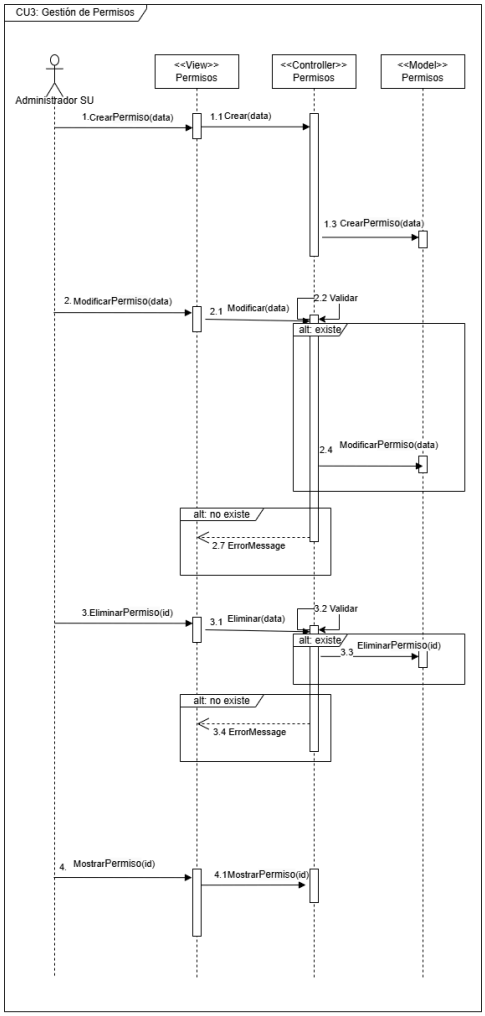


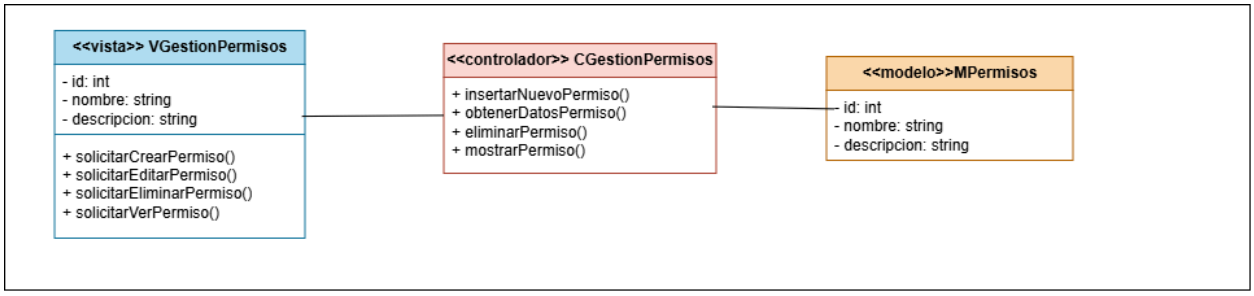
CU2: Gestión de Roles



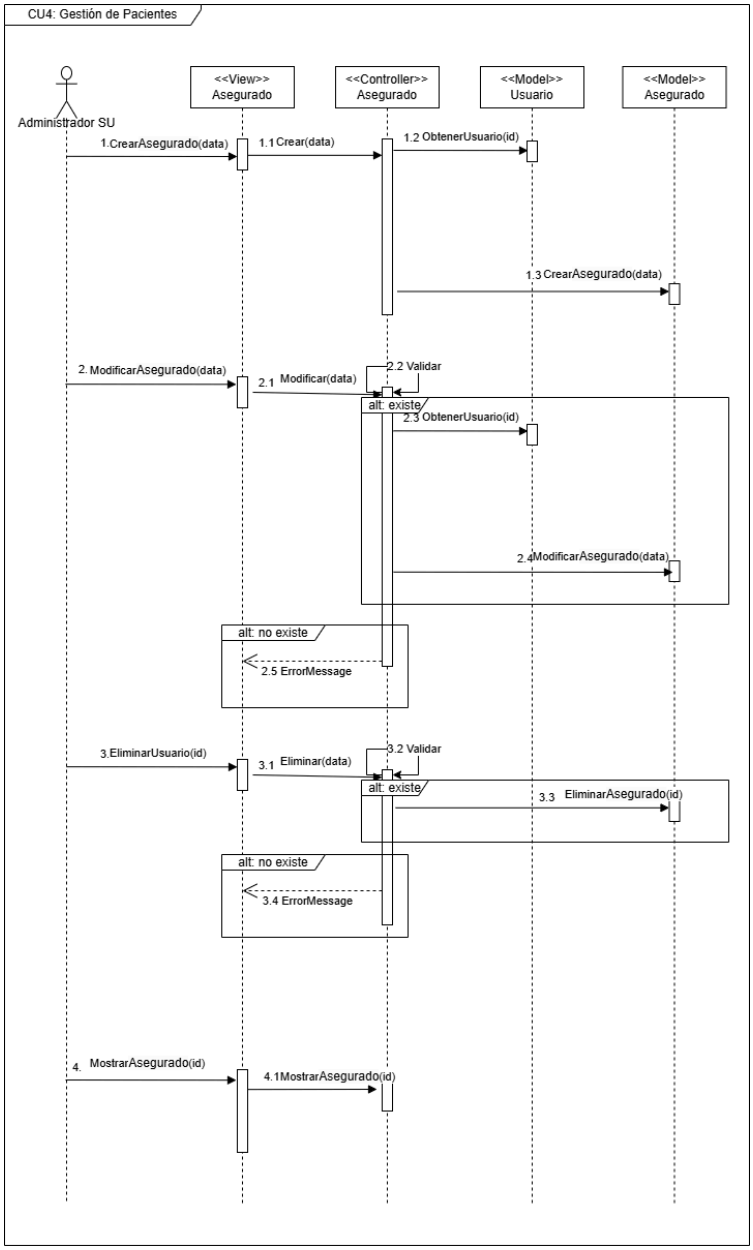


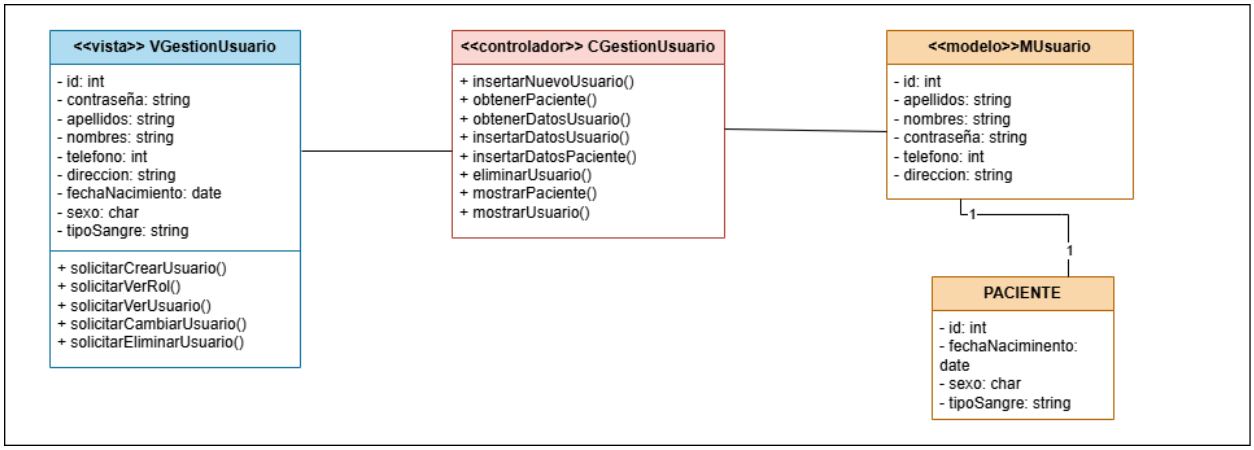
CU3: Gestión de Permisos



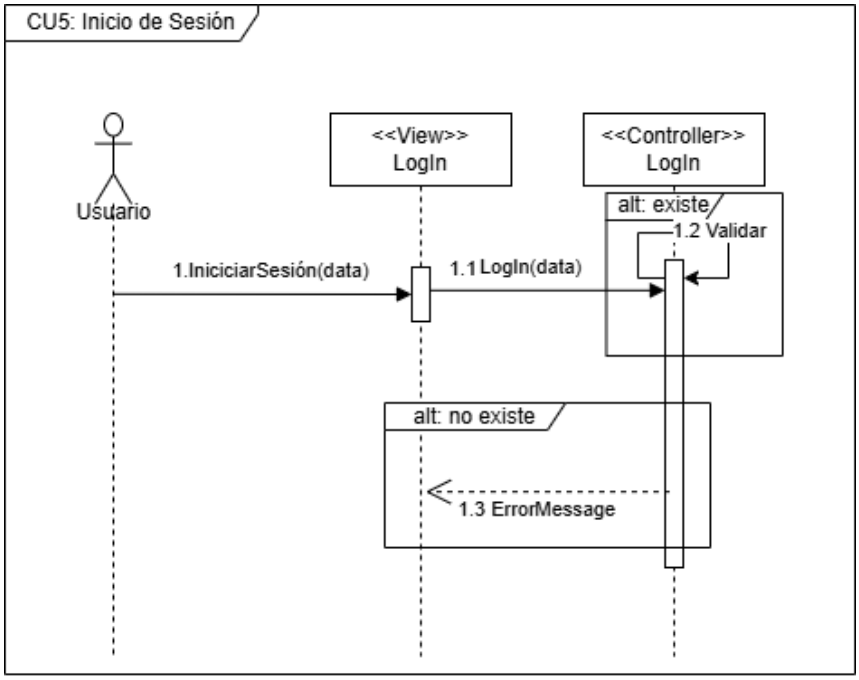


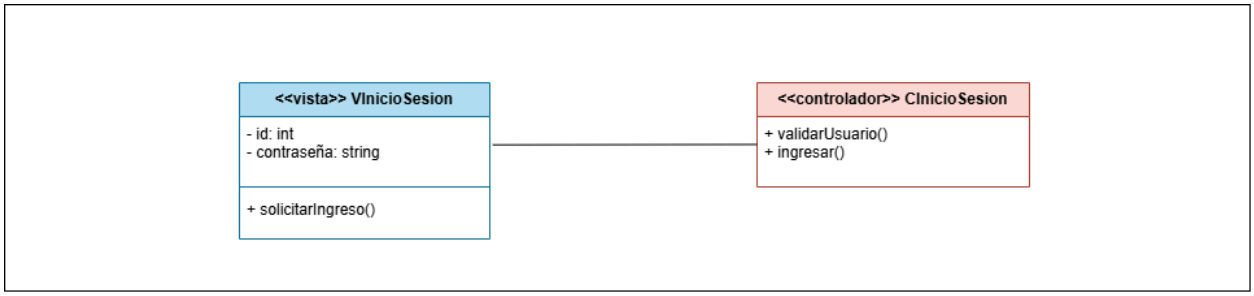
CU4: Gestionar Información de Pacientes



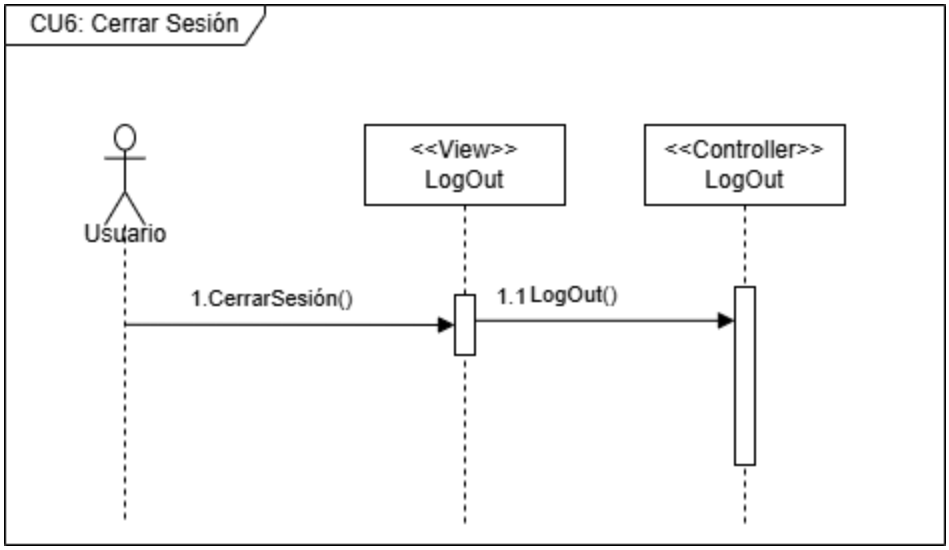


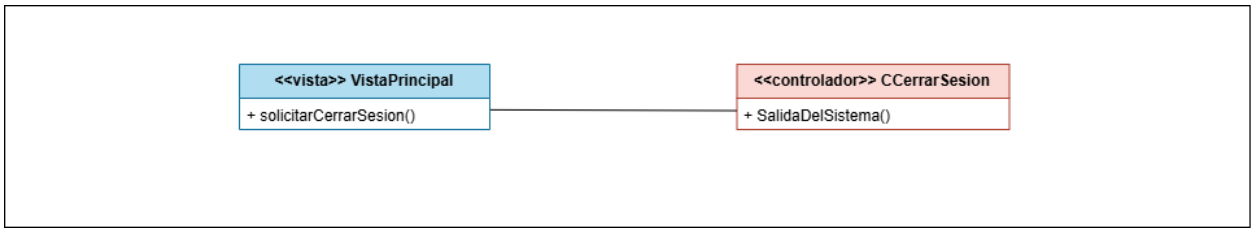
CU5: Iniciar Sesión





CU6: Cerrar Sesión



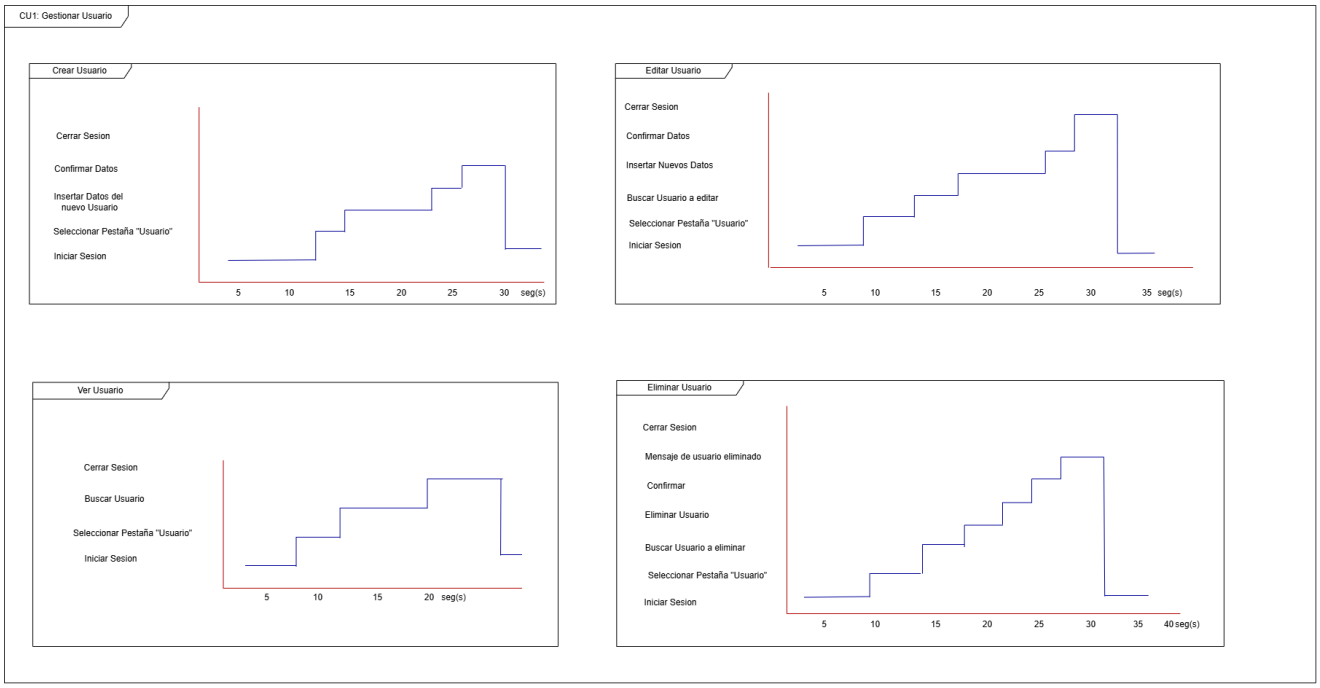


b) Diagrama de Estado

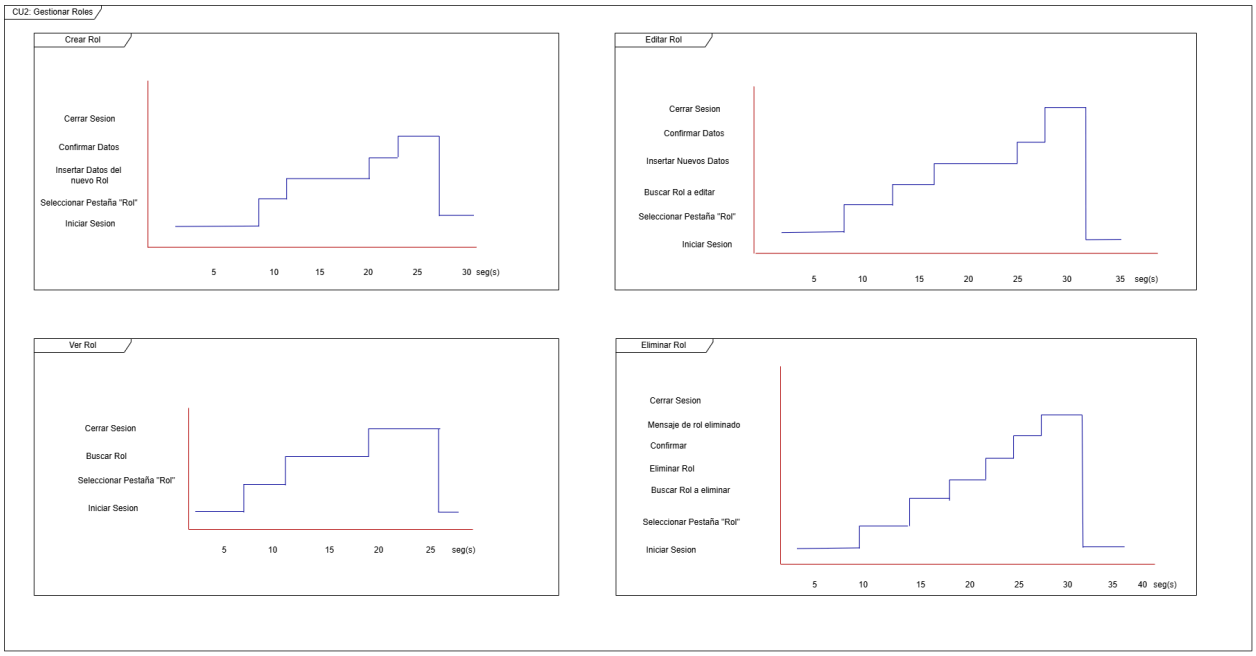
Debido a que en este sprint no se están desarrollando procesos que involucren cambios en los estados de los objetos o entidades, no se requiere la elaboración de diagramas de estados. Los casos de uso actuales se enfocan principalmente en la configuración y gestión de usuarios, roles, permisos y funcionalidades de autenticación y autorización, los cuales no implican transiciones de estados complejos entre las entidades. Por lo tanto, los diagramas de estados no son necesarios en esta fase del proyecto.

c) Diagrama de Tiempo

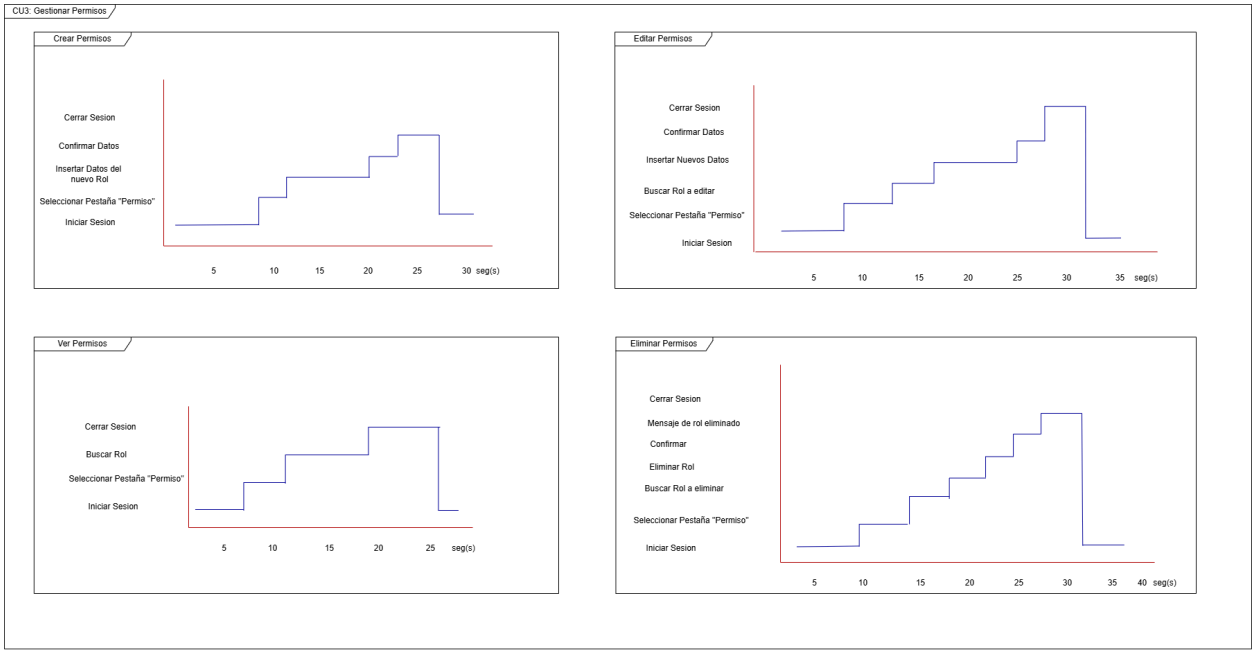
CU1: Gestión de Usuario



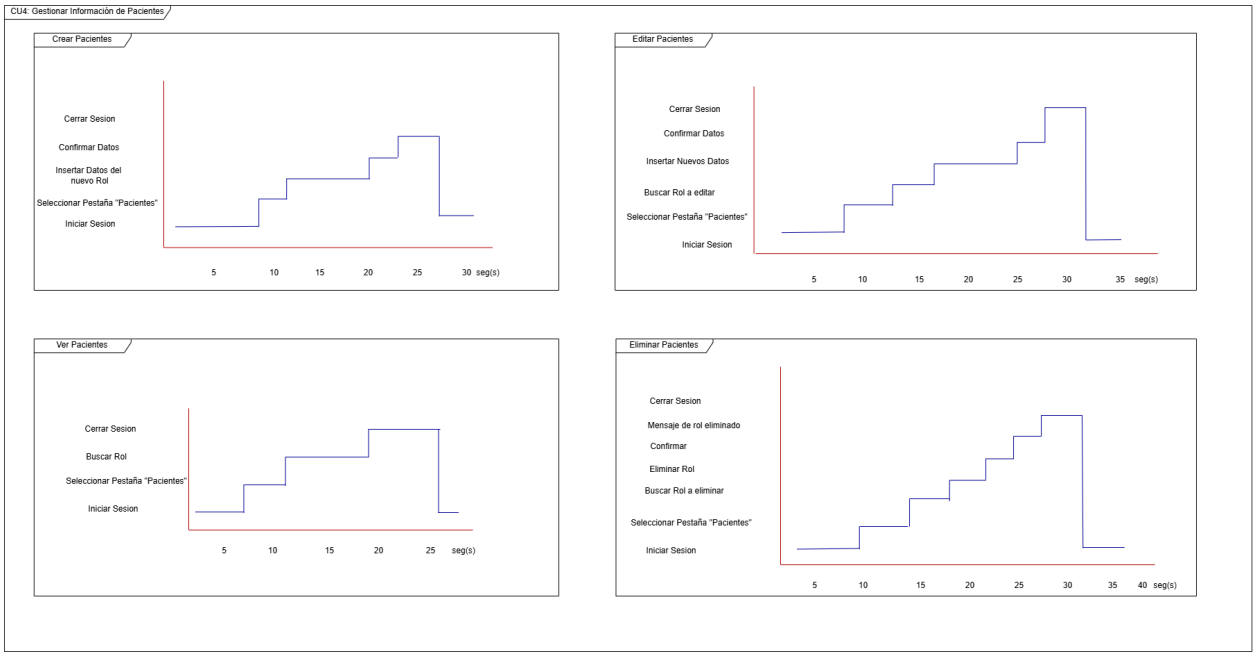
CU2: Gestión de Roles



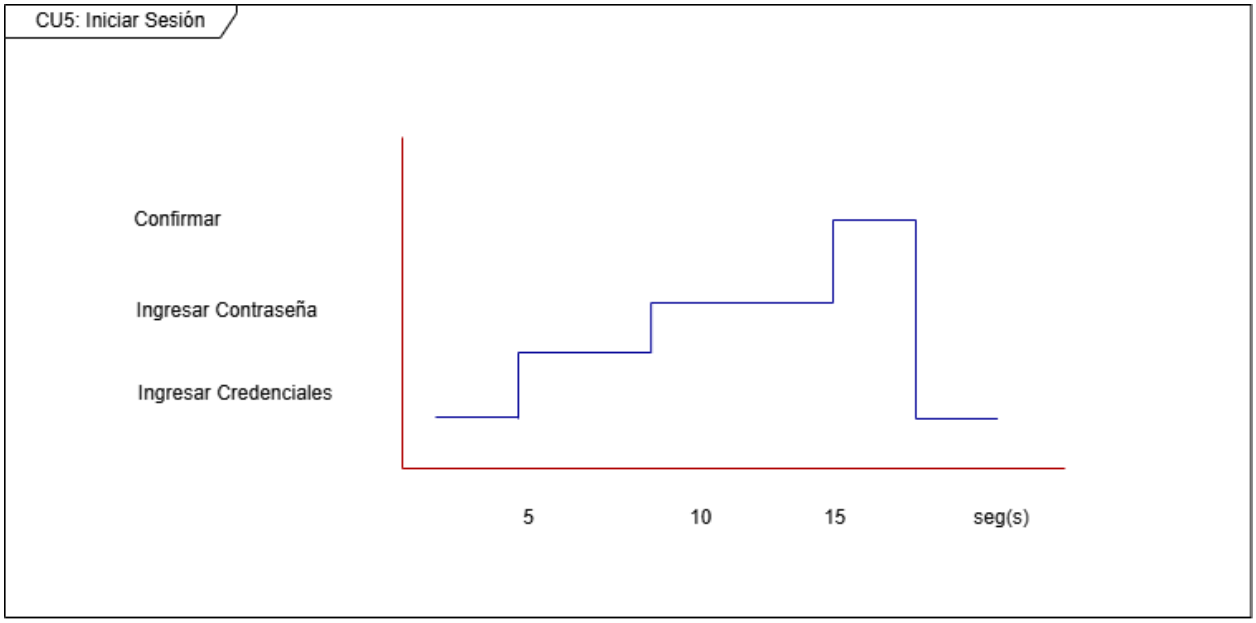
CU3: Gestión de Permisos



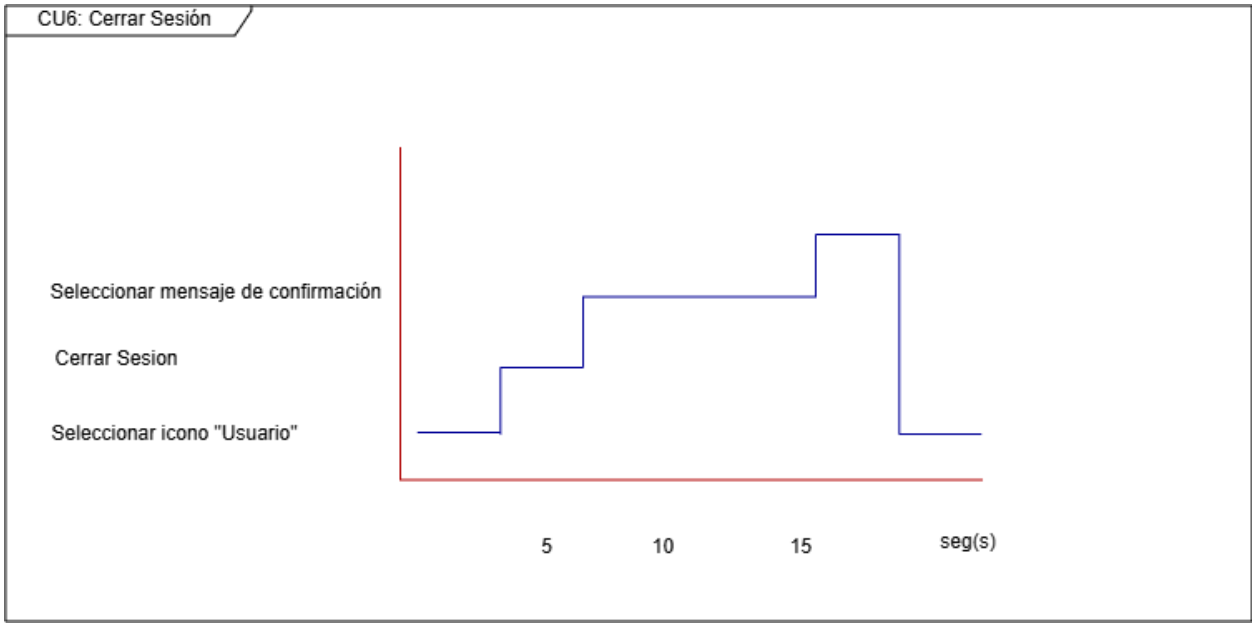
CU4: Gestionar Información de Pacientes



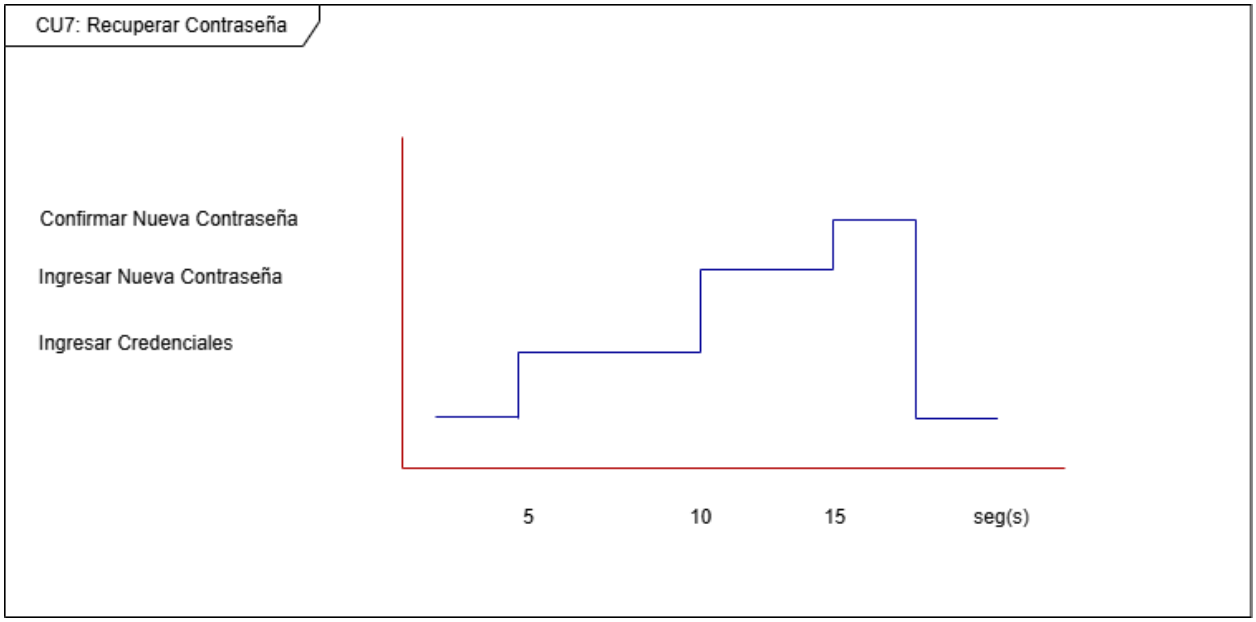
CU5: Iniciar Sesión



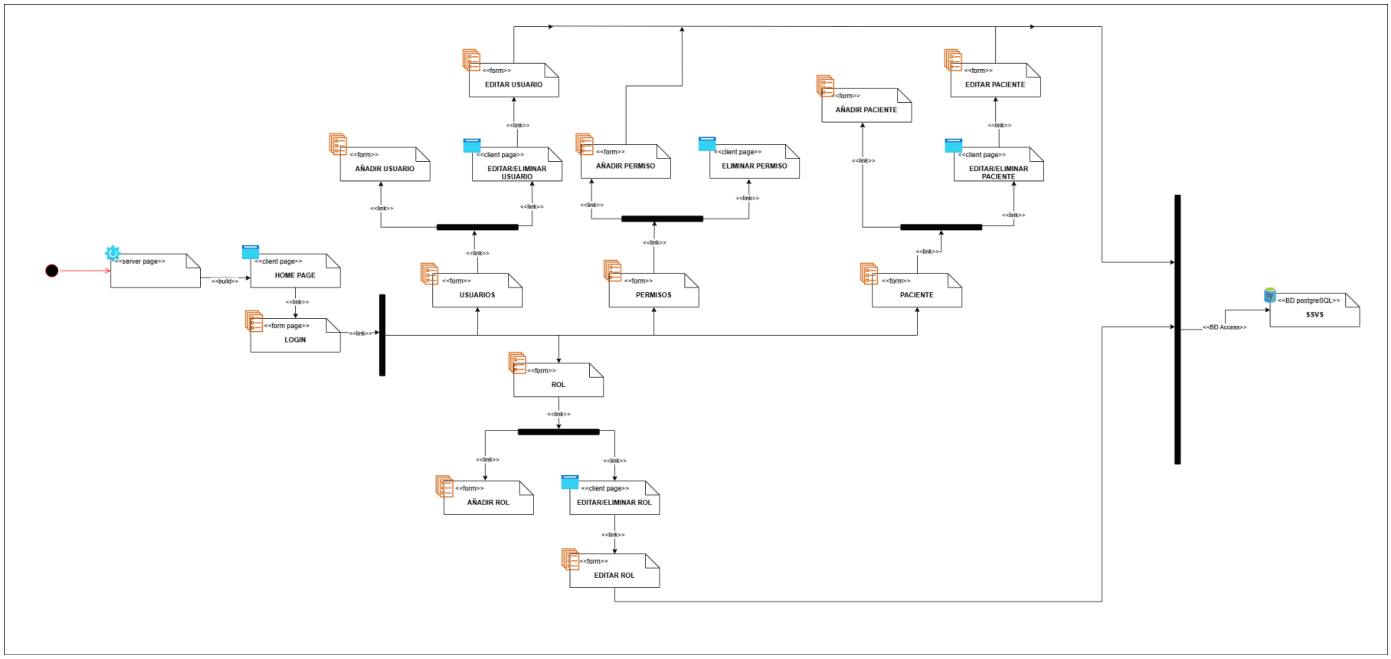
CU6: Cerrar Sesión



CU7: Recuperar Contraseña



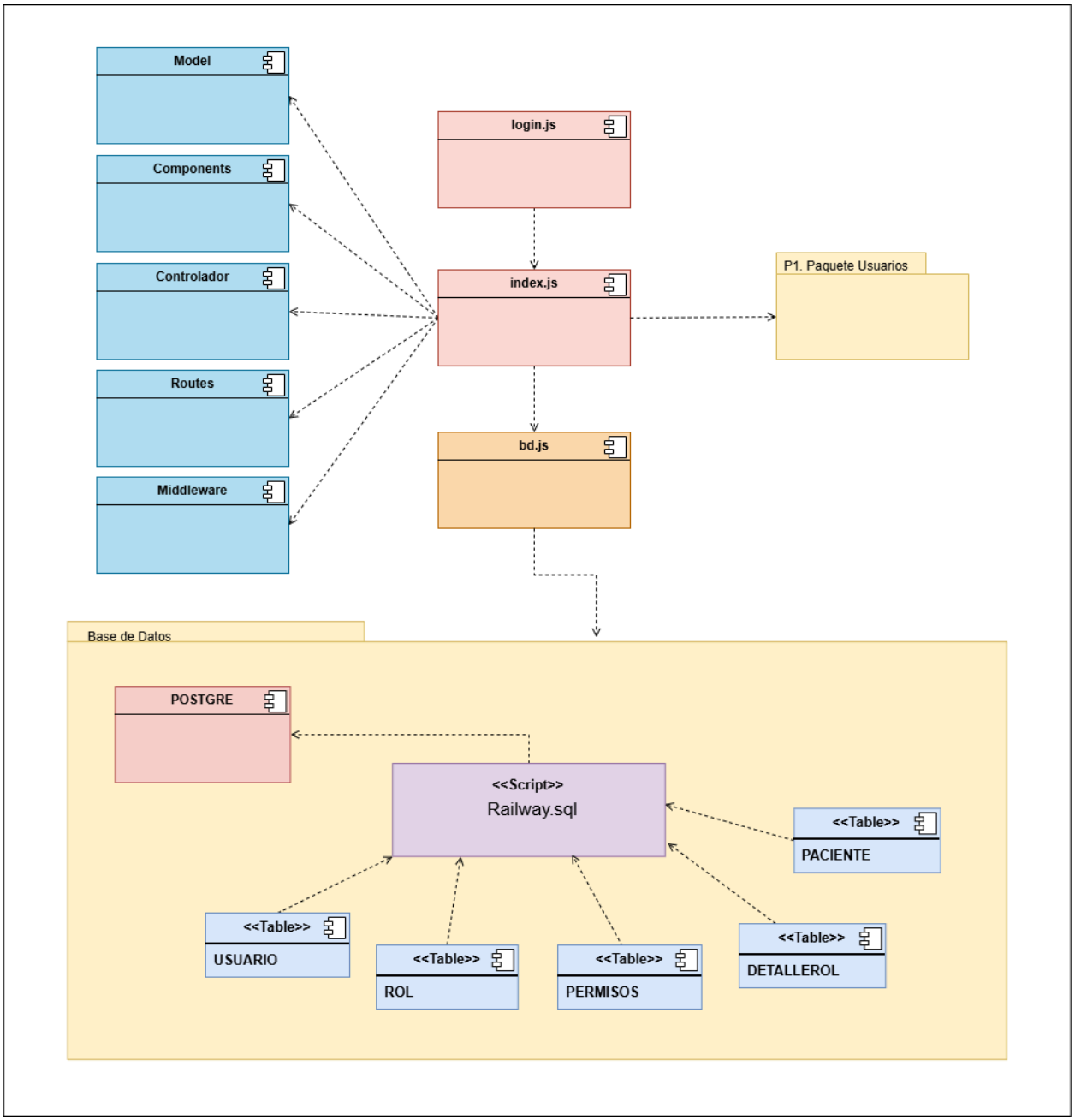
d) Diagrama de Navegación



2.2 Implementación

2.2.1. Componentes y artefactos generados

2.2.1.1 Diagrama de componentes



2.3. Pruebas

2.3.1. Plan de pruebas (criterios de aceptación)

CU1: Gestión de Usuarios

Descripción: Este caso de uso permite al Administrador gestionar los usuarios del sistema, que pueden ser Administradores, Médicos o Pacientes.

Criterios de Aceptación:

- El Administrador puede crear, modificar y eliminar usuarios.
- Los usuarios creados deben tener un rol asignado (Administrador, Médico o Paciente).
- El sistema debe validar la información requerida para crear un usuario (nombre, correo electrónico, contraseña, rol, etc.).
- No se permitirá la creación de usuarios con correos electrónicos duplicados.
- El sistema debe enviar una notificación por correo electrónico al usuario recién creado con las credenciales iniciales.
- El Administrador puede buscar usuarios por nombre, correo electrónico o rol.
- Los datos de los usuarios pueden actualizarse y las modificaciones se deben guardar correctamente.
- Solo los usuarios con el rol de Administrador pueden gestionar otros usuarios.

CU2: Gestión de Roles

Descripción: Permite gestionar los roles de usuario dentro del sistema. El rol define los permisos y accesos a las diferentes funcionalidades del sistema.

Criterios de Aceptación:

- El Administrador puede crear, modificar y eliminar roles.
- Al crear un rol, se debe asignar un conjunto de permisos disponibles.
- Los roles pueden ser editados para añadir o quitar permisos específicos.
- El sistema debe evitar la eliminación de un rol si está asignado a uno o más usuarios activos.
- Los cambios en los permisos asociados a un rol deben reflejarse de inmediato en los accesos de los usuarios que tengan ese rol.
- El Administrador puede listar y consultar los detalles de los roles y sus permisos asociados.

CU3: Gestión de Permisos

Descripción: Este caso de uso permite gestionar los permisos que se asignan a los roles del sistema, controlando el acceso a diferentes funcionalidades.

Criterios de Aceptación:

- El Administrador puede crear, modificar y eliminar permisos.
- Los permisos deben describir claramente la funcionalidad o acceso que controlan (ej. "Gestionar Pacientes", "Acceder a Historias Clínicas").
- No se permitirá la eliminación de un permiso si está asociado a un rol existente.
- El sistema debe permitir listar todos los permisos y consultar los detalles de cada uno.
- Los cambios en los permisos deben reflejarse inmediatamente en los roles asociados.

CU4: Gestionar Información de Pacientes

Descripción: Este caso de uso permite la gestión de la información clínica y personal de los pacientes.

Criterios de Aceptación:

- Los médicos pueden visualizar y actualizar la información clínica de los pacientes asignados a sus fichas de atención.
- El médico puede registrar exámenes, diagnósticos, recetas y tratamientos durante la consulta.
- El médico no puede modificar ni eliminar información previa registrada por él mismo o por otros médicos en consultas anteriores.
- Los datos personales de los pacientes (nombre, dirección, teléfono) pueden ser modificados solo por el Administrador.
- El paciente puede visualizar su información clínica a través del sistema, pero no puede modificarla.
- El sistema debe garantizar que cada paciente tenga una única historia clínica, que pueda ser actualizada en cada consulta médica.

CU5: Iniciar Sesión

Descripción: Los usuarios del sistema (Administrador, Médico, Paciente) pueden iniciar sesión con sus credenciales para acceder al sistema según su rol.

Criterios de Aceptación:

- El usuario puede iniciar sesión proporcionando un correo electrónico y contraseña válidos.
- El sistema debe verificar que las credenciales ingresadas coincidan con un usuario registrado.

- Si las credenciales son incorrectas, el sistema debe mostrar un mensaje de error adecuado.
- Los usuarios con credenciales correctas deben ser redirigidos a la interfaz adecuada según su rol (Administrador, Médico o Paciente).
- El sistema debe bloquear la cuenta temporalmente después de un número determinado de intentos fallidos (p. ej., cinco intentos fallidos).
- El sistema debe generar y validar tokens JWT para mantener la autenticación en la sesión.

CU6: Cerrar Sesión

Descripción: Permite que el usuario cierre su sesión en el sistema de manera segura.

Criterios de Aceptación:

- El usuario puede cerrar sesión de manera explícita utilizando una opción en la interfaz.
- Al cerrar sesión, el sistema debe invalidar el token JWT asociado con la sesión del usuario.
- El sistema debe redirigir al usuario a la página de inicio de sesión tras cerrar sesión.
- El sistema debe asegurar que el token de autenticación no pueda reutilizarse después de cerrar sesión.

CU7: Recuperar Contraseña

Descripción: Este caso de uso permite a los usuarios recuperar su contraseña en caso de que la hayan olvidado.

Criterios de Aceptación:

- El usuario puede solicitar la recuperación de su contraseña proporcionando su correo electrónico registrado.
- El sistema debe enviar un correo electrónico con un enlace de recuperación de contraseña al correo proporcionado, si está registrado.
- El enlace debe permitir al usuario restablecer su contraseña y debe ser válido solo por un tiempo limitado (p. ej., 24 horas).
- El sistema debe permitir que el usuario ingrese y confirme una nueva contraseña.
- El sistema debe validar la seguridad de la nueva contraseña (longitud mínima, complejidad).

- Una vez restablecida, el sistema debe permitir al usuario iniciar sesión con la nueva contraseña.

2.3.2. Reporte de prueba

CU1: Gestión de Usuarios

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de creación de usuario	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Usuarios". 3. Crear un nuevo usuario proporcionando los datos requeridos.	El sistema permite crear un nuevo usuario con todos los datos válidos, y muestra un mensaje de confirmación.	El usuario es creado exitosamente y se muestra un mensaje de confirmación.	El usuario fue creado correctamente, y se mostró el mensaje esperado.	Aprobado
Prueba de modificación de usuario	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Usuarios". 3. Editar un usuario existente modificando su nombre.	El sistema permite editar los datos de un usuario y guarda los cambios correctamente.	El usuario es actualizado con éxito, y se reflejan los cambios en el sistema.	Los cambios fueron aplicados correctamente.	Aprobado
Prueba de eliminación de usuario	1. Iniciar sesión como Administrador. 2. Navegar al módulo	El sistema permite eliminar usuarios, y los datos	El usuario es eliminado correctamente y	El usuario fue eliminado exitosamente.	Aprobado

	"Gestión de Usuarios". 3. Eliminar un usuario existente.	desaparecen del listado sin generar errores.	desaparece del sistema.		

CU2: Gestión de Roles

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de creación de rol	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Roles". 3. Crear un nuevo rol asignando los permisos requeridos.	El sistema permite crear roles con un conjunto de permisos y muestra un mensaje de confirmación.	El rol es creado con éxito y se refleja en el sistema.	El rol fue creado correctamente y se reflejan los permisos.	Aprobado
Prueba de modificación de rol	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Roles". 3. Modificar los permisos asignados a un rol existente.	El sistema permite modificar roles y actualizar los permisos asociados.	Los cambios en los permisos del rol son aplicados y reflejados en los usuarios asignados.	Los permisos fueron modificados correctamente.	Aprobado
Prueba de eliminación de rol	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Roles". 3. Eliminar un rol existente	El sistema elimina el rol correctamente solo si no está asignado a usuarios.	El rol es eliminado si no está asignado, y muestra un error en caso contrario.	El rol fue eliminado sin problemas.	Aprobado

	sin que esté asignado a usuarios activos.				

CU3: Gestión de Permisos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de creación de permiso	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Permisos". 3. Crear un nuevo permiso.	El sistema permite crear un nuevo permiso y muestra un mensaje de confirmación.	El permiso es creado con éxito y se refleja en el sistema.	El permiso fue creado correctamente.	Aprobado
Prueba de modificación de permiso	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Permisos". 3. Editar un permiso existente modificando su descripción.	El sistema permite modificar un permiso existente y guardar los cambios.	El permiso es modificado y los cambios se reflejan correctamente.	Los cambios fueron aplicados correctamente.	Aprobado
Prueba de eliminación de permiso	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Permisos". 3. Eliminar un permiso que no esté asignado	El sistema permite eliminar permisos no asignados a roles.	El permiso es eliminado correctamente si no está asignado a roles.	El permiso fue eliminado correctamente.	Aprobado

	a roles.				

CU4: Gestionar Información de Pacientes

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de creación de Paciente	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Paciente". 3. Crear un nuevo paciente proporcionando los datos requeridos.	El sistema permite crear un nuevo paciente con todos los datos válidos, y muestra un mensaje de confirmación.	El paciente es creado exitosamente y se muestra un mensaje de confirmación.	El paciente fue creado correctamente, y se mostró el mensaje esperado.	Aprobado
Prueba de modificación de Paciente	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Paciente". 3. Editar un paciente existente modificando su nombre.	El sistema permite editar los datos de un paciente y guarda los cambios correctamente.	El paciente es actualizado con éxito, y se reflejan los cambios en el sistema.	Los cambios fueron aplicados correctamente.	Aprobado
Prueba de eliminación de Paciente	1. Iniciar sesión como Administrador. 2. Navegar al módulo "Gestión de Paciente". 3. Eliminar un paciente existente.	El sistema permite eliminar paciente, y los datos desaparecen del listado sin generar errores.	El paciente es eliminado correctamente y desaparece del sistema.	El paciente fue eliminado exitosamente.	Aprobado

CU5: Iniciar Sesión

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de inicio de sesión	1. Navegar a la página de inicio de sesión. 2. Ingresar correo y contraseña válidos. 3. Hacer clic en "Iniciar sesión".	El sistema valida las credenciales y permite el acceso.	El usuario inicia sesión correctamente y es redirigido a la página correspondiente según su rol.	El inicio de sesión fue exitoso.	Aprobado
Prueba de error por credenciales incorrectas	1. Navegar a la página de inicio de sesión. 2. Ingresar un correo o contraseña incorrectos. 3. Hacer clic en "Iniciar sesión".	El sistema muestra un mensaje de error si las credenciales son incorrectas.	Se muestra un mensaje de error indicando credenciales incorrectas.	El error fue mostrado correctamente.	Aprobado

CU6: Cerrar Sesión

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de cierre de sesión	1. Iniciar sesión. 2. Hacer clic en "Cerrar sesión".	El sistema cierra la sesión del usuario y lo redirige a la página de inicio de sesión.	La sesión se cierra correctamente y el usuario es redirigido.	El cierre de sesión fue correcto.	Aprobado

CU7: Recuperar Contraseña

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Prueba de recuperación de contraseña	1. Hacer clic en "¿Olvidaste tu contraseña?" en la página de inicio de sesión. 2. Ingresar el correo registrado. 3. Restablecer la contraseña mediante el enlace enviado.	El sistema envía un correo con un enlace para restablecer la contraseña y permite cambiarla correctamente.	El correo de recuperación es enviado y la contraseña es restablecida correctamente.	El correo fue enviado correctamente y la contraseña fue cambiada.	Aprobado

3. Daily Scrum

Daily Scrum 21/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para iniciar y cerrar sesión	Ninguno	Implementación del backend para la gestión de pacientes
Fabiana Manzoni Bravo	Implementación del frontend para iniciar y cerrar sesión	Ninguno	Implementación del frontend para la gestión de usuarios

Daily Scrum 22/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*

Jorge Arturo Aliaga Valencia	Implementación del backend para la gestión de pacientes	Ninguno	Implementación del backend para la recuperación de contraseña
Fabiana Manzoni Bravo	Implementación del frontend para la gestión de usuarios	Ninguno	Implementación del frontend para la asignación de permisos

Daily Scrum 23/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para la recuperación de contraseña	Ninguno	Implementación del frontend para la gestión de roles
Fabiana Manzoni Bravo	Implementación del frontend para la asignación de permisos	Ninguno	Implementación del frontend para la gestión de roles

Daily Scrum 24/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del frontend para la gestión de pacientes	Ninguno	Revisión y pruebas del backend para recuperación de contraseña
Fabiana Manzoni Bravo	Implementación del frontend para la gestión de roles	Ninguno	Revisión del frontend para gestión de usuarios

Daily Scrum 25/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Revisión del backend para la gestión de pacientes	Ninguno	Implementación del backend para la asignación de permisos
Fabiana Manzoni Bravo	Revisión del frontend para la gestión de usuarios	Ninguno	Pruebas finales de gestión de usuarios en frontend

Daily Scrum 26/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para la asignación de permisos	Ninguno	Revisión final del backend para la asignación de roles
Fabiana Manzoni Bravo	Pruebas finales de gestión de usuarios en frontend	Ninguno	Revisión del frontend para la recuperación de contraseñas

Daily Scrum 27/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Revisión final del backend para la asignación de roles	Ninguno	Implementación de pruebas completas del backend

Fabiana Manzoni Bravo	Revisión del frontend para la recuperación de contraseñas	Ninguno	Implementación de pruebas finales en el frontend

4. Sprint Review

Revisión de Sprint	
Objetivo del Sprint: El objetivo del sprint es desarrollar e implementar las funcionalidades centrales del Sistema de Gestión de Historias Clínicas Electrónicas (HCE) para el Seguro de Salud Vida Sana (SSVS), enfocándose en la gestión de usuarios, roles, permisos y la recuperación de contraseñas. Además, se incluirán funcionalidades críticas relacionadas con la gestión de pacientes.	
Participantes	
Nombre	**Rol**
Jorge Arturo Aliaga Valencia	Desarrollador
Fabiana Manzoni Bravo	Desarrollador
Presentación del incremento	
Función Presentada	**Retroalimentación**
Implementación del frontend para iniciar y cerrar sesion.	Funcionalidad Terminada
Implementacion del backend para iniciar y cerrar sesion.	Funcionalidad Terminada
Implementación movil para iniciar y cerrar sesion.	Funcionalidad Terminada
Implementación del frontend para la gestión de usuarios.	Funcionalidad Terminada
Implementación del backend para la gestión de usuarios.	Funcionalidad Terminada
Implementación del frontend para la gestión de pacientes.	Funcionalidad Terminada

Implementación del backend para la gestión de pacientes.	Funcionalidad Terminada
Implementación del frontend para la gestión de roles.	Funcionalidad Terminada
Implementación del backend para la gestión de roles.	Funcionalidad Terminada
Implementación del frontend para la asignación de permisos.	Funcionalidad Terminada
Implementación del backend para la asignación de permisos.	Funcionalidad Terminada
Implementación del backend para Recuperar Contraseña.	Funcionalidad Terminada
Implementación del frontend para Recuperar Contraseña.	Funcionalidad Terminada
Implementación del frontend para iniciar y cerrar sesion.	Funcionalidad Terminada
Implementación del backend para iniciar y cerrar sesión.	Funcionalidad Terminada
Implementación movil para iniciar y cerrar sesión.	Funcionalidad Terminada
Implementación del frontend para la gestión de usuarios.	Funcionalidad Terminada
Implementación del backend para la gestión de usuarios.	Funcionalidad Terminada
Implementación del frontend para la gestión de pacientes.	Funcionalidad Terminada

5. Sprint Retrospective

Sprint Retrospective
Fecha
27/10/2024
Facilitador
Fabiana Manzoni Bravo
Objetivo

Conclusión del Sprint 1
Nombres de asistentes
Jorge Arturo Aliaga Valencia
Temas por tratar
Sprint 1
Discusión
¿Qué salió bien?
1. **Cumplimiento de tareas**: Las funcionalidades clave como la gestión de usuarios, roles, y permisos fueron completadas según lo planeado. 2. **Integración fluida**: La integración entre el backend y frontend (web y móvil) funcionó sin problemas. 3. **Colaboración del equipo**: La distribución de tareas fue clara y permitió un buen flujo de trabajo. 4. **Pruebas exitosas**: Todas las funcionalidades pasaron las pruebas según los criterios de aceptación.
¿Qué no salió bien?
1. **Retrasos en la revisión**: La revisión final de algunas funcionalidades tomó más tiempo del planeado. 2. **Problemas de usabilidad móvil**: Se identificaron pequeños problemas de usabilidad en la app móvil. 3. **Desafíos en la gestión de permisos**: La lógica de permisos requirió más tiempo del previsto para su correcta implementación.
¿Qué haremos de forma diferente?
1. **Pruebas continuas**: Implementaremos pruebas a lo largo del sprint en lugar de concentrarlas al final. 2. **Mejorar la comunicación**: Se reforzará la coordinación entre frontend y backend para evitar retrasos. 3. **Optimización móvil**: Haremos pruebas de usabilidad móvil tempranas para mejorar la experiencia. 4. **Planificación de permisos**: Planificaremos mejor la implementación de la gestión de permisos para evitar problemas futuros.

6. Burndown y BurnUp (Grafica de tareas y Datos de tareas)

Diagrama Burndown

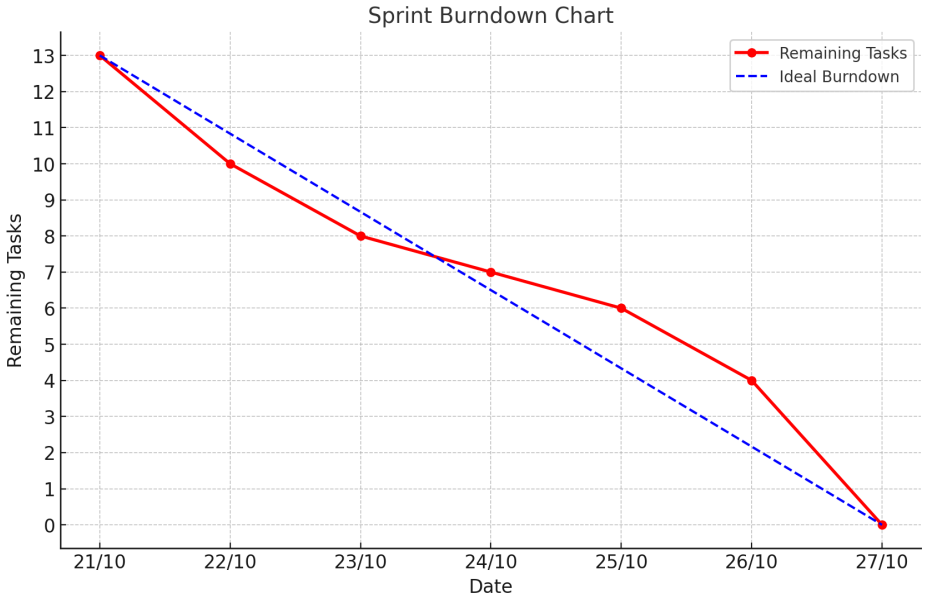
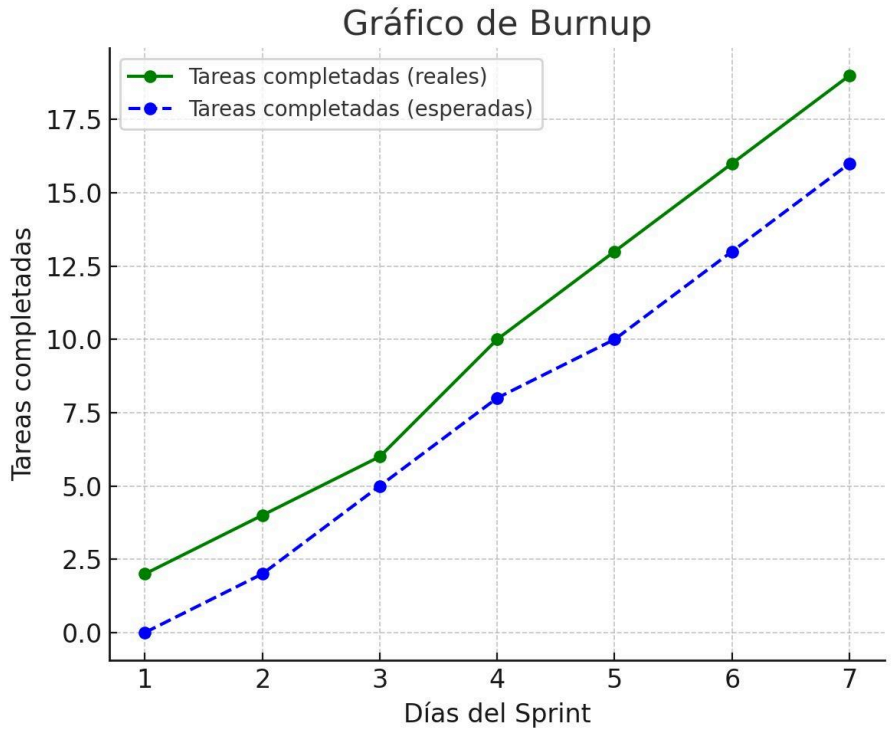
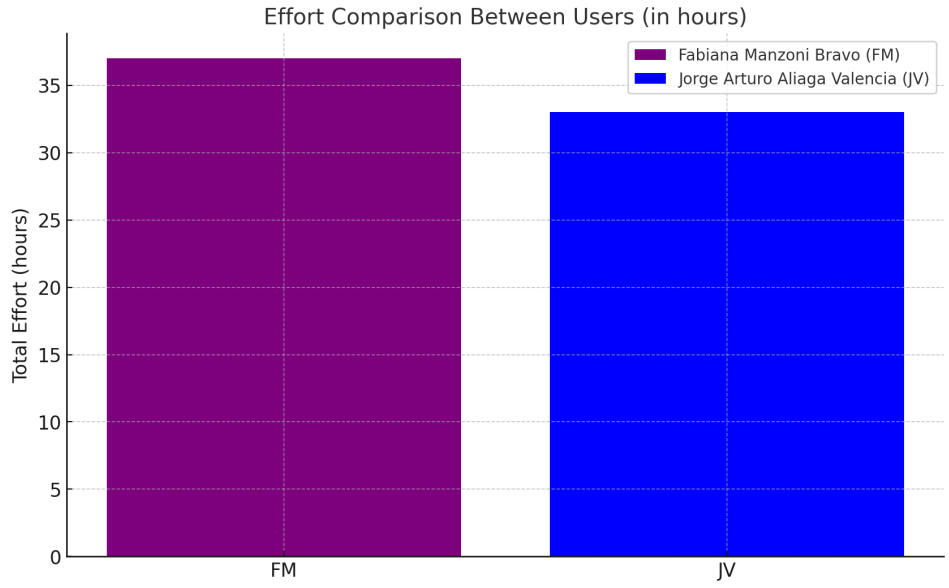


Diagrama BurnUp

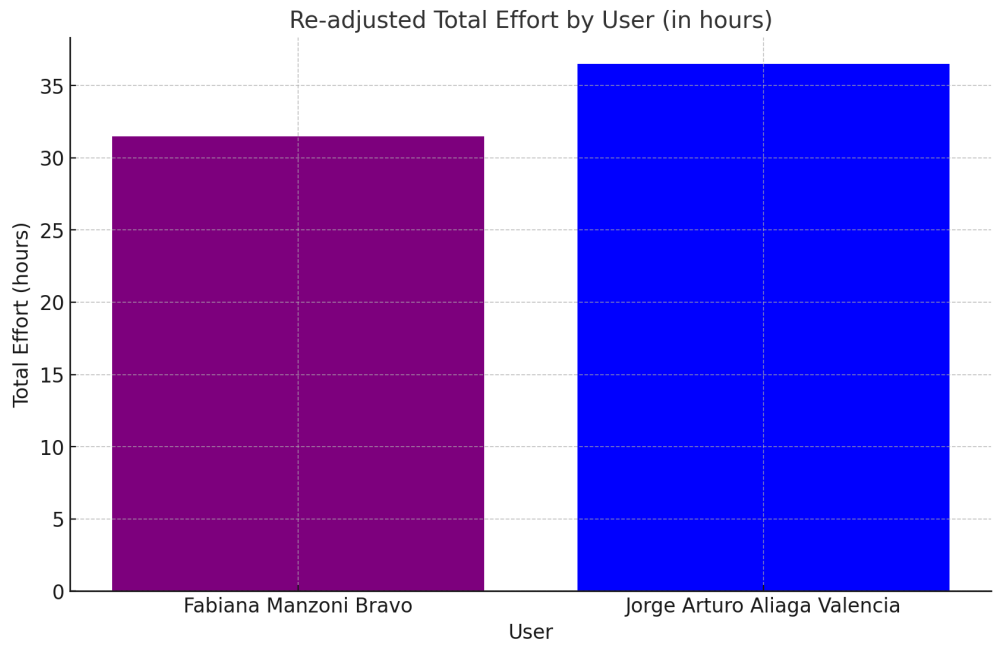


7. Grafica de esfuerzo y Datos de esfuerzo

Grafica de esfuerzo



Datos de esfuerzo



8. Scrum TaskBoard (Backlog, to do, doing, done)

SS-6	Implementacion del frontend para iniciar y cerrar sesion.	LOGIN	DONE	JV
SS-7	Implementacion del backend para iniciar y cerrar sesion.	LOGIN	DONE	JV
SS-8	Implementacion movil para iniciar y cerrar sesion	LOGIN	DONE	FM
SS-9	Implementacion del frontend para la gestion de usuarios.	USERS CRUD	DONE	FM
SS-10	Implementacion del backend para la gestion de usuarios.	USERS CRUD	DONE	FM
SS-11	Implementacion del frontend para la gestion de pacientes.	PATIENT CRUD	DONE	JV
SS-12	Implementacion del backend para la gestion de pacientes.	PATIENT CRUD	DONE	JV
SS-13	Implementacion del frontend para la gestion de roles.	ROLE CRUD	DONE	FM
SS-14	Implementacion del backend para la gestion de roles.	ROLE CRUD	DONE	FM
SS-15	Implementacion del frontend para la asignacion de permisos.	ROLE CRUD	DONE	FM
SS-16	Implementacion del backend para la asignacion de permisos.	ROLE CRUD	DONE	FM
SS-19	Implementacion del backend para Recuperar Contraseña	PASSWORD	DONE	JV
SS-20	Implementacion del frontend para Recuperar Contraseña	PASSWORD	DONE	JV

Sprint 2

1. Sprint Planning

1.1. Objetivos del Sprint

El objetivo del tercer sprint es desarrollar e implementar funcionalidades clave del Sistema de Gestión de Historias Clínicas Electrónicas (HCE), enfocadas en la administración de consultas médicas y registros clínicos. En este sprint, se habilitarán las funcionalidades de gestión de especialidades, médicos y horarios de atención, permitiendo a los pacientes solicitar y gestionar fichas de atención en función de la disponibilidad médica. Además, se implementará la gestión de consultas médicas para que los médicos puedan registrar y actualizar información en la historia clínica de los pacientes durante cada consulta, asegurando un control completo y trazable de la información médica en el Seguro de Salud Vida Sana.

1.1.1 Casos de uso

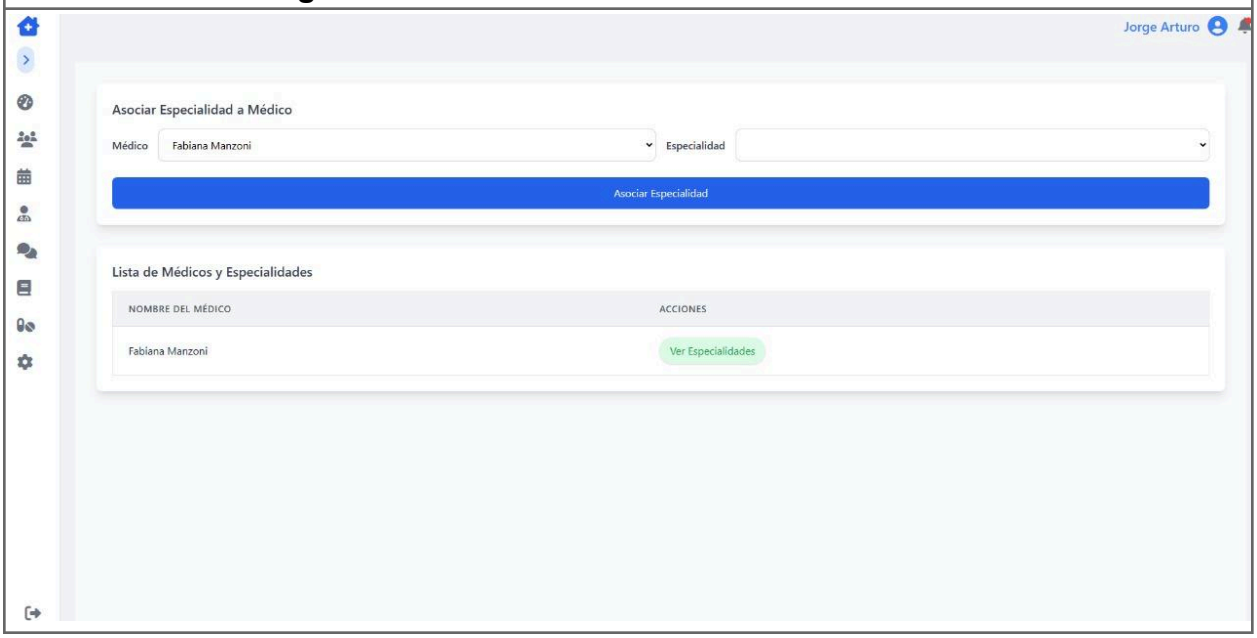
- CU8: Gestión de Especialidades Médicas
- CU9: Gestión de Médicos
- CU10: Gestionar Horarios de Atención de Médicos
- CU11: Gestionar Reserva
- CU12: Gestionar Consultas Médicas
- CU13: Solicitar Permiso
- CU14: Gestionar Postergación
- CU15: Gestionar Receta Médica
- CU16: Programar Laboratorio
- CU17: Gestionar Laboratorio
- CU18: Gestionar Medicamentos
- CU19: VisualizarHistoria Clínica
- CU23: Enviar Recordatorio de Cita al Paciente

1.2. Historias de usuario

CU8: Gestión de Especialidades Médicas

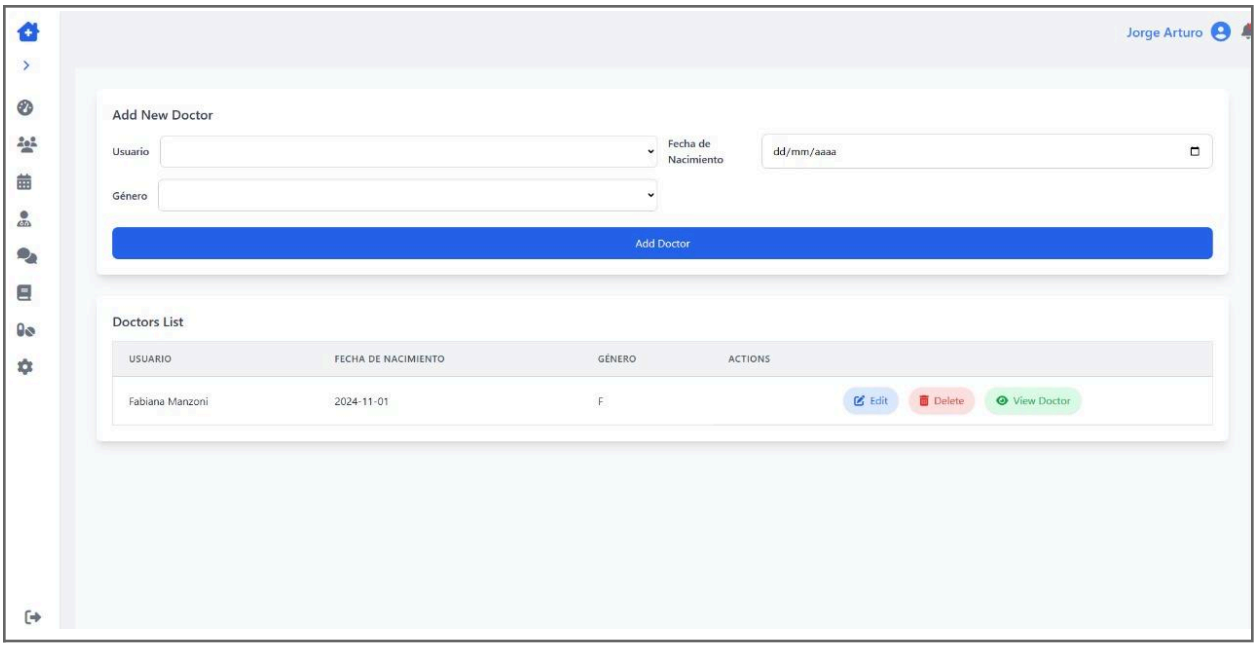
Gestión de Especialidades Médicas		
ID: 8	**Descripción: Gestión de Especialidades Médicas en el sistema**	
Prioridad: Alta		**Estimación PHU: 6hr**

Criterios de aceptación: ● Crear, consultar, modificar y eliminar especialidades médicas en el sistema.
Desarrollador a cargo: Manzoni Bravo Fabiana



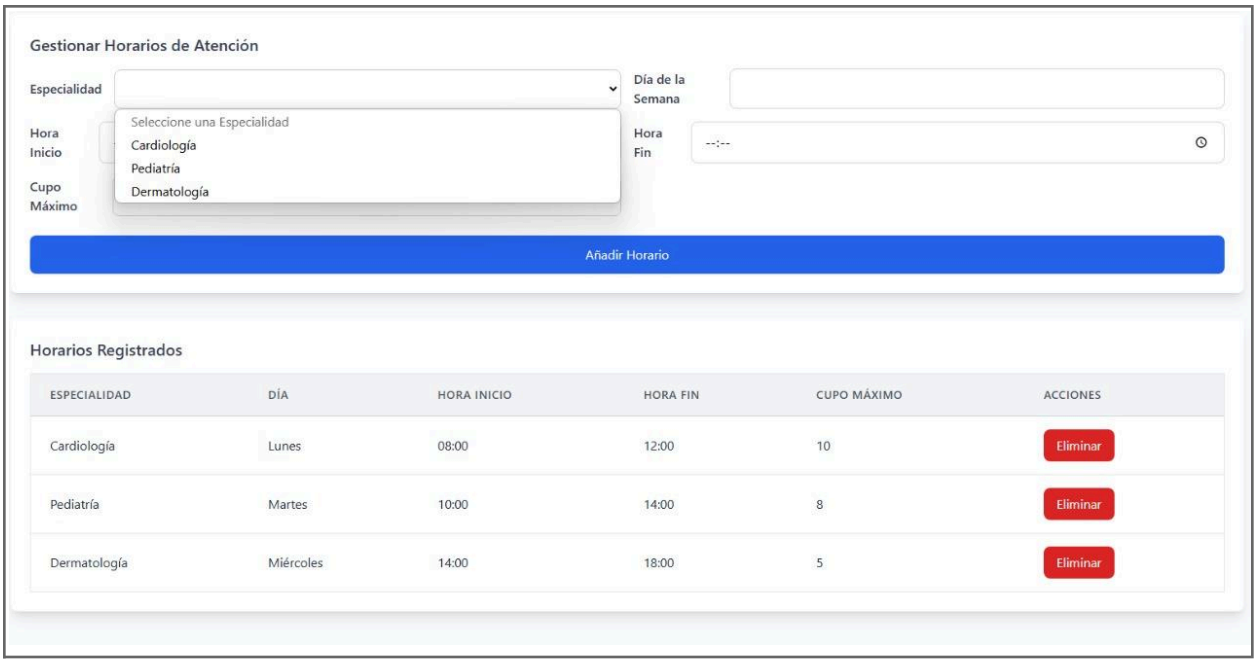
CU9: Gestión de Médicos

Gestión de Médicos	
ID: 9	**Descripción: Gestión de Médicos en el sistema**
Prioridad: Alta	**Estimación PHU: 5hr**
Criterios de aceptación: ● Crear, consultar, modificar y eliminar médicos, asignando especialidades y horarios de atención.	
Desarrollador a cargo: Manzoni Bravo Fabiana	



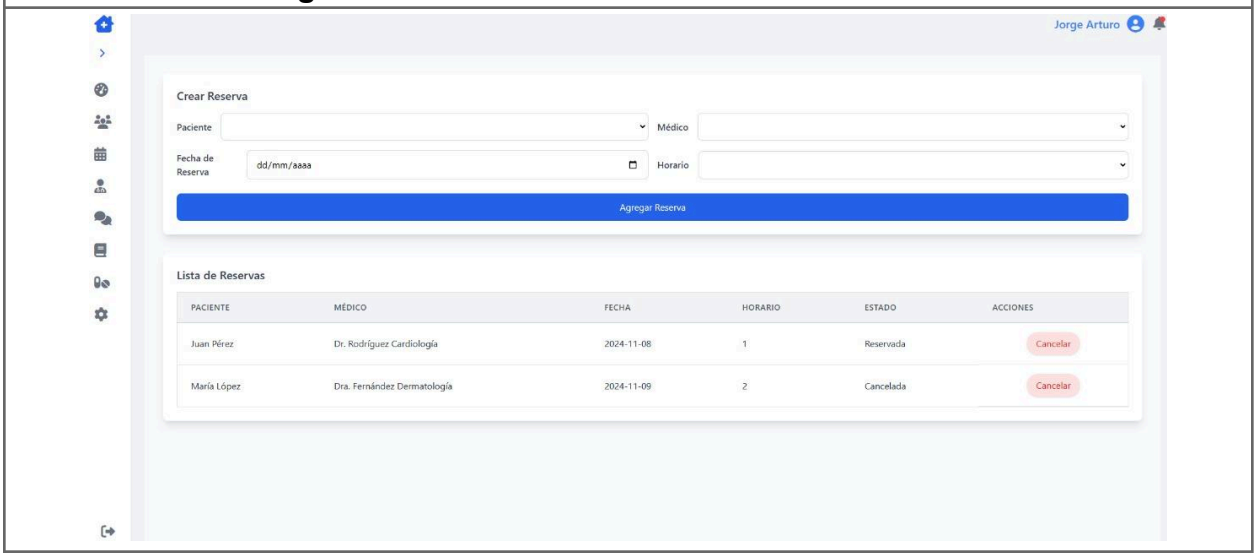
CU10: Gestionar Horarios de Atención de Médicos

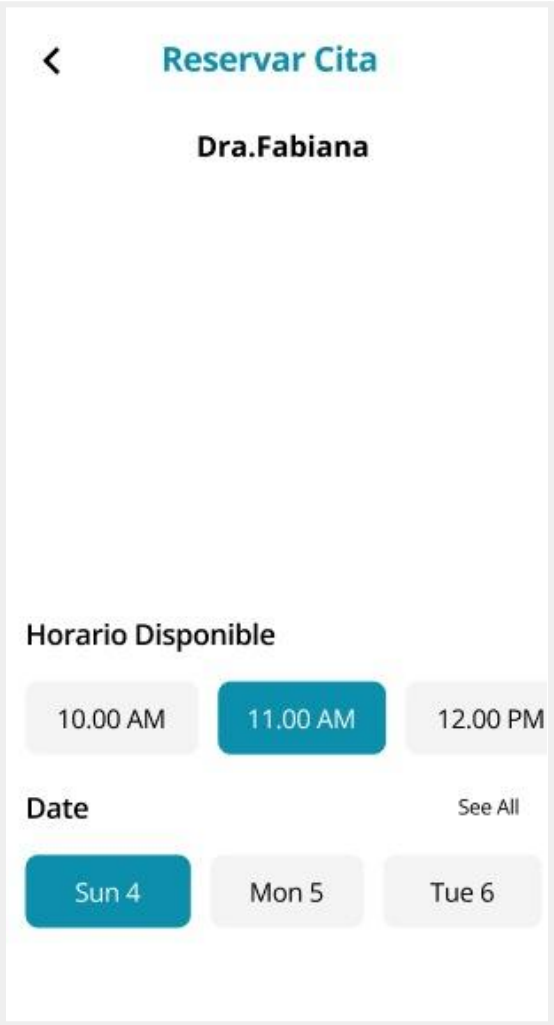
Gestionar Horarios de Atención de Médicos	
ID: 10	**Descripción: Gestionar Horarios de Atención de Médicos en el sistema**
Prioridad: Alta	**Estimación PHU: 8hr**
Criterios de aceptación: ● Configurar, consultar, modificar y eliminar horarios específicos de atención para cada médico.	
Desarrollador a cargo: Manzoni Bravo Fabiana	



CU11: Gestionar Reserva

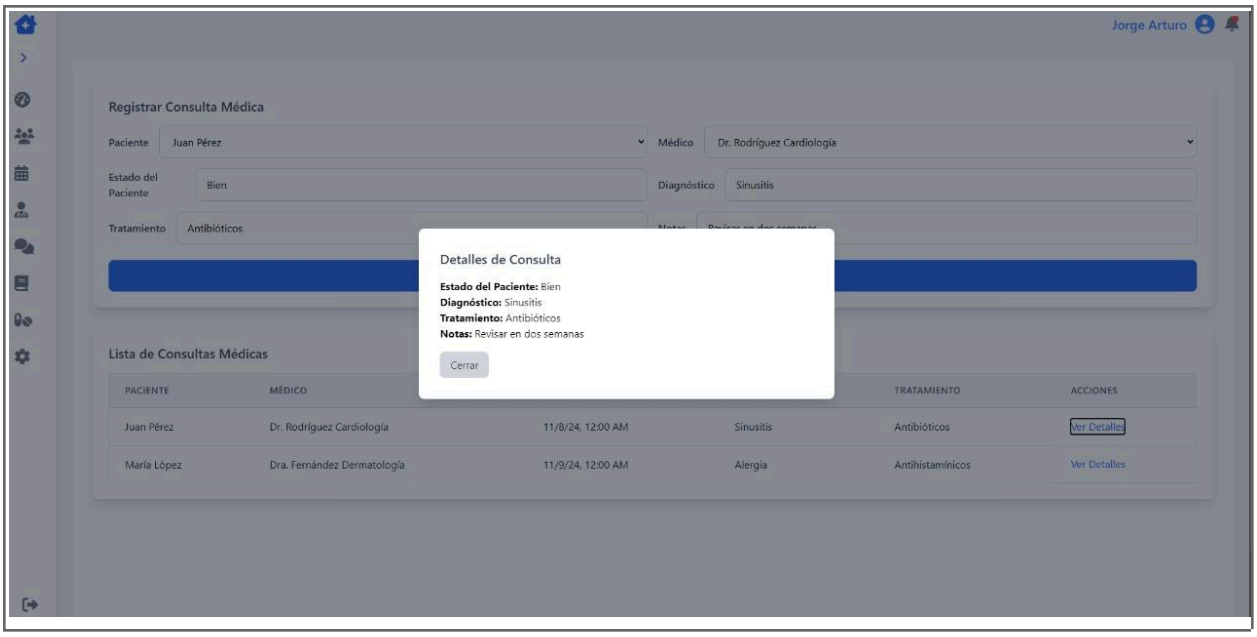
Gestionar Reserva	
ID: 11	**Descripción: Gestionar Reservas en el sistema**
Prioridad: Alta	**Estimación PHU: 8hr**
Criterios de aceptación: ● Consultar Fichas Disponibles: Ver disponibilidad de fichas de atención para los pacientes.	
Desarrollador a cargo: Manzoni Bravo Fabiana	





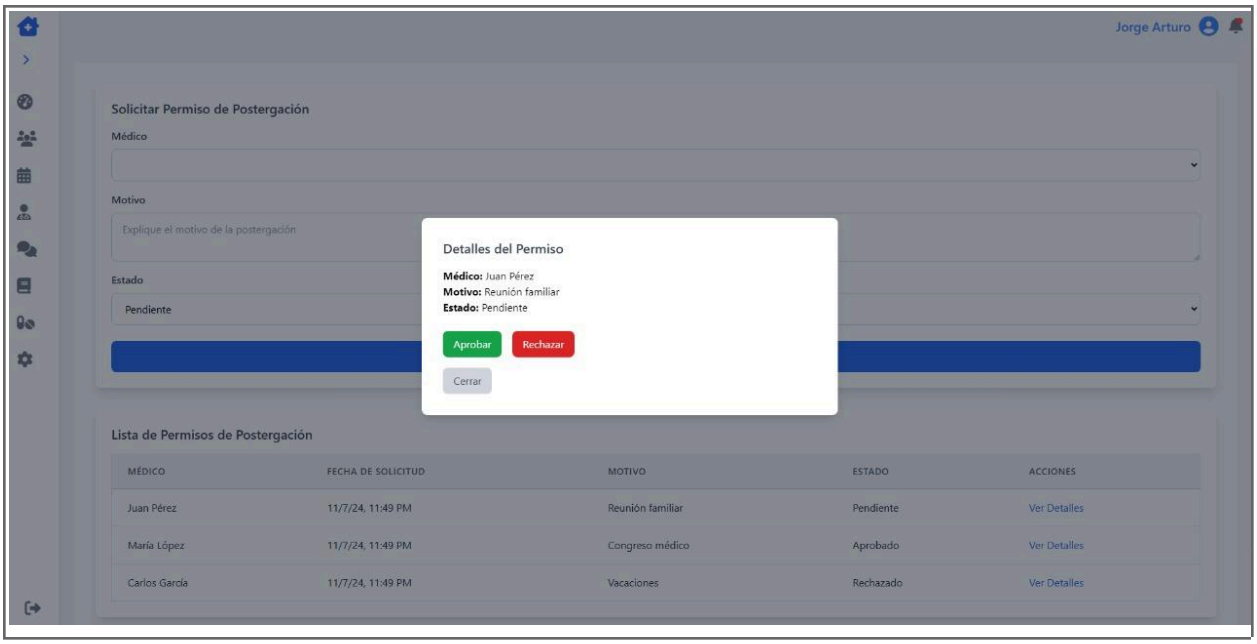
CU12: Gestionar Consultas Médicas

Gestionar Consultas Médicas	
ID: 12	**Descripción: Gestionar Consultas Médicas en el sistema**
Prioridad: Alta	**Estimación PHU:5hr**
Criterios de aceptación: ● Registrar los detalles de la consulta en la historia clínica del paciente, incluyendo diagnóstico y tratamiento.	
Desarrollador a cargo: Manzoni Bravo Fabiana	



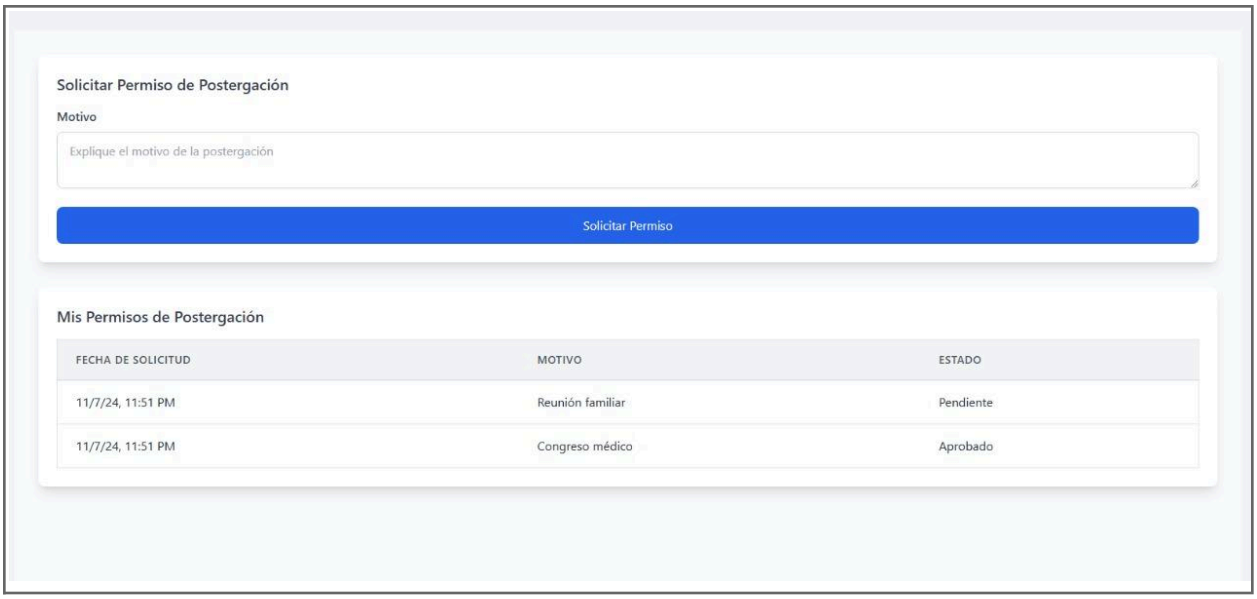
CU13: Solicitar Permiso

Solicitar Permiso	
ID: 13	**Descripción: Solicitar Permiso en el sistema**
Prioridad: Alta	**Estimación PHU: 6hr**
Criterios de aceptación: ● El usuario debe poder seleccionar el tipo de permiso (licencia médica, ausencia justificada, etc.) y proporcionar los detalles necesarios. ● El estado de la solicitud debe estar disponible para seguimiento, con posibles estados como "En revisión", "Aprobado" o "Rechazado".	
Desarrollador a cargo: Manzoni Bravo Fabiana	



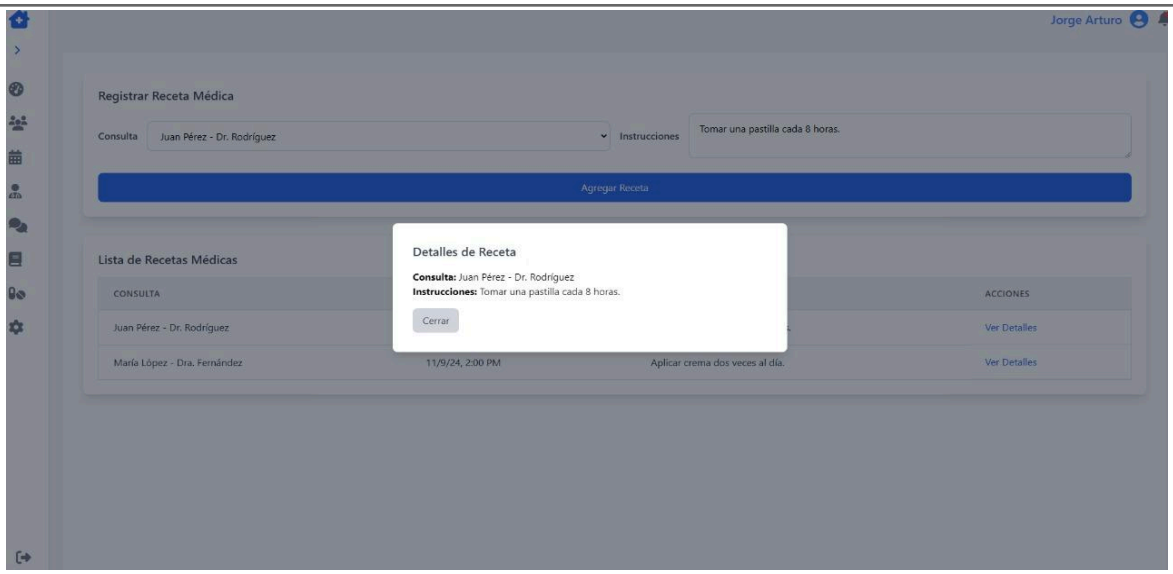
CU14: Gestionar Postergación

Gestionar Postergación	
ID: 14	**Descripción: Gestionar Postergación en el sistema**
Prioridad: Alta	**Estimación PHU: 7hr**
Criterios de aceptación: ● Si la cita o reserva es postergada, debe moverse al dia siguiente de su realización en el mismo horario.	
Desarrollador a cargo: Aliaga Valencia Jorge Arturo	



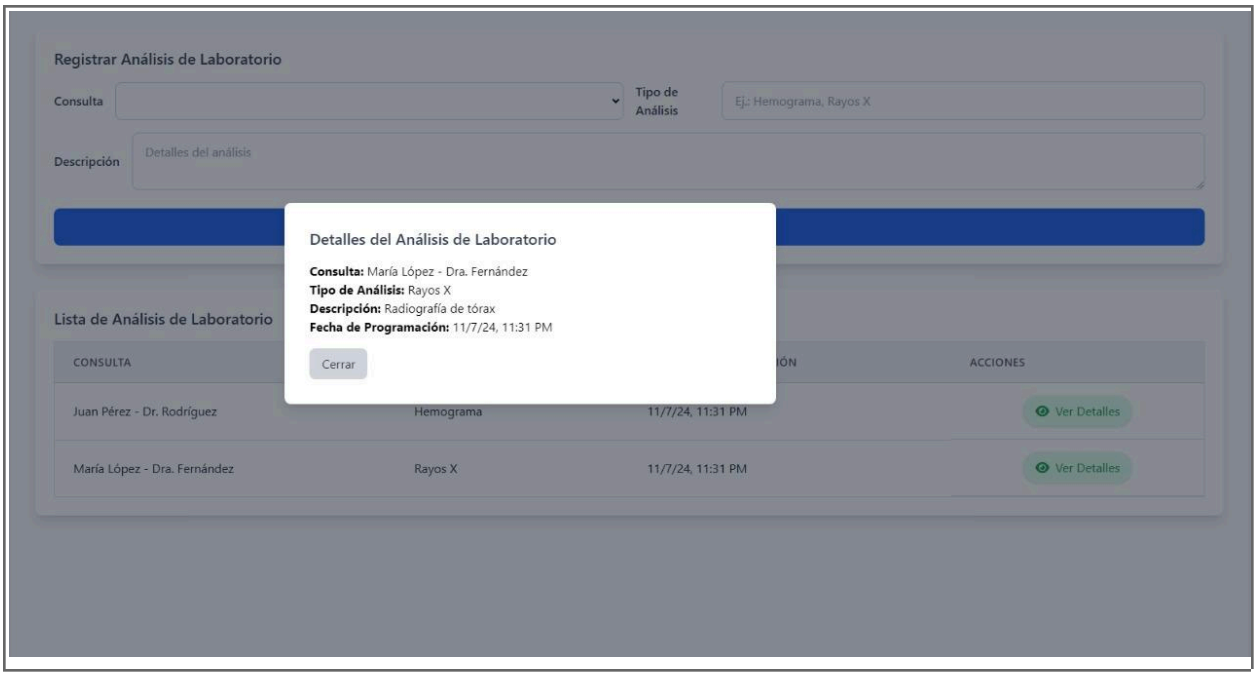
CU15: Gestionar Receta Médica

Gestionar Receta Médica	
ID: 15	**Descripción: Programar Receta Médica en el sistema**
Prioridad: Alta	**Estimación PHU: 4hr**
Criterios de aceptación: ● El médico debe poder crear una receta médica para el paciente durante la consulta. ● La receta debe incluir los medicamentos, dosis, frecuencia y duración del tratamiento.	
Desarrollador a cargo:	



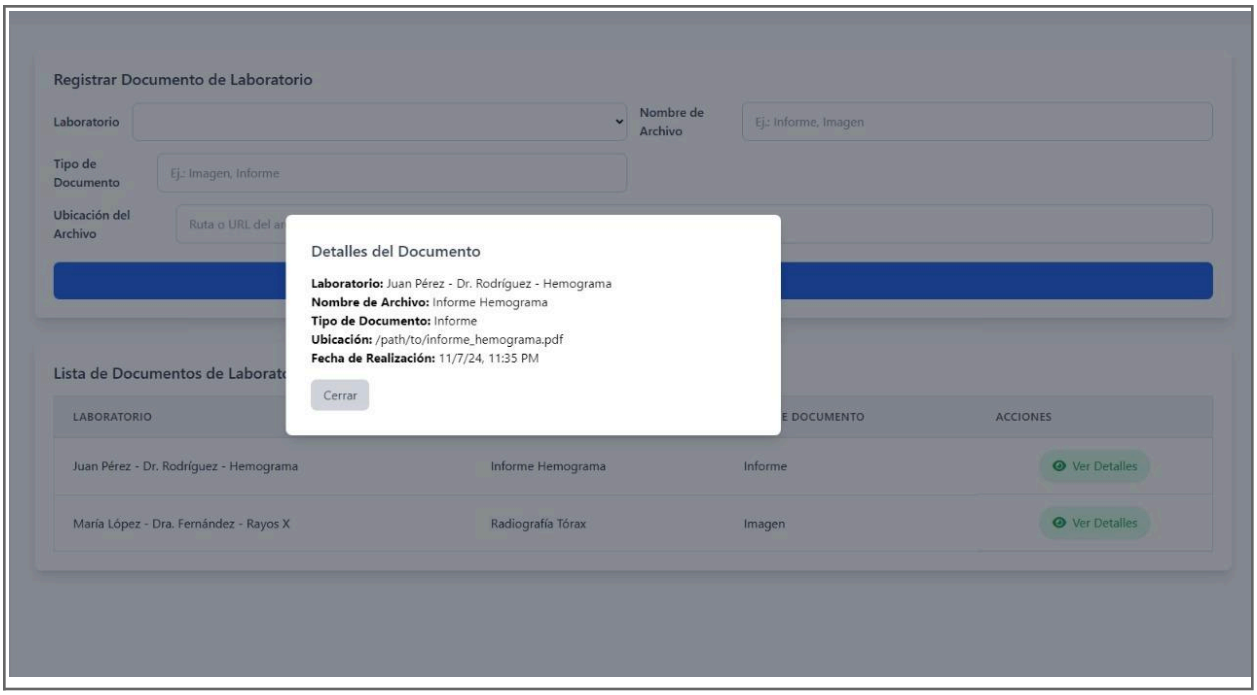
CU16: Programar Laboratorio

Programar Laboratorio	
ID: 16	**Descripción: Programar Laboratorio en el sistema**
Prioridad: Alta	**Estimación PHU: 3hr**
Criterios de aceptación: ● El médico debe poder solicitar la programación de un examen de laboratorio específico.	
Desarrollador a cargo: Aliaga Valencia Jorge Arturo	



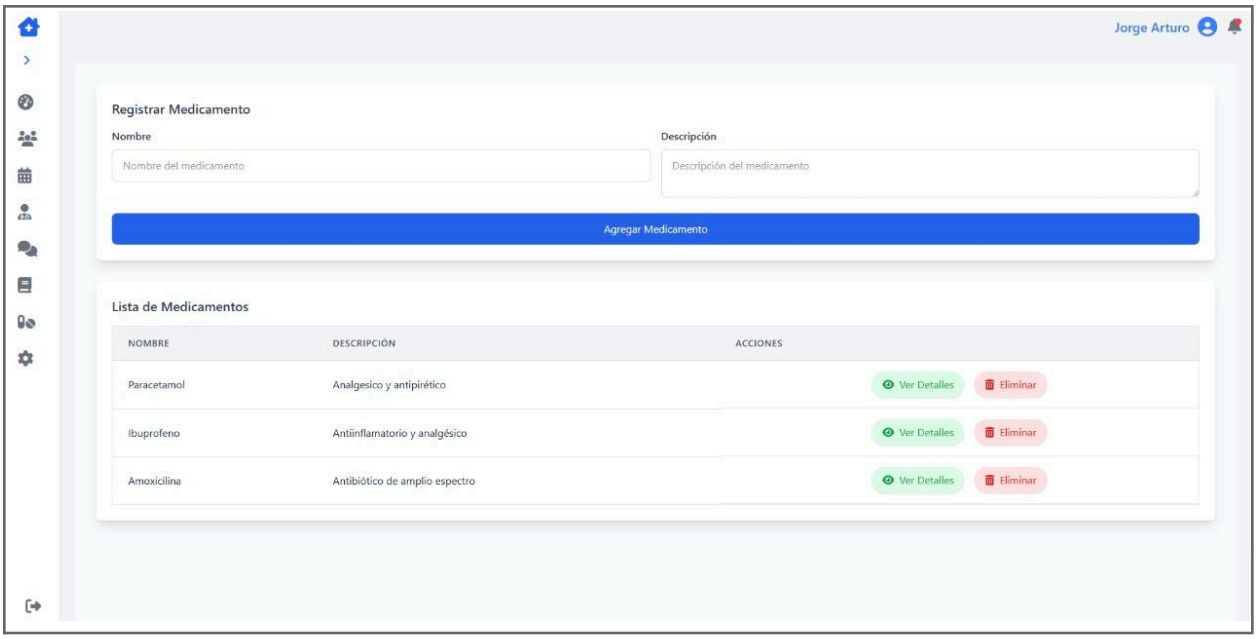
CU17: Gestionar Laboratorio

Gestionar Laboratorio	
ID: 17	**Descripción: Gestionar Laboratorio en el sistema**
Prioridad: Alta	**Estimación PHU: 7hrt**
Criterios de aceptación: ● El sistema debe permitir cargar y almacenar los resultados de los exámenes de laboratorio en el historial clínico del paciente.	
Desarrollador a cargo: Aliaga Valencia Jorge Arturo	



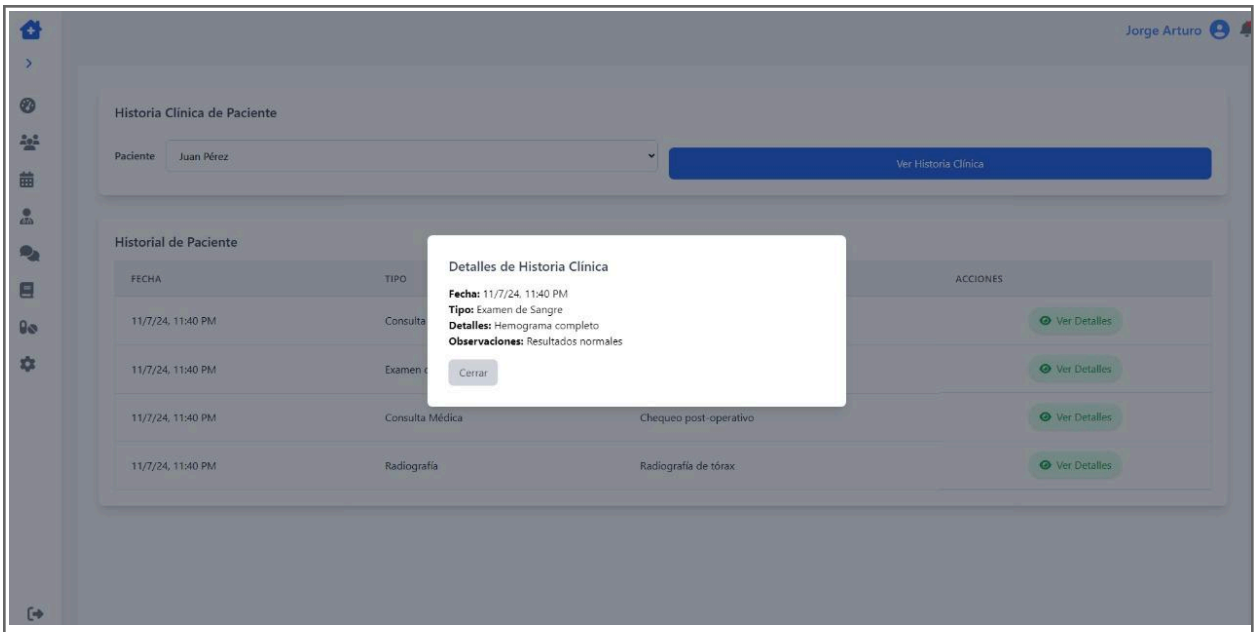
CU18: Gestionar Medicamentos

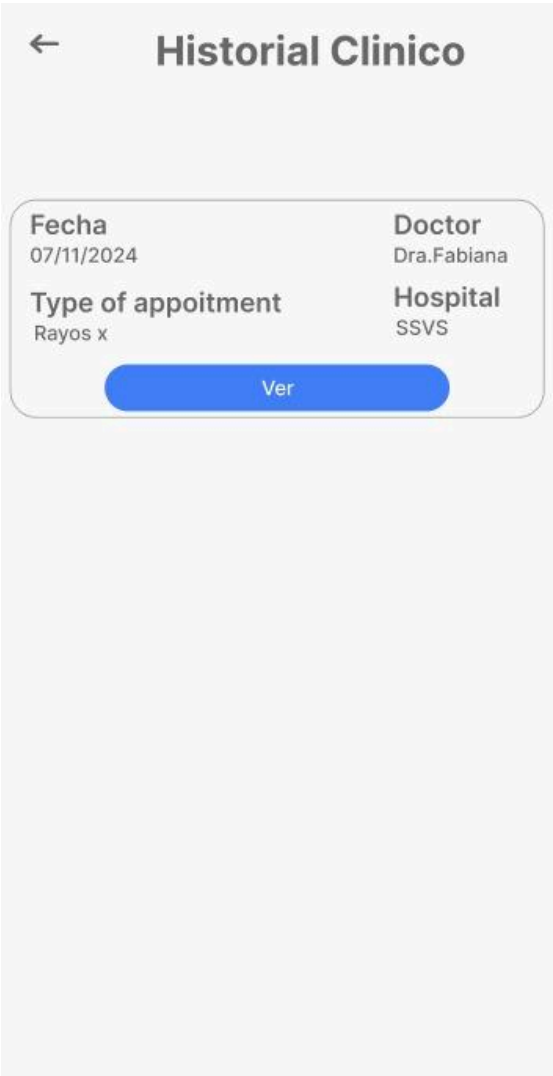
Gestionar Medicamentos	
ID: 18	**Descripción: Gestionar Medicamentos en el sistema**
Prioridad: Alta	**Estimación PHU: 4hr**
Criterios de aceptación: ● El sistema debe permitir al administrador agregar, editar y eliminar medicamentos en el listado de disponibilidad. ● El médico debe poder seleccionar medicamentos desde el listado al momento de generar una receta.	
Desarrollador a cargo: Aliaga Valencia Jorge Arturo	



CU19: Visualizar Historia Clínica

Visualizar Historia Clínica	
ID: 19	**Descripción: Visualizar Historia Clínica en el sistema**
Prioridad: Alta	**Estimación PHU: 4hr**
Criterios de aceptación: ● Visualización por Médicos: Permitir que los médicos vean el historial clínico completo del paciente en solo lectura. ● Visualización por Pacientes: Permitir que los pacientes vean su propio historial clínico completo.	
Desarrollador a cargo: Aliaga Valencia Jorge Arturo	


Jorge Arturo
Historia Clinica de Paciente
Paciente
Juan Pérez
Ver Historia Clínica
Historial de Paciente
FECHA
TIPO
ACCIONES
11/7/24, 11:40 PM
Consulta
Ver Detalles
11/7/24, 11:40 PM
Examen
Ver Detalles
11/7/24, 11:40 PM
Consulta Médica
Chequeo post-operativo
Ver Detalles
11/7/24, 11:40 PM
Radiografía
Radiografía de tórax
Ver Detalles
Detalles de Historia Clínica
Fecha: 11/7/24, 11:40 PM
Tipo: Examen de Sangre
Detalles: Hemograma completo
Observaciones: Resultados normales
Cerrar


Historial Clinico
Fecha
07/11/2024
Doctor
Dra.Fabiana
Type of appoitment
Rayos x
Hospital
SSVS
Ver

CU25: Enviar Recordatorio de Cita al Paciente

Enviar Recordatorio de Cita al Paciente	
ID: 24	**Descripción: Enviar Recordatorio de Cita al Paciente en el sistema**
Prioridad: Alta	**Estimación PHU: 3hr**
Criterios de aceptación: ● El sistema debe enviar un recordatorio de cita al paciente de forma automática antes de la fecha de la cita programada. ● El recordatorio debe incluir la fecha, hora, lugar de la cita y cualquier instrucción importante.	
Desarrollador a cargo: Manzoni Bravo Fabiana	

1.3. Contexto del sistema

El enfoque de este sprint se basara en el desarrollo de la gestión de especialidades, médicos, horarios de atención y fichas de atención, así como registrar y actualizar consultas médicas e historias clínicas de pacientes, garantizando un control integral y seguro de la atención en el sistema HCE del Seguro de Salud Vida Sana.

1.4. Sprint Backlog

Sprint Backlog					
Numero de Sprint: 2			**Tiempo programado: 2 semanas**		
Objetivo: El objetivo del segundo sprint es implementar la gestión de especialidades, médicos, horarios de atención, y fichas de atención, además de habilitar las funcionalidades para registrar y actualizar consultas y la historia clínica de los pacientes, asegurando una administración completa y trazable de las consultas médicas en el sistema HCE del Seguro de Salud Vida Sana.					
Product Owner: No establecido					
Fecha de inicio: 28/10/2024			**Fecha de finalización: 07/11/2024**		
Id	**Tarea**	**Tipo**	**Estimación (hr)**	**Responsable**	**Estado**
1	Diseñar interfaz para Gestión de Especialidades Médicas	Desarrollo	3 hr	Manzoni Bravo Fabiana	Terminado
2	Implementar Gestion de Especialidades Médicas	Desarrollo	7 hr	Manzoni Bravo Fabiana	Terminado
3	Realizar pruebas de Gestion de Especialidades Médicas	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
4	Diseñar interfaz para Gestion de Médicos	Desarrollo	4 hr	Manzoni Bravo Fabiana	Terminado
5	Implementar Gestion de Médicos	Desarrollo	9 hr	Manzoni Bravo Fabiana	Terminado
6	Realizar pruebas de Gestion de Médicos	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
7	Diseñar interfaz para Gestionar Horarios de Atención de Médicos	Desarrollo	3 hr	Manzoni Bravo Fabiana	Terminado

8	Implementar Gestionar Horarios de Atención de Médicos	Desarrollo	8 hr	Manzoni Bravo Fabiana	Terminado
9	Realizar pruebas de Gestionar Horarios de Atención de Médicos	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
10	Diseñar interfaz para Gestionar Reserva	Desarrollo	4 hr	Manzoni Bravo Fabiana	Terminado
11	Implementar Gestionar Reserva	Desarrollo	9 hr	Manzoni Bravo Fabiana	Terminado
12	Realizar pruebas de Gestionar Reserva	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
13	Diseñar interfaz para Gestionar Consultas Médicas	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
14	Implementar Gestionar Consultas Médicas	Desarrollo	6 hr	Manzoni Bravo Fabiana	Terminado
15	Realizar pruebas de Gestionar Consultas Médicas	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
16	Diseñar interfaz para Solicitar Permiso	Desarrollo	3 hr	Manzoni Bravo Fabiana	Terminado
17	Implementar Solicitar Permiso	Desarrollo	6 hr	Manzoni Bravo Fabiana	Terminado
18	Realizar pruebas de Solicitar Permiso	Desarrollo	2 hr	Manzoni Bravo Fabiana	Terminado
19	Diseñar interfaz para Gestionar Postergación	Desarrollo	3 hr	Aliaga Valencia Jorge Arturo	Terminado
20	Implementar Gestionar Postergación	Desarrollo	7 hr	Aliaga Valencia Jorge Arturo	Terminado

21	Realizar pruebas de Gestionar Postergación	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
22	Diseñar interfaz para Gestionar Receta Médica	Desarrollo	4 hr	Aliaga Valencia Jorge Arturo	Terminado
23	Implementar Gestionar Receta Médica	Desarrollo	9 hr	Aliaga Valencia Jorge Arturo	Terminado
24	Realizar pruebas de Gestionar Receta Médica	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
25	Diseñar interfaz para Programar Laboratorio	Desarrollo	3 hr	Aliaga Valencia Jorge Arturo	Terminado
26	Implementar Programar Laboratorio	Desarrollo	8 hr	Aliaga Valencia Jorge Arturo	Terminado
27	Realizar pruebas de Programar Laboratorio	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
28	Diseñar interfaz para Gestionar Laboratorio	Desarrollo	4 hr	Aliaga Valencia Jorge Arturo	Terminado
29	Implementar Gestionar Laboratorio	Desarrollo	9 hr	Aliaga Valencia Jorge Arturo	Terminado
30	Realizar pruebas de Gestionar Laboratorio	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
31	Diseñar interfaz para Gestionar Medicamentos	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
32	Implementar Gestionar Medicamentos	Desarrollo	6 hr	Aliaga Valencia Jorge Arturo	Terminado
33	Realizar pruebas de Gestionar Medicamentos	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
34	Diseñar interfaz para Visualizar Historia Clínica	Desarrollo	3 hr	Aliaga Valencia Jorge Arturo	Terminado
35	Implementar Visualizar Historia Clínica	Desarrollo	5 hr	Aliaga Valencia Jorge Arturo	Terminado

36	Realizar pruebas de Visualizar Historia Clínica	Desarrollo	2 hr	Aliaga Valencia Jorge Arturo	Terminado
37	Diseñar interfaz para Enviar Recordatorio de Cita al Paciente	Desarrollo	5 hr	Manzoni Bravo Fabiana	Terminado
38	Implementar Enviar Recordatorio de Cita al Paciente	Desarrollo	3 hr	Manzoni Bravo Fabiana	Terminado
39	Realizar pruebas de Enviar Recordatorio de Cita al Paciente	Desarrollo	1 hr	Manzoni Bravo Fabiana	Terminado

1.5. Equipo SCRUM

Integrantes	*Rol*	*Entorno*	*Caso de Uso*	*Tareas*	*Nota*
Jorge Arturo Aliaga Valencia	Developer	**Web:** **Movil:**	**CU14:** Gestionar Postergación **CU15:** Gestionar Receta Médica **CU16:** Programar Laboratorio **CU17:** Gestionar Laboratorio **CU18:** Gestionar Medicamentos **CU19:** Visualizar Historia Clínica **CU11:** Gestionar Reserva **CU15:** Gestionar Receta Médica	**19.**Diseñar interfaz para Gestionar Postergación **20.**Implementar Gestionar Postergación **21.**Realizar pruebas de Gestionar Postergación **22.**Diseñar interfaz para Gestionar Receta Médica **23.**Implementar Gestionar Receta Médica **24.**Realizar pruebas de Gestionar Receta Médica **25.**Diseñar interfaz para Programar Laboratorio **26.**Implementar Programar Laboratorio **27.**Realizar pruebas de Programar Laboratorio **28.**Diseñar interfaz para Gestionar Laboratorio	100

				29. Implementar Gestionar Laboratorio **30.**Realizar pruebas de Gestionar Laboratorio **31.**Diseñar interfaz para Gestionar Medicamentos **32.**Implementar Gestionar Medicamentos **33.**Realizar pruebas de Gestionar Medicamentos **34.**Diseñar interfaz para Visualizar Historia Clínica **35.**Implementar Visualizar Historia Clínica **36.**Realizar pruebas de Visualizar Historia Clínica	
Fabiana Manzoni Bravo	Developer	**Web:**

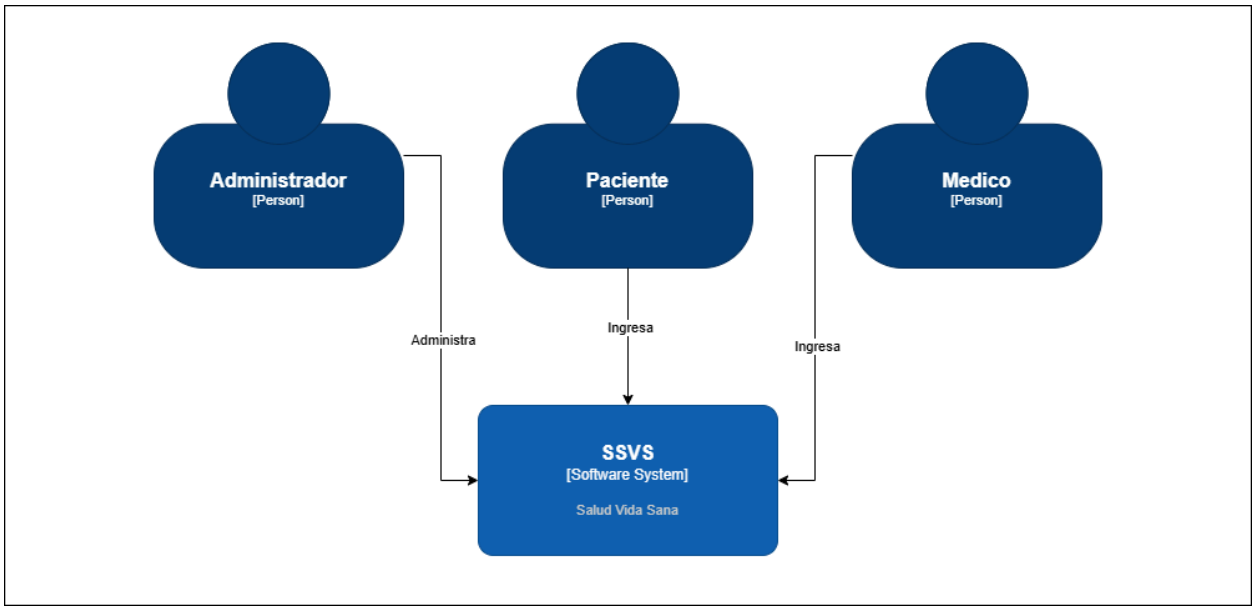
				10. Diseñar interfaz para Gestionar Reserva **11.**Implementar Gestionar Reserva **12.** Realizar pruebas de Gestionar Reserva **13.** Diseñar interfaz para Gestionar Consultas Médicas **14.**Implementar Gestionar Consultas Médicas **15.**Realizar pruebas de Gestionar Consultas Médicas **16.**Diseñar interfaz para Solicitar Permiso **17.**Implementar Solicitar Permiso **18.**Realizar pruebas de Solicitar Permiso **37.** Diseñar interfaz para Enviar Recordatorio de Cita al Paciente **38.** Implementar Enviar Recordatorio de Cita al Paciente **39.** Realizar pruebas de Enviar Recordatorio de Cita al Paciente	

2. Proceso/patrón de desarrollo por Historia de Usuario

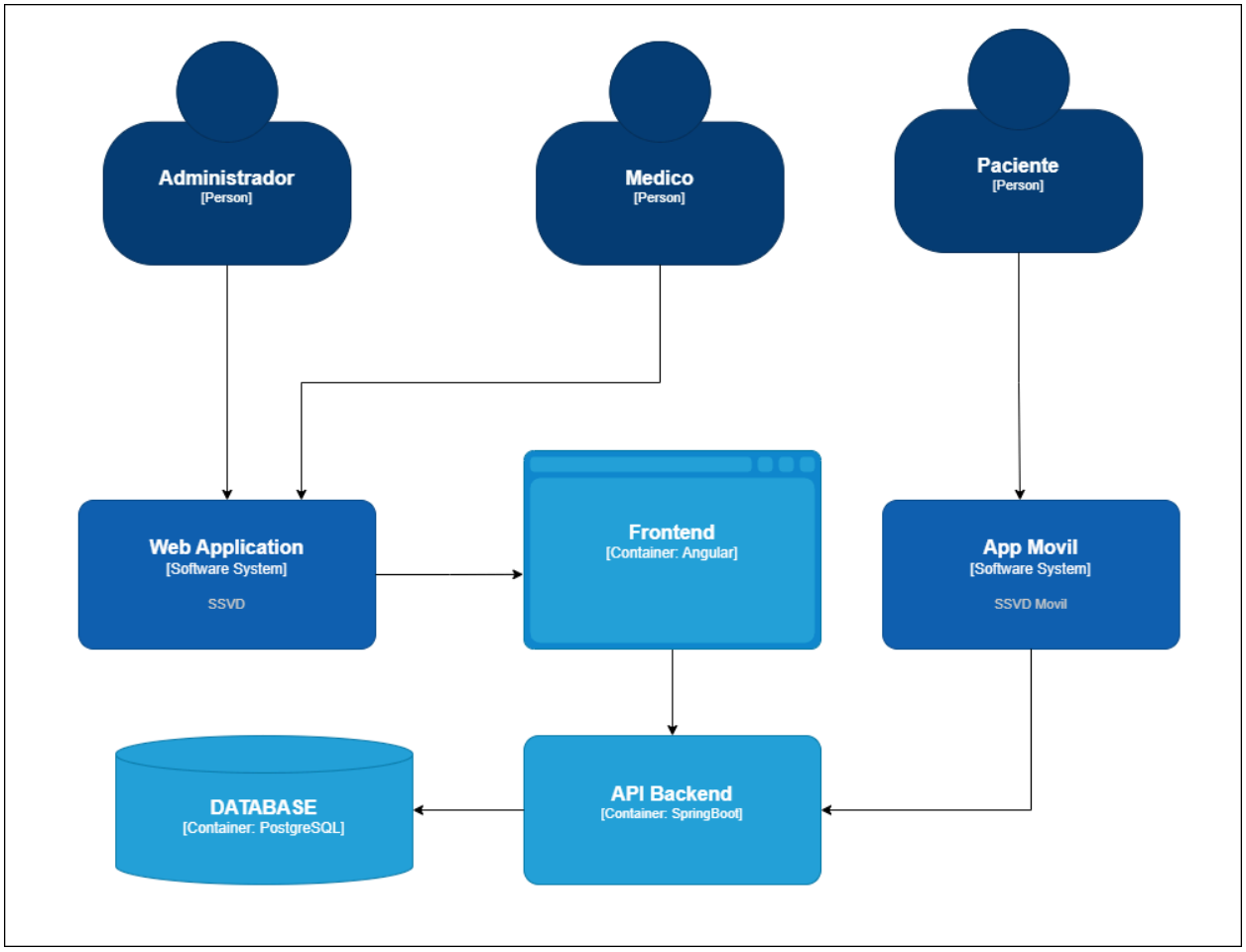
2.1. Diseño

2.1.1. Diseño de la arquitectura

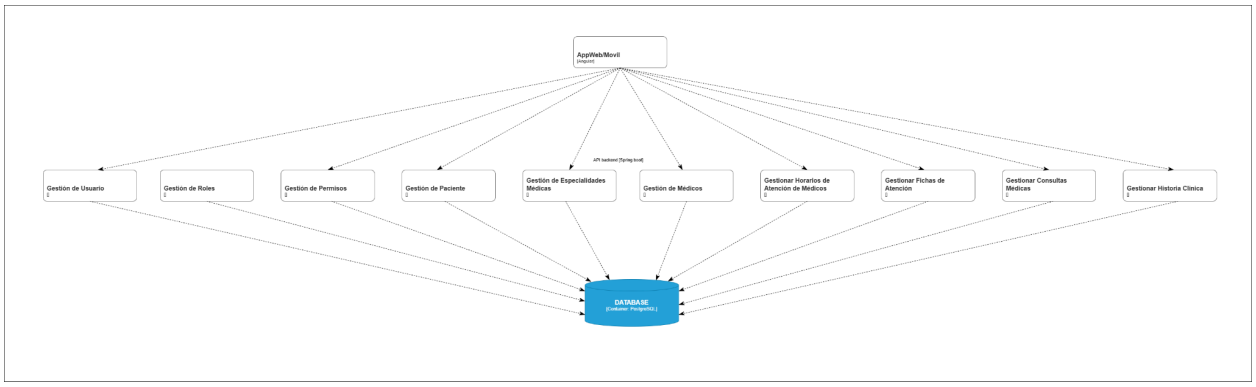
a) Diagrama del Contexto del Sistema



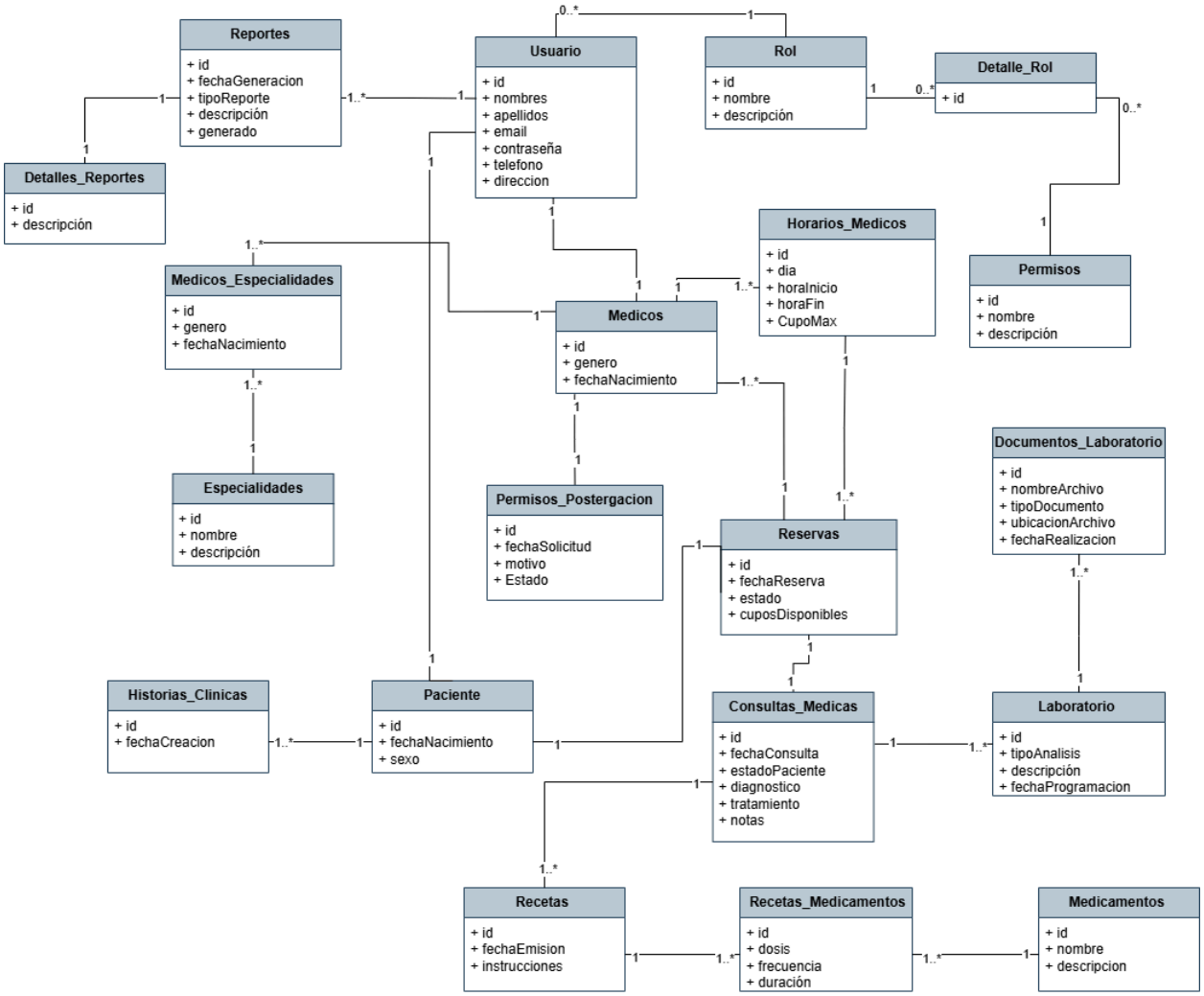
b) Diagrama de Contenedores



c) Diagrama de Componentes

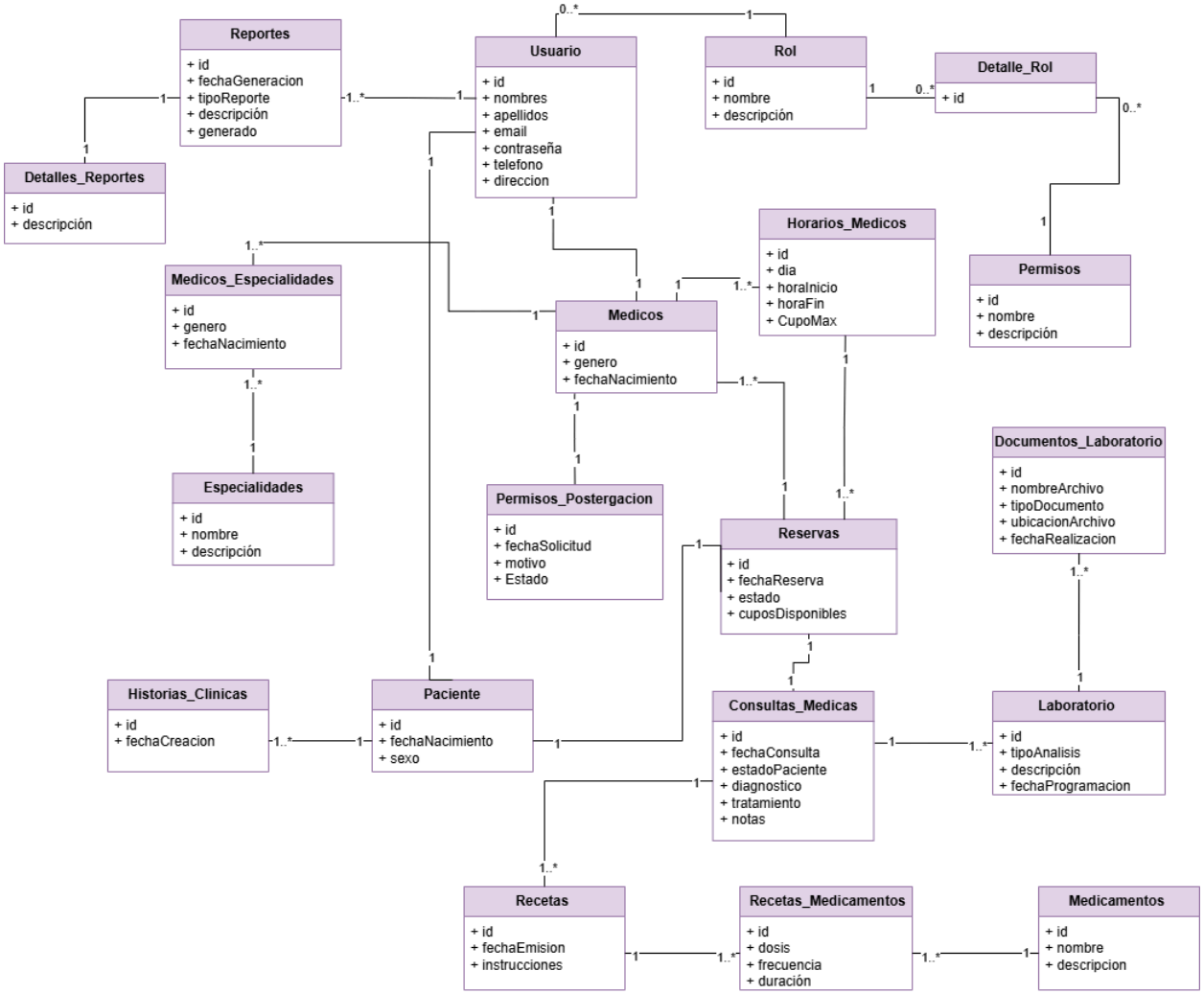


d) Diagrama de Código



2.1.2. Diseño de datos

a) Diagrama de clases



b) Script

```
-- Database: railway

-- DROP DATABASE IF EXISTS railway;

CREATE DATABASE railway
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'en_US.utf8'
    LC_CTYPE = 'en_US.utf8'
    LOCALE_PROVIDER = 'libc'
```

```
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;


CREATE TABLE Roles (
    rol_id SERIAL PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE NOT NULL,
    descripcion TEXT
);

CREATE TABLE Permisos (
    permiso_id SERIAL PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE NOT NULL,
    descripcion TEXT
);

CREATE TABLE Roles_Permisos (
    rol_id INT REFERENCES Roles(rol_id) ON DELETE CASCADE,
    permiso_id INT REFERENCES Permisos(permiso_id) ON DELETE CASCADE,
    PRIMARY KEY (rol_id, permiso_id)
);

CREATE TABLE Usuarios (
    usuario_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    contraseña TEXT NOT NULL,
    telefono VARCHAR(20),
    direccion TEXT,
    rol_id INT REFERENCES Roles(rol_id) ON DELETE SET NULL
);

CREATE TABLE Pacientes (
    paciente_id SERIAL PRIMARY KEY,
    usuario_id INT REFERENCES Usuarios(usuario_id) ON DELETE CASCADE,
    fecha_nacimiento DATE,
    genero VARCHAR(10)
);
```

```
CREATE TABLE Especialidades (
   especialidad_id SERIAL PRIMARY KEY,
   nombre VARCHAR(100) UNIQUE NOT NULL,
   descripcion TEXT
);

CREATE TABLE Medicos (
   medico_id SERIAL PRIMARY KEY,
   usuario_id INT REFERENCES Usuarios(usuario_id) ON DELETE CASCADE,
   genero VARCHAR(50),
   fecha_nacimiento DATE
);

CREATE TABLE Medicos_Especialidades (
   medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
   especialidad_id INT REFERENCES Especialidades(especialidad_id) ON DELETE
CASCADE,
   PRIMARY KEY (medico_id, especialidad_id)
);

CREATE TABLE Horarios_Medicos (
   horario_id SERIAL PRIMARY KEY,
   medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
   dia_semana VARCHAR(20) NOT NULL,
   hora_inicio TIME NOT NULL,
   hora_fin TIME NOT NULL,
   cupo_maximo INT NOT NULL  -- Número máximo de reservas para el horario
);

CREATE TABLE Permisos_Postergacion (
   postergacion_id SERIAL PRIMARY KEY,
   medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
   fecha_solicitud TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   motivo TEXT,
   estado VARCHAR(20) DEFAULT 'Pendiente'
);

CREATE TABLE Reservas (
   reserva_id SERIAL PRIMARY KEY,
```

```
    paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
    medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
    fecha_reserva DATE NOT NULL,
    horario_id INT REFERENCES Horarios_Medicos(horario_id) ON DELETE SET
NULL,
    estado VARCHAR(20) DEFAULT 'Reservada',
    cupos_disponibles INT  -- Cupos restantes para este horario en el día específico
);

CREATE TABLE Historias_Clinicas (
    historia_id SERIAL PRIMARY KEY,
    paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Consultas_Medicas (
    consulta_id SERIAL PRIMARY KEY,
    reserva_id INT REFERENCES Reservas(reserva_id) ON DELETE CASCADE,
    medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
    paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
    historia_id INT REFERENCES Historias_Clinicas(historia_id) ON DELETE
CASCADE,
    fecha_consulta TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado_paciente TEXT,
    diagnostico TEXT,
    tratamiento TEXT,
    notas TEXT
);

CREATE TABLE Recetas (
    receta_id SERIAL PRIMARY KEY,
    consulta_id INT REFERENCES Consultas_Medicas(consulta_id) ON DELETE
CASCADE,
    fecha_emision TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    instrucciones TEXT
);

CREATE TABLE Medicamentos (
    medicamento_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
```

```
    descripcion TEXT
);

CREATE TABLE Recetas_Medicamentos (
    receta_id INT REFERENCES Recetas(receta_id) ON DELETE CASCADE,
    medicamento_id INT REFERENCES Medicamentos(medicamento_id) ON DELETE
CASCADE,
    dosis VARCHAR(50),
    frecuencia VARCHAR(50),
    duracion INT,
    PRIMARY KEY (receta_id, medicamento_id)
);

CREATE TABLE Laboratorio (
    laboratorio_id SERIAL PRIMARY KEY,
    consulta_id INT REFERENCES Consultas_Medicas(consulta_id) ON DELETE
CASCADE,
    tipo_analisis VARCHAR(50) NOT NULL, -- Ej.: Hemograma, Rayos X, etc.
    descripcion TEXT,
    fecha_programacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  -- Fecha en
que se programó el análisis
);

CREATE TABLE Documentos_Laboratorio (
    documento_id SERIAL PRIMARY KEY,
    laboratorio_id INT REFERENCES Laboratorio(laboratorio_id) ON DELETE
CASCADE,
    nombre_archivo VARCHAR(100) NOT NULL,
    tipo_documento VARCHAR(50) NOT NULL, -- Ej.: Imagen, Informe
    ubicacion_archivo TEXT NOT NULL,
    fecha_realizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- Fecha en que
se realizó el examen o documento
);

CREATE TABLE Reportes (
    reporte_id SERIAL PRIMARY KEY,
    fecha_generacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    tipo_reporte VARCHAR(50) NOT NULL,
    descripcion TEXT,
    generado_por INT REFERENCES Usuarios(usuario_id) ON DELETE SET NULL
```

```
);

CREATE TABLE Detalles_Reporte (
    detalle_id SERIAL PRIMARY KEY,
    reporte_id INT REFERENCES Reportes(reporte_id) ON DELETE CASCADE,
    descripcion TEXT
);
```

c) Mapeo

ROL

PK	
id	nombre

USUARIO

PK					FK
id	nombres	apellidos	email	contraseña	rol_id

DETALLE_ROL

PK	FK	FK
id	rol_id	permiso_id

PERMISO

PK	
id	nombre

MEDICO

PK			FK	FK
id	genero	fechaNacimiento	usuario_id	reserva_id

PERMISO_POSTERGACION

PK				FK
id	fechaSolicitud	motivo	estado	medico_id

ESPECIALIDAD

PK		
id	nombre	descripcion

MEDICO_ESPECIALIDAD

PK	FK	FK
id	medico_id	especialidad_id

PACIENTE

PK			FK
id	fechaNacimiento	sexo	usuario_id

HISTORIA_CLINICA

PK		FK
id	fechaCreacion	paciente_id

HORARIOS_MEDICOS

PK					FK
id	dia	horaInicio	horaFin	cupoMax	medico_id

RESERVA

PK				FK	FK
id	fechaReserva	estado	cuposDisponibles	horario_medico_id	paciente_id

CONSULTA_MEDICA

PK						FK
id	fechaConsulta	estadoPaciente	diagnostico	tratamiento	notas	reserva_id

LABORATORIO

PK				FK
id	tipoAnalisis	descripcion	fechaProgramacion	consulta_medica_id

DOCUMENTO_LABORATORIO

PK					FK
id	nombreArchivo	tipoDocumento	ubicacionArchivo	fechaRealizacion	laboratorio_id

RECETA

PK			FK
id	fechaEmision	instrucciones	consulta_medica_id

RECETA_MEDICAMENTO

PK				FK	FK
id	dosis	frecuencia	duracion	receta_id	medicamento_id

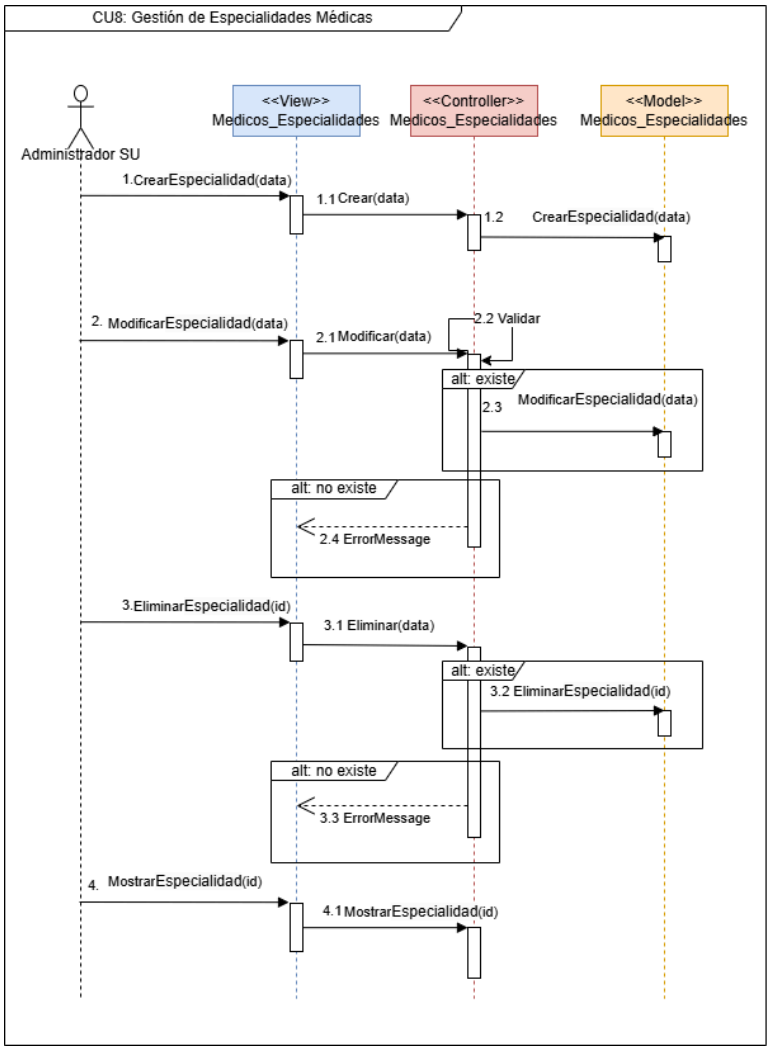
MEDICAMENTO

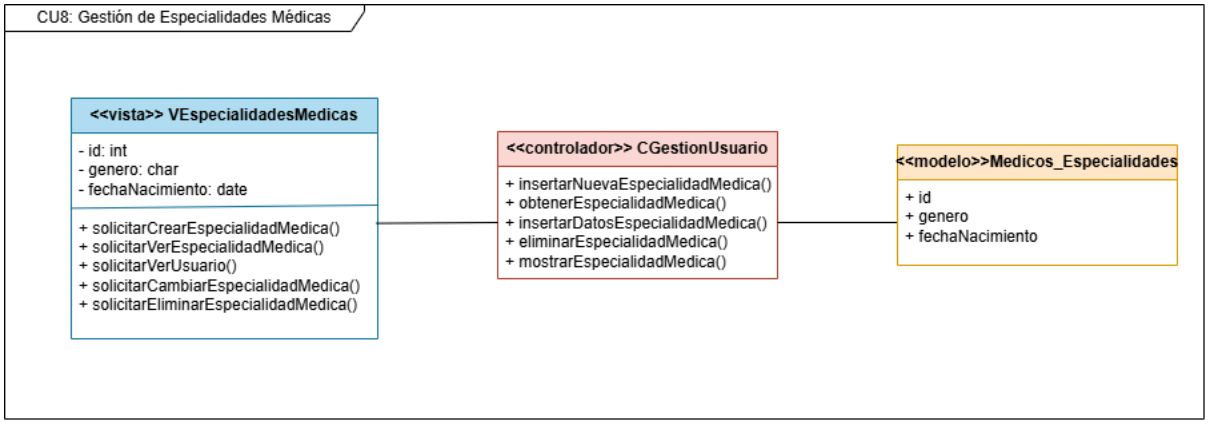
PK		
id	nombre	descripcion

2.1.3. Diseño de la lógica de negocio

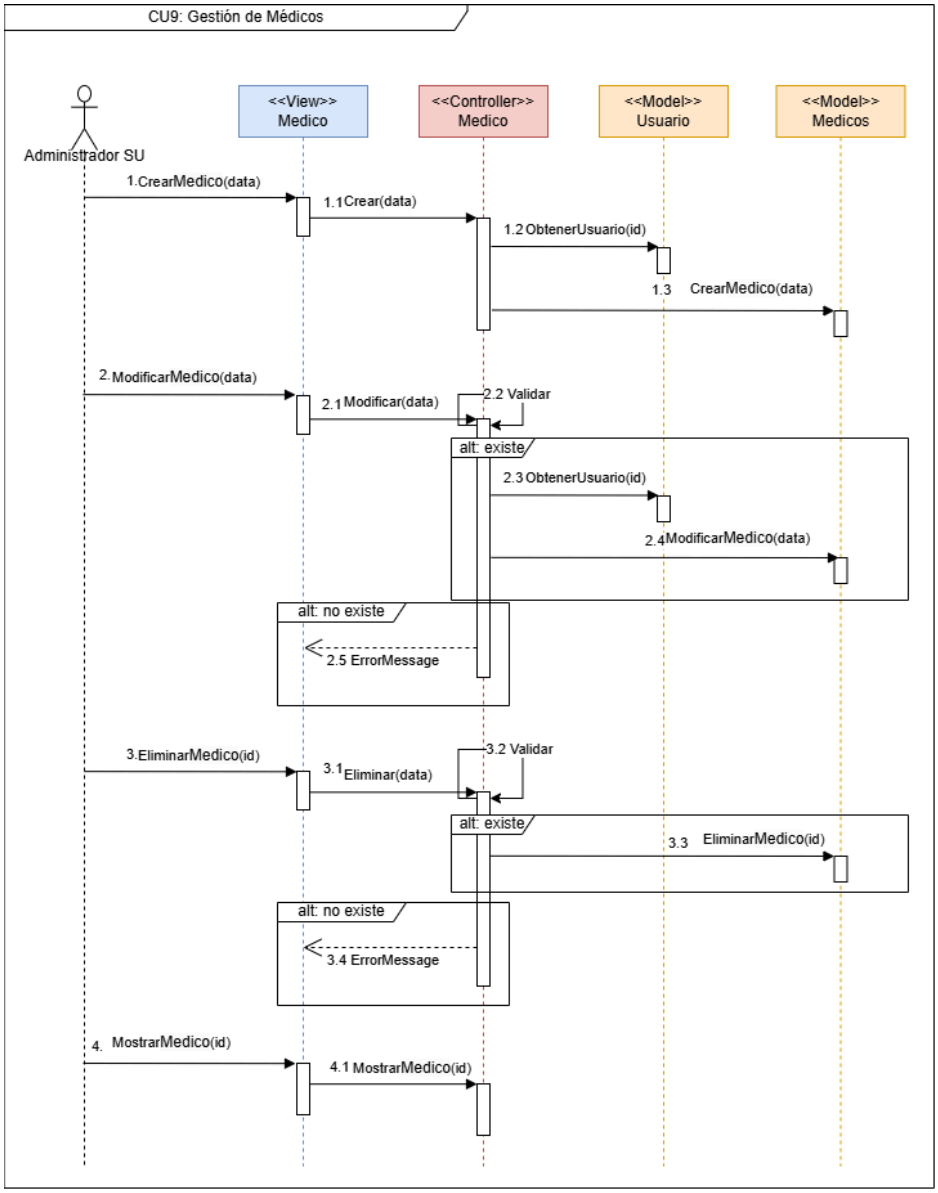
a) Diagrama de Secuencia y Diseño de Clase

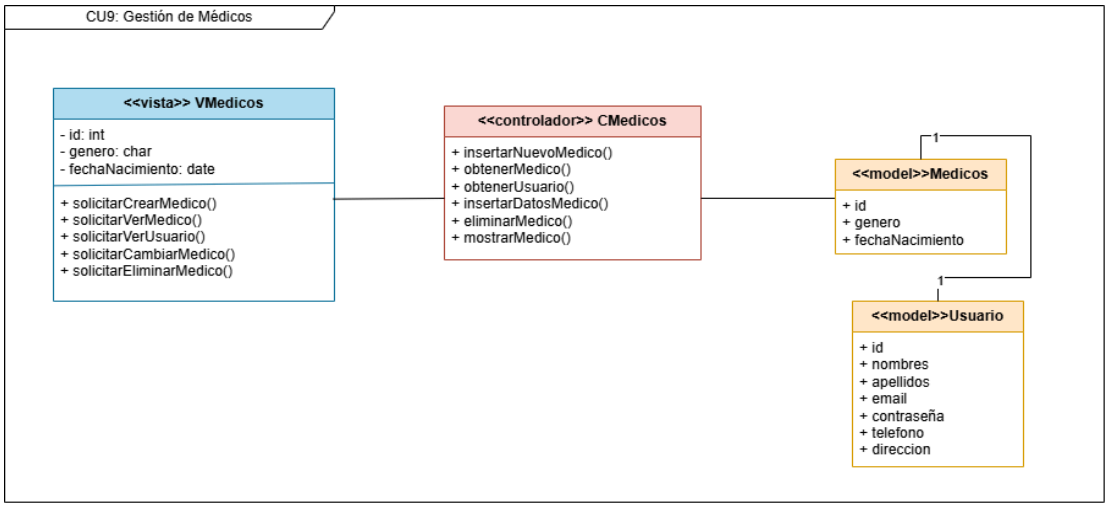
CU8: Gestión de Especialidades Médicas



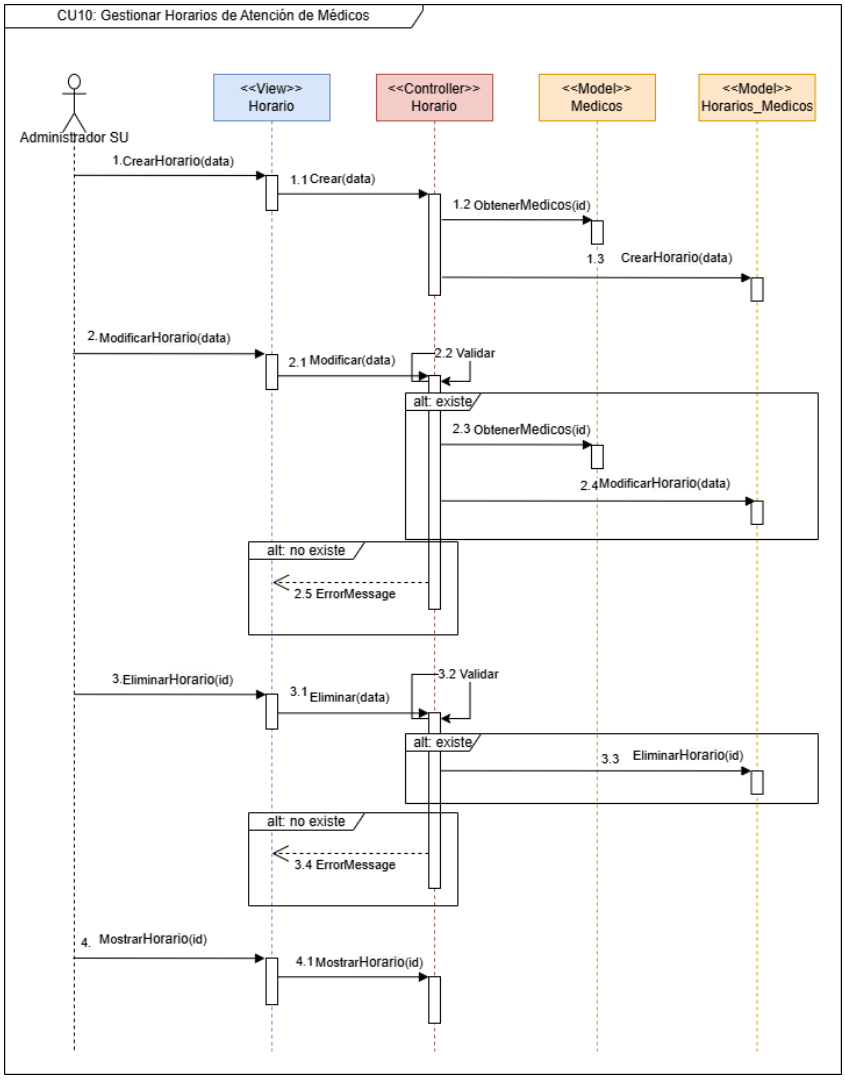


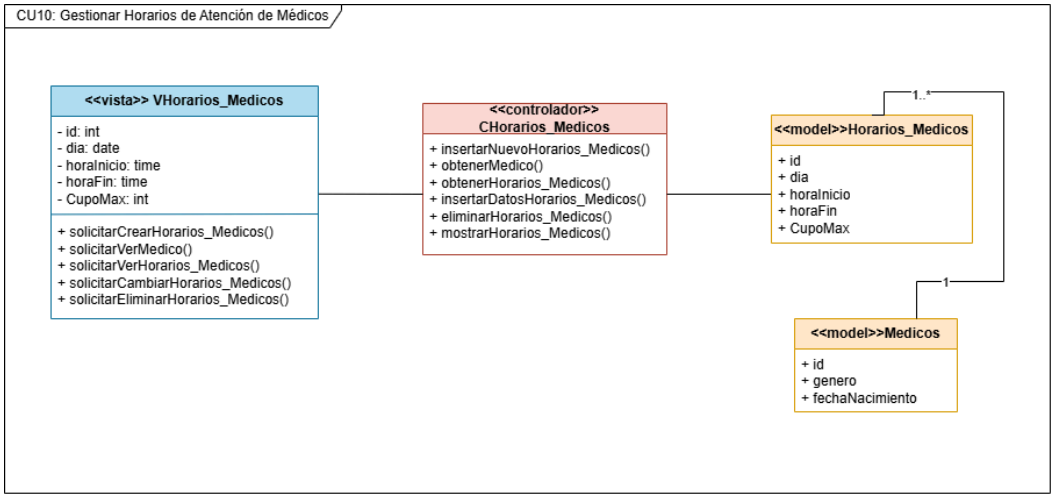
CU9: Gestión de Médicos


CU9: Gestión de Médicos
Administrador SU
<<View>> Medico
<<Controller>> Medico
<<Model>> Usuario
<<Model>> Medicos
1.CrearMedico(data)
1.1Crear(data)
1.2 ObtenerUsuario(id)
1.3 CrearMedico(data)
2. ModificarMedico(data)
2.1 Modificar(data)
2.2 Validar
alt: existe
2.3 ObtenerUsuario(id)
2.4ModificarMedico(data)
alt: no existe
2.5 ErrorMessage
3.EliminarMedico(id)
3.1 Eliminar(data)
3.2 Validar
alt: existe
3.3 EliminarMedico(id)
alt: no existe
3.4 ErrorMessage
4. MostrarMedico(id)
4.1 MostrarMedico(id)

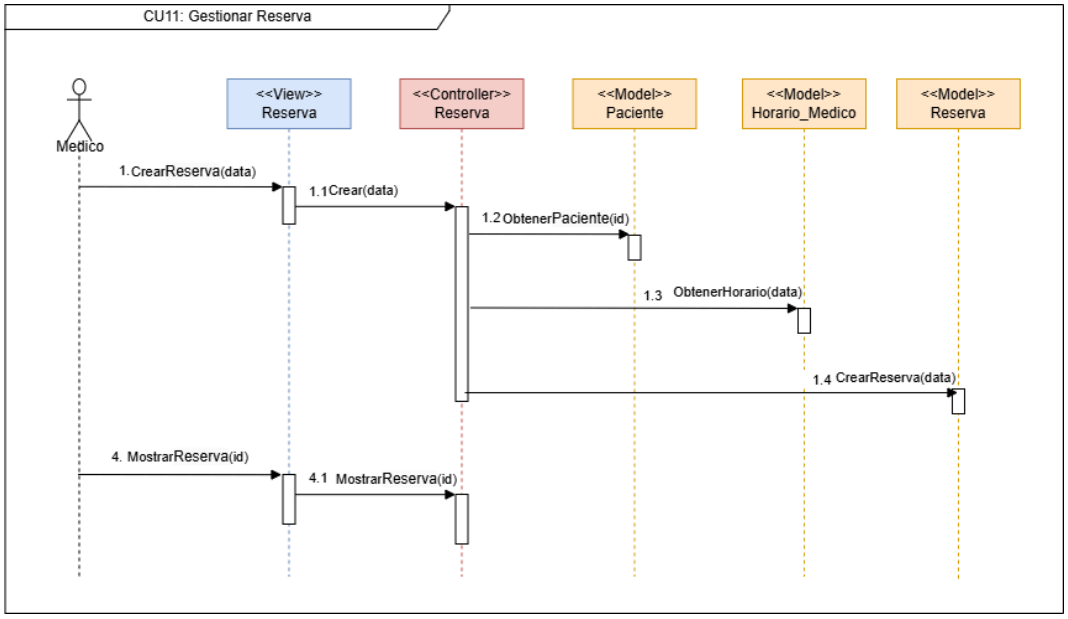

CU9: Gestión de Médicos
<<vista>> VMedicos
- id: int
- genero: char
- fechaNacimiento: date
+ solicitarCrearMedico()
+ solicitarVerMedico()
+ solicitarVerUsuario()
+ solicitarCambiarMedico()
+ solicitarEliminarMedico()
<<controlador>> CMedicos
+ insertarNuevoMedico()
+ obtenerMedico()
+ obtenerUsuario()
+ insertarDatosMedico()
+ eliminarMedico()
+ mostrarMedico()
1
<<model>>Medicos
+ id
+ genero
+ fechaNacimiento
1
<<model>>Usuario
+ id
+ nombres
+ apellidos
+ email
+ contraseña
+ telefono
+ direccion

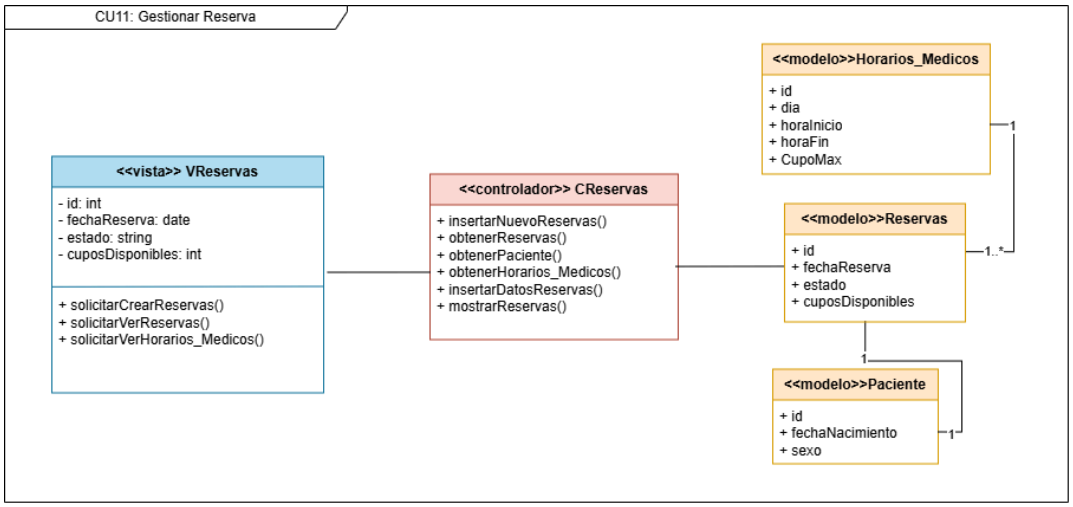
CU10: Gestionar Horarios de Atención de Médicos



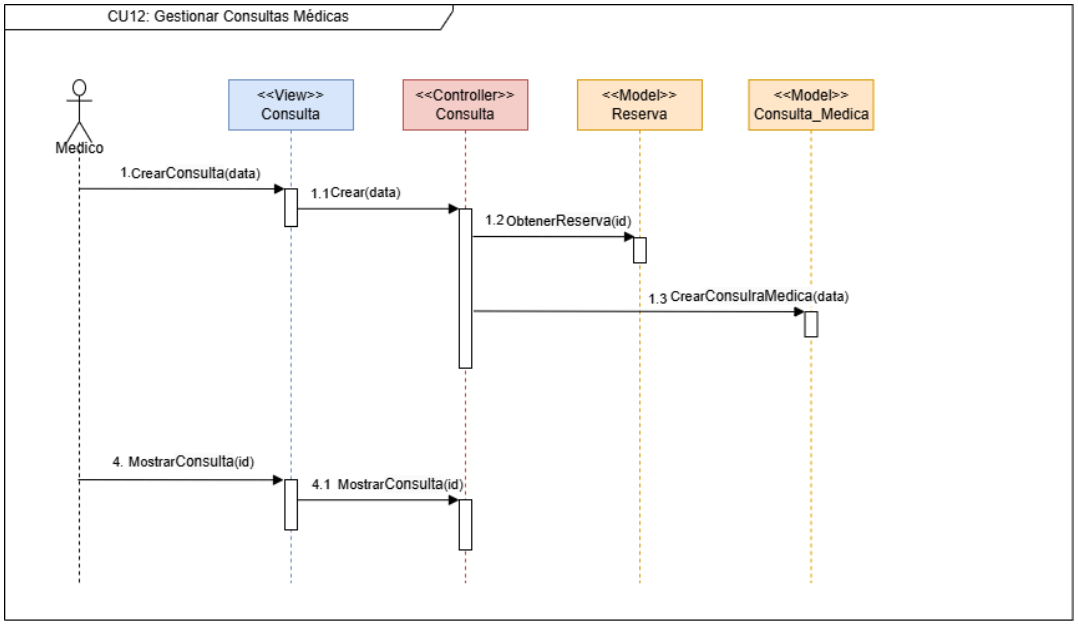


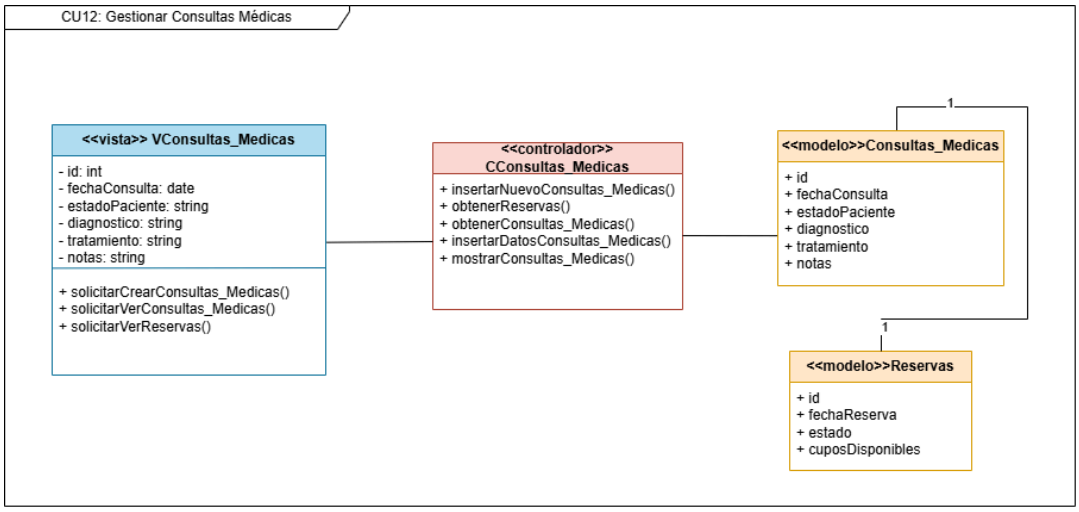
CU11: Gestionar Reserva



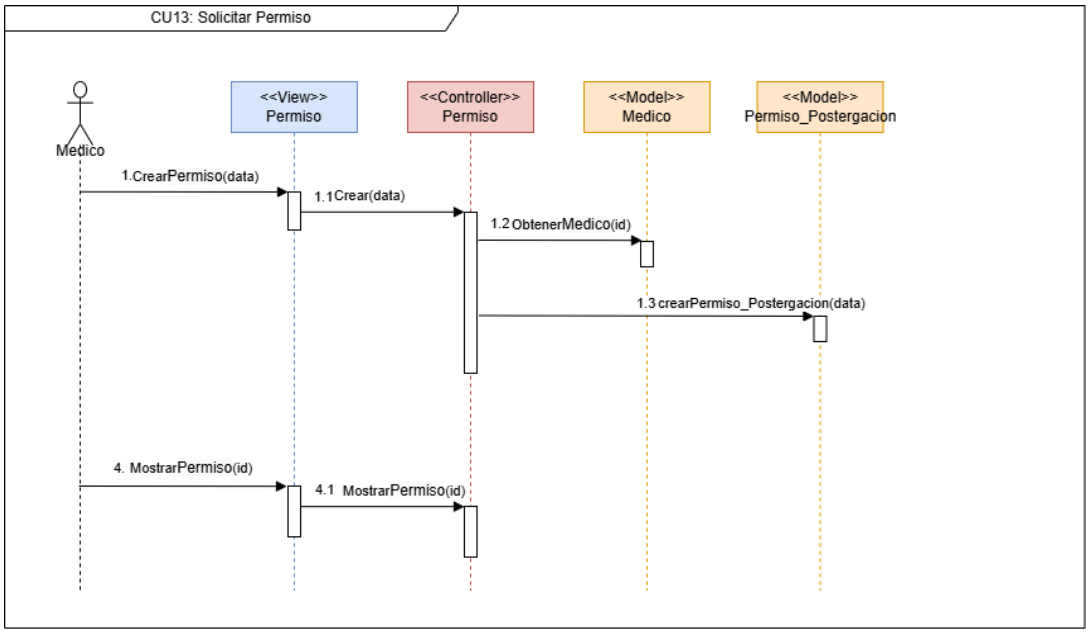


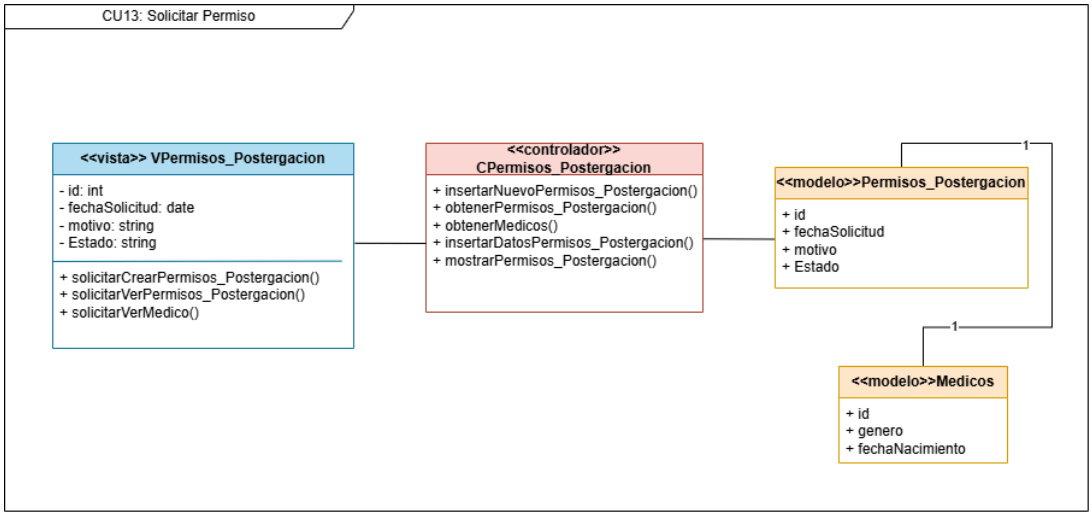
CU12: Gestionar Consultas Médicas



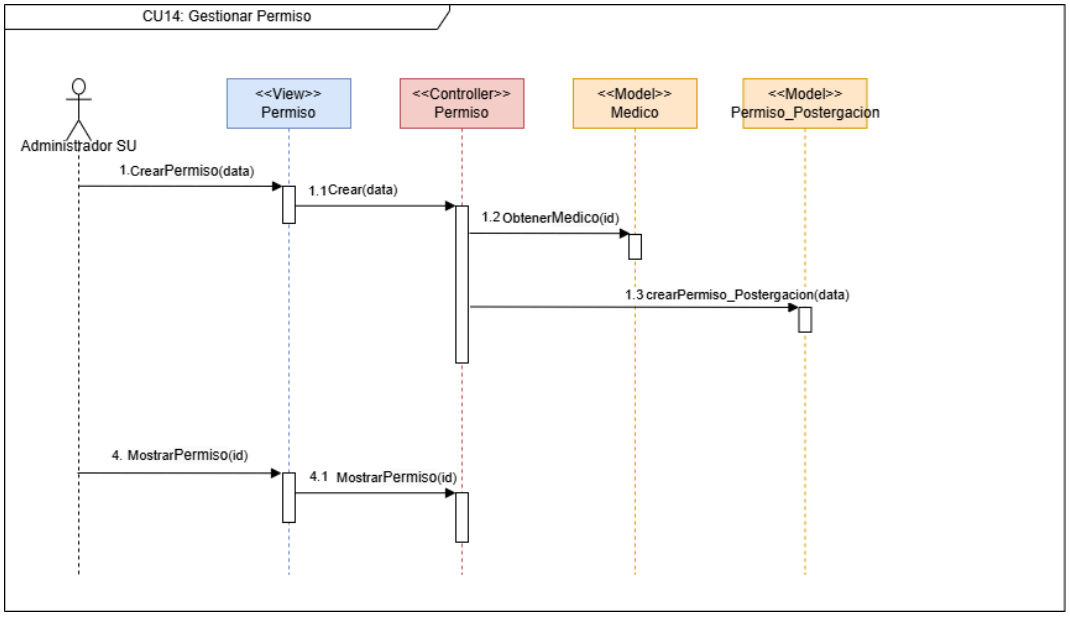


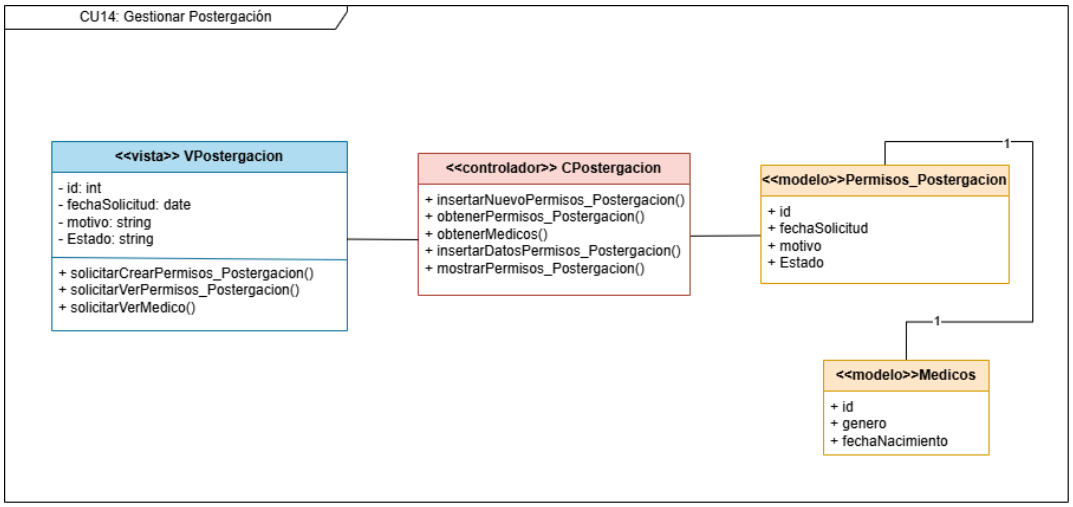
CU13: Solicitar Permiso



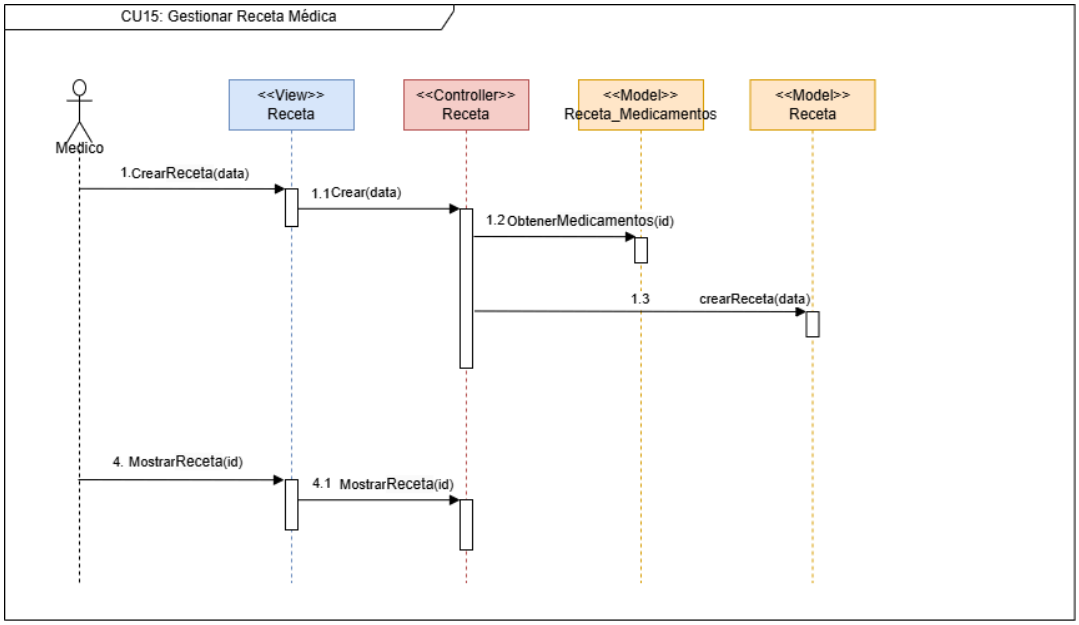


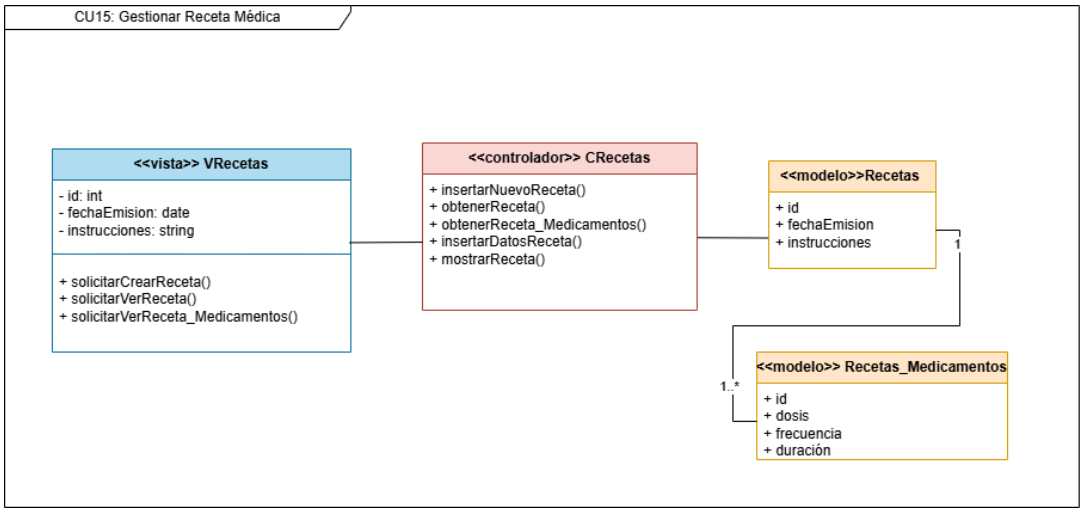
CU14: Gestionar Postergación



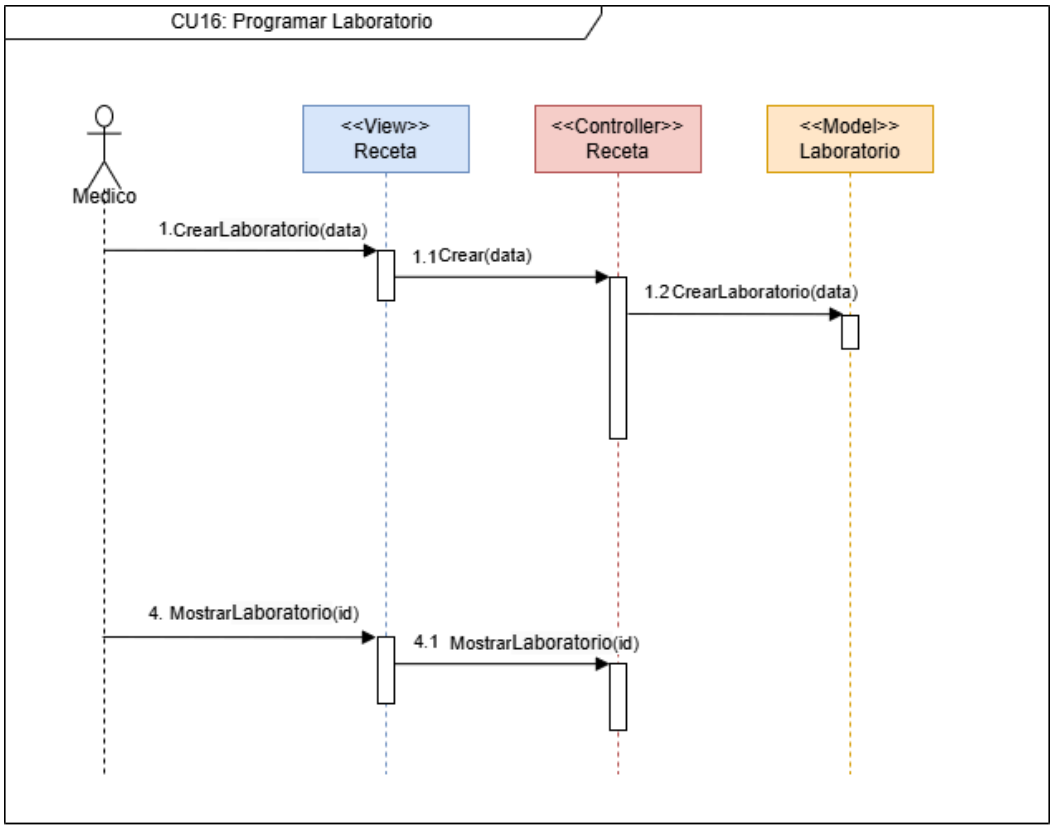


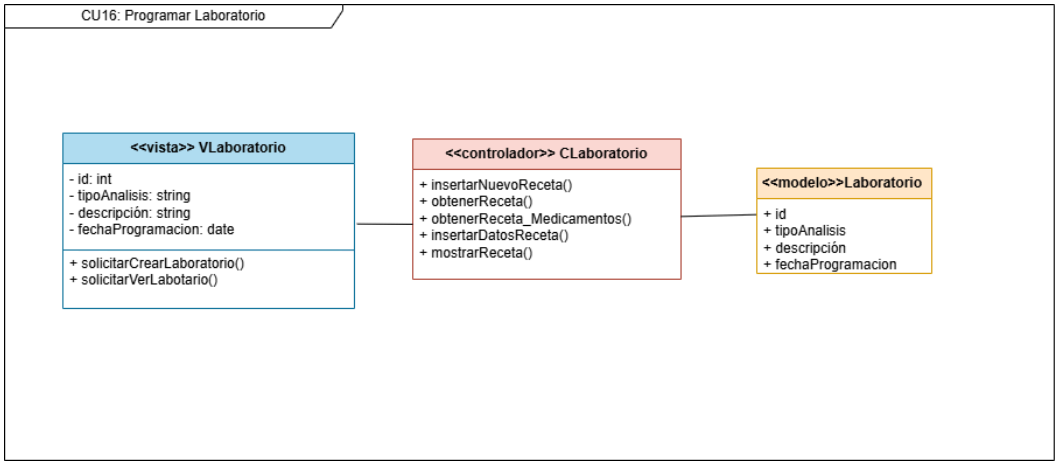
CU15: Gestionar Receta Médica



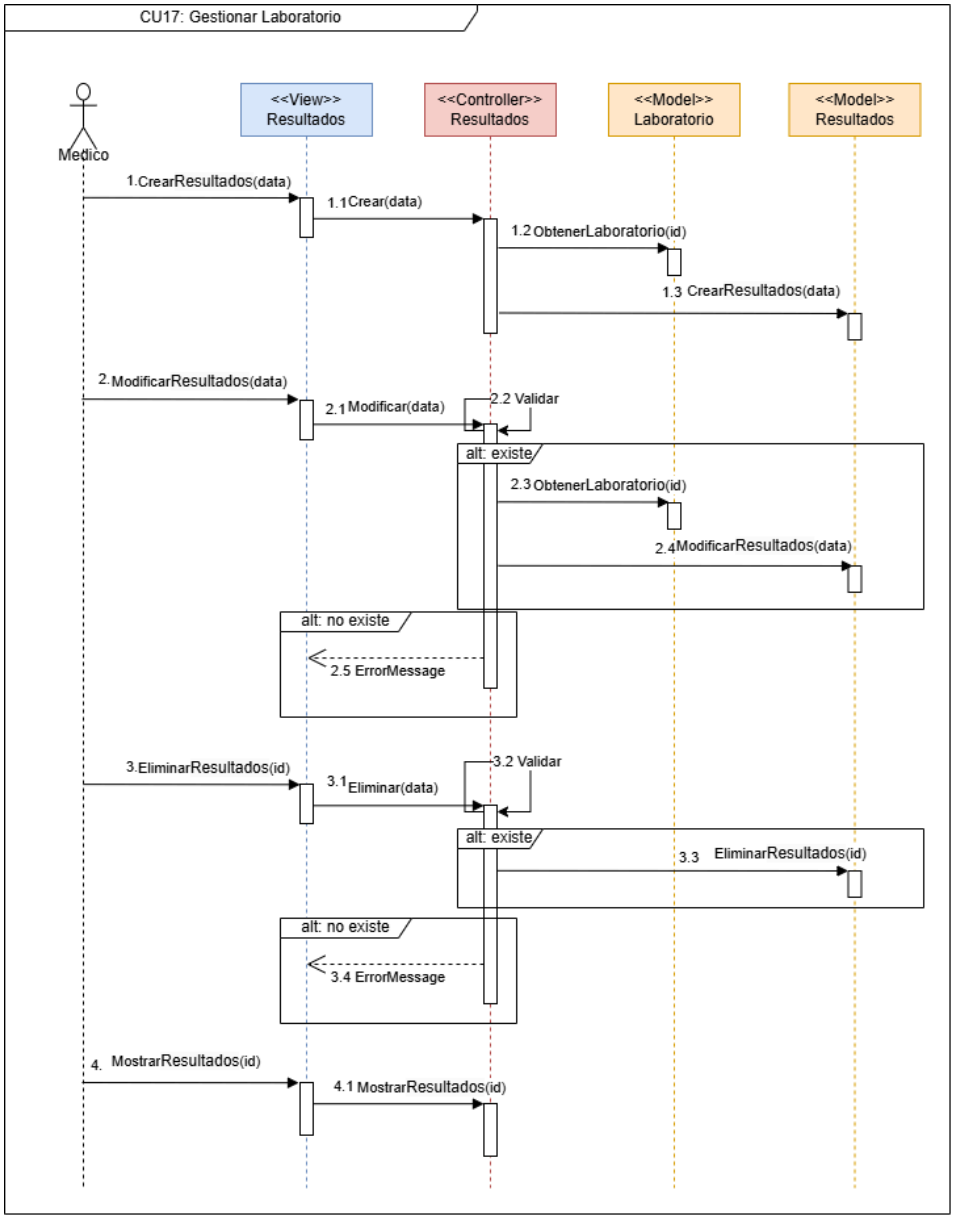


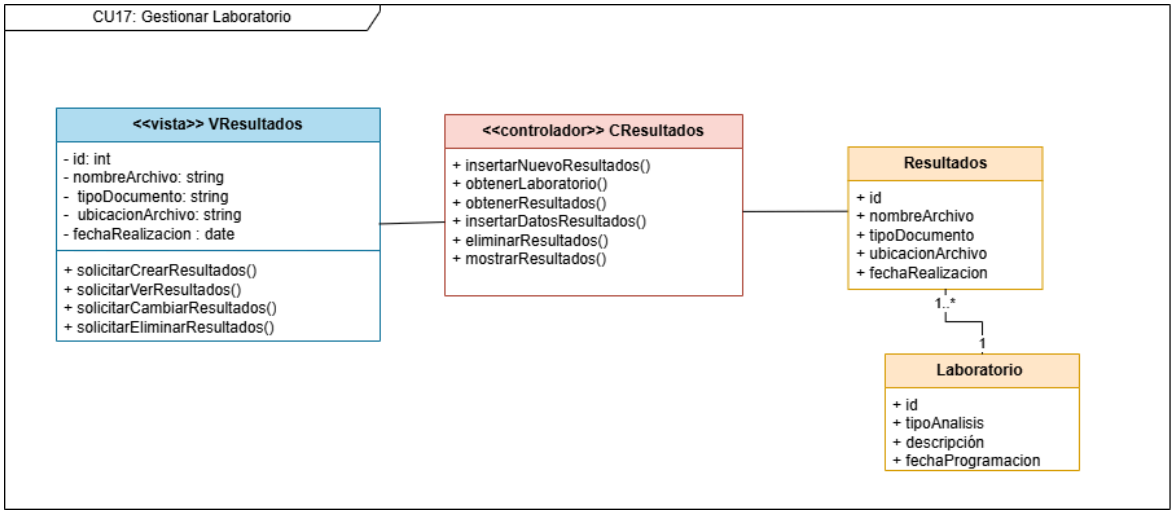
CU16: Programar Laboratorio



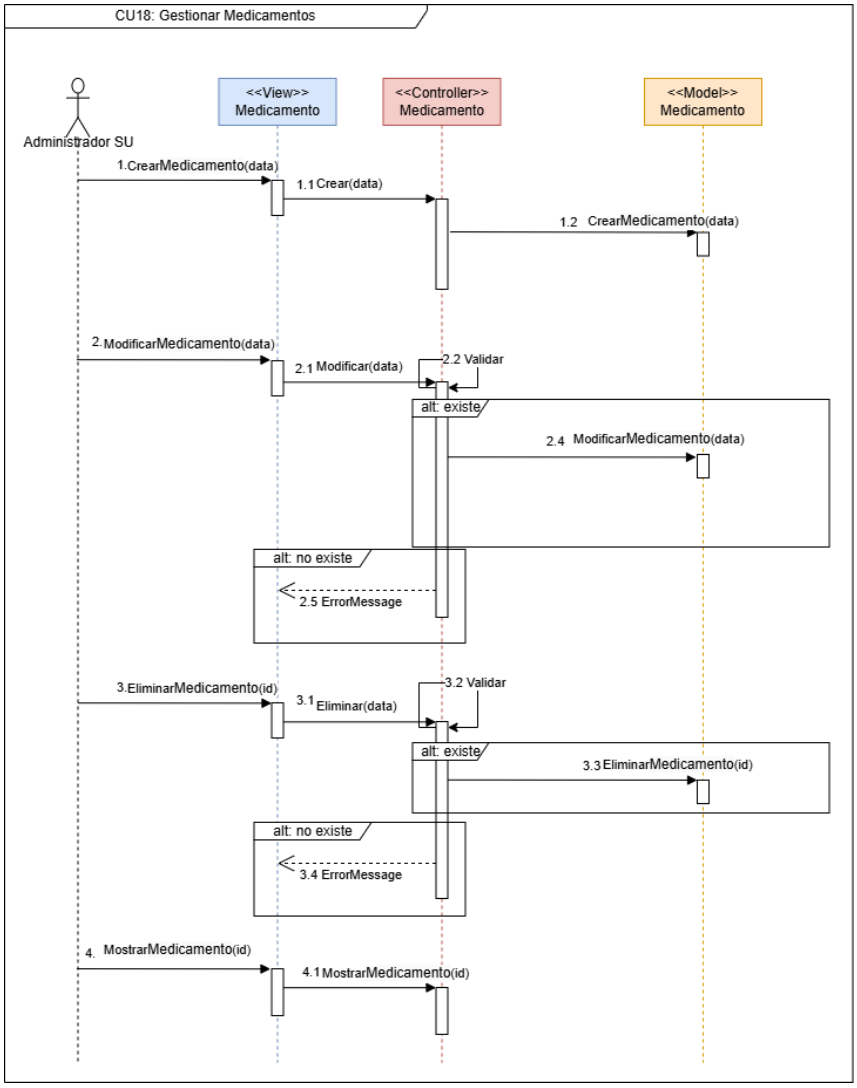


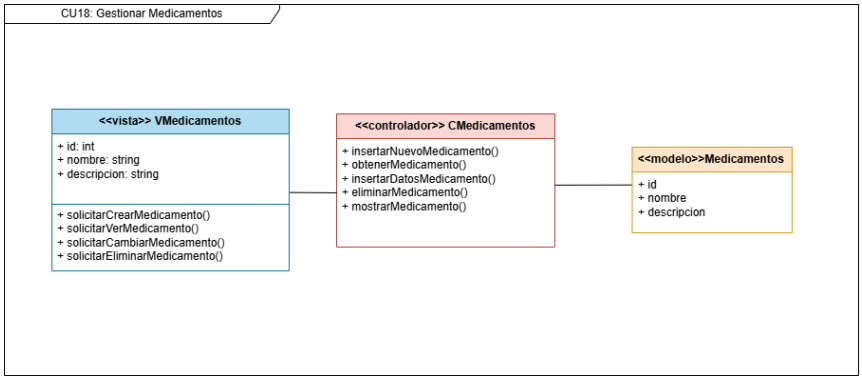
CU17: Gestionar Laboratorio


CU17: Gestionar Laboratorio
Medico
<<View>>
Resultados
<<Controller>>
Resultados
<<Model>>
Laboratorio
<<Model>>
Resultados
1.CrearResultados(data)
1.1Crear(data)
1.2 ObtenerLaboratorio(id)
1.3 CrearResultados(data)
2.ModificarResultados(data)
2.1 Modificar(data)
2.2 Validar
alt: existe
2.3 ObtenerLaboratorio(id)
2.4ModificarResultados(data)
alt: no existe
2.5 ErrorMessage
3.EliminarResultados(id)
3.1Eliminar(data)
3.2 Validar
alt: existe
3.3 EliminarResultados(id)
alt: no existe
3.4 ErrorMessage
4. MostrarResultados(id)
4.1 MostrarResultados(id)

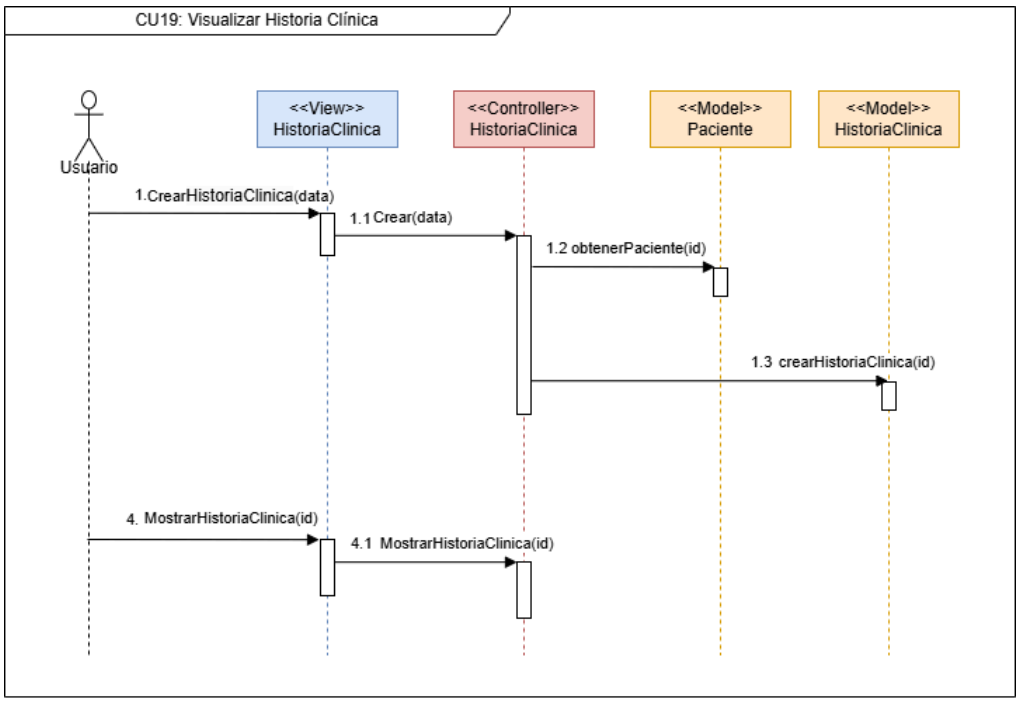


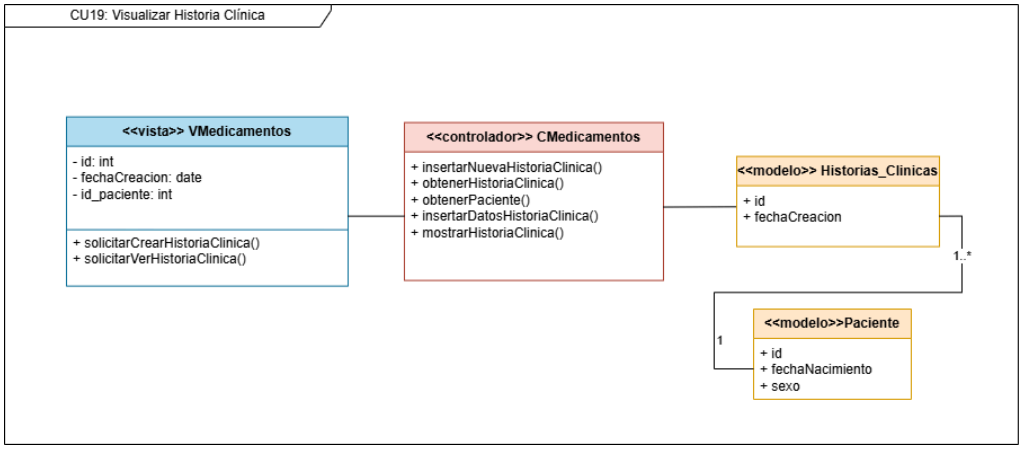
CU18: Gestionar Medicamentos



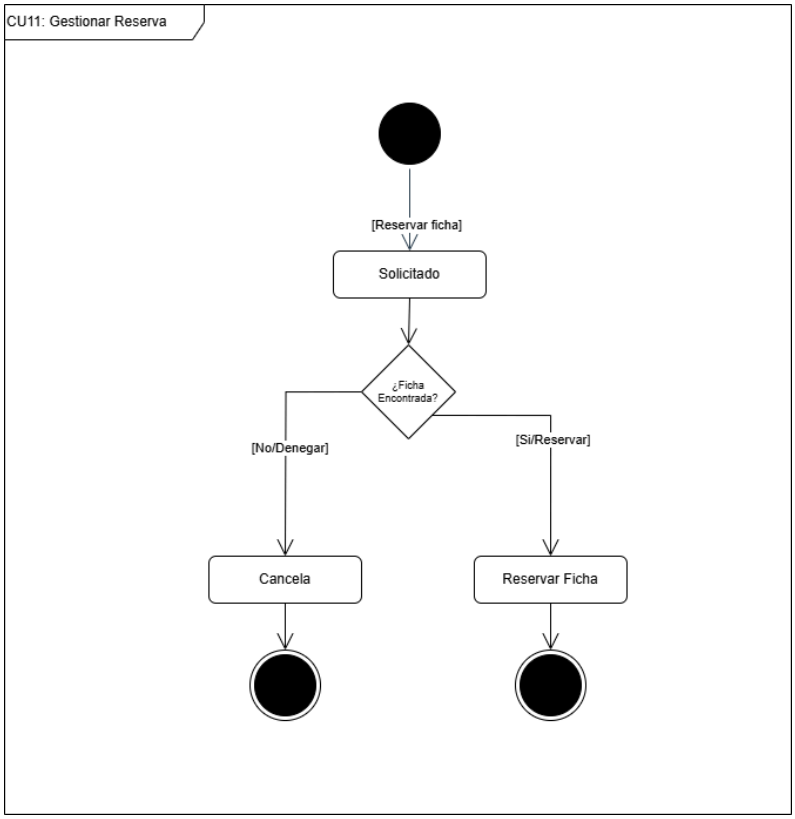


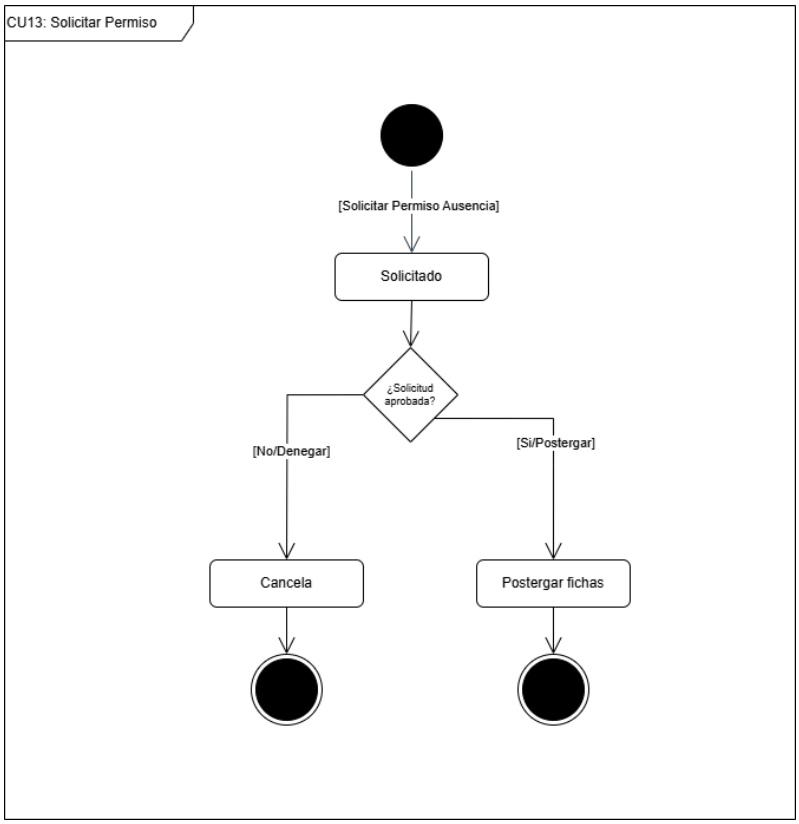
CU19: Visualizar Historia Clínica



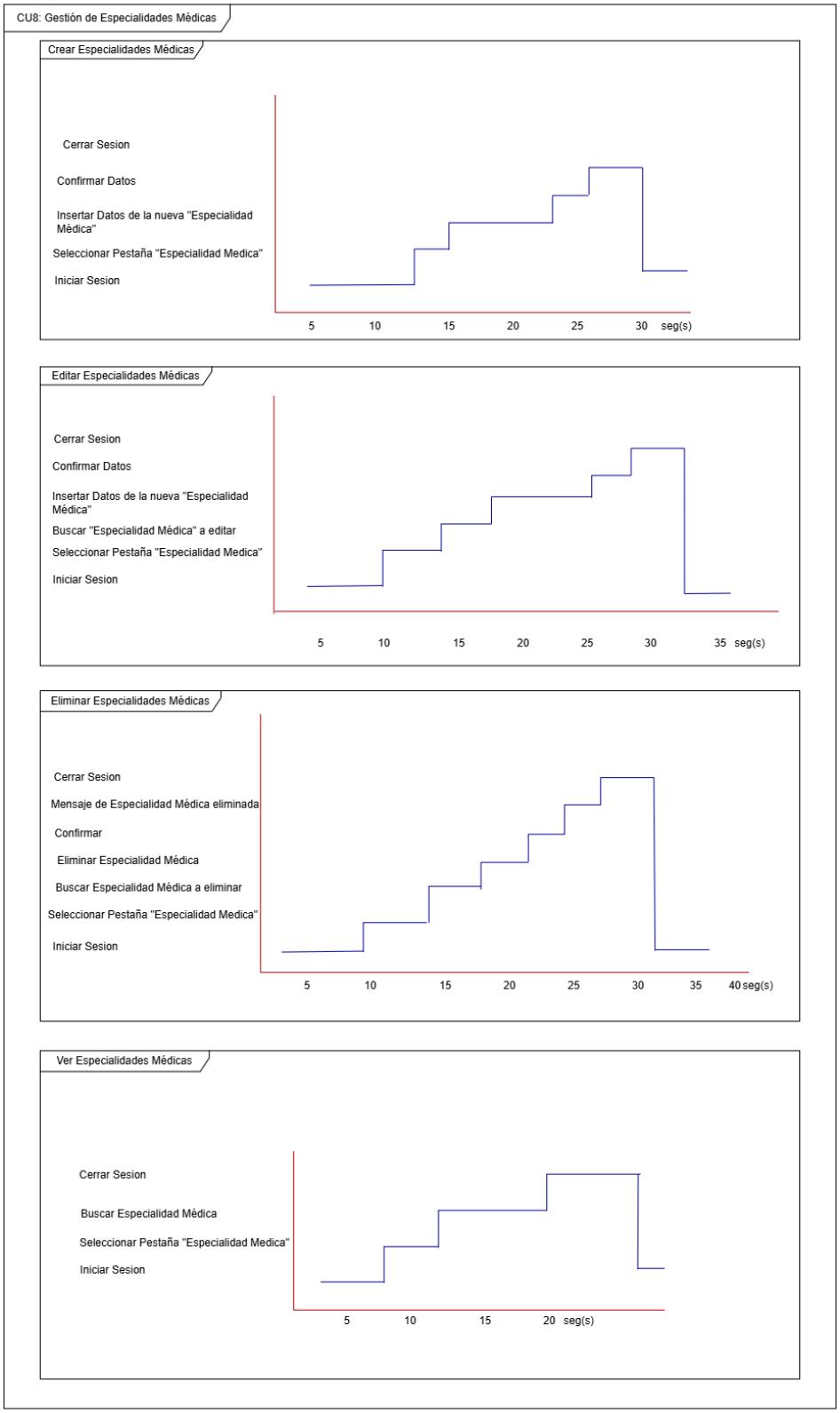


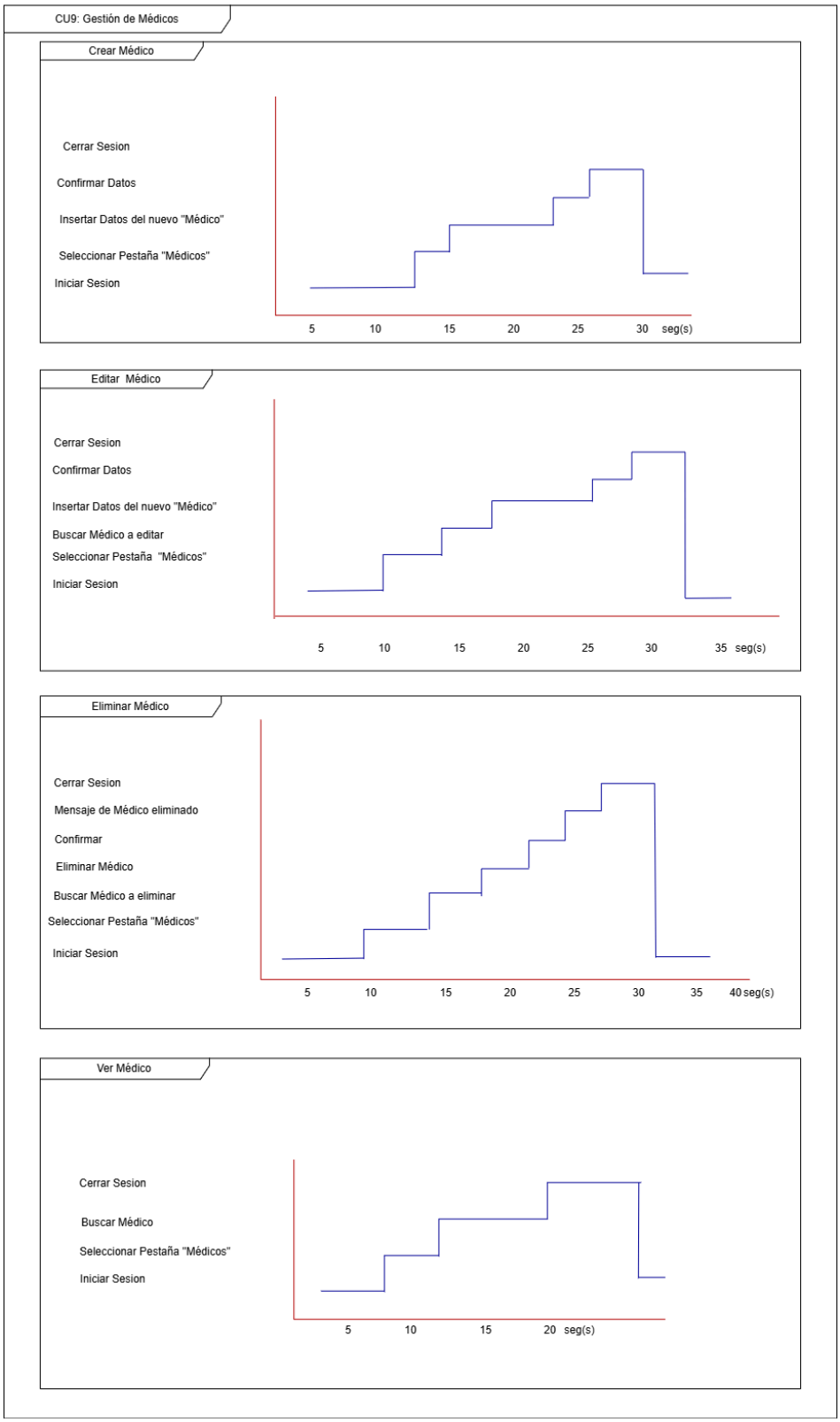
b) Diagrama de Estado

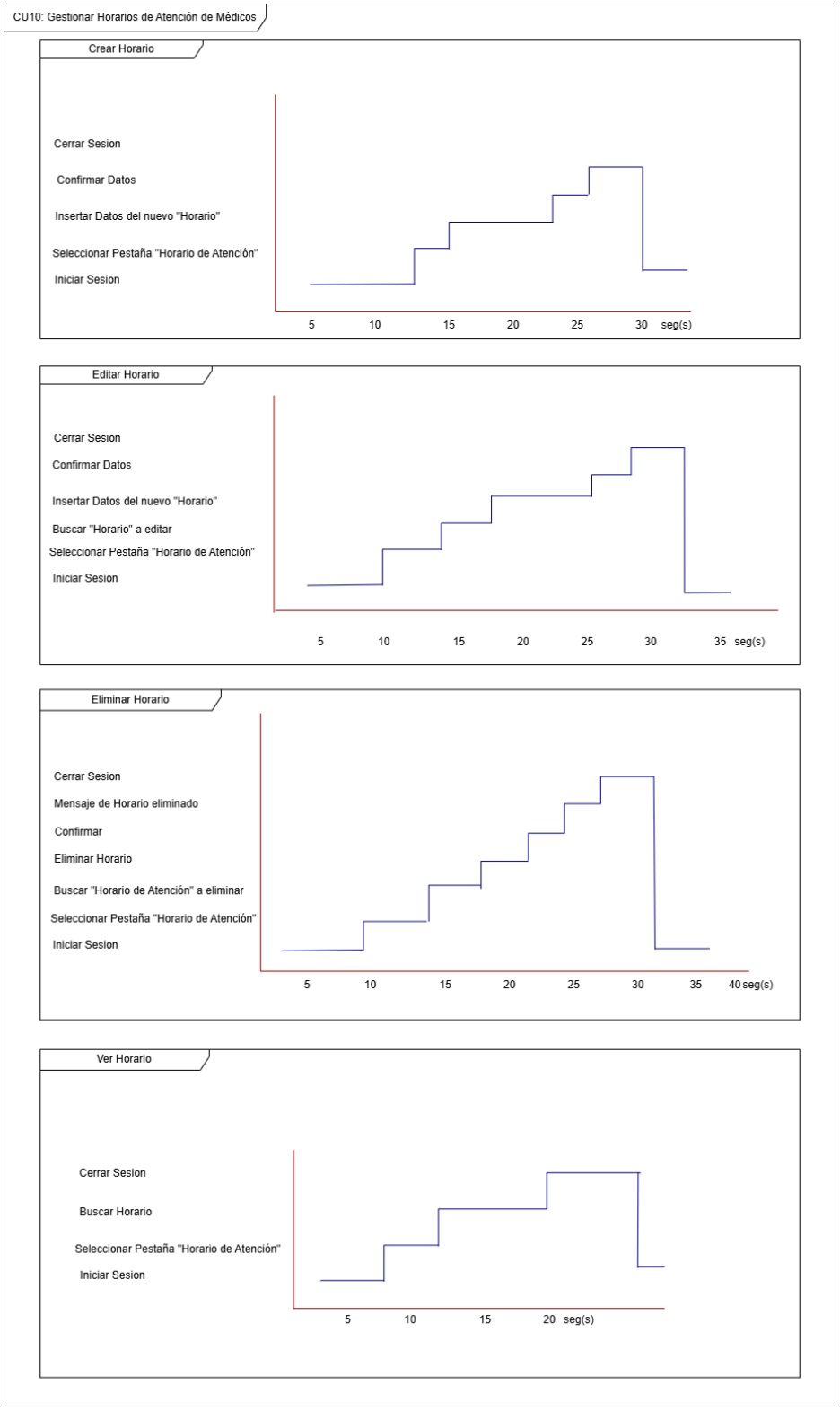


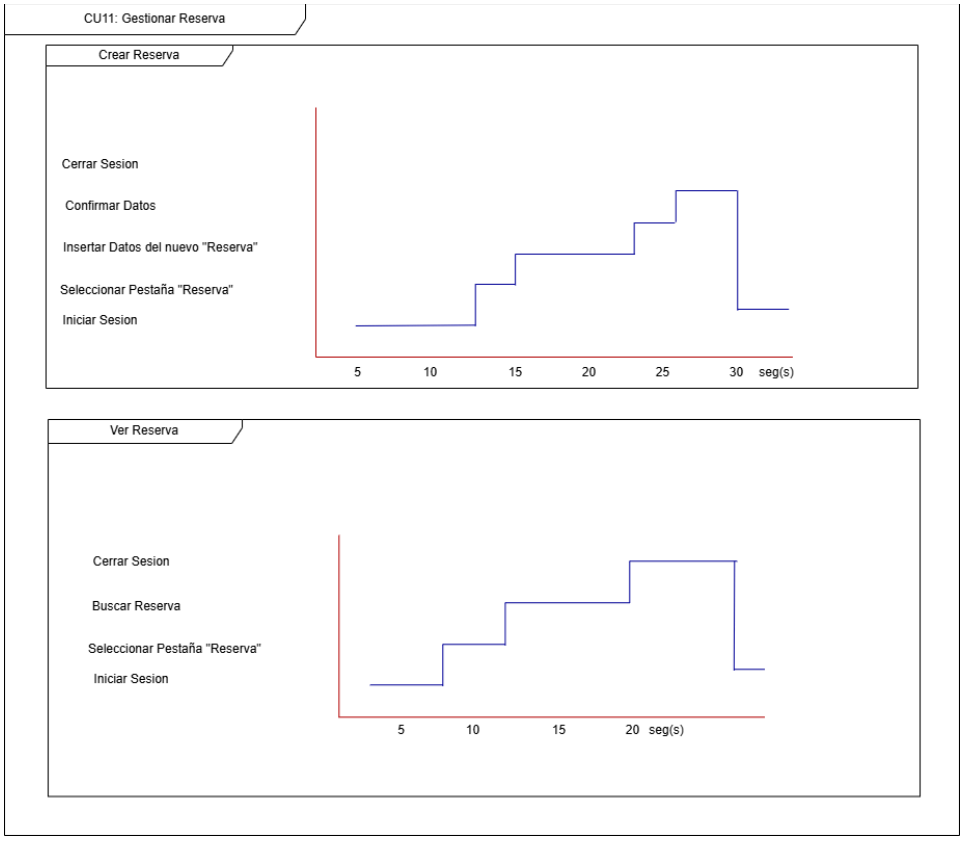


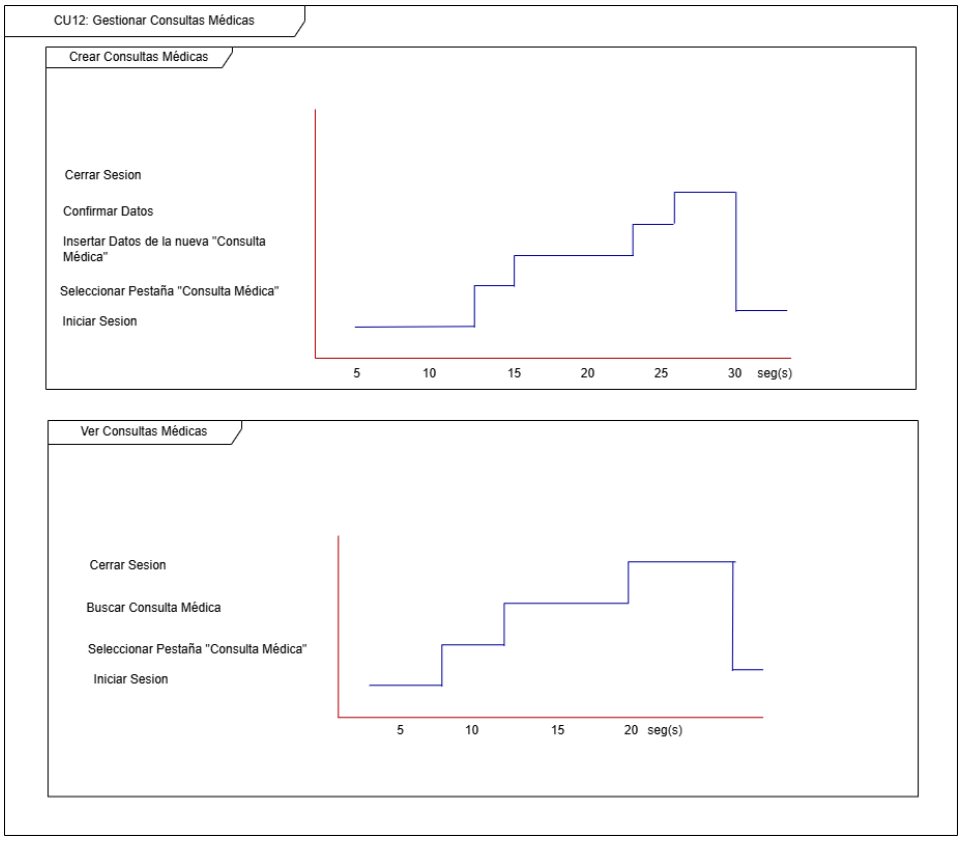
c) Diagrama de Tiempo

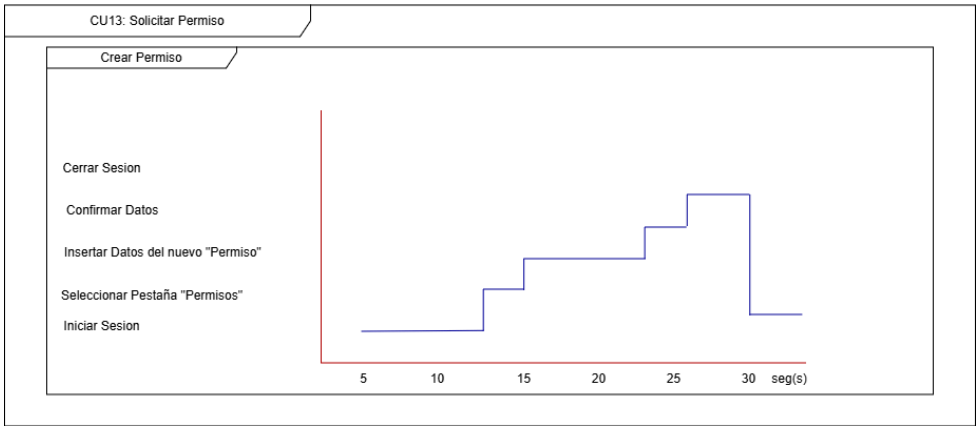


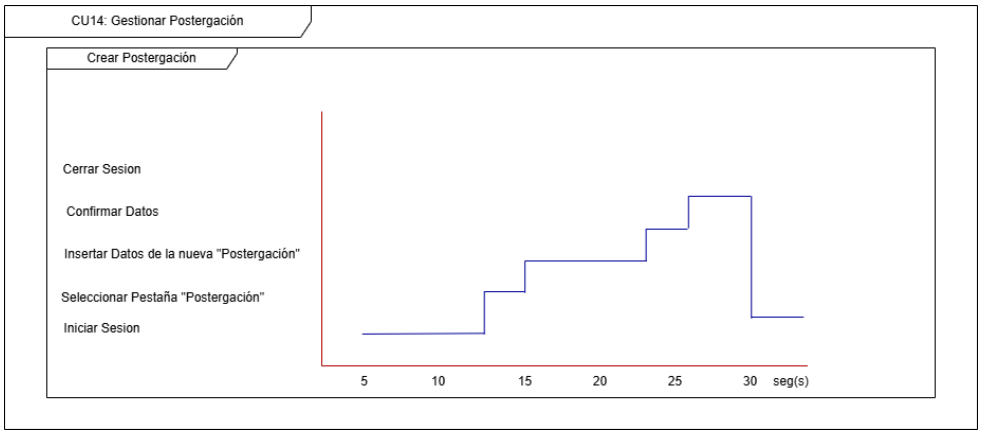

CU9: Gestión de Médicos
Crear Médico
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Médico"
Seleccionar Pestaña "Médicos"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Editar Médico
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Médico"
Buscar Médico a editar
Seleccionar Pestaña "Médicos"
Iniciar Sesion
5
10
15
20
25
30
35
seg(s)
Eliminar Médico
Cerrar Sesion
Mensaje de Médico eliminado
Confirmar
Eliminar Médico
Buscar Médico a eliminar
Seleccionar Pestaña "Médicos"
Iniciar Sesion
5
10
15
20
25
30
35
40
seg(s)
Ver Médico
Cerrar Sesion
Buscar Médico
Seleccionar Pestaña "Médicos"
Iniciar Sesion
5
10
15
20
seg(s)

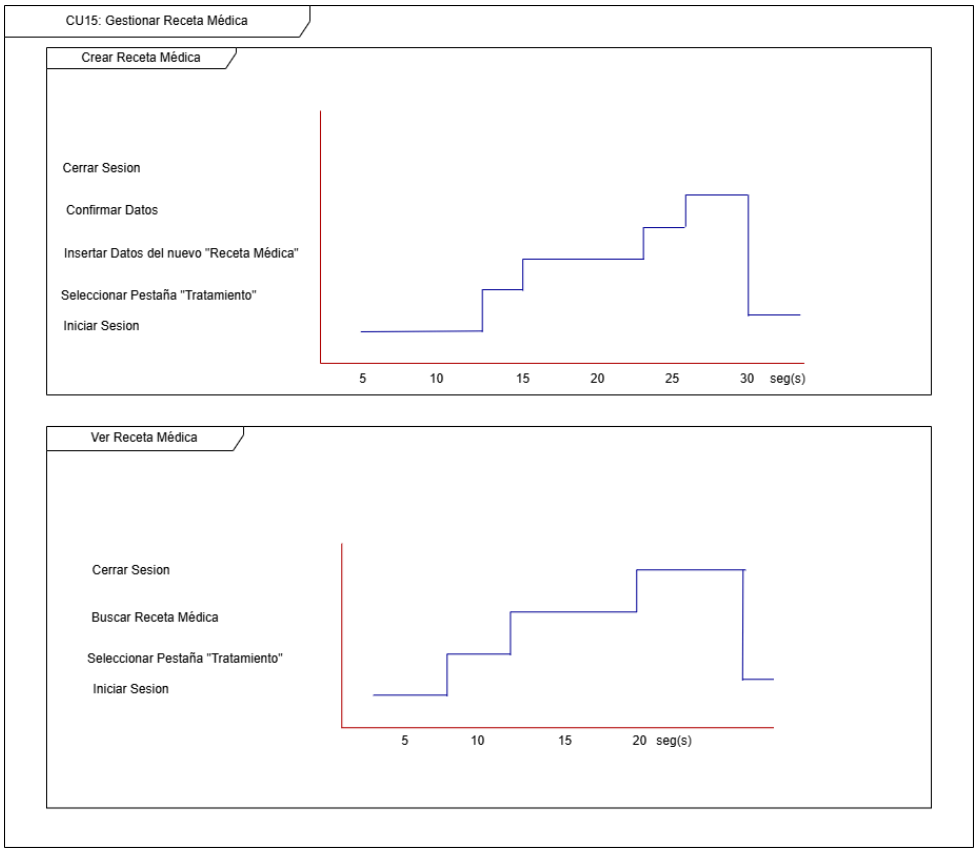

CU10: Gestionar Horarios de Atención de Médicos
Crear Horario
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Horario"
Seleccionar Pestaña "Horario de Atención"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Editar Horario
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Horario"
Buscar "Horario" a editar
Seleccionar Pestaña "Horario de Atención"
Iniciar Sesion
5
10
15
20
25
30
35
seg(s)
Eliminar Horario
Cerrar Sesion
Mensaje de Horario eliminado
Confirmar
Eliminar Horario
Buscar "Horario de Atención" a eliminar
Seleccionar Pestaña "Horario de Atención"
Iniciar Sesion
5
10
15
20
25
30
35
40
seg(s)
Ver Horario
Cerrar Sesion
Buscar Horario
Seleccionar Pestaña "Horario de Atención"
Iniciar Sesion
5
10
15
20
seg(s)

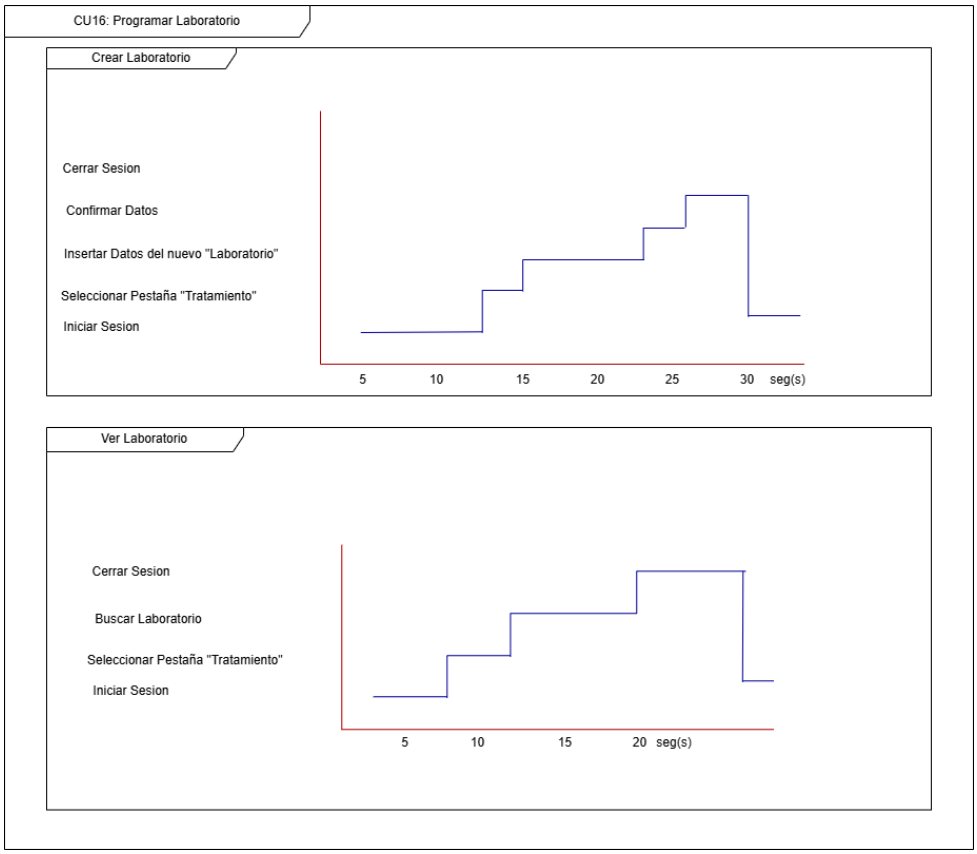

CU11: Gestionar Reserva
Crear Reserva
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Reserva"
Seleccionar Pestaña "Reserva"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Reserva
Cerrar Sesion
Buscar Reserva
Seleccionar Pestaña "Reserva"
Iniciar Sesion
5
10
15
20
seg(s)

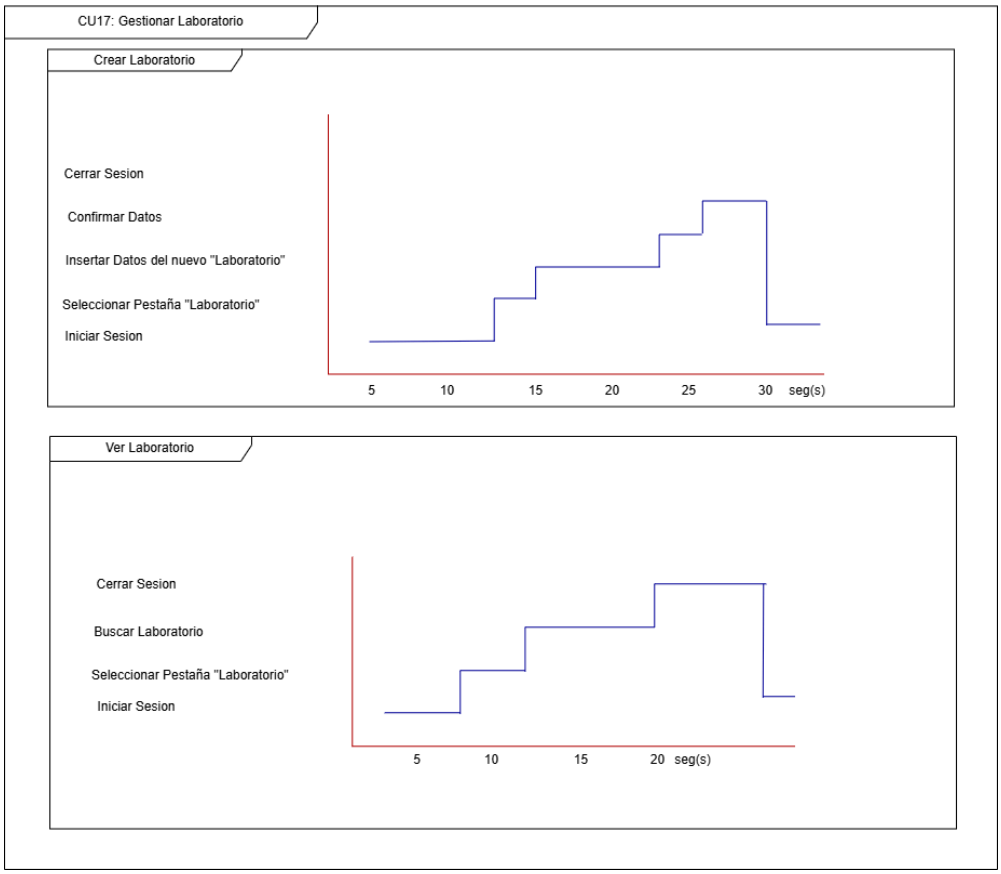

CU12: Gestionar Consultas Médicas
Crear Consultas Médicas
Cerrar Sesion
Confirmar Datos
Insertar Datos de la nueva "Consulta Médica"
Seleccionar Pestaña "Consulta Médica"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Consultas Médicas
Cerrar Sesion
Buscar Consulta Médica
Seleccionar Pestaña "Consulta Médica"
Iniciar Sesion
5
10
15
20
seg(s)

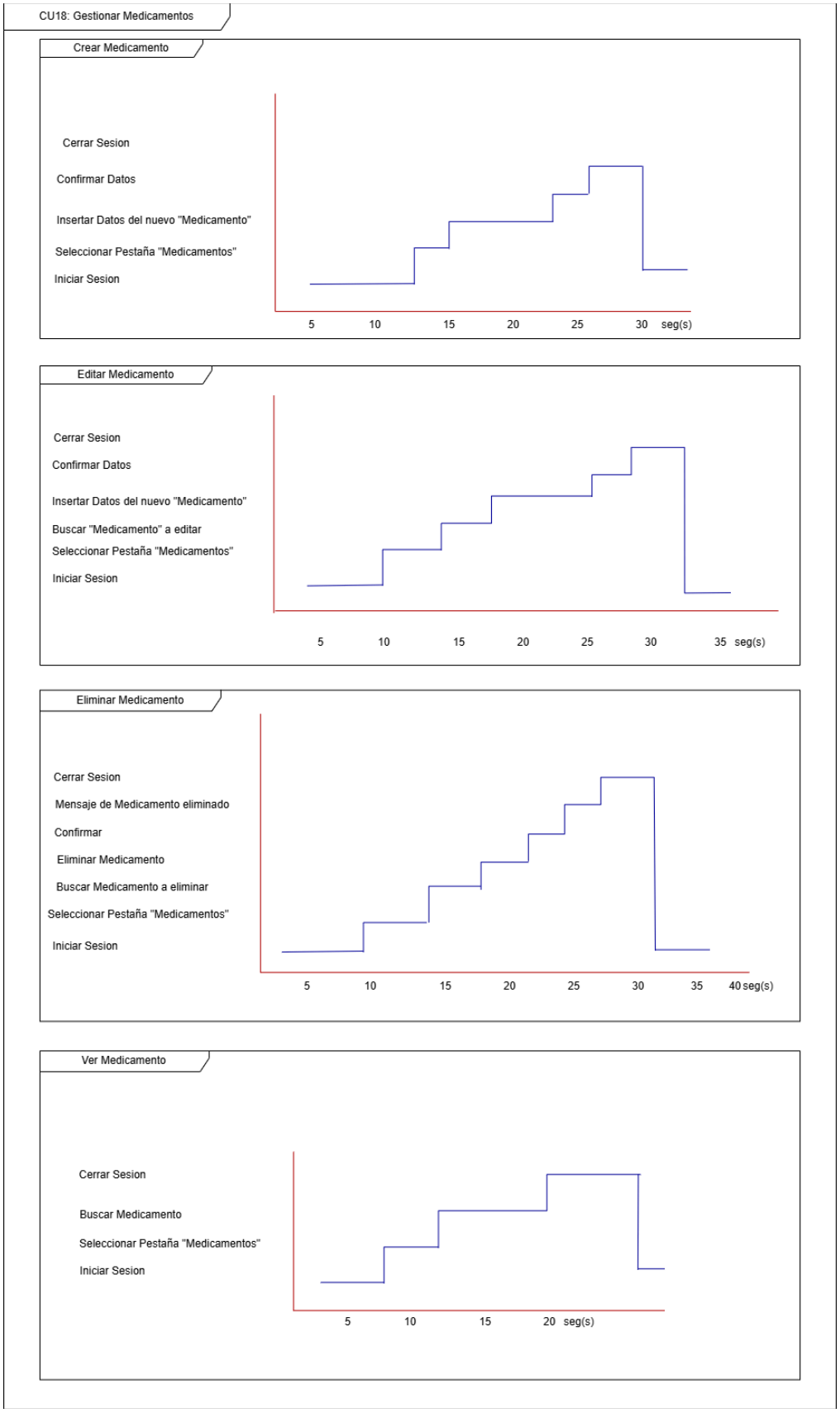

CU13: Solicitar Permiso
Crear Permiso
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Permiso"
Seleccionar Pestaña "Permisos"
Iniciar Sesion
5
10
15
20
25
30
seg(s)

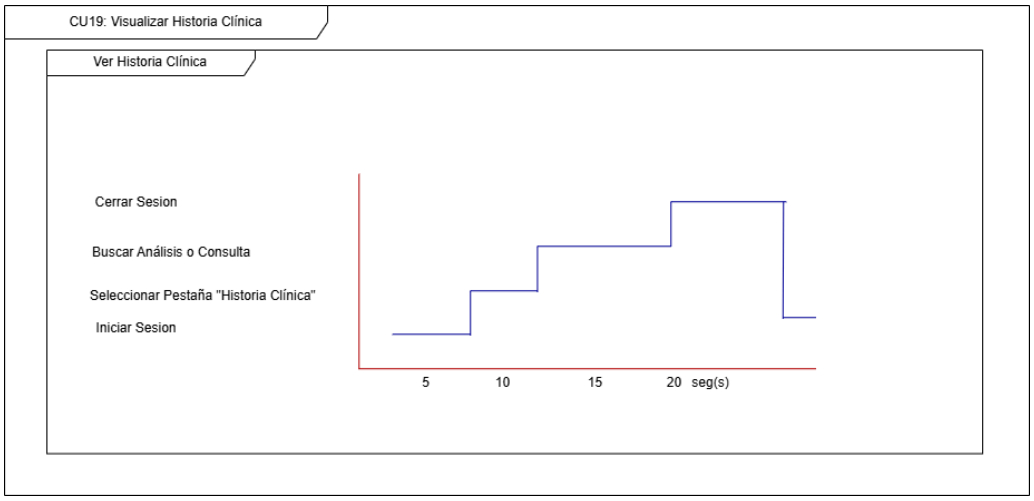

CU14: Gestionar Postergación
Crear Postergación
Cerrar Sesion
Confirmar Datos
Insertar Datos de la nueva "Postergación"
Seleccionar Pestaña "Postergación"
Iniciar Sesion
5
10
15
20
25
30
seg(s)


CU15: Gestionar Receta Médica
Crear Receta Médica
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Receta Médica"
Seleccionar Pestaña "Tratamiento"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Receta Médica
Cerrar Sesion
Buscar Receta Médica
Seleccionar Pestaña "Tratamiento"
Iniciar Sesion
5
10
15
20
seg(s)

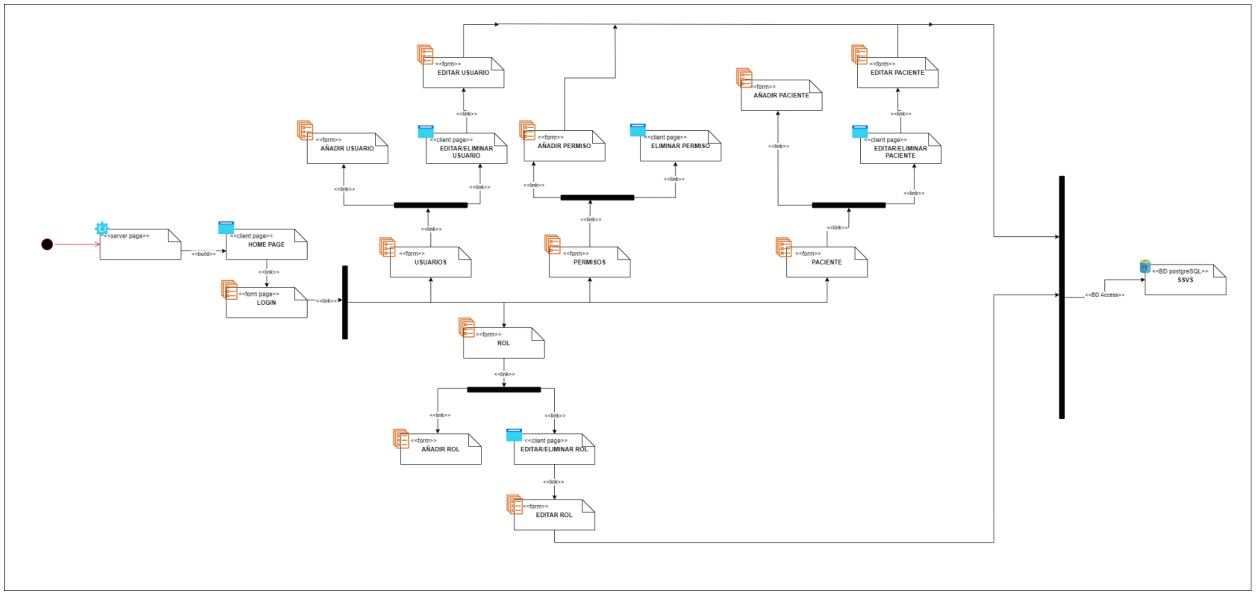

CU16: Programar Laboratorio
Crear Laboratorio
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Laboratorio"
Seleccionar Pestaña "Tratamiento"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Laboratorio
Cerrar Sesion
Buscar Laboratorio
Seleccionar Pestaña "Tratamiento"
Iniciar Sesion
5
10
15
20
seg(s)

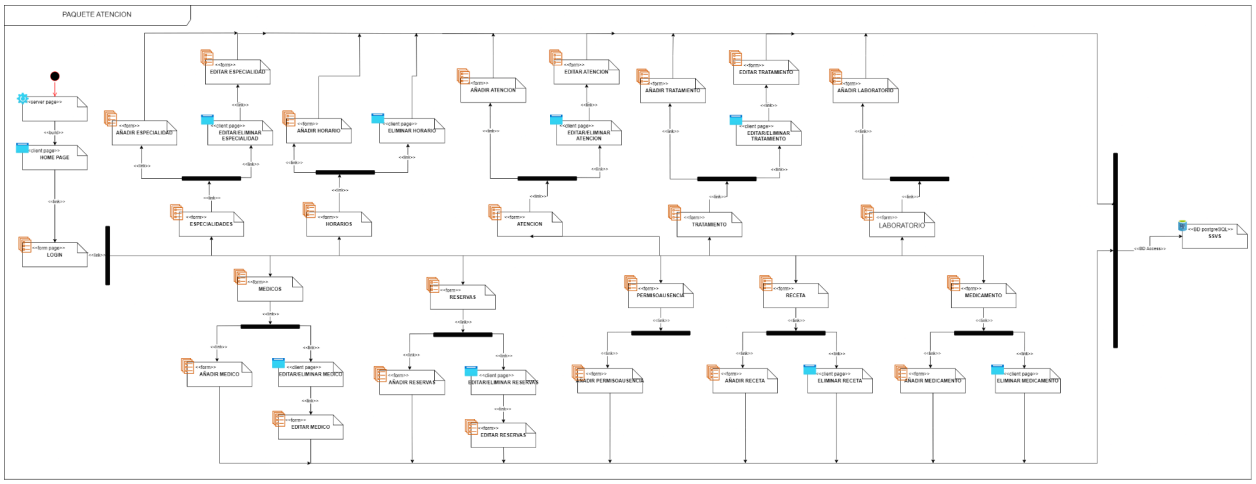

CU17: Gestionar Laboratorio
Crear Laboratorio
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Laboratorio"
Seleccionar Pestaña "Laboratorio"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Laboratorio
Cerrar Sesion
Buscar Laboratorio
Seleccionar Pestaña "Laboratorio"
Iniciar Sesion
5
10
15
20
seg(s)


CU18: Gestionar Medicamentos
Crear Medicamento
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Medicamento"
Seleccionar Pestaña "Medicamentos"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Editar Medicamento
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo "Medicamento"
Buscar "Medicamento" a editar
Seleccionar Pestaña "Medicamentos"
Iniciar Sesion
5
10
15
20
25
30
35
seg(s)
Eliminar Medicamento
Cerrar Sesion
Mensaje de Medicamento eliminado
Confirmar
Eliminar Medicamento
Buscar Medicamento a eliminar
Seleccionar Pestaña "Medicamentos"
Iniciar Sesion
5
10
15
20
25
30
35
40
seg(s)
Ver Medicamento
Cerrar Sesion
Buscar Medicamento
Seleccionar Pestaña "Medicamentos"
Iniciar Sesion
5
10
15
20
seg(s)



d) Diagrama de Navegación

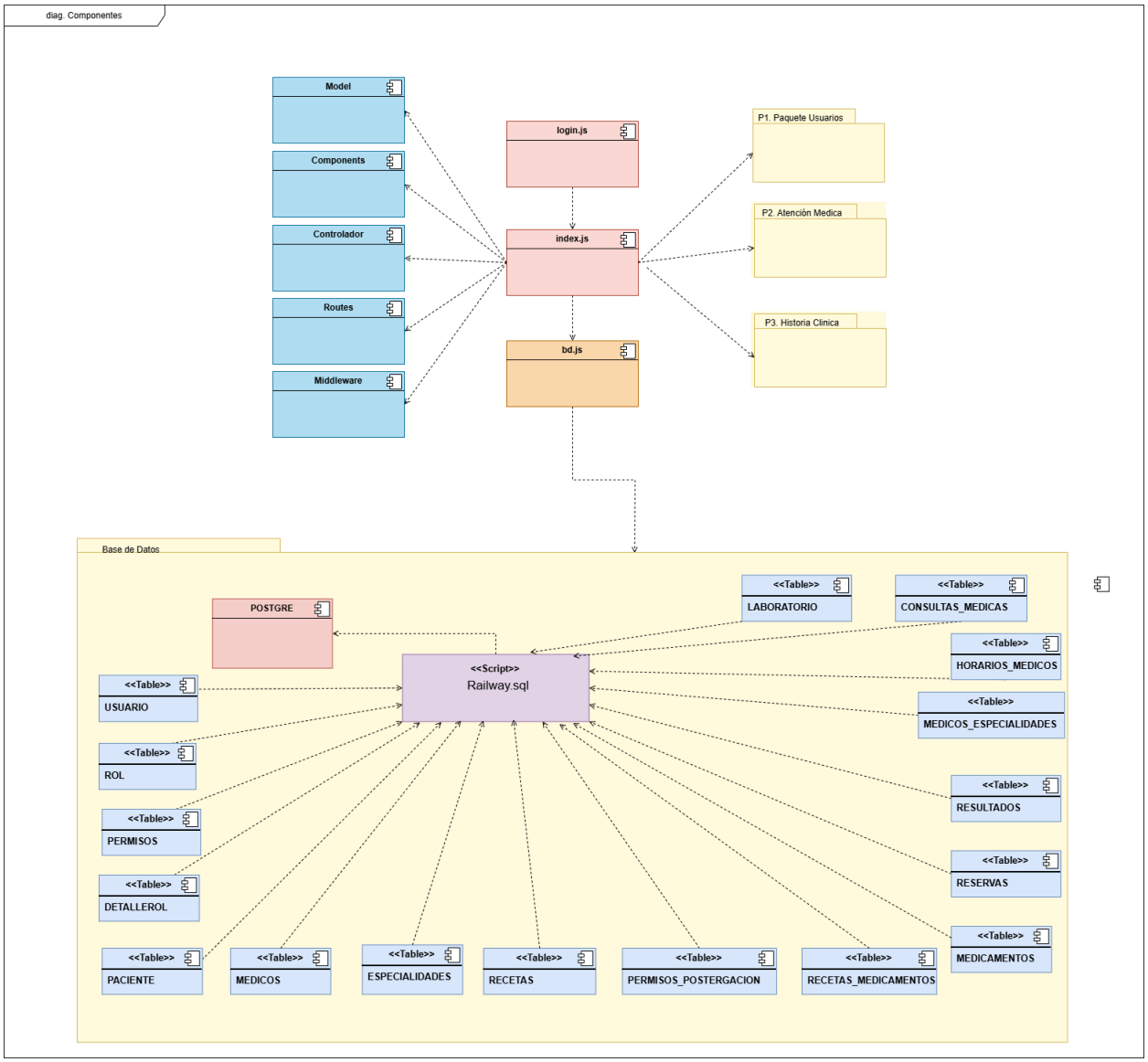




2.2 Implementación

2.2.1. Componentes y artefactos generados

2.2.1.1 Diagrama de componentes



2.3. Pruebas

2.3.1. Plan de pruebas (criterios de aceptación)

CU8: Gestión de Especialidades Médicas

- El sistema debe permitir registrar nuevas especialidades médicas con información completa (nombre, descripción, etc.).
- El sistema debe validar que no se registren especialidades médicas duplicadas.

- El sistema debe permitir editar la información de especialidades médicas existentes.
- El sistema debe permitir eliminar especialidades médicas que no estén asignadas a médicos.

CU9: Gestión de Médicos

- El sistema debe permitir registrar nuevos médicos con información completa (nombre, apellido, especialidad, número de identificación, contacto, etc.).
- El sistema debe validar que no se registren médicos duplicados (por número de identificación o registro profesional).
- El sistema debe permitir asignar una o varias especialidades a un médico.
- El sistema debe permitir editar la información de médicos existentes.
- El sistema debe permitir desactivar o eliminar médicos que no tengan citas programadas o historiales activos.

CU10: Gestionar Horarios de Atención de Médicos

- El sistema debe permitir programar horarios de atención para los médicos, especificando fechas, horas y ubicaciones.
- El sistema debe validar que los horarios de atención no se solapen con otros horarios asignados al mismo médico.
- El sistema debe permitir editar horarios de atención existentes.
- El sistema debe permitir eliminar horarios de atención que no tengan reservas asociadas.
- El sistema debe notificar a los pacientes si se realizan cambios que afecten a sus reservas.

CU11: Gestionar Reserva

- El sistema debe permitir a los pacientes reservar citas médicas en horarios disponibles, seleccionando médico y especialidad.
- El sistema debe validar la disponibilidad del horario al momento de la reserva.
- El sistema debe limitar al paciente a una reserva activa por día o según las políticas establecidas.
- El sistema debe permitir cancelar o reprogramar reservas existentes por parte del paciente o el administrador.
- El sistema debe enviar notificaciones de confirmación y recordatorio al paciente vía correo electrónico o mensaje de texto.

CU12: Gestionar Consultas Médicas

- El sistema debe permitir al médico visualizar su agenda de consultas programadas.
- El sistema debe permitir al médico iniciar y finalizar la consulta, registrando notas, diagnósticos y tratamientos.
- El sistema debe actualizar el estado de la consulta después de la atención (Atendida, No Asistió, Cancelada).

- El sistema debe garantizar que solo el médico asignado pueda acceder a la información de la consulta.

CU13: Solicitar Permiso

- El sistema debe permitir a los médicos solicitar permisos de ausencia, especificando fechas, horarios y motivos.
- El sistema debe permitir a los administradores aprobar o rechazar las solicitudes de permiso de ausencia.
- El sistema debe actualizar automáticamente los horarios de atención y notificar a los pacientes afectados por cambios o cancelaciones.
- El sistema debe registrar un historial de permisos de ausencia para cada médico.
- El sistema debe impedir que se asignen nuevas reservas en los periodos aprobados de ausencia.

CU14: Gestionar Postergación

- El sistema debe permitir que los médicos soliciten la postergación de una consulta programada, indicando motivos y nueva fecha.
- El sistema debe permitir que los administradores aprueben o rechacen la postergación solicitada.
- El sistema debe notificar a los pacientes afectados por la postergación de su consulta.
- El sistema debe actualizar la programación de consultas para reflejar los cambios aprobados.

CU15: Gestionar Receta Médica

- El sistema debe permitir al médico generar recetas médicas electrónicas, con información completa y legible.
- El sistema debe incluir en la receta detalles como medicamentos prescritos, dosis, indicaciones y duración del tratamiento.
- El sistema debe permitir imprimir la receta o enviarla electrónicamente al paciente o a la farmacia correspondiente.
- El sistema debe almacenar una copia de cada receta en la historia clínica del paciente.
- El sistema debe cumplir con las regulaciones y normativas vigentes para la emisión de recetas electrónicas.

CU16: Programar Laboratorio

- El sistema debe permitir al médico programar pruebas de laboratorio para los pacientes, indicando tipo de prueba, fecha y hora.
- El sistema debe validar que las pruebas no se solapen con otros horarios asignados al mismo paciente.
- El sistema debe permitir editar o cancelar las pruebas de laboratorio programadas.

- El sistema debe notificar al paciente y al laboratorio sobre la programación o los cambios en la cita.

CU17: Gestionar Laboratorio

- El sistema debe permitir a los administradores registrar nuevos laboratorios con información completa (nombre, dirección, contacto, etc.).
- El sistema debe permitir gestionar los horarios de atención y disponibilidad de cada laboratorio.
- El sistema debe registrar y asociar los resultados de pruebas de laboratorio a la historia clínica de los pacientes.
- El sistema debe notificar al paciente cuando los resultados de laboratorio estén disponibles.

CU18: Gestionar Medicamentos

- El sistema debe permitir al administrador registrar nuevos medicamentos con información completa (nombre, descripción, dosificación, etc.).
- El sistema debe validar que no se registren medicamentos duplicados.
- El sistema debe permitir editar la información de medicamentos existentes.
- El sistema debe permitir eliminar medicamentos que no estén asociados a recetas activas.

CU19: Visualizar Historia Clínica

- El sistema debe permitir a los médicos y pacientes autorizados visualizar la historia clínica del paciente.
- El sistema debe garantizar la confidencialidad y seguridad de la información clínica, restringiendo el acceso según roles.
- El sistema debe permitir visualizar el historial completo de atenciones, diagnósticos, tratamientos, recetas y resultados de laboratorio.
- El sistema debe permitir adjuntar documentos adicionales (imágenes, resultados de laboratorio, etc.) a la historia clínica.

CU23: Enviar Recordatorio de Cita al Paciente

- El sistema debe permitir programar recordatorios automáticos para las citas médicas de los pacientes.
- El sistema debe enviar recordatorios vía correo electrónico o mensaje de texto al paciente, especificando fecha, hora y ubicación.
- El sistema debe permitir configurar la frecuencia y tipo de recordatorio (diario, semanal, etc.).
- El sistema debe permitir al paciente confirmar su asistencia desde el recordatorio recibido.

2.3.2. Reporte de prueba

CU8: Gestión de Especialidades Médicas

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los administradores gestionar la información de los médicos.	Acceder al módulo de Gestión de Médicos. Intentar registrar un médico con datos incompletos. Intentar registrar un médico duplicado. Registrar un nuevo médico con datos completos. Editar la información del médico registrado. Desactivar o eliminar el médico registrado sin citas programadas.	Permitir registrar nuevos médicos. Validar que no se registren médicos duplicados. Permitir editar la información de médicos existentes. Permitir desactivar o eliminar médicos sin citas activas.	Se validan correctamente los datos completos del médico. No se permite registrar médicos duplicados. La información del médico se edita correctamente. El médico se desactiva o elimina correctamente.	Cumple con los resultados esperados.	Aprobado

CU9: Gestión de Médicos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado

Permite a los administradores programar y gestionar los horarios de atención de los médicos.	Acceder al módulo de Horarios de Atención. Intentar registrar un horario con datos incompletos. Intentar registrar un horario que se solape con otro. Registrar un horario con datos completos. Editar el horario registrado. Eliminar el horario sin reservas asociadas.	Permitir registrar horarios sin solapamientos. Validar que no se registren horarios duplicados. Permitir editar horarios existentes. Permitir eliminar horarios sin reservas asociadas. Notificar cambios a los pacientes afectados.	Se registran y validan los horarios correctamente. No se permite registrar horarios solapados. La edición y eliminación de horarios funciona correctamente. Notificaciones enviadas correctamente.	Cumple con los resultados esperados.	Aprobado

CU10: Gestionar Horarios de Atención de Médicos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los pacientes y administradores gestionar reservas de citas.	Acceder al módulo de Reservas. Intentar reservar en un horario no disponible. Intentar reservar más	Permitir reservas en horarios disponibles. Limitar reservas diarias. Permitir cancelación y	Las reservas se gestionan correctamente, con disponibilidad verificada. Limitación de reservas	Cumple con los resultados esperados.	Aprobado

	de una vez en el mismo día. Realizar una reserva en un horario disponible. Cancelar o reprogramar la reserva.	reprogramación. Enviar confirmación y recordatorio al paciente.	diarias cumplida. Cancelación y reprogramación realizadas correctamente. Notificaciones enviadas correctamente.		

CU11: Gestionar Reserva

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los médicos gestionar la agenda y atención de consultas.	Acceder a la agenda de consultas. Iniciar y finalizar una consulta. Registrar notas y diagnósticos en la consulta. Registrar ausencias de pacientes.	Permitir visualizar y atender las consultas programadas. Registrar notas y diagnósticos en la consulta. Registrar y actualizar el estado de la consulta.	La agenda de consultas se visualiza correctamente. La atención de la consulta se registra con notas y diagnósticos. Ausencias de pacientes registradas correctamente.	Cumple con los resultados esperados.	Aprobado

CU12: Gestionar Consultas Médicas

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los médicos solicitar y gestionar permisos de ausencia.	Acceder al módulo de Permisos. Solicitar un permiso con fechas válidas. Aprobar o rechazar el permiso. Verificar cancelación automática de citas afectadas.	Permitir solicitud de permisos de ausencia. Aprobar o rechazar permisos. Cancelar automáticamente citas afectadas.	El permiso de ausencia se solicita y aprueba correctamente. Citas afectadas canceladas automáticamente.	Cumple con los resultados esperados.	Aprobado

CU13: Solicitar Permiso

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite solicitar y gestionar postergaciones de consultas.	Acceder al módulo de Postergaciones. Solicitar postergación con nueva fecha. Aprobar o rechazar la postergación. Notificar al paciente de la postergación.	Permitir solicitar y aprobar postergaciones. Notificar cambios al paciente.	La postergación se solicita y aprueba correctamente. Notificación enviada al paciente.	Cumple con los resultados esperados.	Aprobado

CU14: Gestionar Postergación

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite al médico gestionar recetas médicas electrónicas.	Generar una receta con detalles completos. Imprimir o enviar la receta. Guardar la receta en la historia clínica.	Generar recetas con información completa. Imprimir o enviar recetas. Guardar recetas en la historia clínica.	La receta se genera y envía correctamente. La receta se almacena en la historia clínica.	Cumple con los resultados esperados.	Aprobado

CU15: Gestionar Receta Médica

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite programar pruebas de laboratorio.	Programar una prueba de laboratorio con fecha y hora. Validar que no haya solapamientos. Notificar la programación al paciente.	Permitir programación de laboratorio. Evitar solapamientos. Notificar al paciente.	Prueba de laboratorio programada y notificada correctamente.	Cumple con los resultados esperados.	Aprobado

CU16: Programar Laboratorio

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite registrar y gestionar laboratorios y resultados.	Registrar un laboratorio. Gestionar disponibilidad del laboratorio.	Permitir registrar laboratorios y disponibilidad.	Registro y gestión de laboratorios correctamente.	Cumple con los resultados esperados.	Aprobado

	Asociar resultados de pruebas a la historia clínica.	Asociar resultados a la historia clínica. Notificar al paciente cuando los resultados estén disponibles.	Resultados asociados y notificados correctamente.		

CU17: Gestionar Laboratorio

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite registrar y gestionar medicamentos.	Registrar un medicamento con información completa. Validar que no haya duplicados. Editar y eliminar medicamentos no asociados a recetas activas.	Permitir registrar y editar medicamentos. Validar duplicados. Permitir eliminar medicamentos no asociados a recetas activas.	Medicamentos gestionados correctamente.	Cumple con los resultados esperados.	Aprobado

CU18: Gestionar Medicamentos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite visualizar la historia clínica del paciente.	Acceder al módulo de Historia Clínica.	Permitir acceso seguro a la historia	La historia clínica se visualiza y maneja	Cumple con los resultados esperados.	Aprobado

	Verificar que solo usuarios autorizados puedan visualizar la historia clínica. Visualizar el historial completo de atenciones, diagnósticos, tratamientos y recetas. Adjuntar y visualizar documentos adicionales (imágenes, resultados de laboratorio, etc.).	clínica según roles. Visualizar el historial completo de atenciones, diagnósticos, tratamientos y recetas. Permitir adjuntar documentos adicionales a la historia clínica.	correctamente según los permisos. El historial completo y los documentos adicionales se visualizan adecuadamente.		

CU19: Visualizar Historia Clínica

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite visualizar la historia clínica completa del paciente, incluyendo atenciones, diagnósticos y documentos adicionales.	Acceder al módulo de Historia Clínica como un usuario autorizado. Visualizar el historial completo de atenciones, diagnósticos, tratamientos y recetas.	Permitir acceso seguro a la historia clínica según roles. Visualizar el historial completo de atenciones, diagnósticos, tratamientos y recetas.	La historia clínica se visualiza adecuadamente y según permisos. Se muestran el historial completo y los documentos adicionales	Cumple con los resultados esperados.	Aprobado

	Adjuntar y visualizar documentos adicionales (imágenes, resultados de laboratorio, etc.). Verificar la seguridad y restricción de acceso a la información clínica.	Permitir adjuntar y visualizar documentos adicionales en la historia clínica.	correctamente. Acceso restringido a usuarios autorizados.		

CU23: Enviar Recordatorio de Cita al Paciente

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite enviar recordatorios de citas a los pacientes.	Programar un recordatorio automático para una cita. Configurar la frecuencia del recordatorio (diario, semanal, etc.). Verificar el envío de recordatorio al paciente por correo electrónico o mensaje de texto. Confirmar que el paciente reciba la	Programar recordatorios automáticos. Configurar frecuencia y tipo de recordatorio. Enviar recordatorios correctamente al paciente. Permitir confirmación de asistencia por parte del paciente.	Los recordatorios se envían correctamente según la frecuencia configurada. Los pacientes reciben y pueden confirmar la asistencia a la cita.	Cumple con los resultados esperados.	Aprobado

	notificación y pueda confirmar su asistencia.				

3. Daily Scrum

Daily Scrum - 28/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Gestionar Postergacion	Ninguno	Continuar con Implementacion del backend para Gestionar Postergacion
Fabiana Manzoni Bravo	Avance en Implementacion del backend de la Gestion de Especialidades Médicas	Ninguno	Continuar con Implementacion del backend de la Gestion de Especialidades Médicas

Daily Scrum - 29/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del frontend para Gestionar Postergacion	Ninguno	Continuar con Implementacion del frontend para Gestionar Postergacion
Fabiana Manzoni Bravo	Avance en Implementacion del frontend de la Gestion de Especialidades Médicas	Ninguno	Continuar con Implementacion del frontend de la Gestion de Especialidades Médicas

Daily Scrum - 30/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Programar Receta Medica	Ninguno	Continuar con Implementacion del backend para Programar Receta Medica
Fabiana Manzoni Bravo	Avance en Implementacion del backend de la Gestion de Médicos	Ninguno	Continuar con Implementacion del backend de la Gestion de Médicos

Daily Scrum - 31/10/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del frontend para Programar Receta Medica	Ninguno	Continuar con Implementacion del frontend para Programar Receta Medica
Fabiana Manzoni Bravo	Avance en Implementacion del frontend de la Gestion de Médicos	Ninguno	Continuar con Implementacion del frontend de la Gestion de Médicos

Daily Scrum - 01/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Programar Laboratorio	Ninguno	Continuar con Implementacion del backend para Programar Laboratorio

Fabiana Manzoni Bravo	Avance en Implementacion del backend para la Gestion de Horarios de Atencion de Médicos	Ninguno	Continuar con Implementacion del backend para la Gestion de Horarios de Atencion de Médicos

Daily Scrum - 02/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del frontend para Programar Laboratorio	Ninguno	Continuar con Implementacion del frontend para Programar Laboratorio
Fabiana Manzoni Bravo	Avance en Implementacion del frontend para la Gestion de Horarios de Atencion de Médicos	Ninguno	Continuar con Implementacion del frontend para la Gestion de Horarios de Atencion de Médicos

Daily Scrum - 03/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Gestionar Laboratorio	Ninguno	Continuar con Implementacion del backend para Gestionar Laboratorio
Fabiana Manzoni Bravo	Avance en Implementacion del backend para la Gestion de Reserva	Ninguno	Continuar con Implementacion del backend para la Gestion de Reserva

Daily Scrum - 04/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del frontend para la Gestion de Reserva	Ninguno	Continuar con Implementacion del frontend para Gestionar Laboratorio
Fabiana Manzoni Bravo	Avance en Implementacion del frontend para la Gestion de Reserva	Ninguno	Avance en Implementacion del frontend para la Gestion de Reserva

Daily Scrum - 05/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Gestionar Medicamentos	Ninguno	Continuar con Implementacion del backend para Gestionar Medicamentos
Fabiana Manzoni Bravo	Avance en Implementacion del backend para la Gestion de Consultas Médicas	Ninguno	Continuar con Implementacion del backend para la Gestion de Consultas Médicas

Daily Scrum - 06/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del frontend para Gestionar Medicamentos	Ninguno	Continuar con Implementacion del frontend para Gestionar Medicamentos

Fabiana Manzoni Bravo	Avance en Implementacion del frontend para la Gestion de Consultas Médicas	Ninguno	Continuar con Implementacion del frontend para la Gestion de Consultas Médicas

Daily Scrum - 07/112024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Avance en Implementacion del backend para Gestionar Historia Clínica	Ninguno	Continuar con Implementacion del backend para Gestionar Historia Clínica
Fabiana Manzoni Bravo	Continuar con Implementacion del backend para Solicitar Permisos	Ninguno	Continuar con Implementacion del backend para Solicitar Permisos

4. Sprint Review

Revisión de Sprint	
Objetivo del Sprint: El objetivo del sprint es desarrollar e implementar las funcionalidades centrales del Sistema de Gestión de Historias Clínicas Electrónicas (HCE) para el Seguro de Salud Vida Sana (SSVS), enfocándose en los modulos de atencion medica y de historia clinica. Pudiendo ser: Gestión de Médicos, Horarios de Atención de Médicos, Reservas, Consultas Médicas, Permisos, Postergación de Fichas, Recetas Médicas, Laboratorios, Medicamentos, Historial Clínico y Recordatorios para las citas médicas.	
Participantes	
Nombre	**Rol**
Jorge Arturo Aliaga Valencia	Desarrollador

Fabiana Manzoni Bravo	Desarrollador
Presentación del incremento	
Función Presentada	**Retroalimentación**
Diseñar interfaz para Gestión de Especialidades Médicas	Funcionalidad Terminada
Implementar Gestion de Especialidades Médicas	Funcionalidad Terminada
Realizar pruebas de Gestion de Especialidades Médicas	Funcionalidad Terminada
Diseñar interfaz para Gestion de Médicos	Funcionalidad Terminada
Implementar Gestion de Médicos	Funcionalidad Terminada
Realizar pruebas de Gestion de Médicos	Funcionalidad Terminada
Diseñar interfaz para Gestionar Horarios de Atención de Médicos	Funcionalidad Terminada
Implementar Gestionar Horarios de Atención de Médicos	Funcionalidad Terminada
Realizar pruebas de Gestionar Horarios de Atención de Médicos	Funcionalidad Terminada
Diseñar interfaz para Gestionar Reserva	Funcionalidad Terminada
Implementar Gestionar Reserva	Funcionalidad Terminada
Realizar pruebas de Gestionar Reserva	Funcionalidad Terminada
Diseñar interfaz para Gestionar Consultas Médicas	Funcionalidad Terminada
Implementar Gestionar Consultas Médicas	Funcionalidad Terminada
Realizar pruebas de Gestionar Consultas Médicas	Funcionalidad Terminada

Diseñar interfaz para Solicitar Permiso	Funcionalidad Terminada
Implementar Solicitar Permiso	Funcionalidad Terminada
Realizar pruebas de Solicitar Permiso	Funcionalidad Terminada
Diseñar interfaz para Gestionar Postergación	Funcionalidad Terminada
Implementar Gestionar Postergación	Funcionalidad Terminada
Realizar pruebas de Gestionar Postergación	Funcionalidad Terminada
Diseñar interfaz para Gestionar Receta Médica	Funcionalidad Terminada
Implementar Gestionar Receta Médica	Funcionalidad Terminada
Realizar pruebas de Gestionar Receta Médica	Funcionalidad Terminada
Diseñar interfaz para Programar Laboratorio	Funcionalidad Terminada
Implementar Programar Laboratorio	Funcionalidad Terminada
Realizar pruebas de Programar Laboratorio	Funcionalidad Terminada
Diseñar interfaz para Gestionar Laboratorio	Funcionalidad Terminada
Implementar Gestionar Laboratorio	Funcionalidad Terminada
Realizar pruebas de Gestionar Laboratorio	Funcionalidad Terminada
Diseñar interfaz para Gestionar Medicamentos	Funcionalidad Terminada
Implementar Gestionar Medicamentos	Funcionalidad Terminada
Realizar pruebas de Gestionar Medicamentos	Funcionalidad Terminada

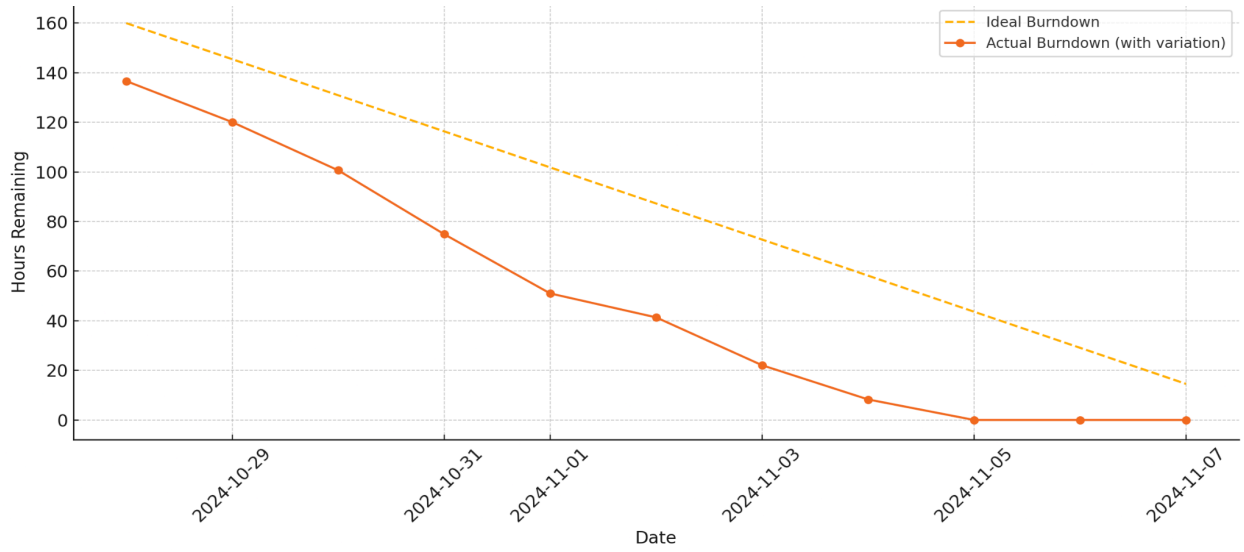
Diseñar interfaz para Visualizar Historia Clínica	Funcionalidad Terminada
Implementar Visualizar Historia Clínica	Funcionalidad Terminada
Realizar pruebas de Visualizar Historia Clínica	Funcionalidad Terminada
Diseñar interfaz para Enviar Recordatorio de Cita al Paciente	Funcionalidad Terminada
Implementar Enviar Recordatorio de Cita al Paciente	Funcionalidad Terminada
Realizar pruebas de Enviar Recordatorio de Cita al Paciente	Funcionalidad Terminada

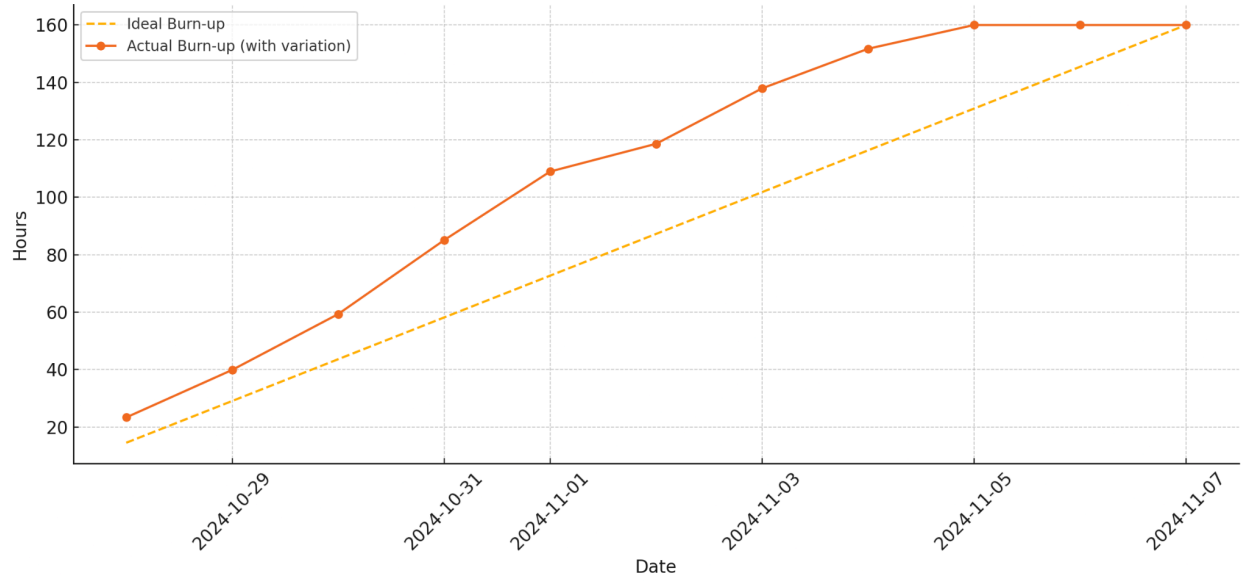
5. Sprint Retrospective

Sprint Retrospective
Fecha
07/11/2024
Facilitador
Fabiana Manzoni Bravo
Objetivo
Conclusión del Sprint 2
Nombres de asistentes
Jorge Arturo Aliaga Valencia
Temas por tratar
Sprint 2
Discusión
¿Qué salió bien?

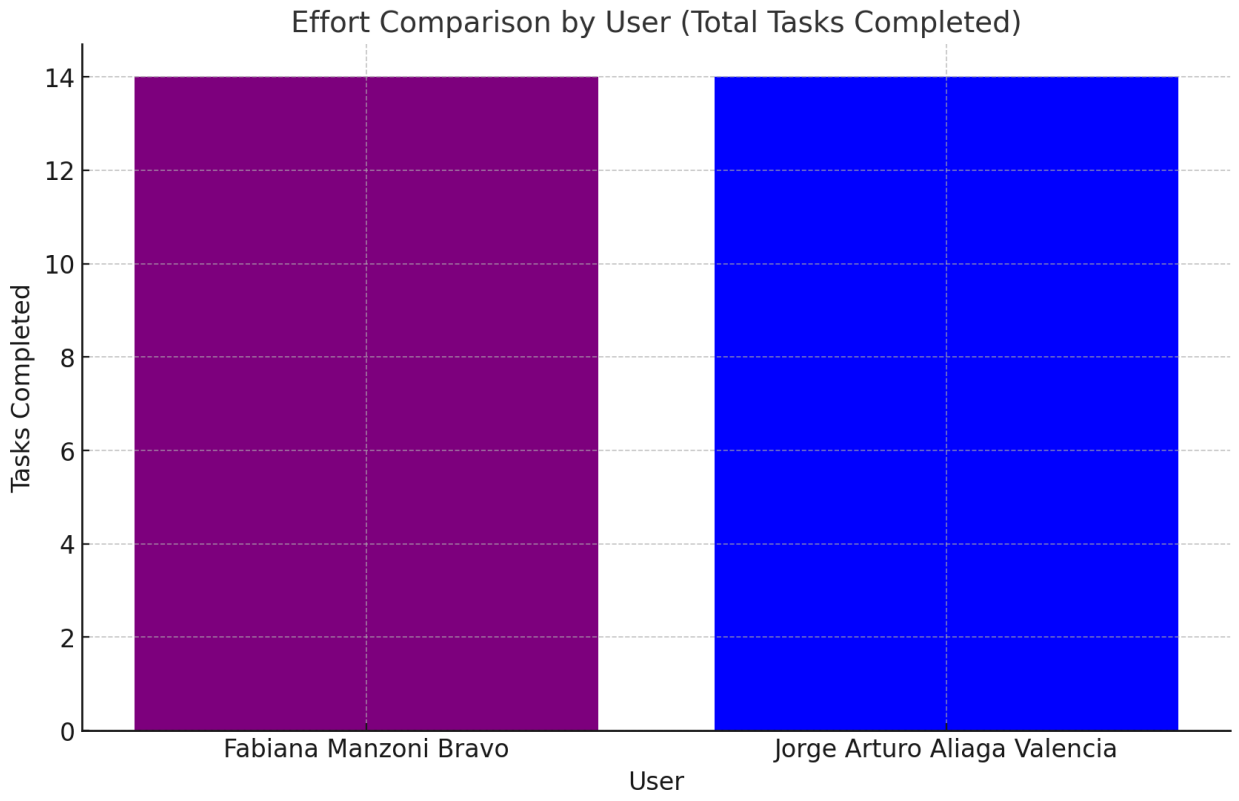
- Logramos completar todas las tareas del backlog de acuerdo a lo planificado, lo cual nos permitió entregar las funcionalidades requeridas para este sprint. - La comunicación entre los miembros del equipo fue fluida. Los Daily Scrums ayudaron a mantener el enfoque y a resolver rápidamente cualquier duda o bloqueo. - A pesar de la presión en algunos momentos, logramos mantener una buena calidad en el código y en las funcionalidades entregadas, lo cual fue validado durante las revisiones de las tareas.
¿Qué no salió bien?
- Algunas tareas tomaron más tiempo del esperado. Esto generó una carga de trabajo extra al final del sprint, y algunos miembros del equipo tuvieron que trabajar el fin de semana para compensar. - La documentación se fue realizando junto con el desarrollo, pero al final del sprint tuvimos que dedicar tiempo extra para dejarla completa y bien estructurada, lo que generó algo de presión adicional.
¿Qué haremos de forma diferente?
- En los próximos sprints, dedicaremos más tiempo a revisar y ajustar las estimaciones de cada tarea. Nos enfocaremos en evaluar las tareas con mayor precisión para evitar trabajar fuera de horario y así cumplir los objetivos dentro del tiempo laboral. - Vamos a establecer un hábito de actualizar la documentación en tiempo real, de modo que no se acumule al final del sprint. Revisaremos diariamente para asegurar que esté al día y evitar hacer ajustes de último minuto.

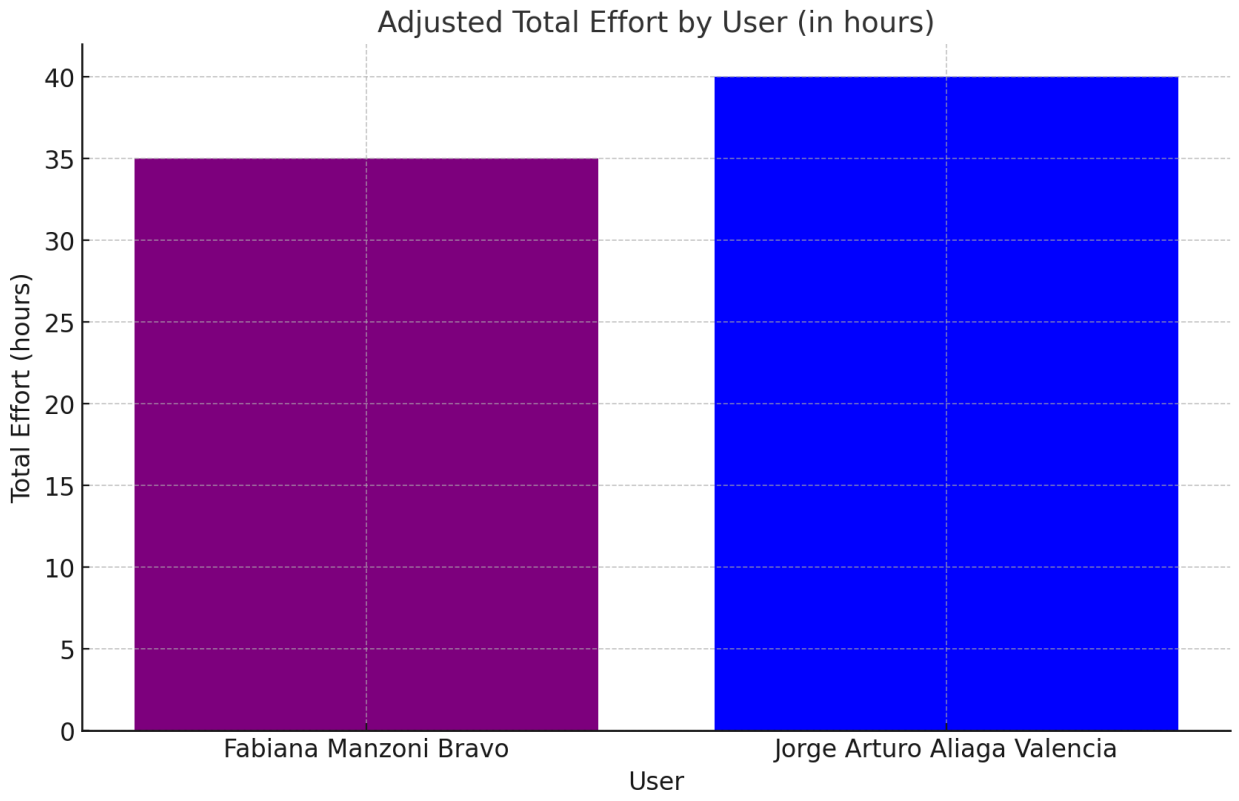
6. Burndown y BurnUp (Grafica de tareas y Datos de tareas)



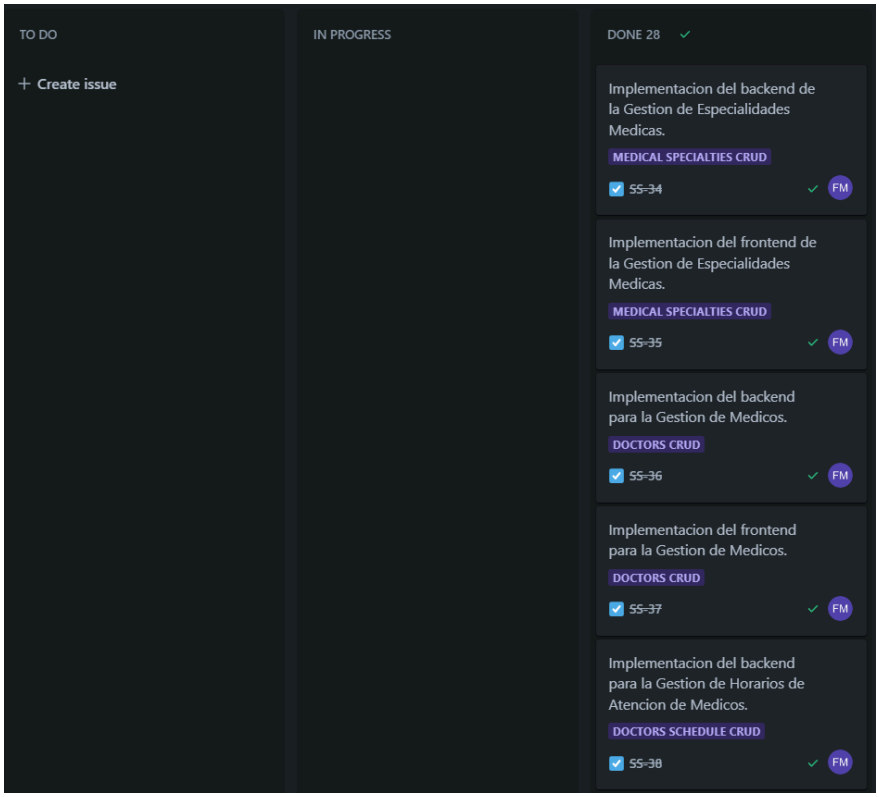


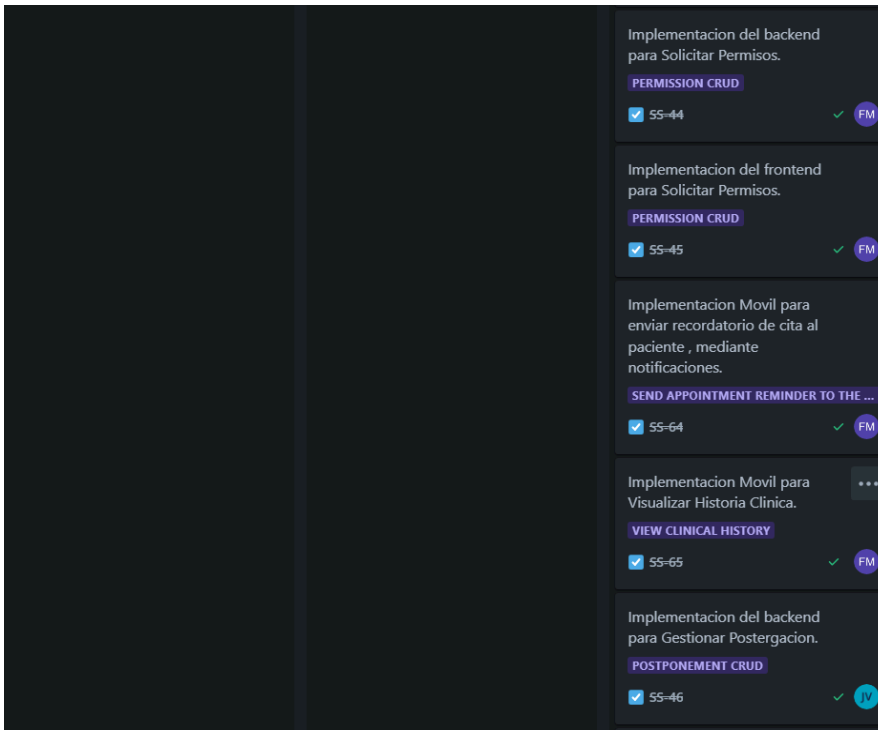
7. Grafica de esfuerzo y Datos de esfuerzo





8. Scrum TaskBoard (Backlog, to do, doing, done)





TO DO	IN PROGRESS	DONE 28
		Implementacion del frontend para Gestionar Postergacion. POSTPONEMENT CRUD ~~SS-47~~ JV
		Implementacion del backend para Programar Receta Medica. PRESCRIPTION CRUD ~~SS-48~~ JV
		Implementacion del frontend para Programar Receta Medica. PRESCRIPTION CRUD ~~SS-49~~ JV
		Implementacion del backend para Programar Laboratorio. SCHEDULE LAB ~~SS-50~~ JV
		Implementacion del frontend para Programar Laboratorio. SCHEDULE LAB ~~SS-51~~ JV
		Implementacion del backend para Gestionar Laboratorio. LAB CRUD ~~SS-52~~ JV

TO DO	IN PROGRESS	DONE 28
		Implementacion del frontend para Gestionar Laboratorio. LAB CRUD ~~SS-53~~ JV
		Implementacion del backend para Gestionar Medicamentos. MANAGE MEDICATIONS ~~SS-54~~ JV
		Implementacion del frontend para Gestionar Medicamentos. MANAGE MEDICATIONS ~~SS-55~~ JV
		Implementacion del backend para Gestionar Historia Clinica. MANAGE CLINICAL HISTORY ~~SS-56~~ JV
		Implementacion del frontend para Gestionar Historia Clinica. MANAGE CLINICAL HISTORY ~~SS-57~~ JV
		Implementacion movil para Visualizar la Reserva. VIEW RESERVATION ~~SS-62~~ JV

Sprint 3

1. Sprint Planning

1.1. Objetivos del Sprint

El objetivo del sprint es implementar funcionalidades avanzadas del sistema que permitan una administración completa de reportes, bitácoras, evaluaciones y pagos, así como garantizar el acceso seguro y trazable a la información médica. Durante este sprint, se desarrollará la gestión de reportes personalizados para facilitar la toma de decisiones, la visualización de la bitácora para rastrear acciones del sistema, la evaluación de médicos por parte de los pacientes, y la exportación de historias clínicas en formatos estándar. Además, se incluirá la gestión de pagos de seguros, asegurando un manejo eficiente y transparente de las transacciones realizadas por los asegurados. Estas funcionalidades garantizarán un control integral y eficiente del sistema HCE.

1.1.1 Casos de uso

- **CU20:** Gestionar Reportes Personalizados
- **CU21:** Visualizar Bitácora
- **CU22:** Evaluación de Médicos
- **CU24:** Exportar historia clínica
- **CU25:** Gestión de Pagos de Seguros
- **CU26:** Gestión de Método de Pago
- **CU27:** Visualizar Historial de Pagos

1.2. Historias de usuario

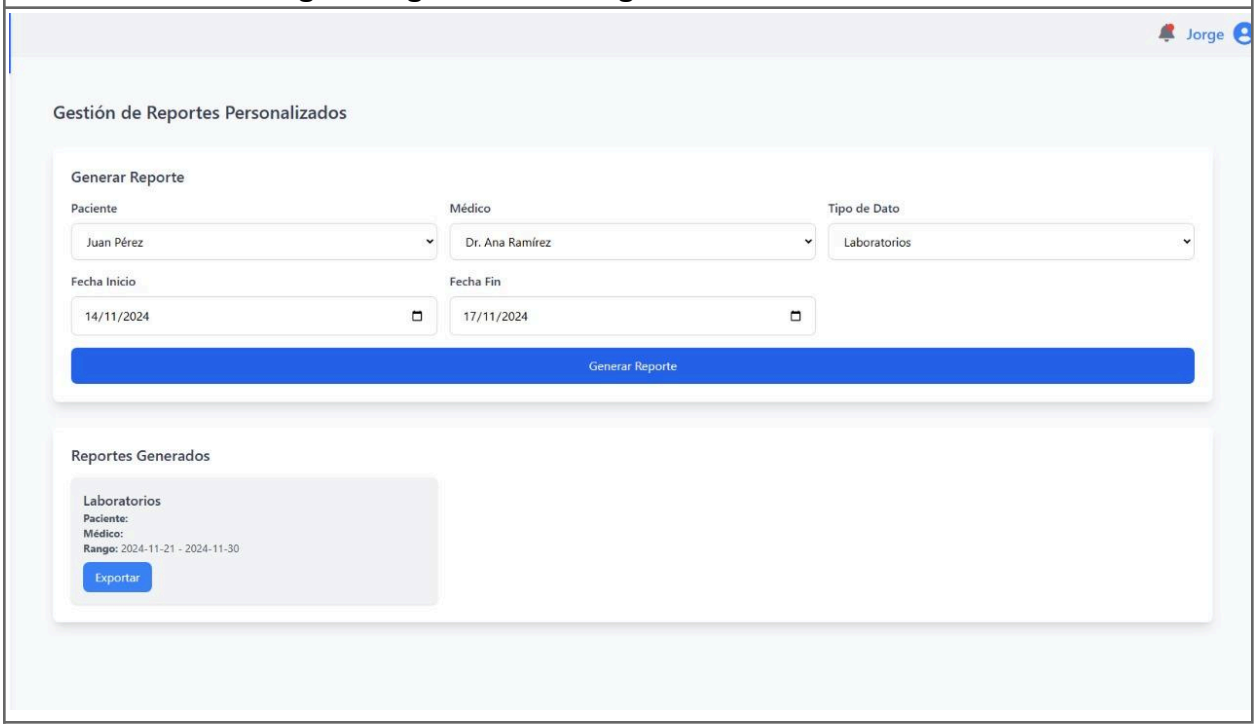
CU20: Gestionar Reportes Personalizados

Gestionar Reportes Personalizados		
ID:	**Descripción: Gestion de Reportes Personalizados en el sistema**	
Prioridad: Alta		**Estimación PHU: 6hr**

Criterios de aceptación:

- El administrador puede seleccionar criterios de búsqueda como rango de fechas, médico, especialidad y asegurado.
- Los reportes generados muestran información detallada, completa y actualizada según los filtros aplicados.
- Los reportes se pueden visualizar en pantalla y descargar en formatos PDF o Excel.

Desarrollador a cargo: Jorge Arturo Aliaga Valencia



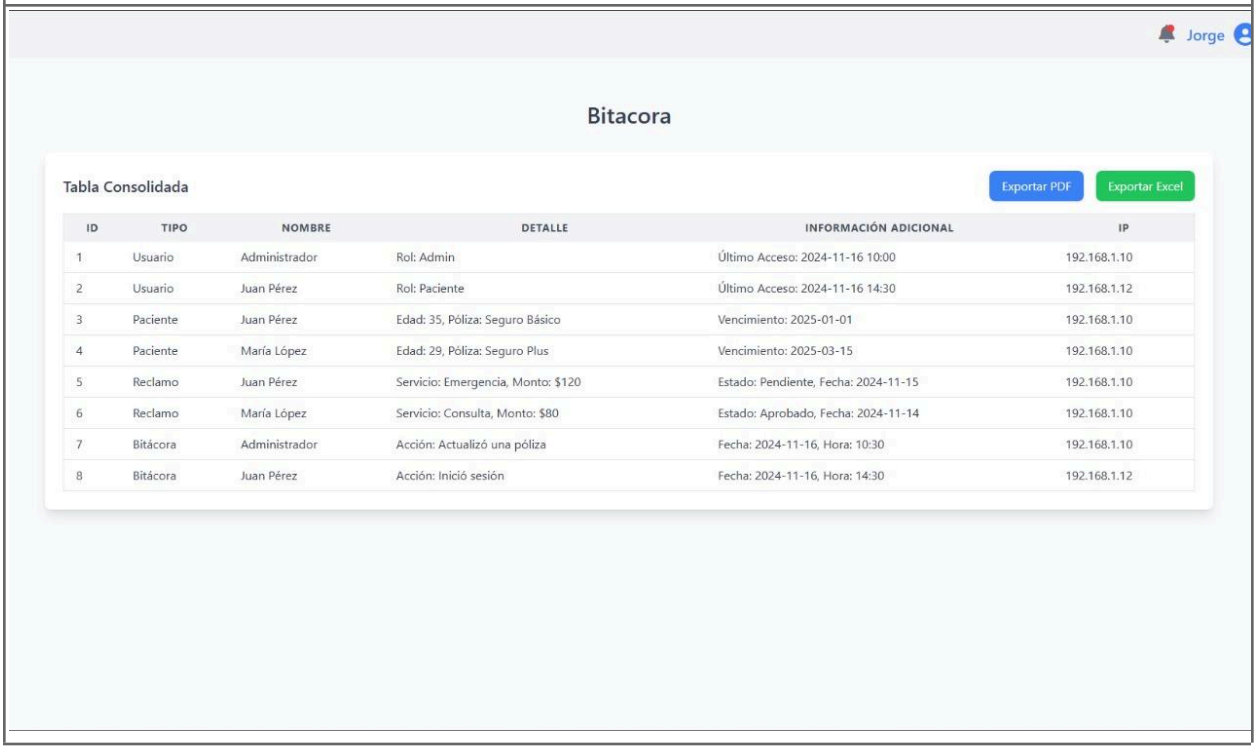
CU21: Visualizar Bitácora

Visualizar Bitácora	
ID:	**Descripción: Visualizar Bitácora en el sistema**
Prioridad: Alta	**Estimación PHU: 6hr**

Criterios de aceptación:

- Solo los administradores pueden acceder al módulo de bitácora.
- La bitácora muestra eventos del sistema con detalles como usuario, acción realizada, fecha, hora e IP.
- Los datos en la bitácora son ordenados cronológicamente y se pueden filtrar por criterios como tipo de acción, fecha y usuario.

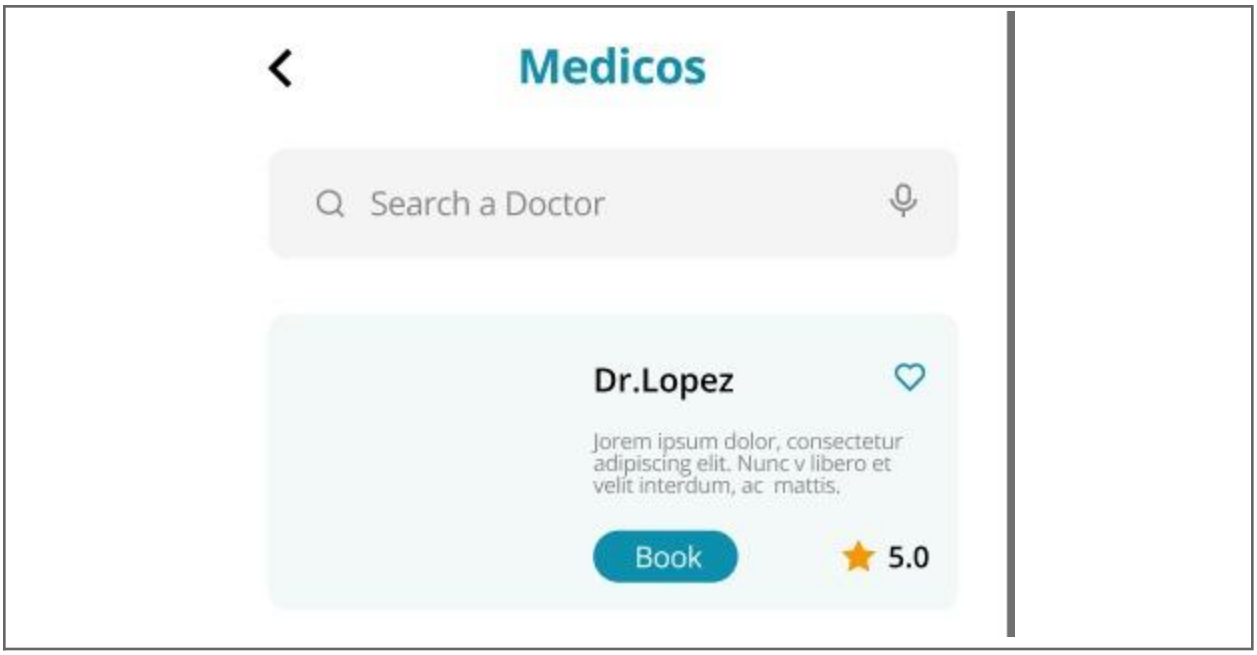
Desarrollador a cargo: Jorge Arturo Aliaga Valencia



ID	TIPO	NOMBRE	DETALLE	INFORMACIÓN ADICIONAL	IP
1	Usuario	Administrador	Rol: Admin	Último Acceso: 2024-11-16 10:00	192.168.1.10
2	Usuario	Juan Pérez	Rol: Paciente	Último Acceso: 2024-11-16 14:30	192.168.1.12
3	Paciente	Juan Pérez	Edad: 35, Póliza: Seguro Básico	Vencimiento: 2025-01-01	192.168.1.10
4	Paciente	María López	Edad: 29, Póliza: Seguro Plus	Vencimiento: 2025-03-15	192.168.1.10
5	Reclamo	Juan Pérez	Servicio: Emergencia, Monto: $120	Estado: Pendiente, Fecha: 2024-11-15	192.168.1.10
6	Reclamo	María López	Servicio: Consulta, Monto: $80	Estado: Aprobado, Fecha: 2024-11-14	192.168.1.10
7	Bitácora	Administrador	Acción: Actualizó una póliza	Fecha: 2024-11-16, Hora: 10:30	192.168.1.10
8	Bitácora	Juan Pérez	Acción: Inició sesión	Fecha: 2024-11-16, Hora: 14:30	192.168.1.12

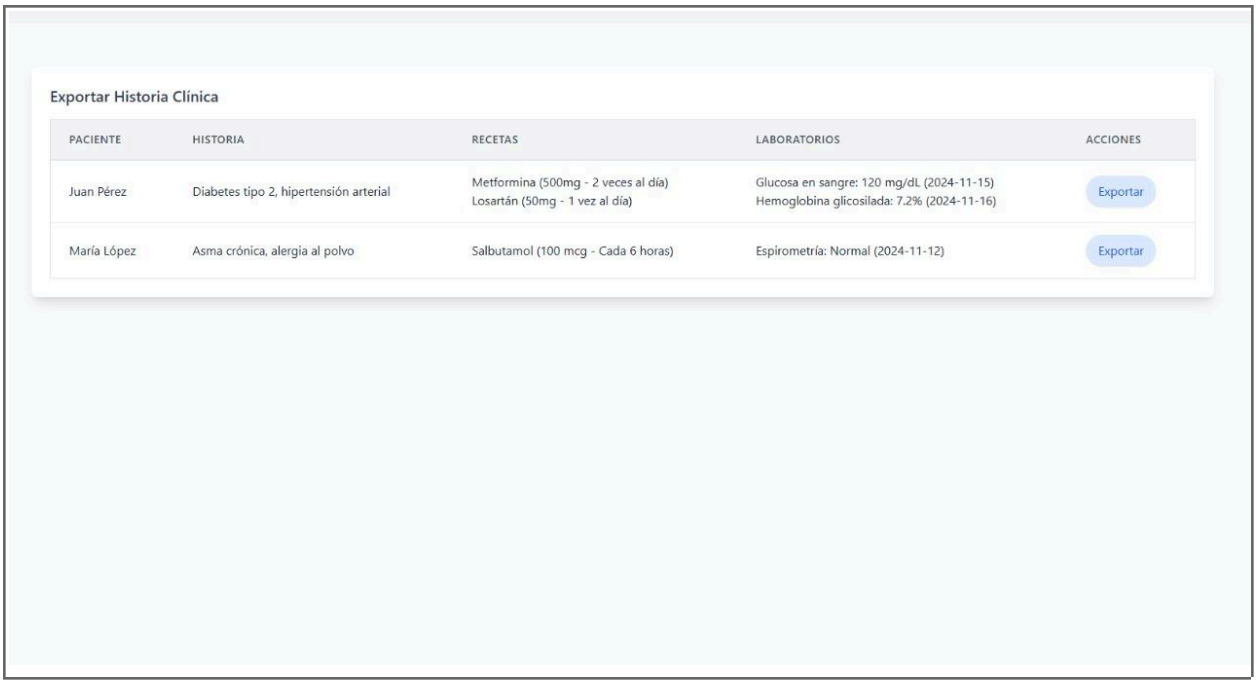
CU22: Evaluación de Médicos

Evaluación de Médicos	
ID:	**Descripción: Evaluación de Médicos en el sistema**
Prioridad: Alta	**Estimación PHU: 3hr**
Criterios de aceptación: ● Los pacientes pueden evaluar a los médicos después de una consulta, utilizando un sistema de puntuación o comentarios.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



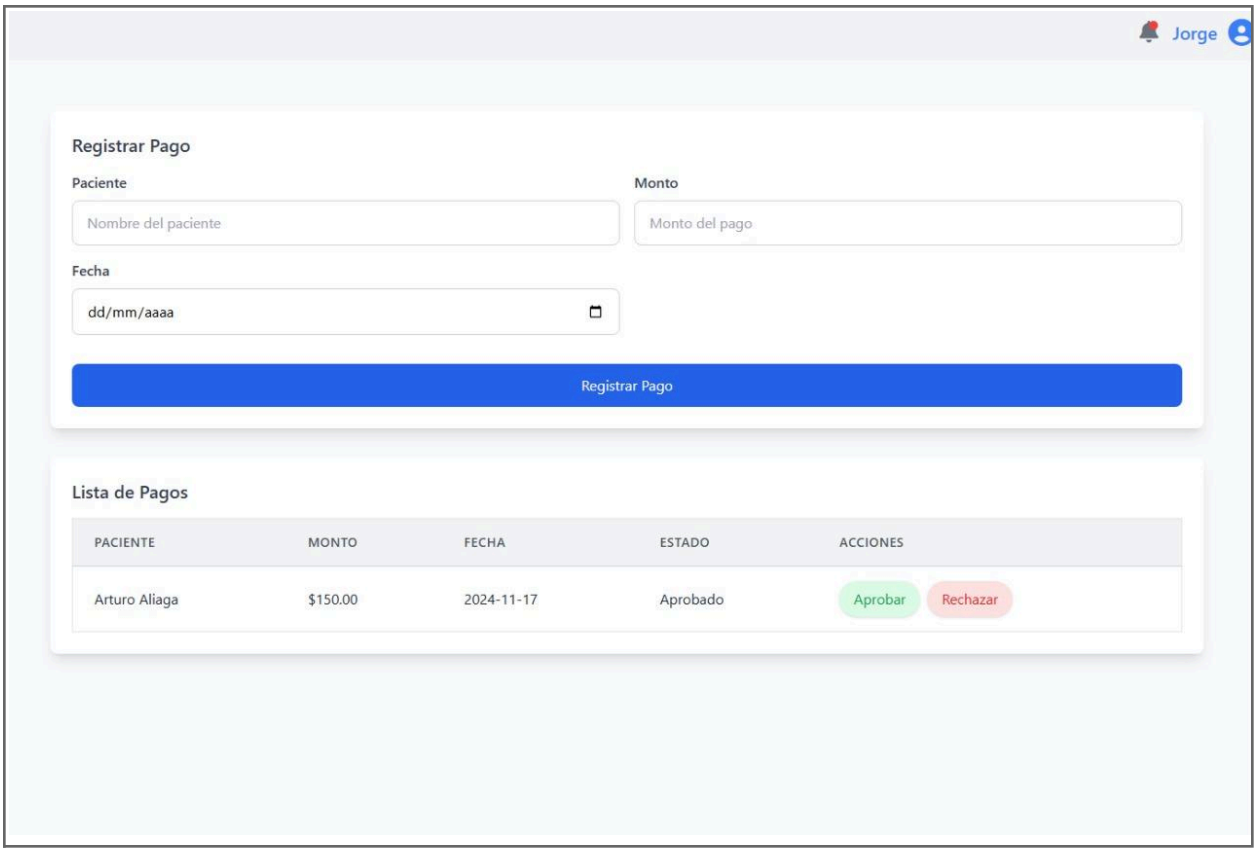
CU24: Exportar historia clínica

Exportar historia clínica	
ID:	**Descripción: Exportar historia clínica en el sistema**
Prioridad: Alta	**Estimación PHU: 3hr**
Criterios de aceptación: ● .Los médicos y pacientes pueden solicitar la exportación de la historia clínica del paciente. ● La historia clínica exportada incluye datos actualizados como diagnósticos, tratamientos, exámenes y recetas. ● Los archivos exportados están disponibles en formatos PDF o XML, según sea necesario.	
Desarrollador a cargo: Jorge Arturo Aliaga Valencia	



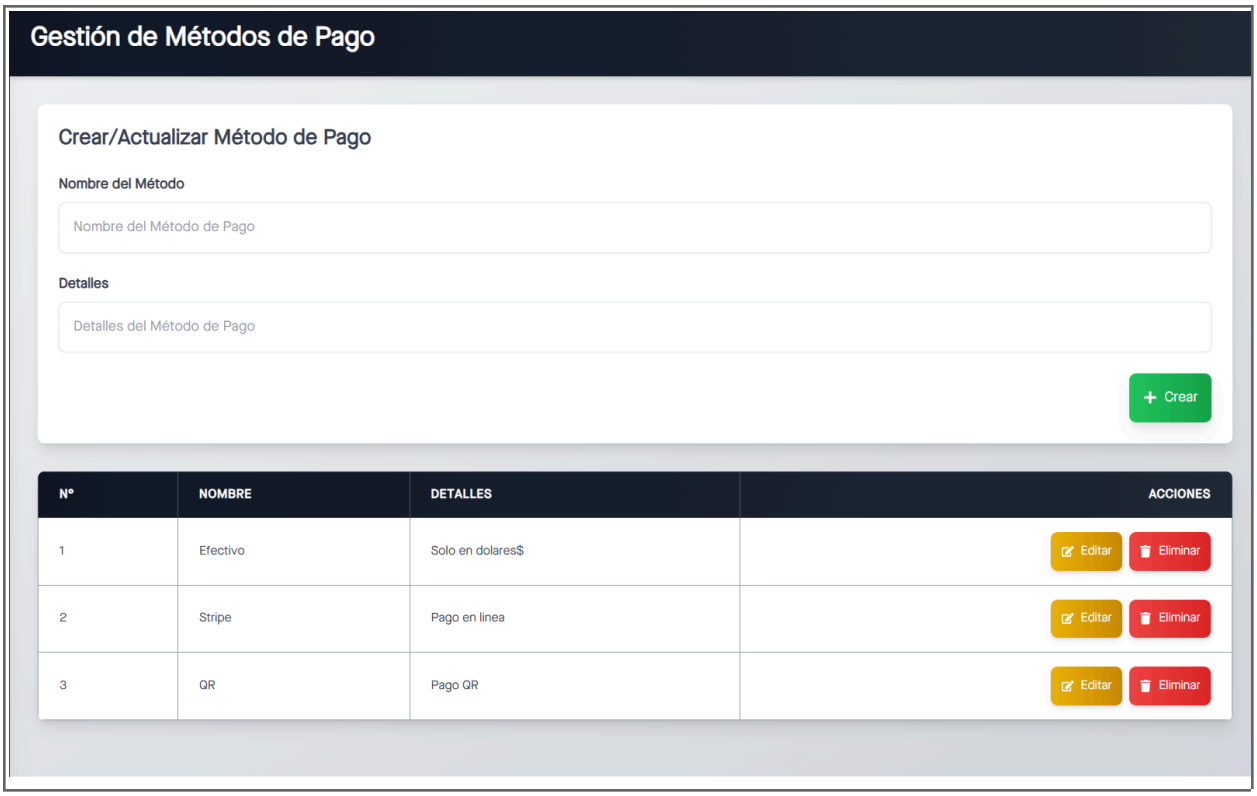
CU25: Gestión de Pagos de Seguros

Gestión de Pagos de Seguros	
ID:	**Descripción: Gestión de Pagos de Seguros en el sistema**
Prioridad: Alta	**Estimación PHU: 8hr**
Criterios de aceptación: ● Los asegurados pueden consultar sus pagos pendientes y realizar pagos a través de los métodos habilitados (tarjeta de crédito, transferencia bancaria, etc.). ● El sistema valida los detalles del pago y registra correctamente el monto, método y fecha del pago. ● Los pagos realizados actualizan automáticamente el estado del seguro como "vigente". ● Los administradores pueden visualizar un historial de pagos por asegurado o periodo y generar reportes.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



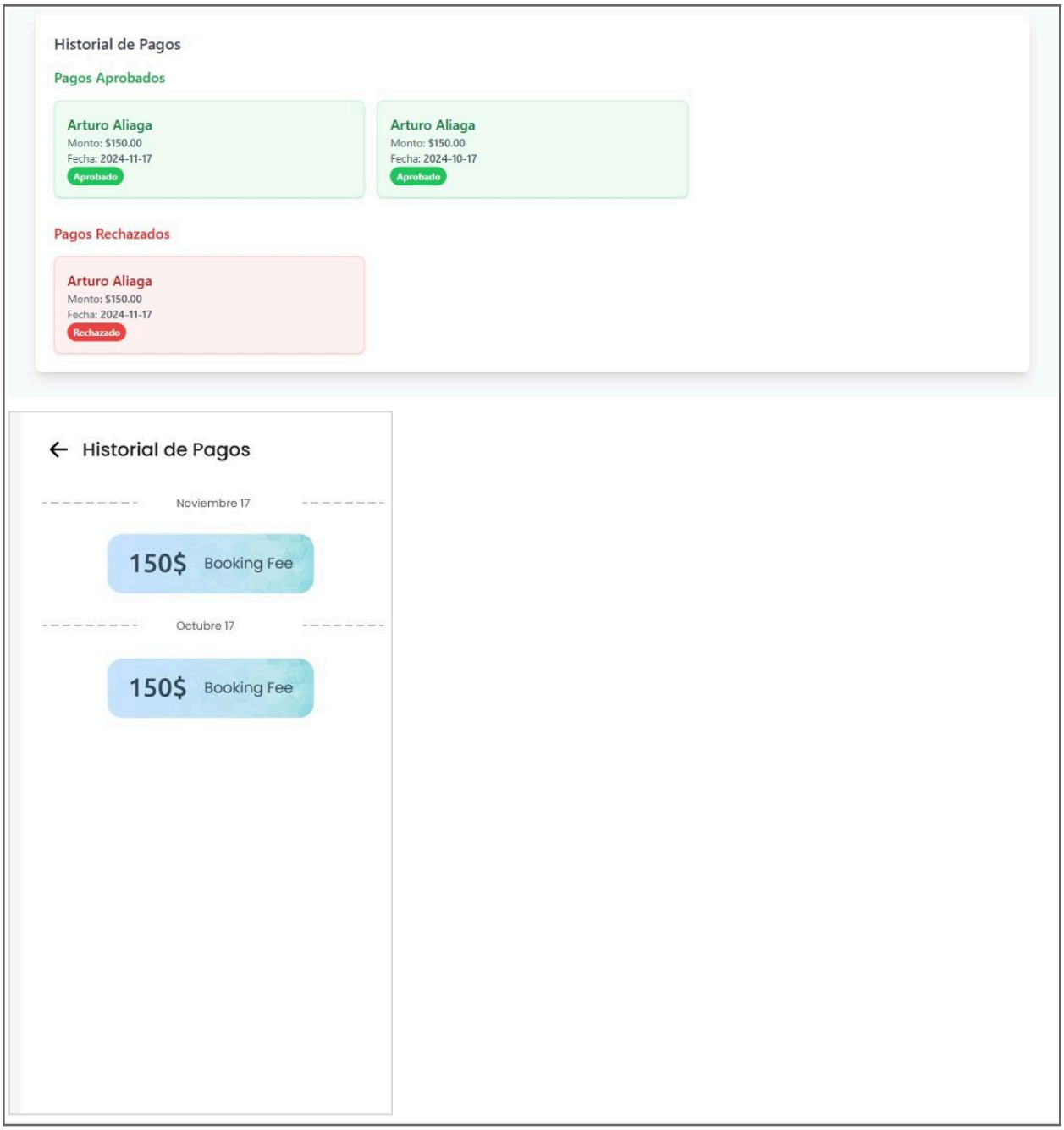
CU26: Gestión de Método de Pago

Gestión de Método de Pago	
ID:	**Descripción: Gestión de Método de Pago en el sistema**
Prioridad: Alta	**Estimación PHU: 8hr**
Criterios de aceptación: ● Permitir al administrador registrar nuevos métodos de pago con información válida (nombre, descripción, tipo). ● Mostrar un mensaje de confirmación al registrar, editar o eliminar métodos de pago.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



CU27: Visualizar Historial de Pagos

Visualizar Historial de Pagos	
ID:	**Descripción: Visualizar Historial de Pagos en el sistema**
Prioridad: Alta	**Estimación PHU: 8hr**
Criterios de aceptación: ● Permitir al paciente y al administrador consultar el historial de pagos con información detallada (fecha, monto, método, estado). ● Validar que los filtros aplicados (por fecha, estado, método de pago) funcionen correctamente. ● Asegurar que los datos mostrados en el historial sean precisos y estén actualizados. ● Restringir el acceso al historial de pagos para que solo el paciente correspondiente o el administrador puedan visualizarlo. ● Mostrar un mensaje claro en caso de que no existan pagos registrados para los criterios seleccionados.	
Desarrollador a cargo: Fabiana Manzoni Bravo	



1.3. Contexto del sistema

1.4. Sprint Backlog

Sprint Backlog	
Numero de Sprint: 3	**Tiempo programado: 2 semanas**

Objetivo: El objetivo del sprint es desarrollar la gestión de reportes personalizados, bitácoras, evaluaciones de médicos, exportación de historias clínicas y pagos de seguros, asegurando funcionalidades avanzadas para un control eficiente y transparente en el sistema HCE.					
Product Owner: No establecido					
Fecha de inicio: 08/11/2024			**Fecha de finalización: 17/11/2024**		
Id	**Tarea**	**Tipo**	**Estimación (hr)**	**Responsable**	**Estado**
1	Diseñar interfaz para Gestionar Reportes Personalizados	Desarrollo	3hr	Jorge Arturo Aliaga Valencia	Terminado
2	Implementar Gestionar Reportes Personalizados	Desarrollo	2hr	Jorge Arturo Aliaga Valencia	Terminado
3	Realizar pruebas de Gestionar Reportes Personalizados	Desarrollo	2hr	Jorge Arturo Aliaga Valencia	Terminado
4	Diseñar interfaz para Visualizar Bitácora	Desarrollo	4hr	Jorge Arturo Aliaga Valencia	Terminado
5	Implementar Visualizar Bitácora	Desarrollo	3hr	Jorge Arturo Aliaga Valencia	Terminado
6	Realizar pruebas de Visualizar Bitácora	Desarrollo	2hr	Jorge Arturo Aliaga Valencia	Terminado
7	Diseñar interfaz para Evaluación de Médicos	Desarrollo	6hr	Fabiana Manzoni Bravo	Terminado
8	Implementar Evaluación de Médicos	Desarrollo	5hr	Fabiana Manzoni Bravo	Terminado
9	Realizar pruebas de Evaluación de Médicos	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado
10	Diseñar interfaz Exportar historia clínica	Desarrollo	2hr	Jorge Arturo Aliaga Valencia	Terminado

11	Implementar Exportar historia clínica	Desarrollo	3hr	Jorge Arturo Aliaga Valencia	Terminado
12	Realizar pruebas de Exportar historia clínica	Desarrollo	2hr	Jorge Arturo Aliaga Valencia	Terminado
13	Diseñar interfaz para Gestión de Pagos de Seguros	Desarrollo	6hr	Fabiana Manzoni Bravo	Terminado
14	Implementar Gestión de Pagos de Seguros	Desarrollo	5hr	Fabiana Manzoni Bravo	Terminado
15	Realizar pruebas de Gestión de Pagos de Seguros	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado
16	Diseñar interfaz para Gestión de Método de Pago	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado
17	Implementar Gestión de Método de Pago	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado
18	Realizar pruebas de Gestión de Método de Pago	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado
19	Diseñar interfaz para Visualizar Historial de Pagos	Desarrollo	3hr	Fabiana Manzoni Bravo	Terminado
20	Implementar Visualizar Historial de Pagos	Desarrollo	3hr	Fabiana Manzoni Bravo	Terminado
21	Realizar pruebas de Visualizar Historial de Pagos	Desarrollo	2hr	Fabiana Manzoni Bravo	Terminado

1.5. Equipo SCRUM

Integrantes	*Rol*	*Entorno*	*Caso de Uso*	*Tareas*	*Nota*
Jorge Arturo Aliaga Valencia	Developer	**Web:** **Movil:**	**CU20:** Gestionar Reportes Personalizados **CU21:** Visualizar Bitácora **CU24:** Exportar historia clínica **CU22:** Evaluación de Médicos **CU24:** Exportar historia clínica	1.Diseñar interfaz para Gestionar Reportes Personalizados 2.Implementar Gestionar Reportes Personalizados 3.Realizar pruebas de Gestionar Reportes Personalizados 4.Diseñar interfaz para Visualizar Bitácora 5.Implementar Visualizar Bitácora 6.Realizar pruebas de Visualizar Bitácora 7.Diseñar interfaz Exportar historia clínica 8.Implementar Exportar historia clínica 9.Realizar pruebas de Exportar historia clínica	100
Fabiana Manzoni Bravo	Developer	**Web:** **Movil:**	**CU22:** Evaluación de Médicos **CU25:** Gestión de Pagos de Seguros **CU26:** Gestión de Método de Pago **CU27:** Visualizar Historial de Pagos	10.Diseñar interfaz para Gestión de Pagos de Seguros 11.Implementar Gestión de Pagos de Seguros 12.Realizar pruebas de Gestión de Pagos de Seguros 13.Diseñar interfaz para Gestión de Método de Pago 14.Implementar Gestión de Método de Pago 15.Realizar pruebas de Gestión de Método de Pago 16.Diseñar interfaz para Visualizar Historial de Pagos	95

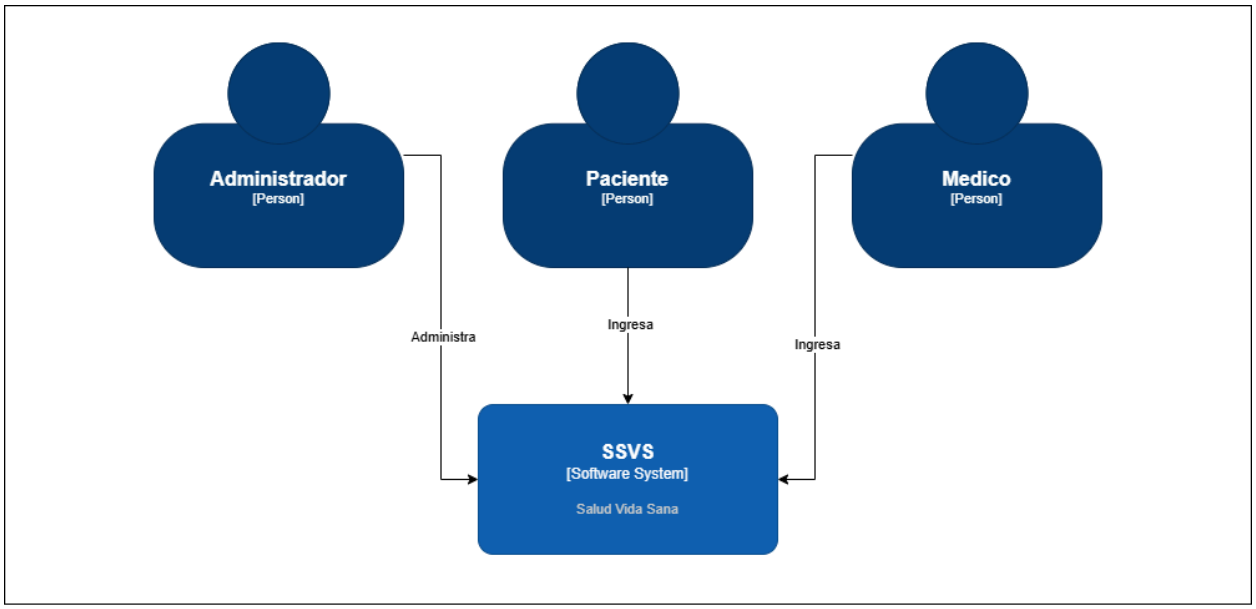
				17.Implementar Visualizar Historial de Pagos 18.Realizar pruebas de Visualizar Historial de Pagos	

2. Proceso/patrón de desarrollo por Historia de Usuario

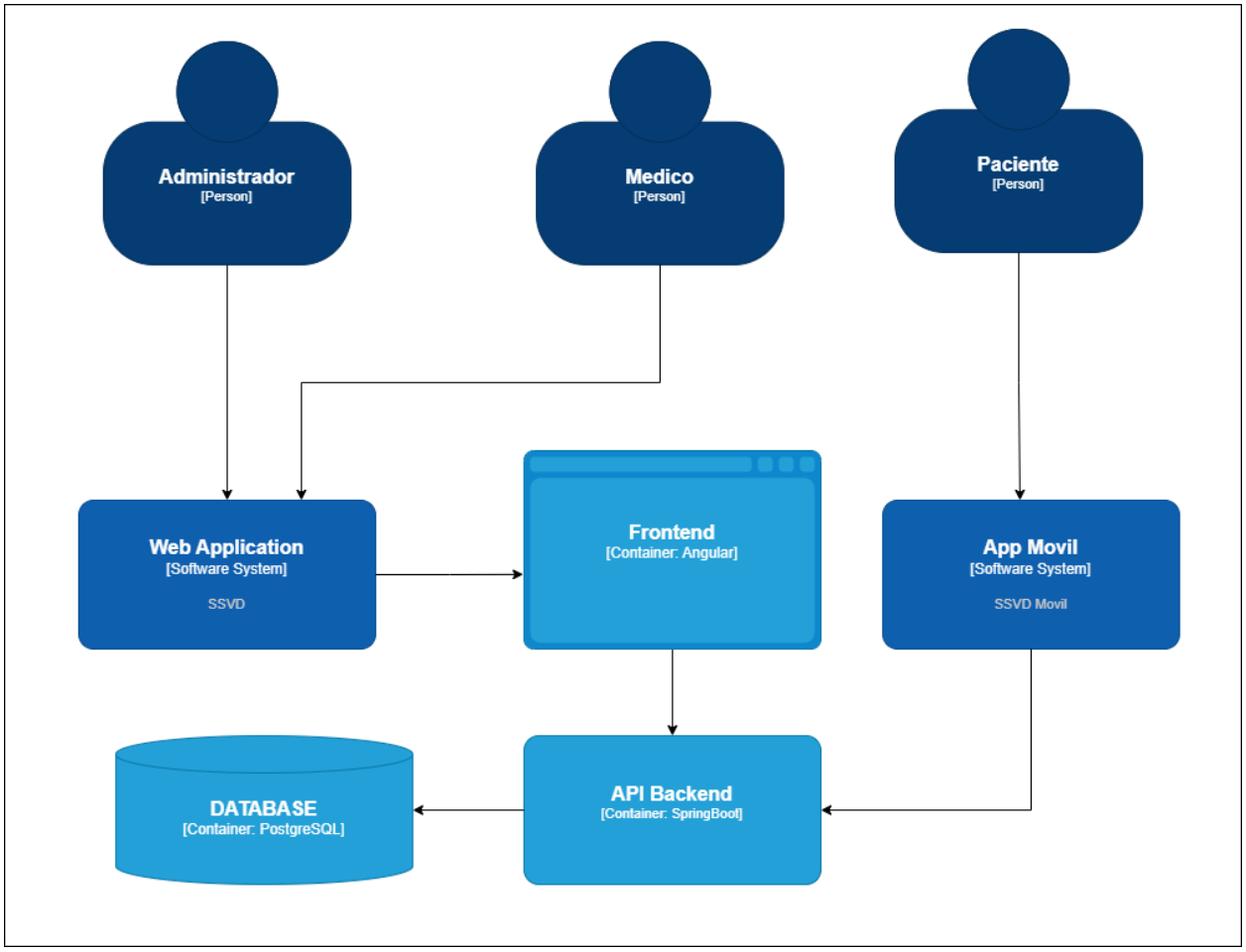
2.1. Diseño

2.1.1. Diseñar de la arquitectura

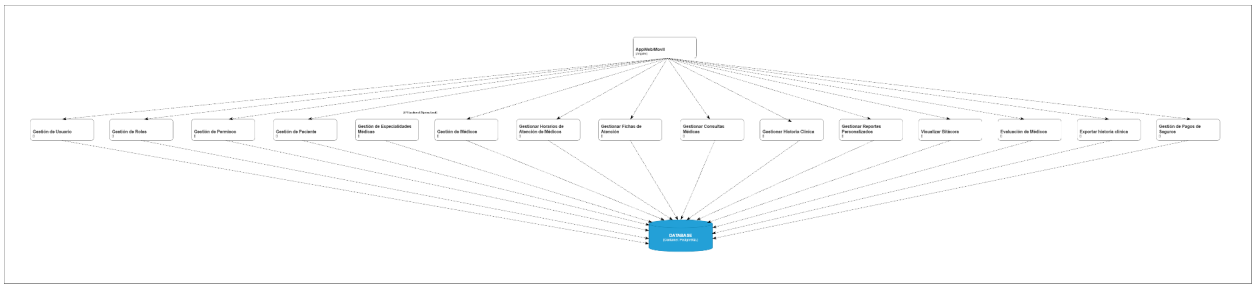
a) Diagrama del Contexto del Sistema



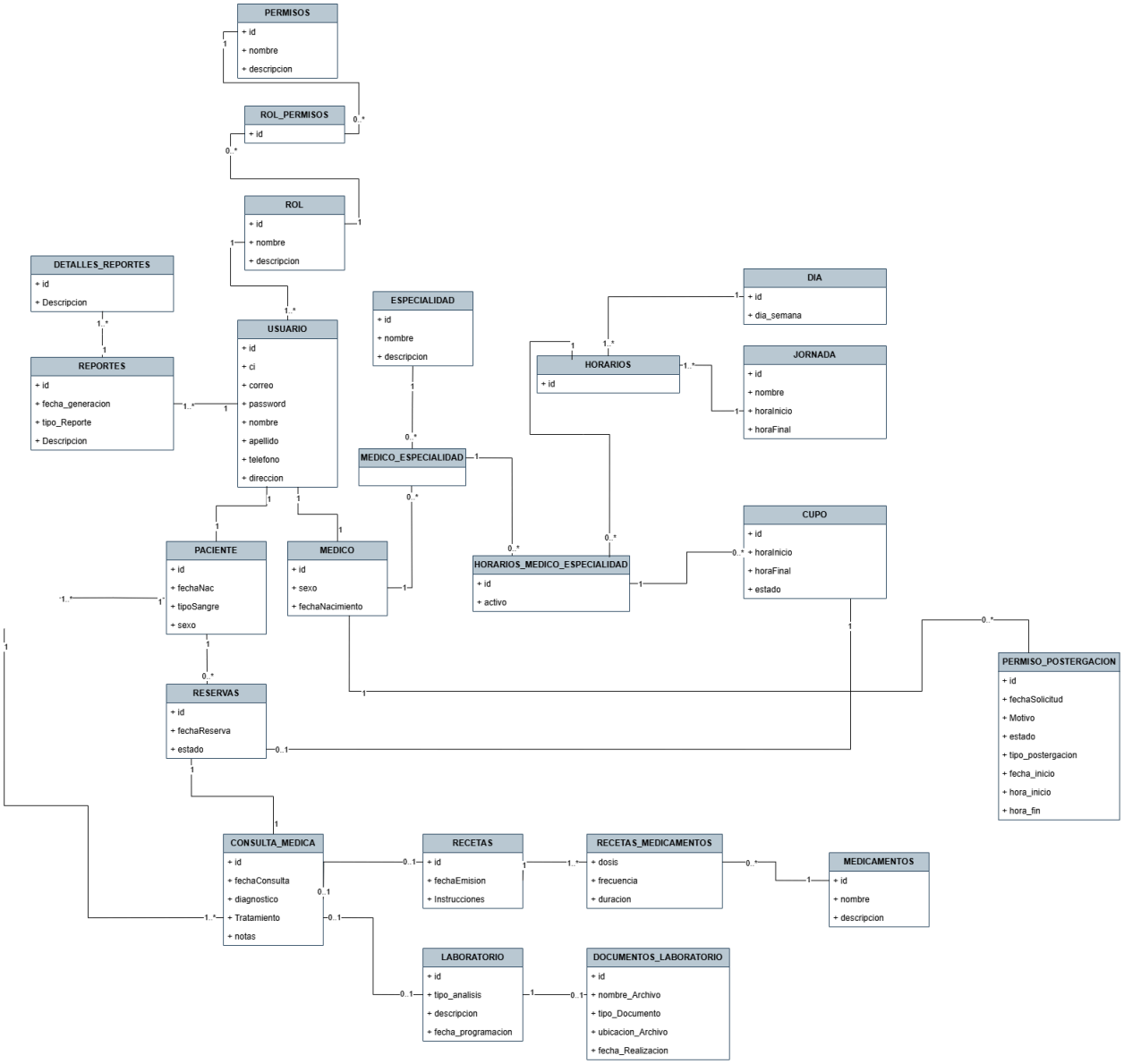
b) Diagrama de Contenedores



c) Diagrama de Componentes

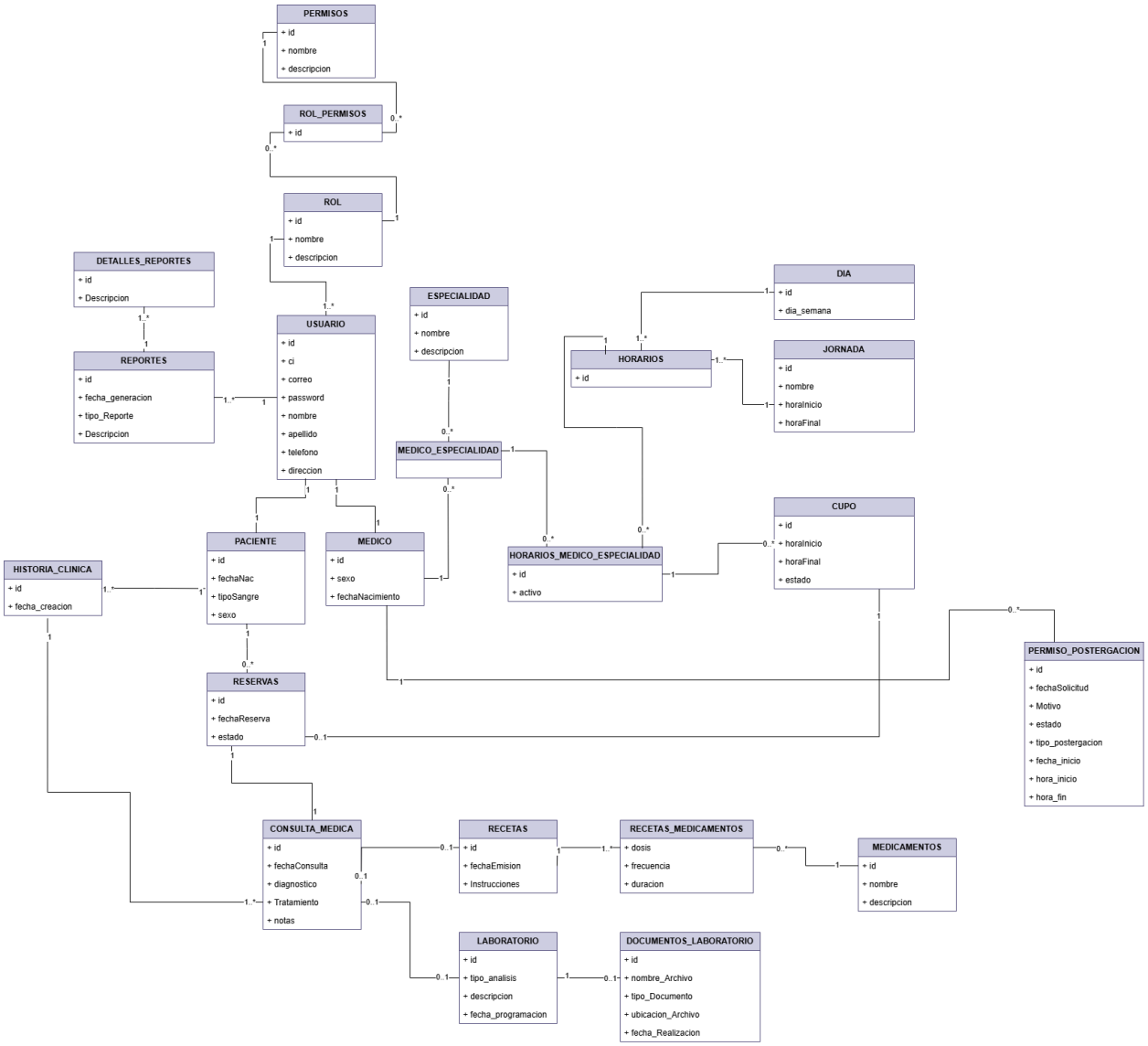


d) Diagrama de Código



2.1.2. Diseño de datos

a) Diagrama de clases



b) Script

```
-- Database: railway

-- DROP DATABASE IF EXISTS railway;

CREATE DATABASE railway
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
```

```
    LC_COLLATE = 'en_US.utf8'
    LC_CTYPE = 'en_US.utf8'
    LOCALE_PROVIDER = 'libc'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;


CREATE TABLE Roles (
    rol_id SERIAL PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE NOT NULL,
    descripcion TEXT
);

CREATE TABLE Permisos (
    permiso_id SERIAL PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE NOT NULL,
    descripcion TEXT
);

CREATE TABLE Roles_Permisos (
    rol_id INT REFERENCES Roles(rol_id) ON DELETE CASCADE,
    permiso_id INT REFERENCES Permisos(permiso_id) ON DELETE CASCADE,
    PRIMARY KEY (rol_id, permiso_id)
);

CREATE TABLE Usuarios (
    usuario_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    contraseña TEXT NOT NULL,
    telefono VARCHAR(20),
    direccion TEXT,
    rol_id INT REFERENCES Roles(rol_id) ON DELETE SET NULL
);

CREATE TABLE Pacientes (
    paciente_id SERIAL PRIMARY KEY,
    usuario_id INT REFERENCES Usuarios(usuario_id) ON DELETE CASCADE,
```

```
    fecha_nacimiento DATE,
    genero VARCHAR(10)
);

CREATE TABLE Especialidades (
    especialidad_id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) UNIQUE NOT NULL,
    descripcion TEXT
);

CREATE TABLE Medicos (
    medico_id SERIAL PRIMARY KEY,
    usuario_id INT REFERENCES Usuarios(usuario_id) ON DELETE CASCADE,
    genero VARCHAR(50),
    fecha_nacimiento DATE
);

CREATE TABLE Medicos_Especialidades (
    medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
    especialidad_id INT REFERENCES Especialidades(especialidad_id) ON DELETE
CASCADE,
    PRIMARY KEY (medico_id, especialidad_id)
);

CREATE TABLE Horarios_Medicos (
    horario_id SERIAL PRIMARY KEY,
    medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
    dia_semana VARCHAR(20) NOT NULL,
    hora_inicio TIME NOT NULL,
    hora_fin TIME NOT NULL,
    cupo_maximo INT NOT NULL  -- Número máximo de reservas para el horario
);

CREATE TABLE Permisos_Postergacion (
    postergacion_id SERIAL PRIMARY KEY,
    medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
    fecha_solicitud TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    motivo TEXT,
    estado VARCHAR(20) DEFAULT 'Pendiente'
);
```

```
CREATE TABLE Reservas (
   reserva_id SERIAL PRIMARY KEY,
   paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
   medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
   fecha_reserva DATE NOT NULL,
   horario_id INT REFERENCES Horarios_Medicos(horario_id) ON DELETE SET
NULL,
   estado VARCHAR(20) DEFAULT 'Reservada',
   cupos_disponibles INT  -- Cupos restantes para este horario en el día específico
);

CREATE TABLE Historias_Clinicas (
   historia_id SERIAL PRIMARY KEY,
   paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
   fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Consultas_Medicas (
   consulta_id SERIAL PRIMARY KEY,
   reserva_id INT REFERENCES Reservas(reserva_id) ON DELETE CASCADE,
   medico_id INT REFERENCES Medicos(medico_id) ON DELETE CASCADE,
   paciente_id INT REFERENCES Pacientes(paciente_id) ON DELETE CASCADE,
   historia_id INT REFERENCES Historias_Clinicas(historia_id) ON DELETE
CASCADE,
   fecha_consulta TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   estado_paciente TEXT,
   diagnostico TEXT,
   tratamiento TEXT,
   notas TEXT
);

CREATE TABLE Recetas (
   receta_id SERIAL PRIMARY KEY,
   consulta_id INT REFERENCES Consultas_Medicas(consulta_id) ON DELETE
CASCADE,
   fecha_emision TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   instrucciones TEXT
);
```

```
CREATE TABLE Medicamentos (
   medicamento_id SERIAL PRIMARY KEY,
   nombre VARCHAR(100) NOT NULL,
   descripcion TEXT
);

CREATE TABLE Recetas_Medicamentos (
   receta_id INT REFERENCES Recetas(receta_id) ON DELETE CASCADE,
   medicamento_id INT REFERENCES Medicamentos(medicamento_id) ON DELETE
CASCADE,
   dosis VARCHAR(50),
   frecuencia VARCHAR(50),
   duracion INT,
   PRIMARY KEY (receta_id, medicamento_id)
);

CREATE TABLE Laboratorio (
   laboratorio_id SERIAL PRIMARY KEY,
   consulta_id INT REFERENCES Consultas_Medicas(consulta_id) ON DELETE
CASCADE,
   tipo_analisis VARCHAR(50) NOT NULL, -- Ej.: Hemograma, Rayos X, etc.
   descripcion TEXT,
   fecha_programacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  -- Fecha en
que se programó el análisis
);

CREATE TABLE Documentos_Laboratorio (
   documento_id SERIAL PRIMARY KEY,
   laboratorio_id INT REFERENCES Laboratorio(laboratorio_id) ON DELETE
CASCADE,
   nombre_archivo VARCHAR(100) NOT NULL,
   tipo_documento VARCHAR(50) NOT NULL, -- Ej.: Imagen, Informe
   ubicacion_archivo TEXT NOT NULL,
   fecha_realizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- Fecha en que
se realizó el examen o documento
);

CREATE TABLE Reportes (
   reporte_id SERIAL PRIMARY KEY,
   fecha_generacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
    tipo_reporte VARCHAR(50) NOT NULL,
    descripcion TEXT,
    generado_por INT REFERENCES Usuarios(usuario_id) ON DELETE SET NULL
);

CREATE TABLE Detalles_Reporte (
    detalle_id SERIAL PRIMARY KEY,
    reporte_id INT REFERENCES Reportes(reporte_id) ON DELETE CASCADE,
    descripcion TEXT
);
```

c) Mapeo

ROL

PK	
id	nombre

USUARIO

PK					FK
id	nombres	apellidos	email	contraseña	rol_id

DETALLE_ROL

PK	FK	FK
id	rol_id	permiso_id

PERMISO

PK	
id	nombre

MEDICO

PK			FK	FK
id	genero	fechaNacimiento	usuario_id	reserva_id

PERMISO_POSTERGACION

PK				FK
id	fechaSolicitud	motivo	estado	medico_id

ESPECIALIDAD

PK		
id	nombre	descripcion

MEDICO_ESPECIALIDAD

PK	FK	FK
id	medico_id	especialidad_id

PACIENTE

PK			FK
id	fechaNacimiento	sexo	usuario_id

HISTORIA_CLINICA

PK		FK
id	fechaCreacion	paciente_id

HORARIOS_MEDICOS

PK					FK
id	dia	horaInicio	horaFin	cupoMax	medico_id

RESERVA

PK				FK	FK
id	fechaReserva	estado	cuposDisponibles	horario_medico_id	paciente_id

CONSULTA_MEDICA

PK						FK
id	fechaConsulta	estadoPaciente	diagnostico	tratamiento	notas	reserva_id

LABORATORIO

PK				FK
id	tipoAnalisis	descripcion	fechaProgramacion	consulta_medica_id

DOCUMENTO_LABORATORIO

PK					FK
id	nombreArchivo	tipoDocumento	ubicacionArchivo	fechaRealizacion	laboratorio_id

RECETA

PK			FK
id	fechaEmision	instrucciones	consulta_medica_id

RECETA_MEDICAMENTO

PK				FK	FK
id	dosis	frecuencia	duracion	receta_id	medicamento_id

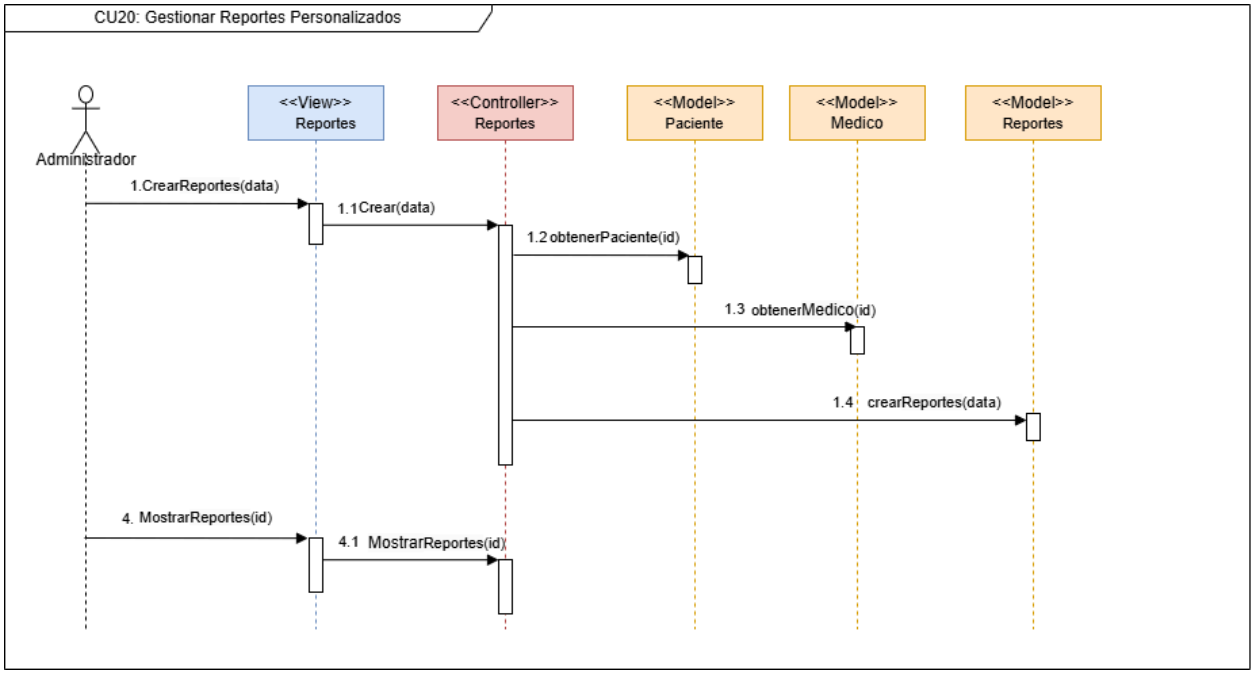
MEDICAMENTO

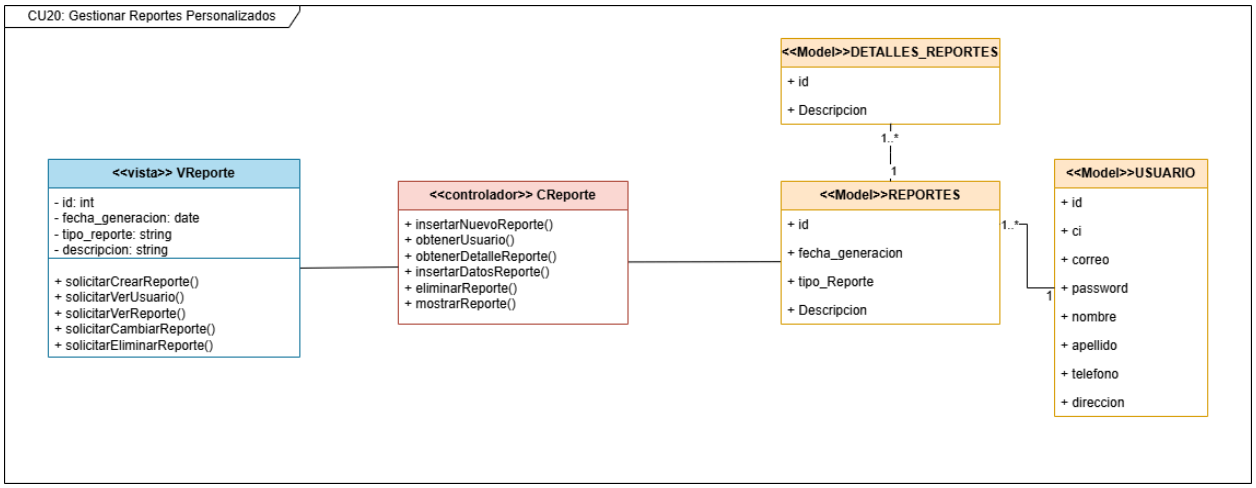
PK		
id	nombre	descripcion

2.1.3. Diseño de la lógica de negocio

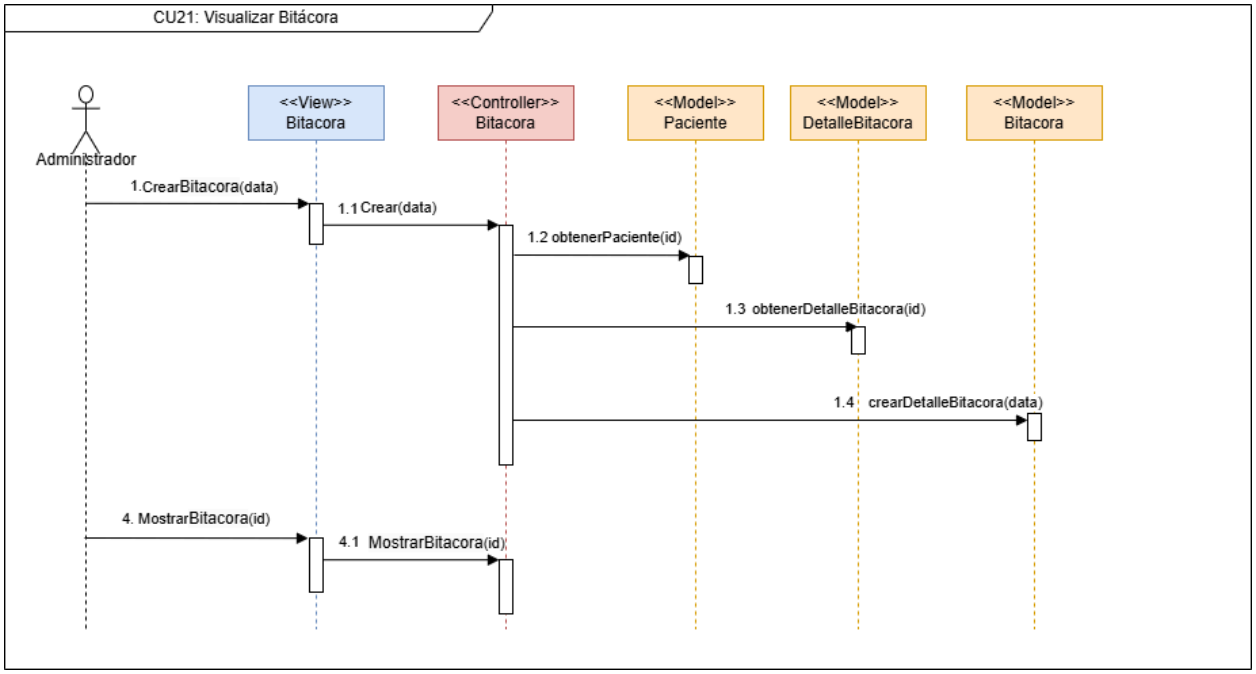
a) Diagrama de Secuencia y Diseño de Clase

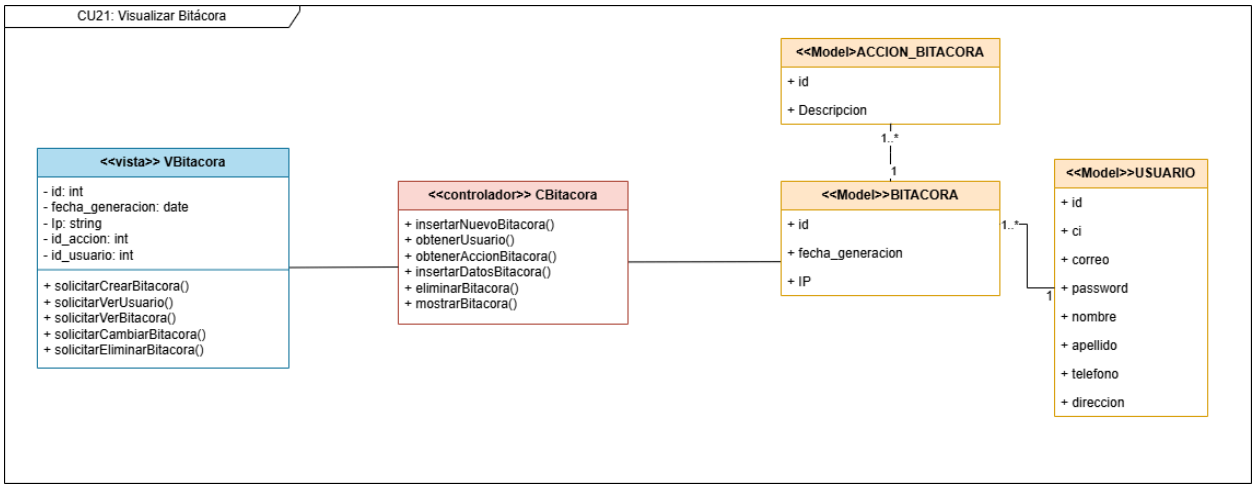
CU20: Gestionar Reportes Personalizados



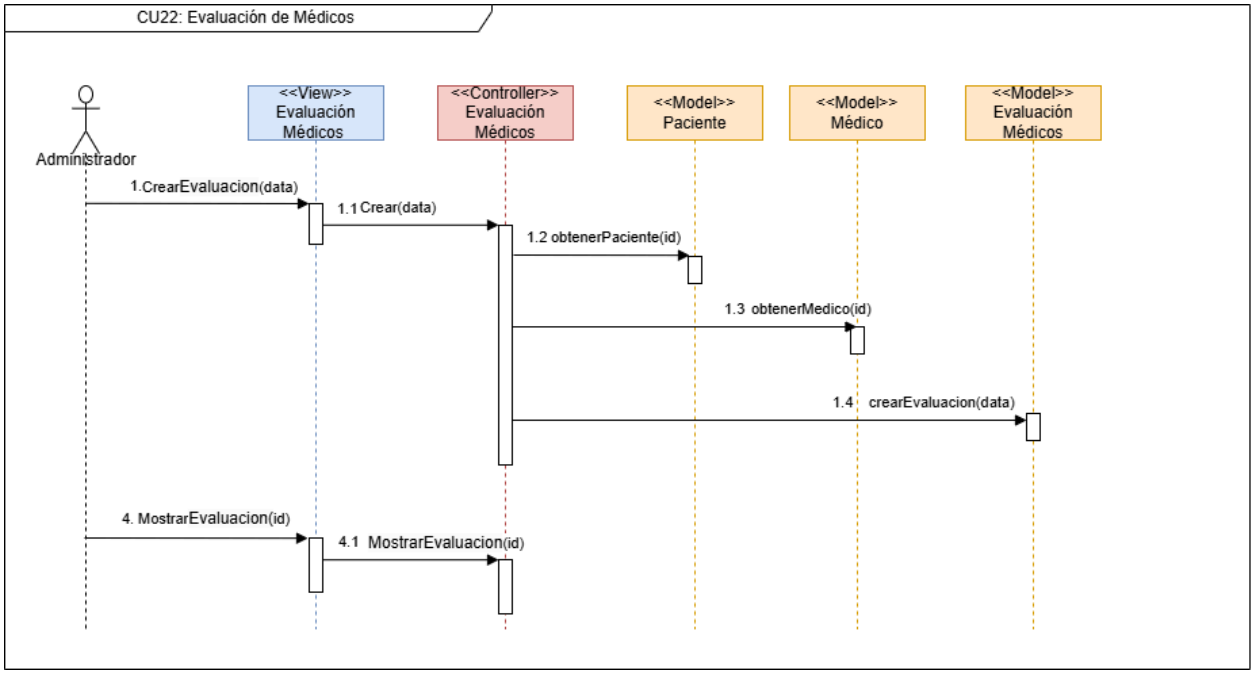


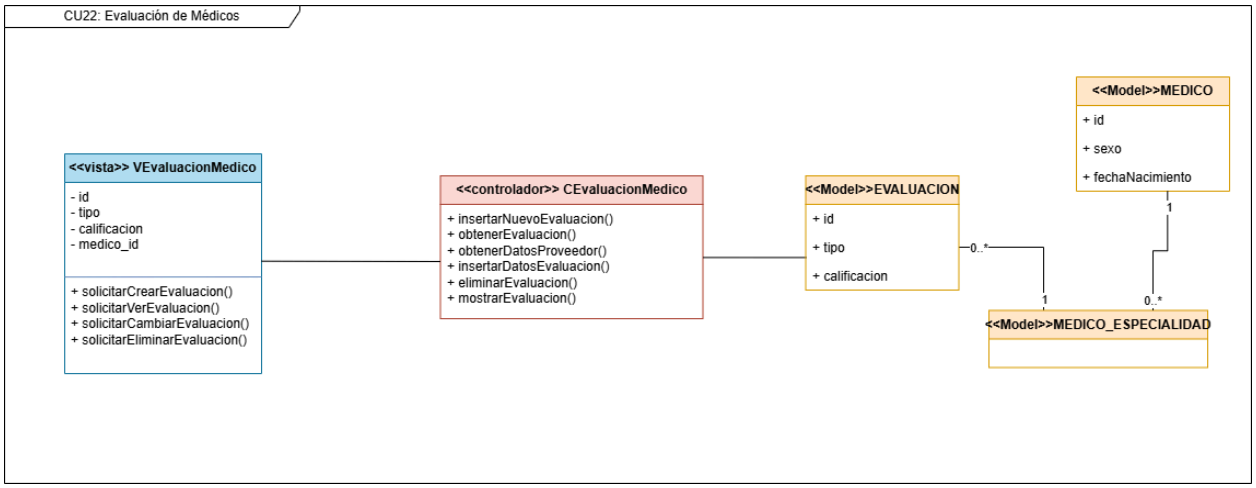
CU21: Visualizar Bitácora



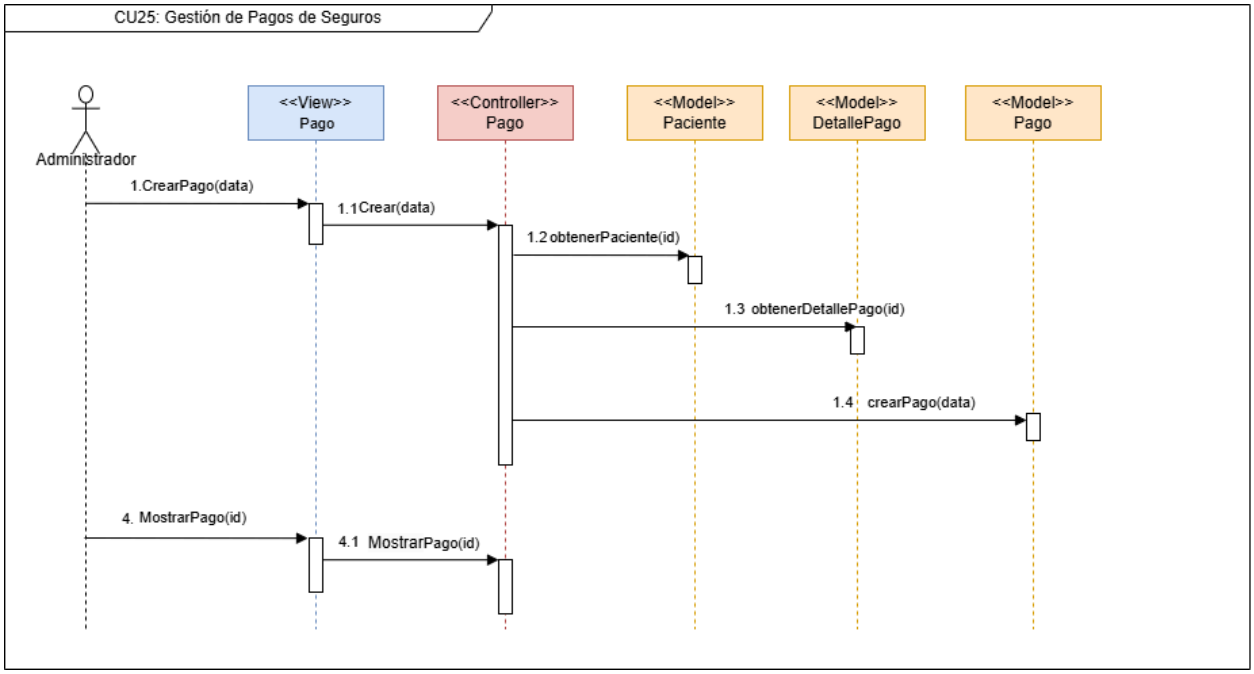


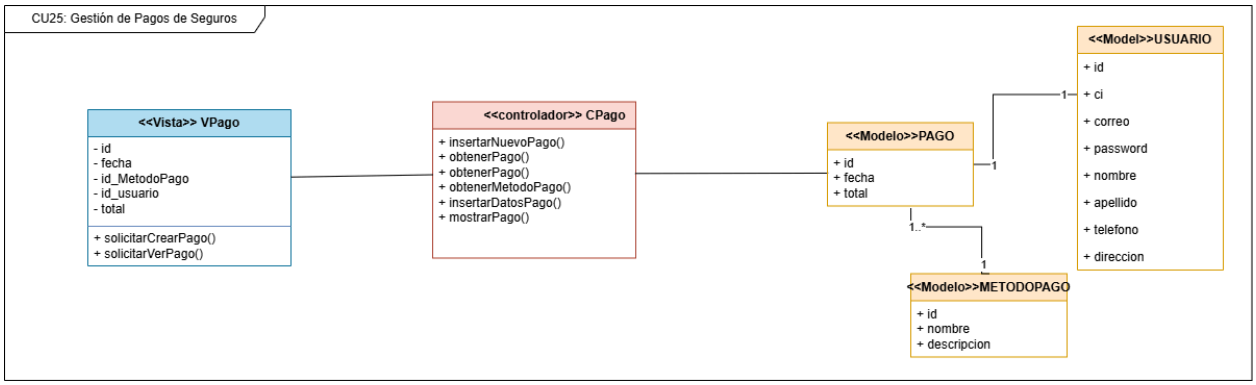
CU22: Evaluación de Médicos



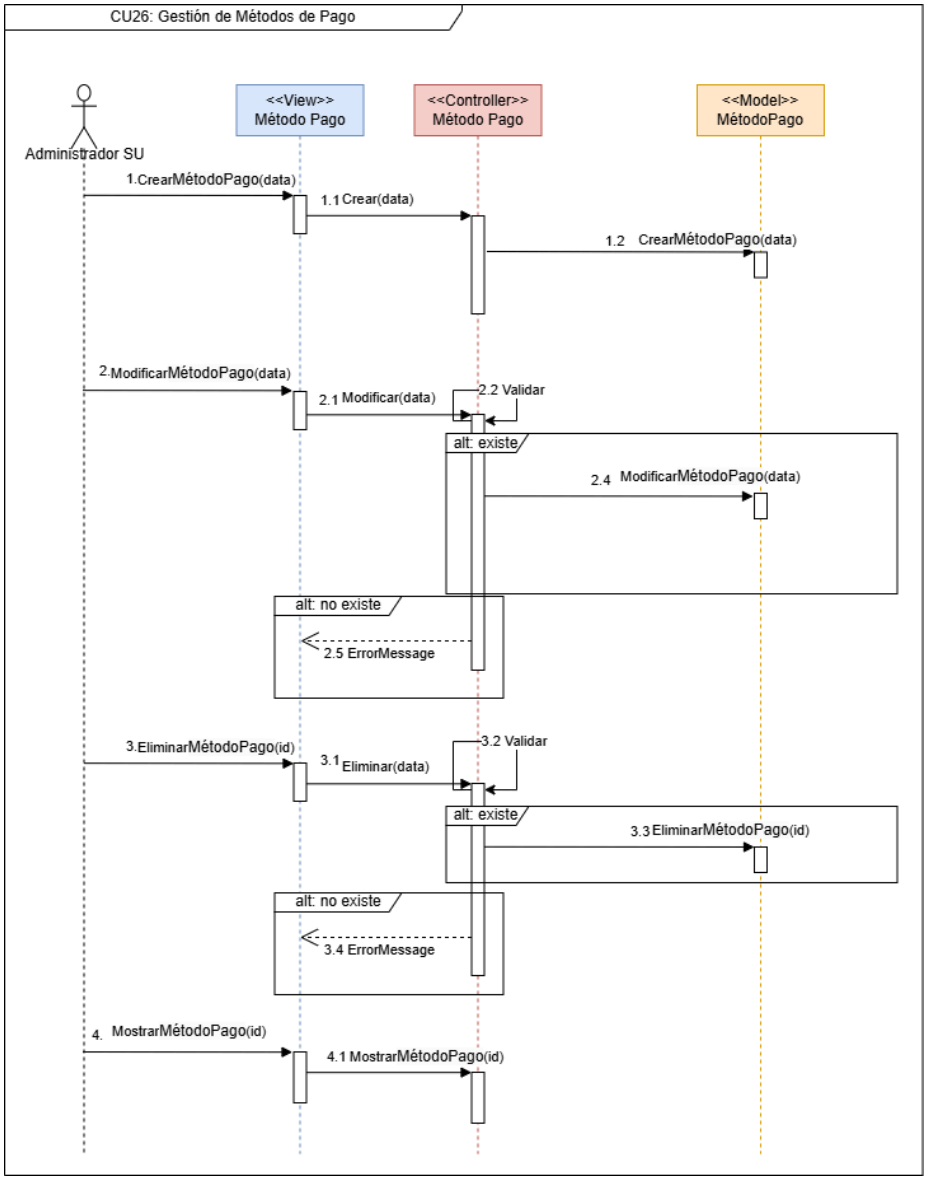


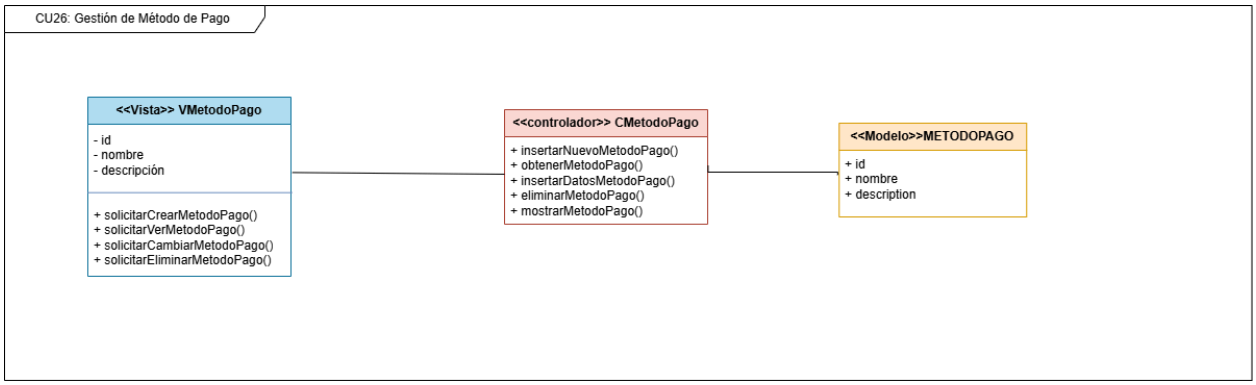
CU25: Gestión de Pagos de Seguros



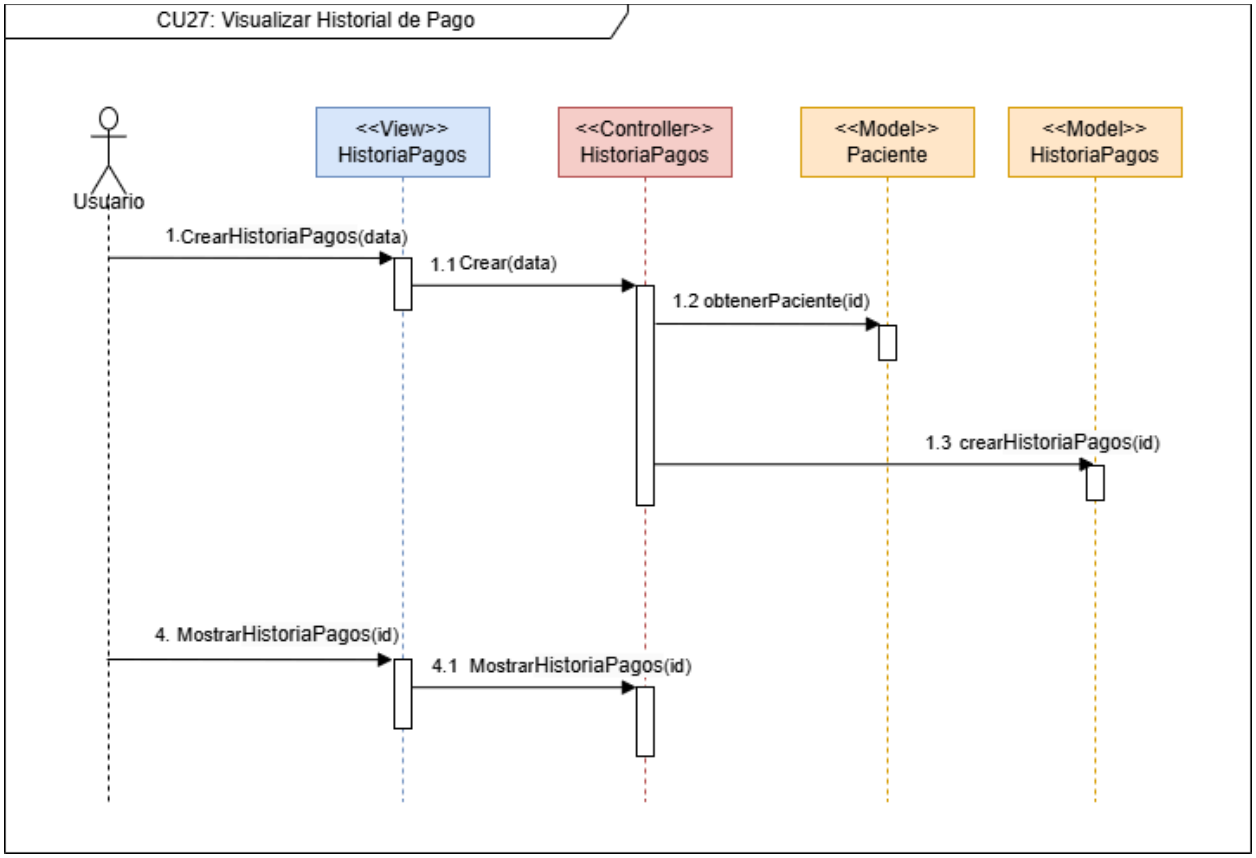


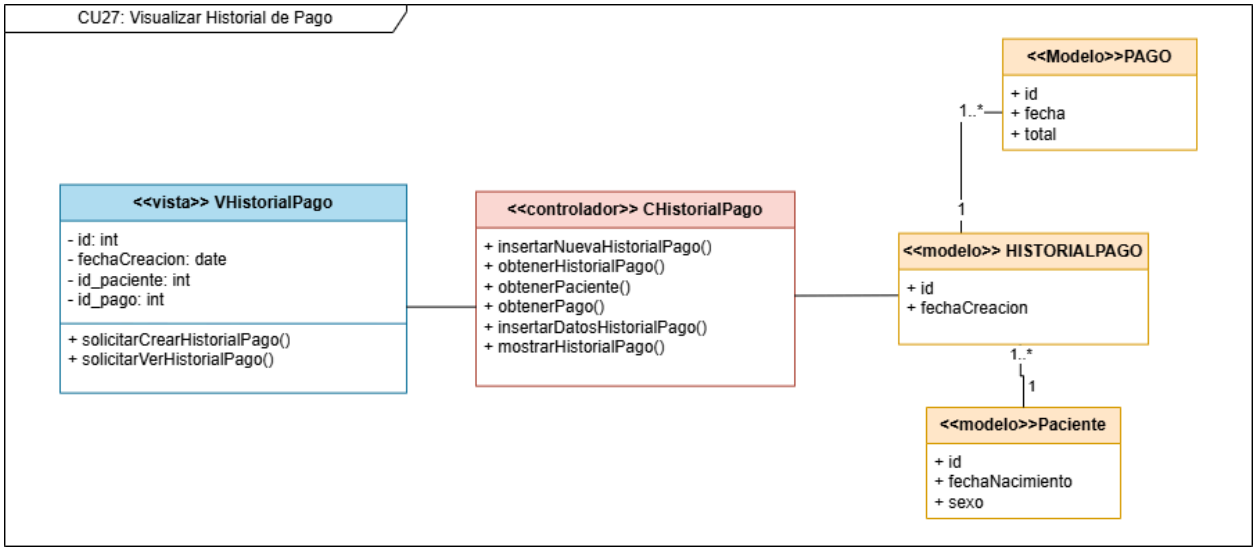
CU26: Gestión de Métodos de Pago



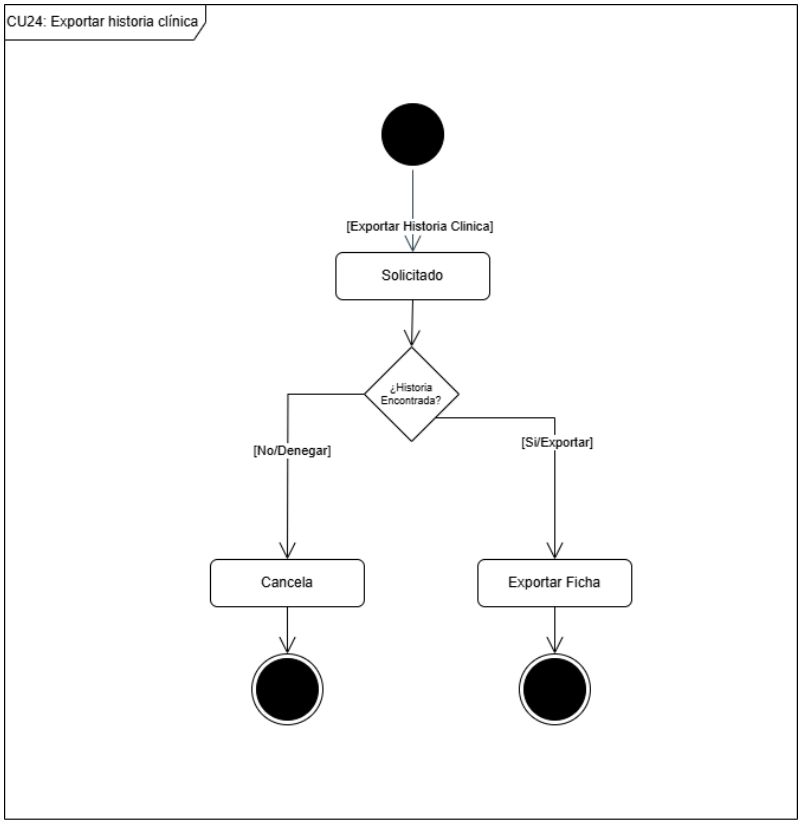


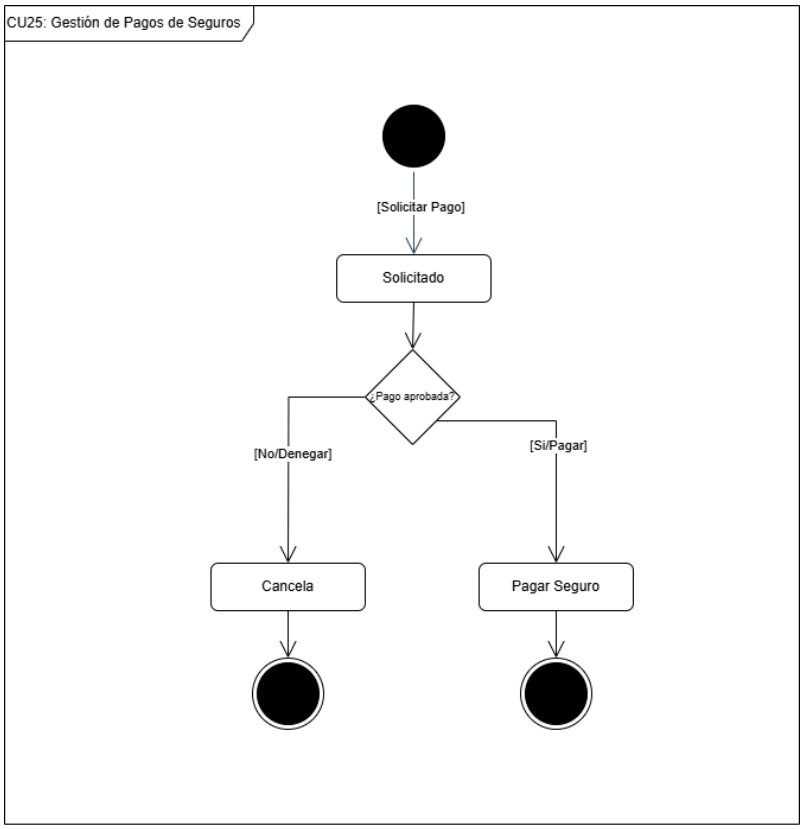
CU27: Visualizar Historial de Pago


CU27: Visualizar Historial de Pago
Usuario
<<View>>
HistoriaPagos
<<Controller>>
HistoriaPagos
<<Model>>
Paciente
<<Model>>
HistoriaPagos
1.CrearHistoriaPagos(data)
1.1 Crear(data)
1.2 obtenerPaciente(id)
1.3 crearHistoriaPagos(id)
4. MostrarHistoriaPagos(id)
4.1 MostrarHistoriaPagos(id)

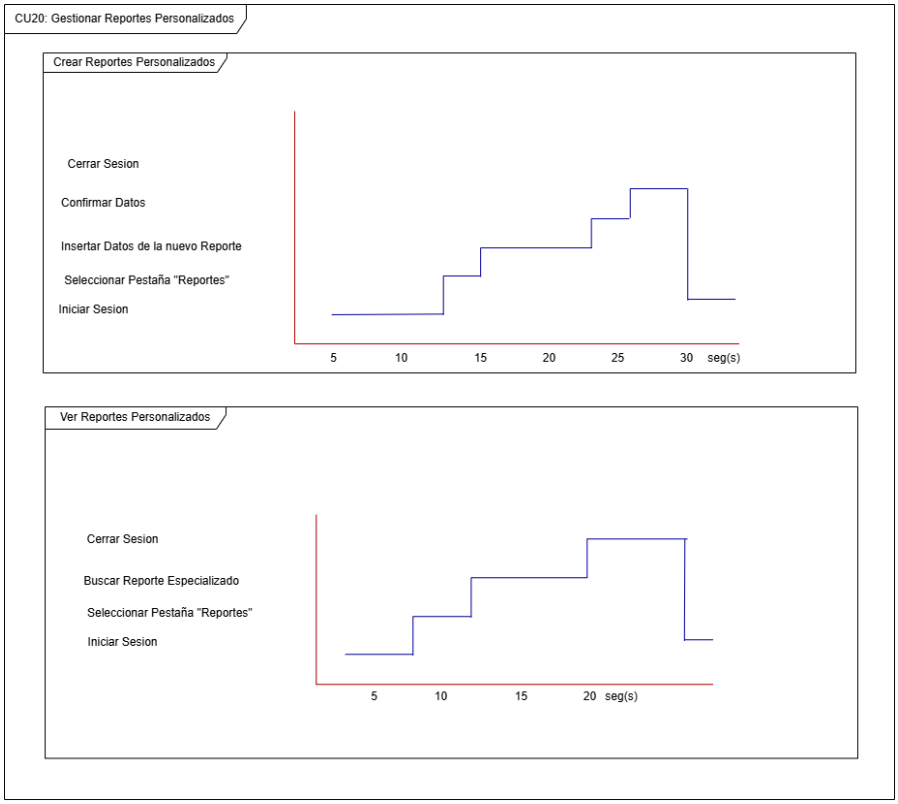

CU27: Visualizar Historial de Pago
<<Modelo>>PAGO
+ id
+ fecha
+ total
1..*
1
<<vista>> VHistorialPago
- id: int
- fechaCreacion: date
- id_paciente: int
- id_pago: int
+ solicitarCrearHistorialPago()
+ solicitarVerHistorialPago()
<<controlador>> CHistorialPago
+ insertarNuevaHistorialPago()
+ obtenerHistorialPago()
+ obtenerPaciente()
+ obtenerPago()
+ insertarDatosHistorialPago()
+ mostrarHistorialPago()
<<modelo>> HISTORIALPAGO
+ id
+ fechaCreacion
1..*
1
<<modelo>>Paciente
+ id
+ fechaNacimiento
+ sexo

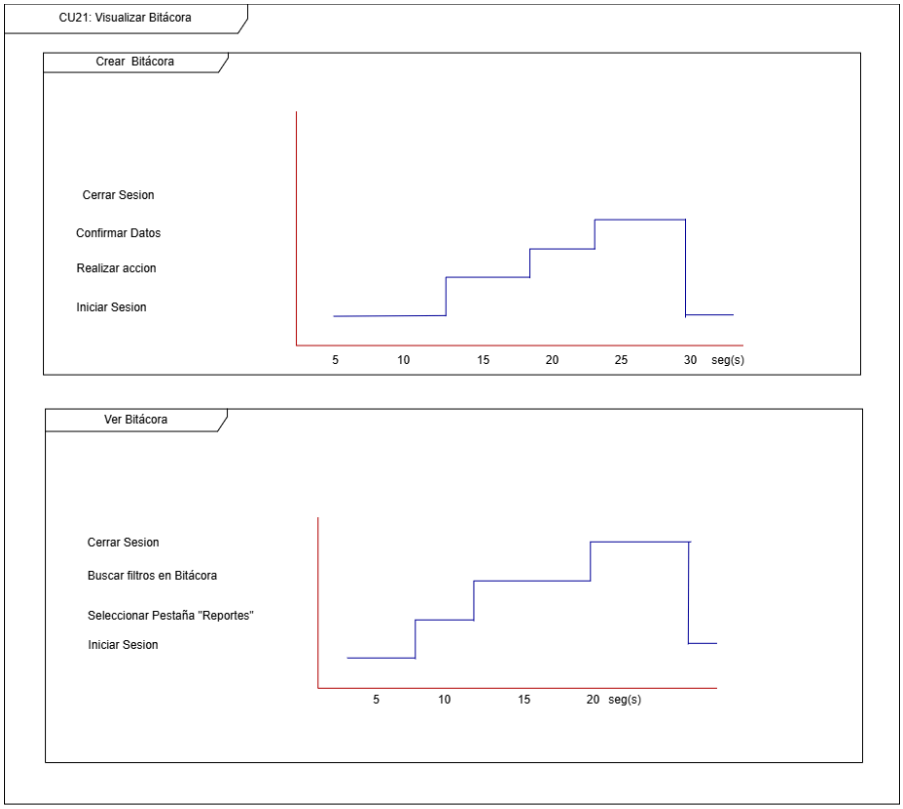
b) Diagrama de Estado

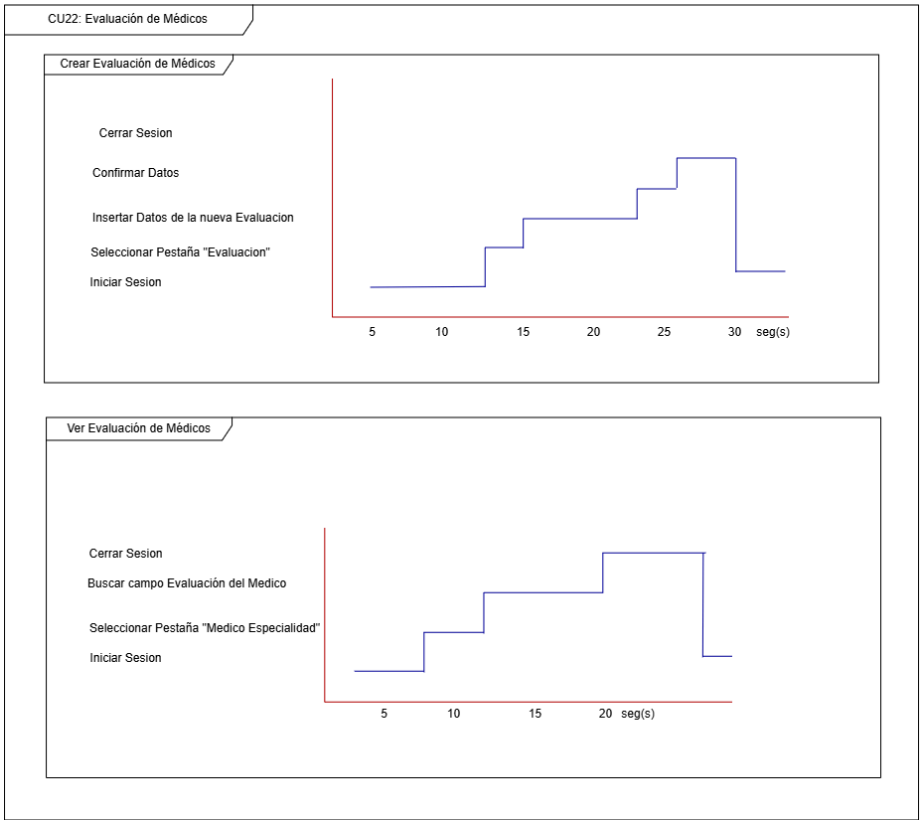


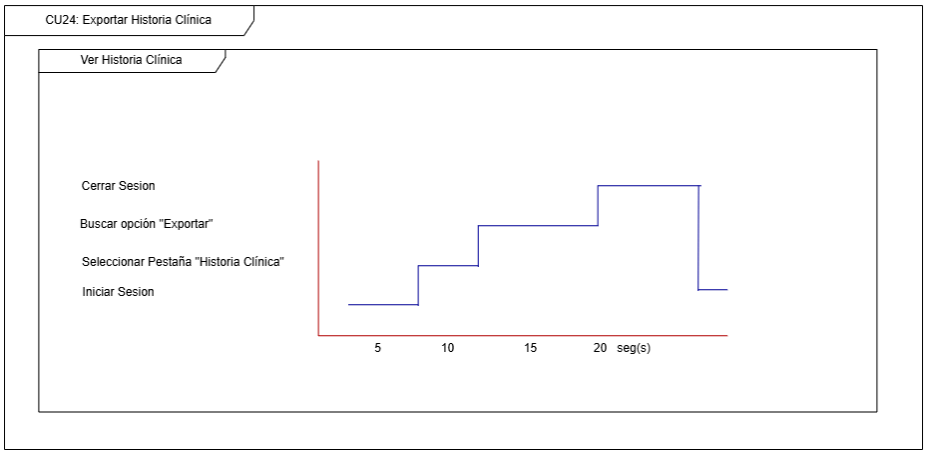


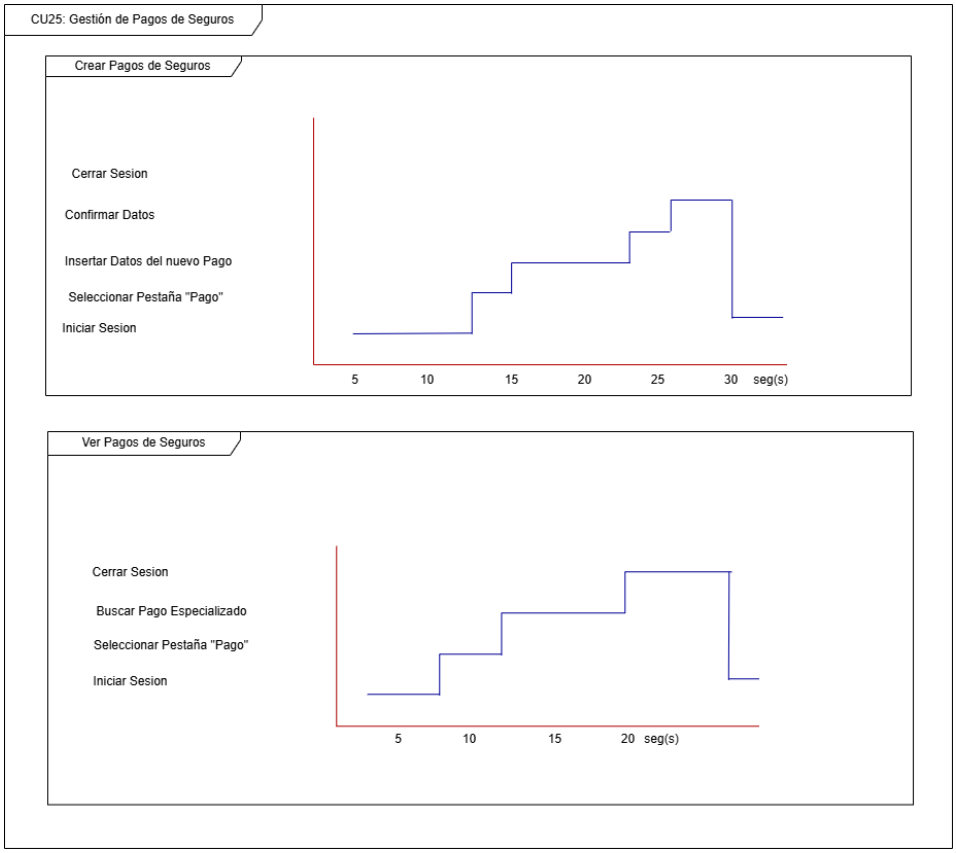
c) Diagrama de Tiempo





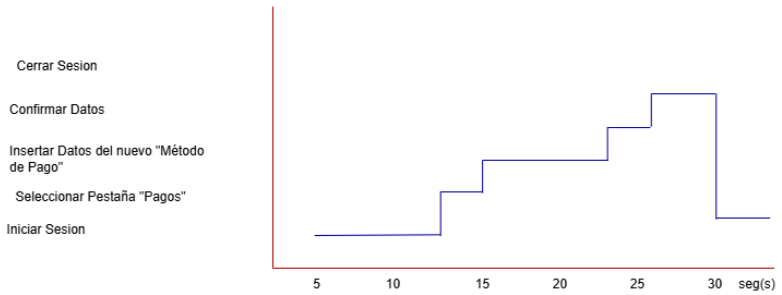

CU22: Evaluación de Médicos
Crear Evaluación de Médicos
Cerrar Sesion
Confirmar Datos
Insertar Datos de la nueva Evaluacion
Seleccionar Pestaña "Evaluacion"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Evaluación de Médicos
Cerrar Sesion
Buscar campo Evaluación del Medico
Seleccionar Pestaña "Medico Especialidad"
Iniciar Sesion
5
10
15
20
seg(s)


CU24: Exportar Historia Clínica
Ver Historia Clínica
Cerrar Sesion
Buscar opción "Exportar"
Seleccionar Pestaña "Historia Clínica"
Iniciar Sesion
5
10
15
20
seg(s)

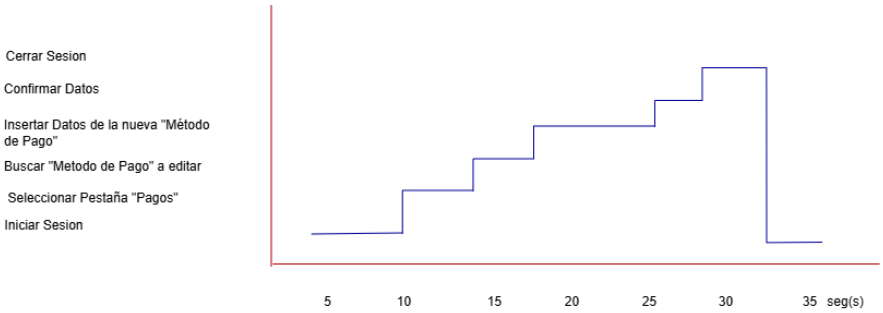

CU25: Gestión de Pagos de Seguros
Crear Pagos de Seguros
Cerrar Sesion
Confirmar Datos
Insertar Datos del nuevo Pago
Seleccionar Pestaña "Pago"
Iniciar Sesion
5
10
15
20
25
30
seg(s)
Ver Pagos de Seguros
Cerrar Sesion
Buscar Pago Especializado
Seleccionar Pestaña "Pago"
Iniciar Sesion
5
10
15
20
seg(s)

CU26: Gestión de Método de Pago

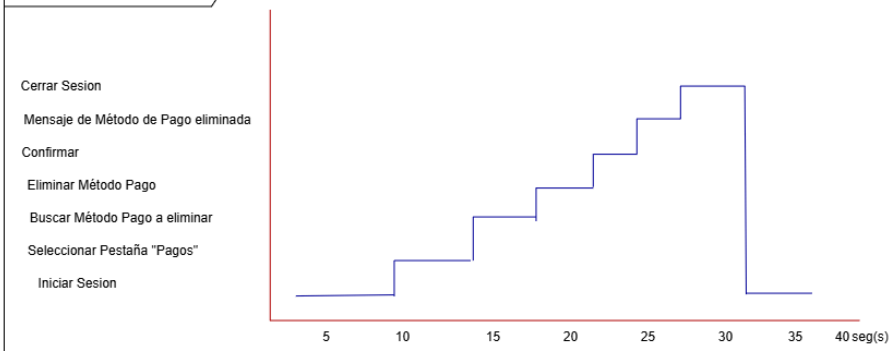
Crear Método de Pago



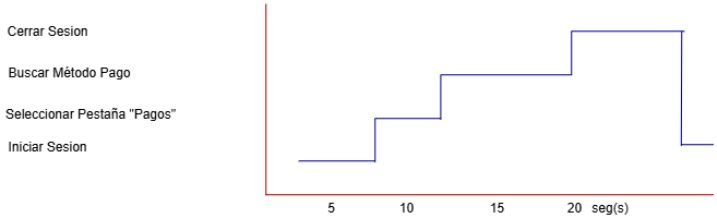
Editar Método de Pago

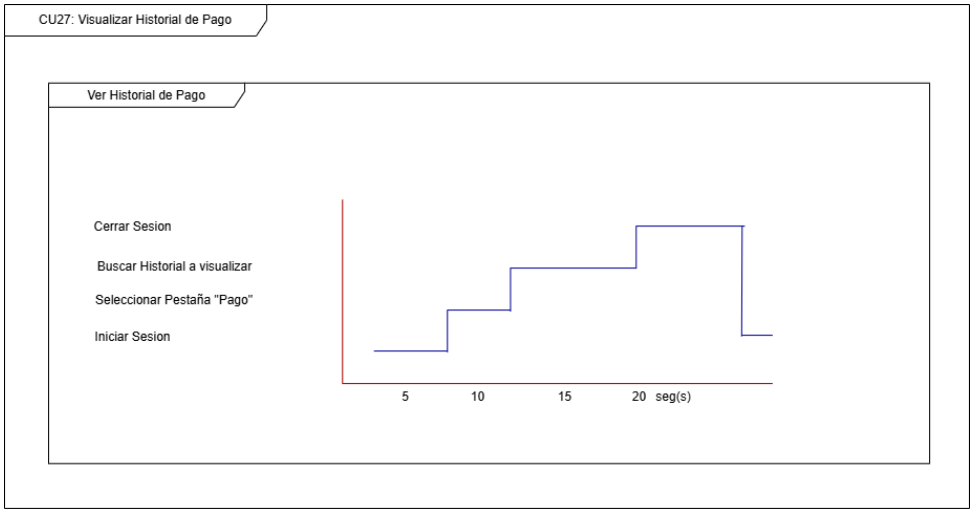


Eliminar Método de Pago

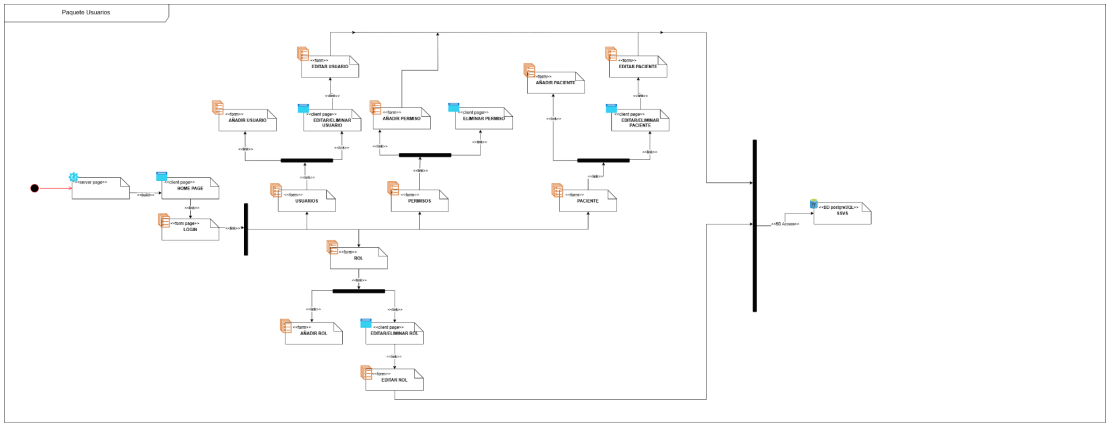


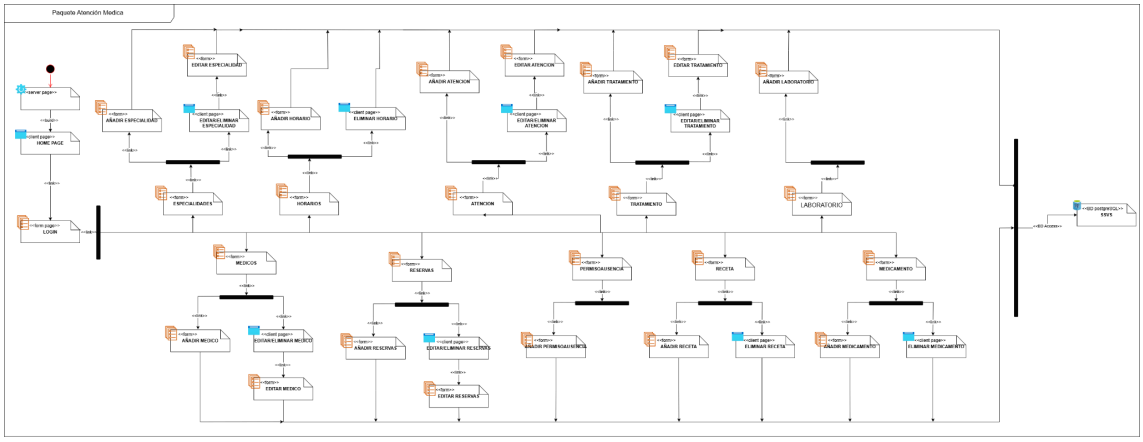
Ver Método de Pago

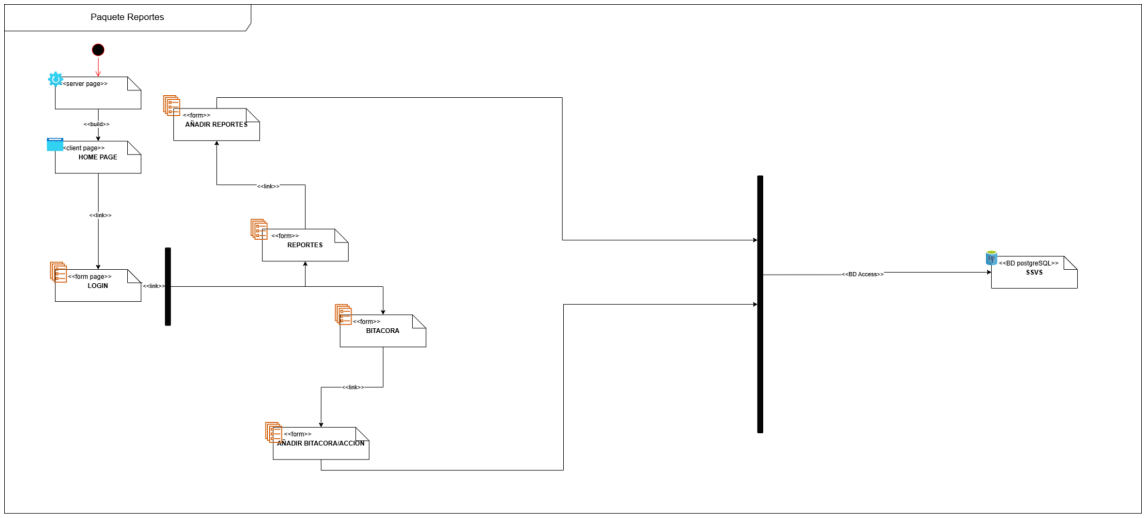


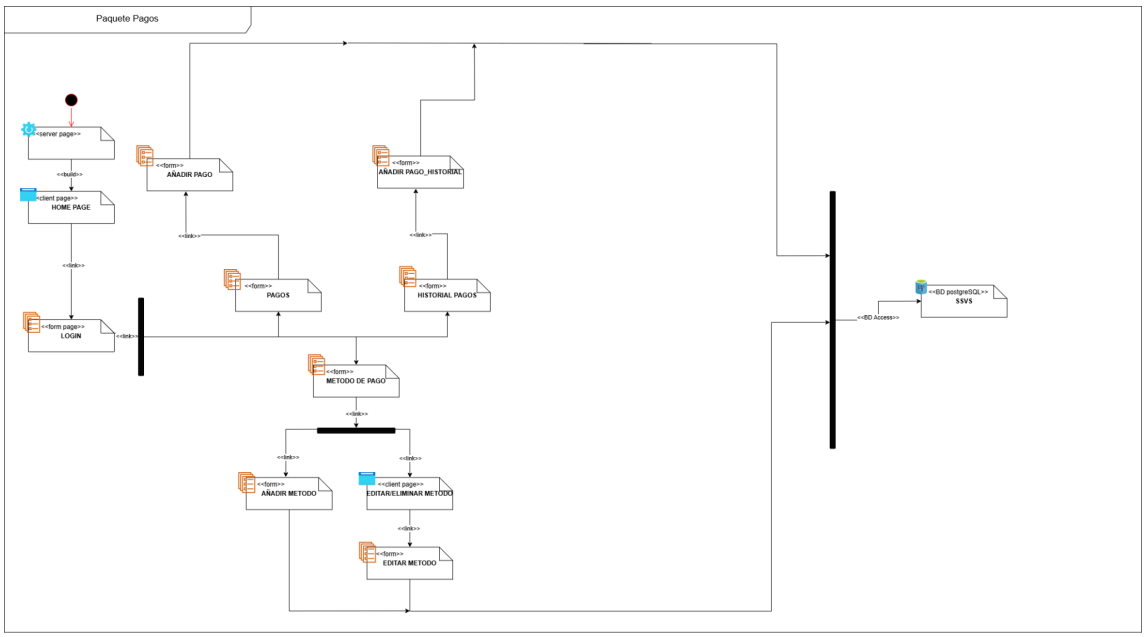


d) Diagrama de Navegación





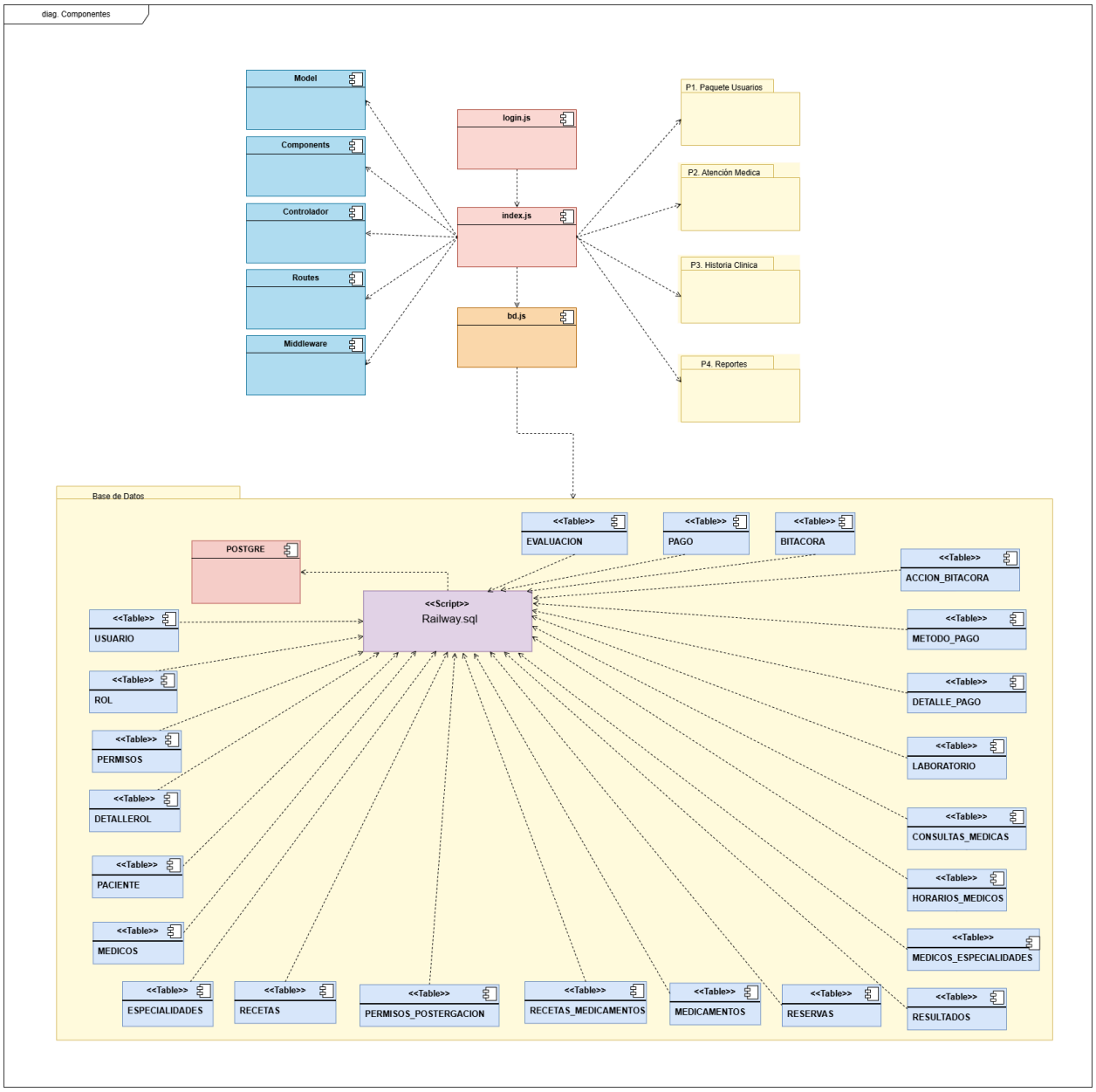

Paquete Reportes
<<server page>>
<<build>>
<<client page>>
HOME PAGE
<<link>>
<<form page>>
LOGIN
<<link>>
<<form>>
AÑADIR REPORTES
<<link>>
<<form>>
REPORTES
<<form>>
BITACORA
<<link>>
<<form>>
AÑADIR BITACORA/ACCIÓN
<<BD Access>>
<<BD postgreSQL>>
SV


Paquete Pagos
<<server page>>
<<build>>
<<client page>>
HOME PAGE
<<link>>
<<form page>>
LOGIN
<<link>>
<<form>>
AÑADIR PAGO
<<form>>
AÑADIR PAGO_HISTORIAL
<<link>>
<<form>>
PAGOS
<<link>>
<<form>>
HISTORIAL PAGOS
<<form>>
METODO DE PAGO
<<link>>
<<link>>
<<form>>
AÑADIR METODO
<<link>>
<<client page>>
EDITAR/ELIMINAR METODO
<<link>>
<<form>>
EDITAR METODO
<<BD Access>>
<<BD postgreSQL>>
SV

2.2 Implementación

2.2.1. Componentes y artefactos generados

2.2.1.1 Diagrama de componentes



2.3. Pruebas

2.3.1. Plan de pruebas (criterios de aceptación)

CU20: Gestionar Reportes Personalizados

1. El administrador puede seleccionar criterios de búsqueda como rango de fechas, médico, especialidad y asegurado.
2. Los reportes generados muestran información detallada, completa y actualizada según los filtros aplicados.
3. Los reportes se pueden visualizar en pantalla y descargar en formatos PDF o Excel.
4. El sistema asegura la integridad y exactitud de los datos generados.
5. Los reportes generados respetan las restricciones de acceso según el rol del usuario.

CU21: Visualizar Bitácora

1. Solo los administradores pueden acceder al módulo de bitácora.
2. La bitácora muestra eventos del sistema con detalles como usuario, acción realizada, fecha, hora e IP.
3. Los datos en la bitácora son ordenados cronológicamente y se pueden filtrar por criterios como tipo de acción, fecha y usuario.
4. La información es clara y permite rastrear acciones específicas de usuarios en el sistema.
5. La visualización de la bitácora no afecta el rendimiento del sistema, incluso con grandes volúmenes de datos.

CU22: Evaluación de Médicos

1. Los pacientes pueden evaluar a los médicos después de una consulta, utilizando un sistema de puntuación o comentarios.
2. El sistema garantiza que solo los pacientes que tuvieron consultas con un médico puedan evaluarlo.
3. Las evaluaciones se registran correctamente y son visibles solo para el administrador.
4. Los médicos no pueden visualizar ni modificar sus evaluaciones.

CU24: Exportar Historia Clínica

1. Los médicos y pacientes pueden solicitar la exportación de la historia clínica del paciente.
2. La historia clínica exportada incluye datos actualizados como diagnósticos, tratamientos, exámenes y recetas.
3. Los archivos exportados están disponibles en formatos PDF o XML, según sea necesario.
4. El sistema garantiza que solo usuarios autorizados (el paciente o médicos que lo atendieron) puedan exportar la historia clínica.

5. La exportación se realiza de forma rápida y asegura la privacidad de los datos del paciente.

CU25: Gestión de Pagos de Seguros

1. Los asegurados pueden consultar sus pagos pendientes y realizar pagos a través de los métodos habilitados (tarjeta de crédito, transferencia bancaria, etc.).
2. El sistema valida los detalles del pago y registra correctamente el monto, método y fecha del pago.
3. Los pagos realizados actualizan automáticamente el estado del seguro como "vigente".
4. Los administradores pueden visualizar un historial de pagos por asegurado o periodo y generar reportes.

CU26: Gestión de Métodos de Pago

- Permitir al administrador registrar nuevos métodos de pago con información válida (nombre, descripción, tipo).
- Mostrar un mensaje de confirmación al registrar, editar o eliminar métodos de pago.

CU27: Visualizar Historial de Pago

- Permitir al paciente y al administrador consultar el historial de pagos con información detallada (fecha, monto, método, estado).
- Validar que los filtros aplicados (por fecha, estado, método de pago) funcionen correctamente.
- Asegurar que los datos mostrados en el historial sean precisos y estén actualizados.
- Restringir el acceso al historial de pagos para que solo el paciente correspondiente o el administrador puedan visualizarlo.
- Mostrar un mensaje claro en caso de que no existan pagos registrados para los criterios seleccionados.

2.3.2. Reporte de prueba

CU20: Gestionar Reportes Personalizados

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado

Permite al administrador generar reportes personalizados.	Acceder al módulo de reportes. Seleccionar criterios de búsqueda (fechas, médico, especialidad, asegurado). Generar reporte y descargar en PDF o Excel.	Permitir generar reportes con filtros aplicados. Permitir descarga en formatos seleccionados. Mostrar datos actualizados y completos.	El reporte se genera con datos correctos y se descarga en el formato elegido.	Los reportes cumplen con los filtros y se descargan correctamente.	Aprobado

CU21: Visualizar Bitácora

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los administradores visualizar la bitácora del sistema.	Acceder al módulo de bitácora. Filtrar por tipo de acción, fecha y usuario. Consultar el historial de acciones.	Permitir filtrar y visualizar registros por criterios aplicados. Mostrar datos como usuario, acción, fecha, hora e IP.	Los registros se visualizan correctamente y se aplican los filtros.	Los registros son precisos y los filtros funcionan correctamente.	Aprobado

CU22: Evaluación de Médicos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los pacientes evaluar a los médicos.	Finalizar una consulta médica. Acceder al módulo de evaluaciones. Enviar una	Permitir solo a pacientes con consultas completadas evaluar. Registrar la evaluación	La evaluación se registra y está disponible para el	Las evaluaciones se registraron correctamente y se asociaron al	Aprobado

	puntuación y/o comentario.	correctamente.	administrador.	médico evaluado.	

CU24: Exportar historia clínica

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite exportar la historia clínica de un paciente.	Acceder al perfil del paciente (como médico o paciente). Solicitar la exportación de la historia clínica en PDF.	Generar la historia clínica completa con datos actualizados (diagnósticos, tratamientos, recetas). Permitir exportación en PDF.	La historia clínica se genera con los datos correctos y se descarga en formato PDF.	La exportación se realiza correctamente con todos los datos requeridos.	Aprobado

CU25: Gestión de Pagos de Seguros

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite gestionar pagos de seguros médicos.	Acceder al módulo de pagos como paciente. Consultar pagos pendientes. Realizar un pago y verificar la actualización del estado del seguro.	Validar detalles del pago (monto, método, fecha). Actualizar el estado del seguro como "vigente" tras el pago.	El pago se registra correctamente y el seguro pasa a estado "vigente".	Los pagos fueron registrados y los estados se actualizaron correctamente.	Aprobado

CU26: Gestión de Método de Pago

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite al administrador gestionar los métodos de pago disponibles en el sistema.	Acceder al módulo de métodos de pago. Registrar un nuevo método de pago (e.g., tarjeta de crédito, transferencia). Editar o eliminar un método de pago existente.	Permitir registrar, editar y eliminar métodos de pago. Asegurar que solo se eliminen métodos no utilizados.	Los métodos de pago se registran, editan o eliminan correctamente según las acciones realizadas.	Los métodos de pago fueron gestionados correctamente y las restricciones funcionaron adecuadamente.	Aprobado

CU27: Visualizar Historial de Pagos

Descripción de la Prueba	Procedimiento de Prueba	Criterios de Aceptación	Resultado Esperado	Resultado Obtenido	Estado
Permite a los pacientes y administradores consultar el historial de pagos.	Acceder al módulo de historial de pagos. Filtrar por periodo o estado (pagado, pendiente). Consultar los detalles de cada pago realizado o pendiente.	Permitir visualizar los pagos realizados y pendientes con detalles (fecha, monto, método). Aplicar filtros correctamente.	El historial muestra todos los pagos según los filtros aplicados y con datos precisos.	Los pagos se visualizan correctamente y los filtros funcionan adecuadamente.	Aprobado

3. Daily Scrum

Daily Scrum - 09/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Pagos de Seguros.	Ninguno	Implementación del backend para Reportes Personalizados.
Fabiana Manzoni Bravo	Implementación del frontend para Pagos de Seguros.	Ninguno	Implementación del frontend para Reportes Personalizados.

Daily Scrum - 10/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Reportes Personalizados.	Ninguno	Implementación del backend para Evaluación de Médicos.
Fabiana Manzoni Bravo	Implementación del frontend para Reportes Personalizados.	Ninguno	Implementación del frontend para Evaluación de Médicos.

Daily Scrum - 11/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Evaluación de Médicos.	Ninguno	Implementación del backend para Exportar Historia Clínica.

Fabiana Manzoni Bravo	Implementación de frontend para Evaluación de Médicos.	Ninguno	Implementación del frontend para Exportar Historia Clínica.

Daily Scrum - 12/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Exportar Historia Clínica.	Ninguno	Implementación del backend para Métodos de Pago.
Fabiana Manzoni Bravo	Implementación del frontend para Exportar Historia Clínica.	Ninguno	Implementación del frontend para Métodos de Pago.

Daily Scrum - 13/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Métodos de Pago.	Ninguno	Implementación del backend para Historial de Pagos..
Fabiana Manzoni Bravo	Implementación del frontend para Métodos de Pago.	Ninguno	Implementación del frontend para Historial de Pagos.

Daily Scrum - 14/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación del backend para Historial de Pagos.	Ninguno	Implementación móvil para Evaluación de Médicos.

Fabiana Manzoni Bravo	Implementación del frontend para Historial de Pagos.	Ninguno	Implementación móvil para Historial de Pagos..

Daily Scrum - 15/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación móvil para Evaluación de Médicos.	Ninguno	Implementación móvil para Exportar Historia Clínica.
Fabiana Manzoni Bravo	Implementación móvil para Historial de Pagos.	Ninguno	Ninguna.

Daily Scrum - 16/11/2024

Integrante	*Actividad Realizada*	*Impedimentos*	*Próxima Actividad*
Jorge Arturo Aliaga Valencia	Implementación móvil para Exportar Historia Clínica.	Ninguno	Ninguna.
Fabiana Manzoni Bravo	Ninguna	Ninguno	Ninguna

4. Sprint Review

Revisión de Sprint	
Objetivo del Sprint: El objetivo del sprint es desarrollar la gestión de reportes personalizados, bitácoras, evaluaciones de médicos, exportación de historias clínicas y pagos de seguros, asegurando funcionalidades avanzadas para un control eficiente y transparente en el sistema HCE.	
Participantes	
Nombre	**Rol**
Jorge Arturo Aliaga Valencia	Desarrollador
Fabiana Manzoni Bravo	Desarrollador
Presentación del incremento	
Función Presentada	**Retroalimentación**
Diseñar interfaz para Gestionar Reportes Personalizados	Funcionalidad Terminada
Implementar Gestionar Reportes Personalizados	Funcionalidad Terminada
Realizar pruebas de Gestionar Reportes Personalizados	Funcionalidad Terminada
Diseñar interfaz para Visualizar Bitácora	Funcionalidad Terminada
Implementar Visualizar Bitácora	Funcionalidad Terminada
Realizar pruebas de Visualizar Bitácora	Funcionalidad Terminada
Diseñar interfaz para Evaluación de Médicos	Funcionalidad Terminada
Implementar Evaluación de Médicos	Funcionalidad Terminada
Realizar pruebas de Evaluación de Médicos	Funcionalidad Terminada

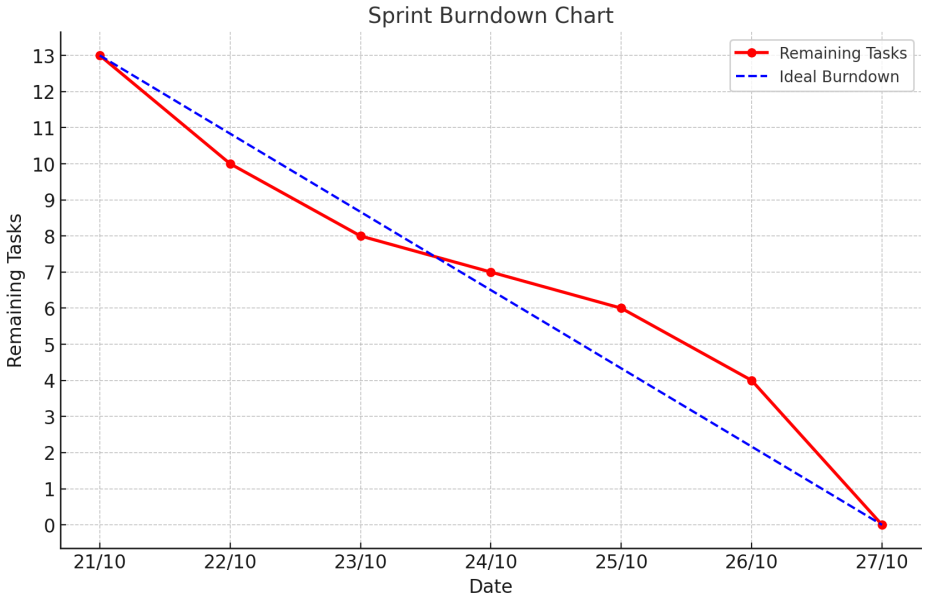
Diseñar interfaz Exportar historia clínica	Funcionalidad Terminada
Implementar Exportar historia clínica	Funcionalidad Terminada
Realizar pruebas de Exportar historia clínica	Funcionalidad Terminada
Diseñar interfaz para Gestión de Pagos de Seguros	Funcionalidad Terminada
Implementar Gestión de Pagos de Seguros	Funcionalidad Terminada
Realizar pruebas de Gestión de Pagos de Seguros	Funcionalidad Terminada
Diseñar interfaz para Gestión de Método de Pago	Funcionalidad Terminada
Implementar Gestión de Método de Pago	Funcionalidad Terminada
Realizar pruebas de Visualizar Historial de Pagos	Funcionalidad Terminada
Diseñar interfaz para Visualizar Historial de Pagos	Funcionalidad Terminada
Implementar Visualizar Historial de Pagos	Funcionalidad Terminada
Realizar pruebas de Visualizar Historial de Pagos	Funcionalidad Terminada

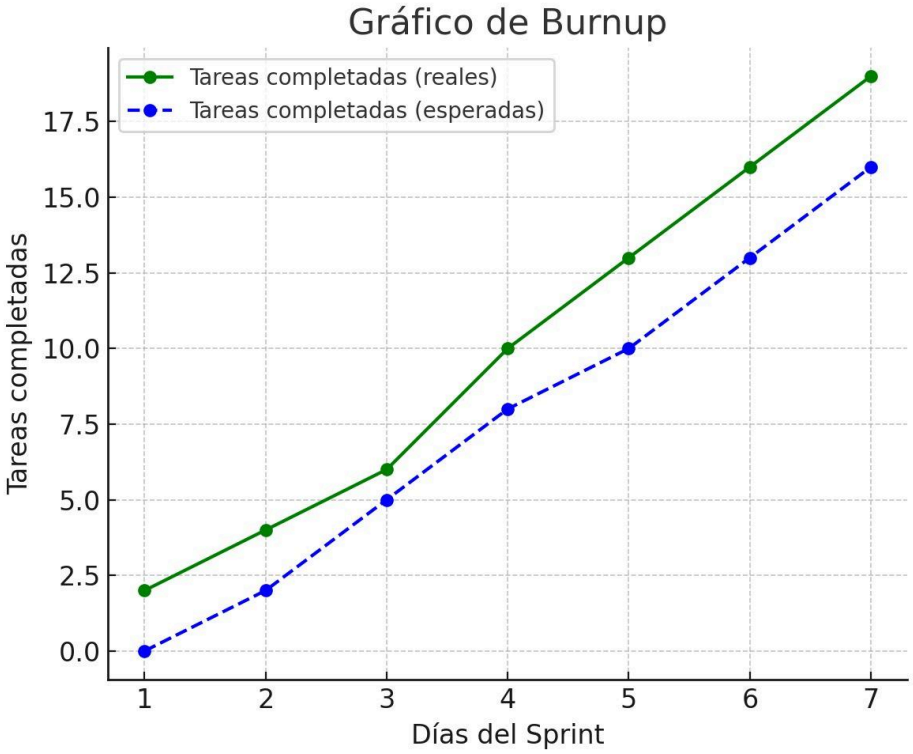
5. Sprint Retrospective

Sprint Retrospective
Fecha
17/11/2024
Facilitador
Jorge Arturo Aliaga Valencia

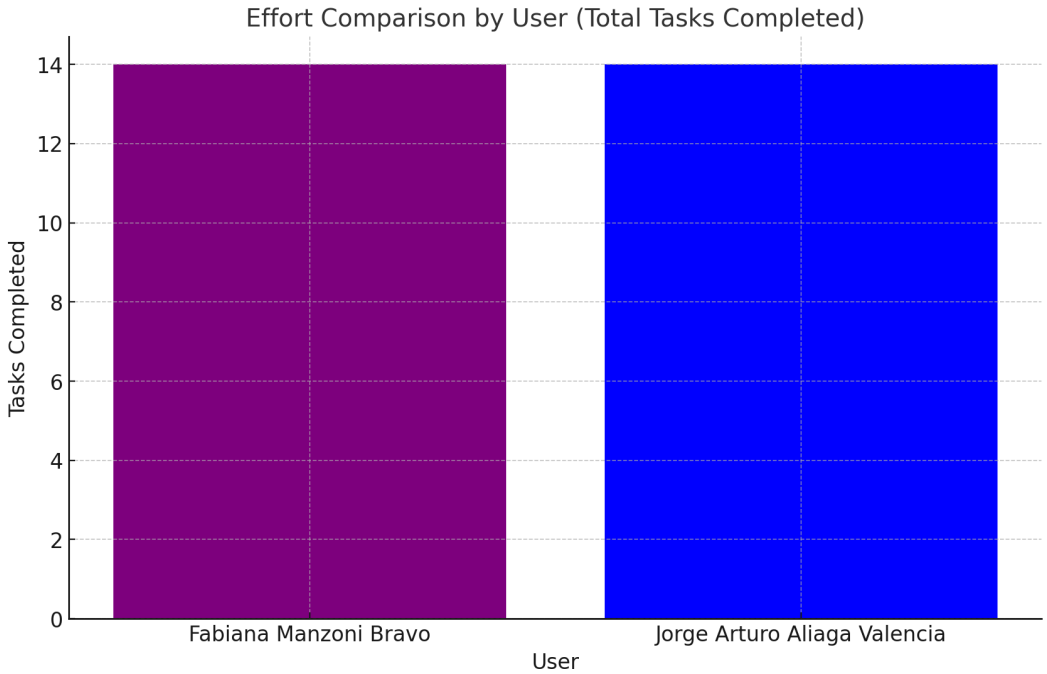
Objetivo
Conclusión del Sprint 3
Nombres de asistentes
Fabiana Manzoni Bravo
Temas por tratar
Sprint 3
Discusión
¿Qué salió bien?
- Se completaron las funcionalidades clave del sprint, incluyendo la gestión de reportes personalizados y la exportación de historias clínicas, cumpliendo con los criterios de aceptación. - La integración entre módulos se realizó sin contratiempos, garantizando un flujo de datos consistente. - Las pruebas de funcionalidad fueron exitosas, con resultados positivos en las validaciones críticas de seguridad y trazabilidad.
¿Qué no salió bien?
- La documentación se fue realizando junto con el desarrollo, pero al final del sprint tuvimos que dedicar tiempo extra para dejarla completa y bien estructurada, lo que generó algo de presión adicional.
¿Qué haremos de forma diferente?
- Nada. El proyecto está finalizado con éxito.

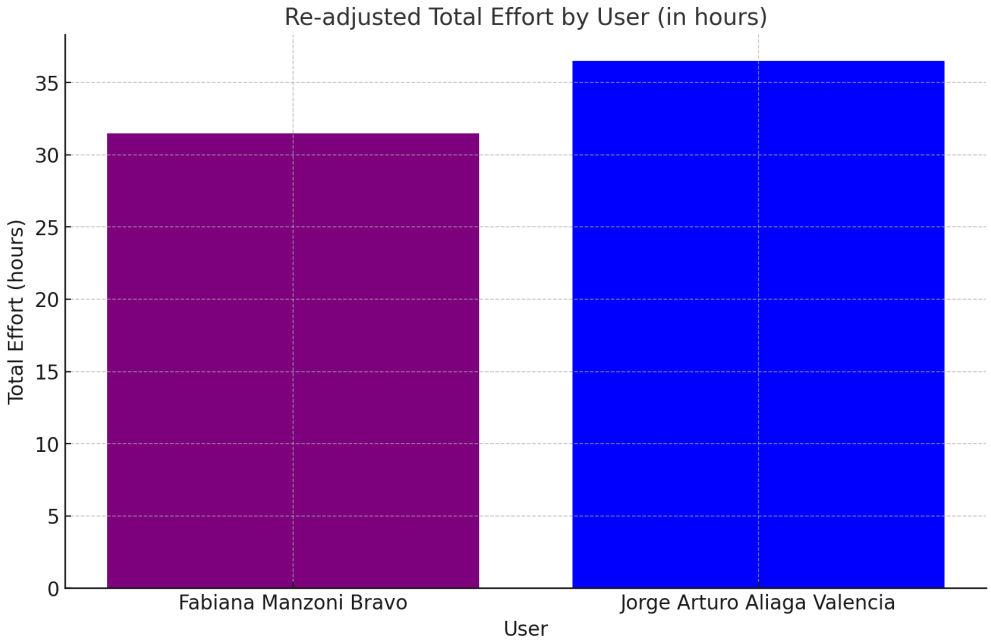
6. Burndown y BurnUp (Grafica de tareas y Datos de tareas)



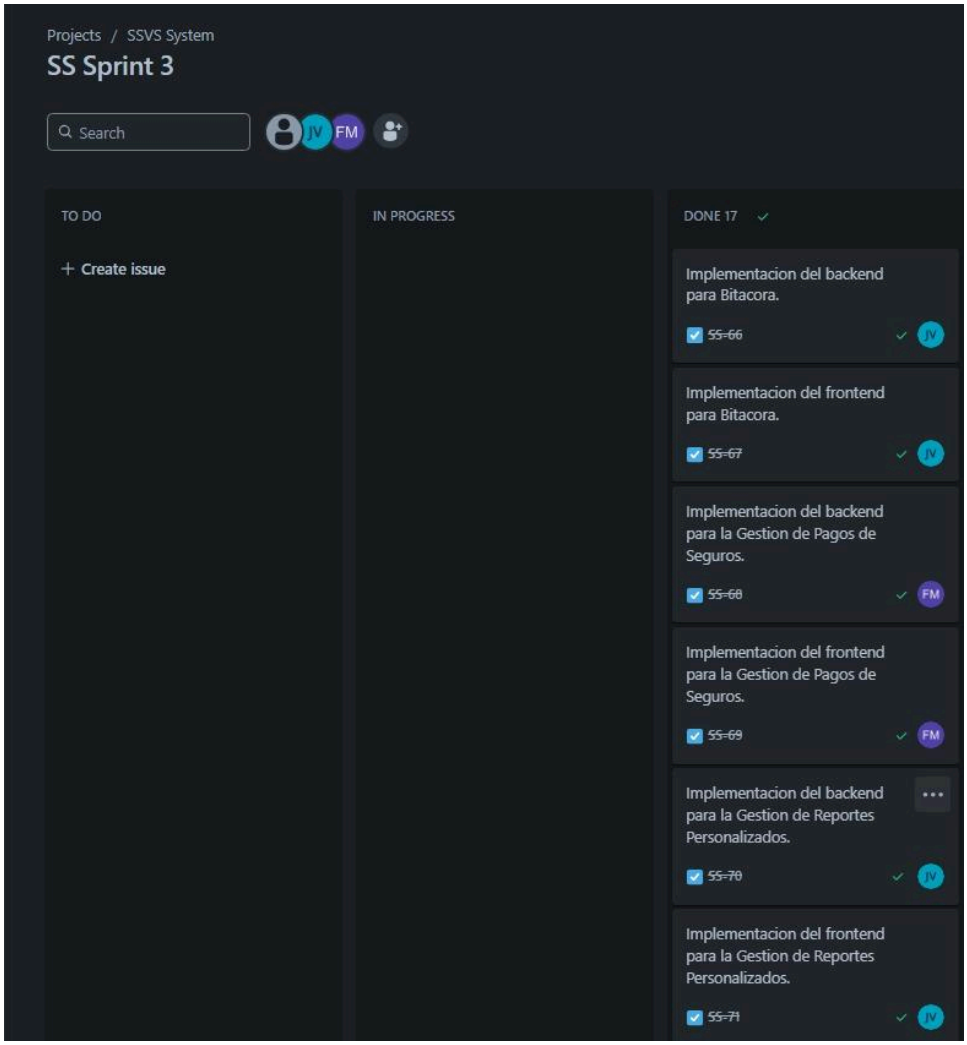


7. Gráfica de esfuerzo y Datos de esfuerzo





8. Scrum TaskBoard (Backlog, to do, doing, done)



TO DO	IN PROGRESS	DONE 17
		Implementacion del backend para la Evaluacion de Medicos. ~~SS-72~~ FM
		Implementacion del frontend para la Evaluacion de Medicos. ~~SS-73~~ FM
		Implementacion del backend para la Exportacion del Historial Clinico. ~~SS-74~~ JV
		Implementacion del frontend para la Exportacion del Historial Clinico. ~~SS-75~~ JV
		Implementacion del backend para la Gestion de Metodo de Pago. ~~SS-76~~ FM
		Implementacion del frontend para la Gestion de Metodo de Pago ~~SS-77~~ FM
		Implementacion del backend para Visualizar Historial de Pago. ~~SS-78~~ FM
		Implementacion del backend para Visualizar Historial de Pago. ~~SS-79~~ FM
		Implementacion Movil para la Evaluacion de Medicos. ~~SS-80~~ JV
		Implementacion Movil para exportar Historial Clinico. ~~SS-81~~ JV
		Implementacion Movil para Visualizar Historial de Pagos. ~~SS-82~~ FM

CONCLUSIONES

El desarrollo del sistema de gestión de historias clínicas electrónicas con reservas, utilizando Angular, Spring Boot y Flutter, fue una experiencia desafiante y enriquecedora. Este proyecto combina una app móvil, diseñada en Flutter para los pacientes, con una plataforma web en Angular y un backend en Spring Boot para la administración eficiente de usuarios, roles, horarios y consultas. Esta estructura asegura una experiencia optimizada tanto para los pacientes como para el personal médico, cumpliendo con altos estándares de accesibilidad y seguridad.

Como equipo de dos integrantes, trabajamos en conjunto para implementar múltiples módulos que funcionan de manera integrada, priorizando la usabilidad y escalabilidad del sistema. Las tecnologías seleccionadas permitieron desarrollar un sistema flexible y moderno, capaz de adaptarse a las necesidades cambiantes del sector salud.

En conclusión, este proyecto no solo facilita la gestión médica, sino que también destaca como una solución tecnológica innovadora y escalable. La experiencia adquirida en su desarrollo nos prepara para enfrentar nuevos retos en el ámbito de la transformación digital en la salud.

BIBLIOGRAFIA

Harris, M. (2021). *Spring Boot in Action*. Manning Publications.

Gehrke, J., & Silberschatz, A. (2010). *Database System Concepts* (6th ed.). McGraw-Hill.

Wampler, D. (2020). *Flutter in Action*. Manning Publications.

Ducrohet, X., & Moore, C. (2021). *Pro Angular 10* (4th ed.). Apress.

SOFTWARE (Avance en cada presentación)

Sprint 1: Fundamentos del Sistema

En este sprint se desarrollaron las bases fundamentales del sistema, enfocándose en los módulos de **Gestión de Usuarios**, **Roles** y **Permisos**, junto con las funcionalidades de inicio y cierre de sesión. El módulo de **Gestión de Usuarios** permitió implementar un CRUD completo para la creación, edición, eliminación y listado de usuarios, utilizando Angular en el frontend y Spring Boot en el backend. También se integraron validaciones básicas para asegurar la calidad de los datos ingresados.

El módulo de **Gestión de Roles** incluyó la capacidad de asignar roles a los usuarios y asociar permisos específicos a cada rol. Esto fue complementado con un sistema de **Gestión de Permisos**, que permite definir y administrar permisos de manera granular. Además, se implementó la funcionalidad de **Inicio de Sesión** con autenticación JWT, que asegura la seguridad del acceso al sistema. La opción de **Cerrar Sesión** permite invalidar el token JWT y registrar el evento en la bitácora, capturando detalles como IP, fecha y hora. Finalmente, se añadió la funcionalidad de **Recuperar Contraseña**, completando los requisitos básicos para la gestión de accesos.

Este sprint sentó las bases del sistema al implementar funcionalidades críticas de seguridad y administración, utilizando tablas dinámicas, formularios reactivos y servicios RESTful que permiten una interacción eficiente con el sistema.

Sprint 2: Funcionalidades Médicas

En el segundo sprint, el desarrollo se centró en los módulos médicos, implementando los siguientes casos de uso:

1. **Gestión de Especialidades Médicas**: Se desarrolló un CRUD para mantener un catálogo de especialidades médicas, asegurando que no existan duplicados.
2. **Gestión de Médicos**: Se creó un módulo que permite registrar médicos, asociarlos a especialidades y mostrar sus horarios.
3. **Gestión de Horarios de Atención**: Se implementaron horarios dinámicos vinculados a médicos y especialidades, con validaciones para evitar traslapes.
4. **Gestión de Reservas de Fichas**: Los asegurados pueden realizar reservas desde la app móvil en Flutter, seleccionando horarios según disponibilidad.
5. **Gestión de Consultas Médicas**: Los médicos pueden registrar consultas y actualizar la historia clínica de los asegurados.
6. **Gestión de Postergaciones**: Se añadió una funcionalidad para que los médicos soliciten permisos de ausencia, con lógica para reprogramar automáticamente las citas afectadas y notificar a los asegurados.
7. **Gestión de Recetas y Medicamentos**: Los médicos pueden generar recetas, asociarlas a medicamentos y compartirlas con los asegurados a través del sistema.
8. **Gestión de Laboratorios**: Permite programar exámenes de laboratorio y registrar resultados asociados a consultas médicas.

Este sprint permitió la automatización de procesos médicos clave, integrando lógica avanzada como validaciones de traslapes, reprogramación automática de horarios y notificaciones para asegurados.

Sprint 3: Reportes y Funcionalidades Avanzadas

El tercer sprint se enfocó en funcionalidades avanzadas para finalizar el sistema, implementando módulos como:

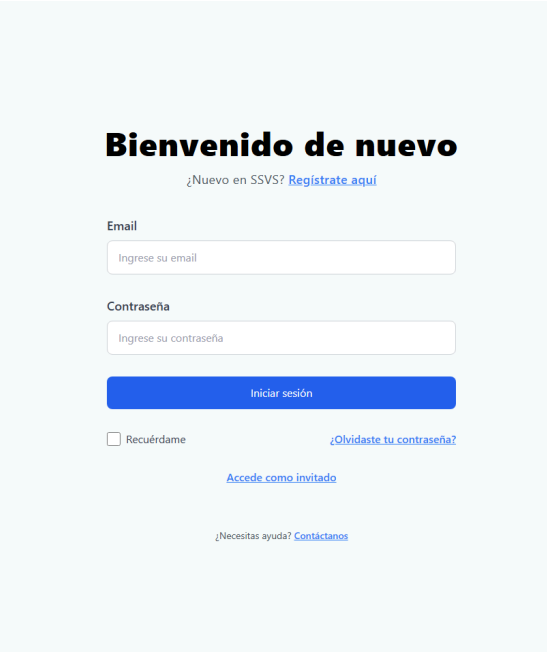
1. **Gestión de Reportes**: Se desarrollaron reportes en PDF y Excel para generar estadísticas sobre usuarios, reservas y consultas, disponibles para administradores.
2. **Visualización de Bitácora**: Se implementó un módulo para consultar eventos registrados, como inicios de sesión, acciones críticas y cierres de sesión, mejorando la trazabilidad.
3. **Notificaciones Dinámicas**: Se añadió la funcionalidad de notificaciones automáticas y visuales para asegurar la comunicación entre el sistema y los usuarios.
4. **Valoración de Médicos**: Se creó un módulo para que los asegurados puedan calificar a los médicos tras las consultas, mostrando un promedio de las valoraciones en el perfil del médico.
5. **Gestión de Pagos**: Desde la web, los administradores pueden gestionar los pagos realizados por los asegurados, mientras que la app móvil permite visualizar el historial de pagos, sin incluir opciones para realizar transacciones.

Este sprint consolidó el sistema como una solución integral, mejorando la trazabilidad mediante reportes y bitácoras, e incluyendo componentes interactivos como la valoración de médicos y notificaciones en tiempo real.

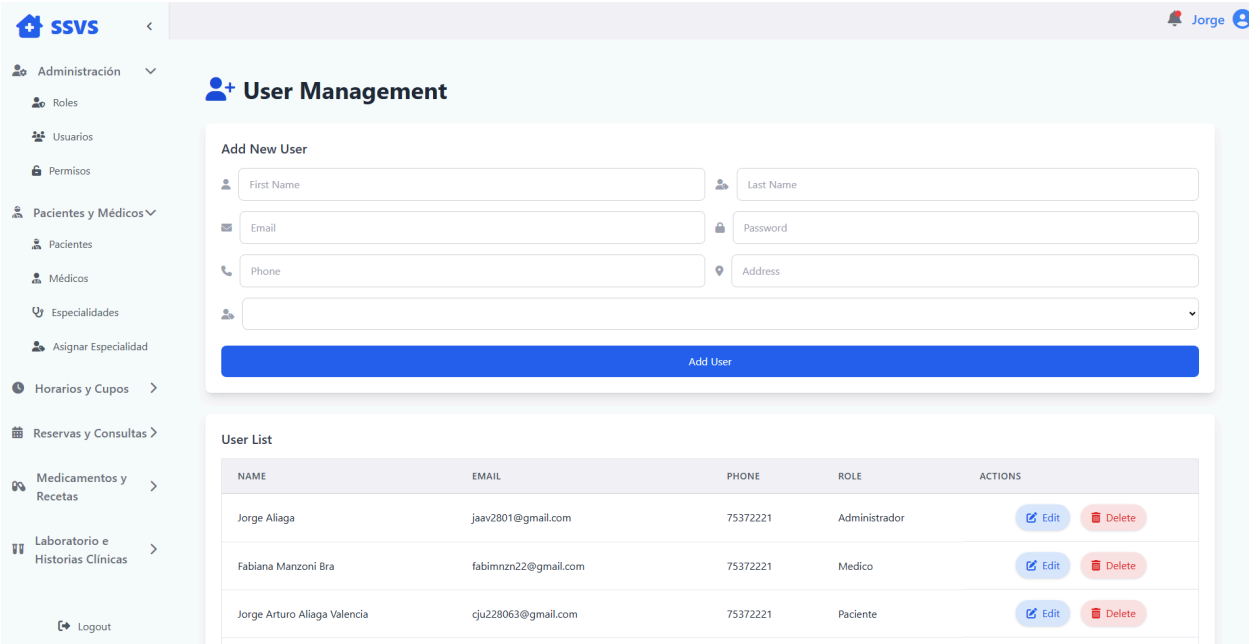
PEQUEÑO MANUAL

Administrador

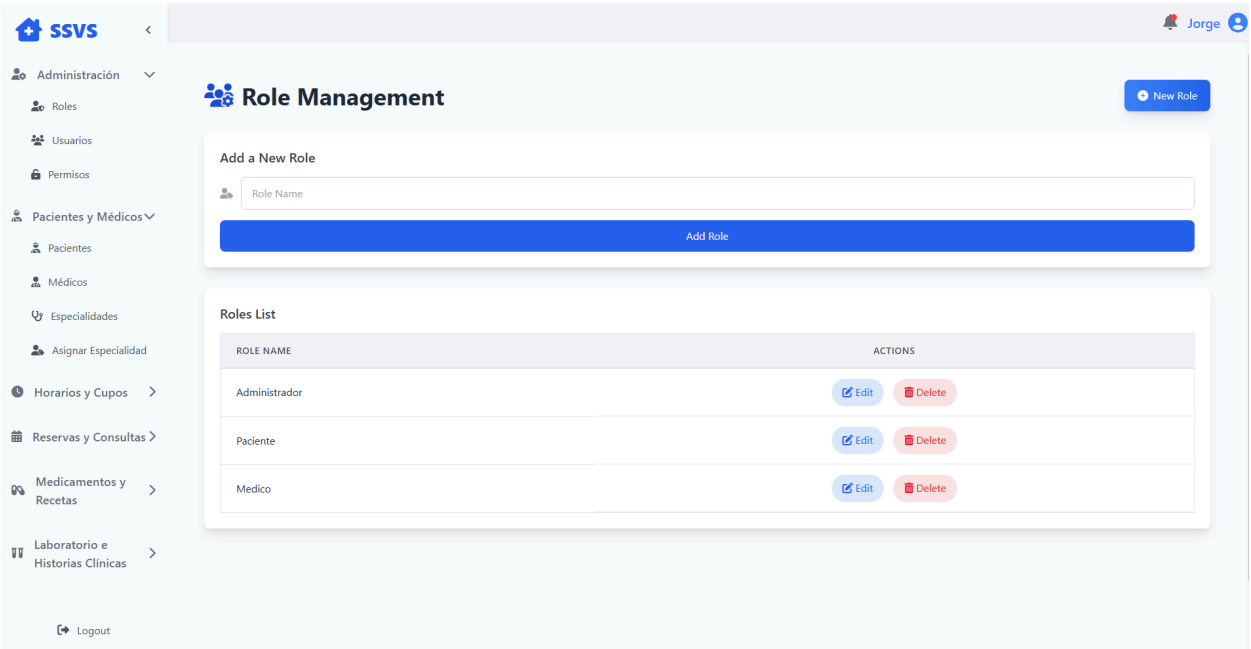
Para iniciar sesión, es necesario ingresar correctamente las credenciales de un perfil asignado con este rol.



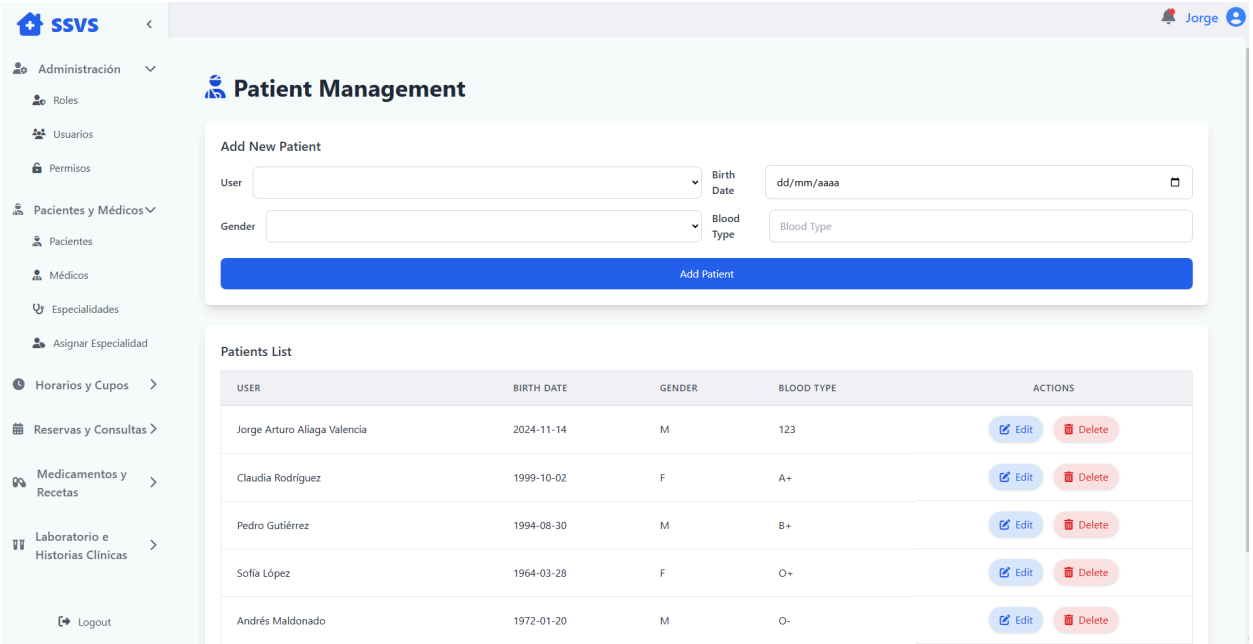
Una vez dentro, se dispondrá de un sidebar. Al hacer clic en "Usuario", se abrirá la pestaña destinada a la gestión de usuarios, permitiendo crear un nuevo usuario o modificar uno existente.



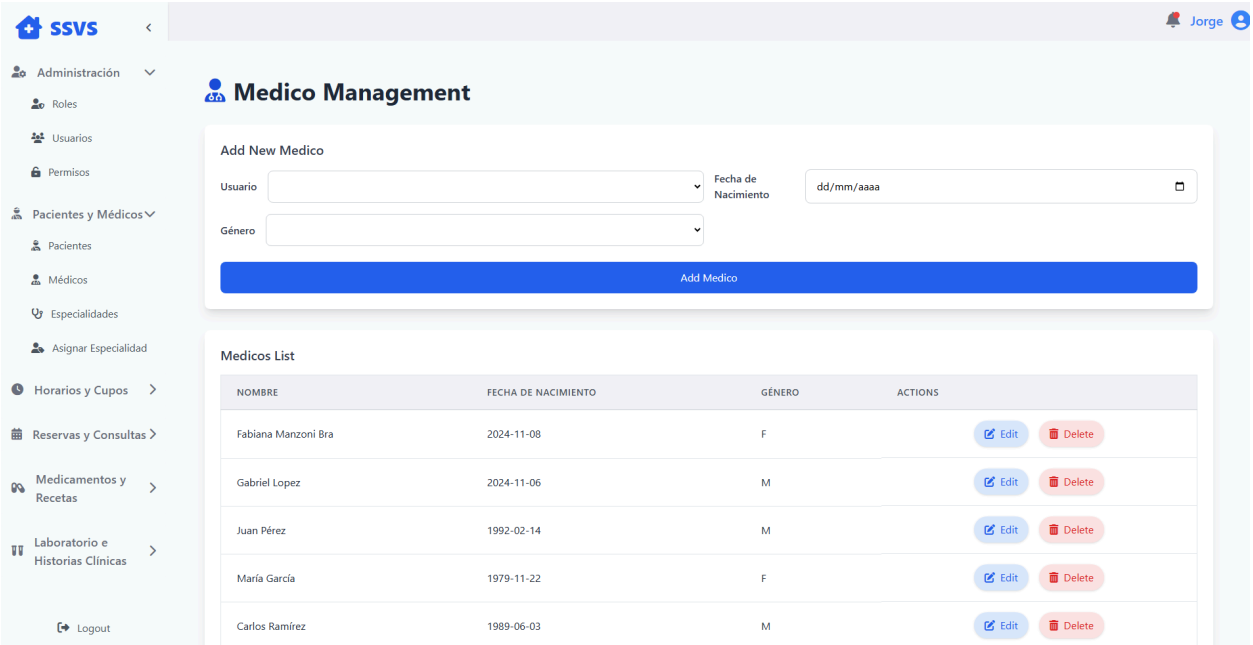
Al hacer clic en "Roles", se abrirá la pestaña destinada a la gestión de roles, permitiendo crear un nuevo rol o modificar uno existente.



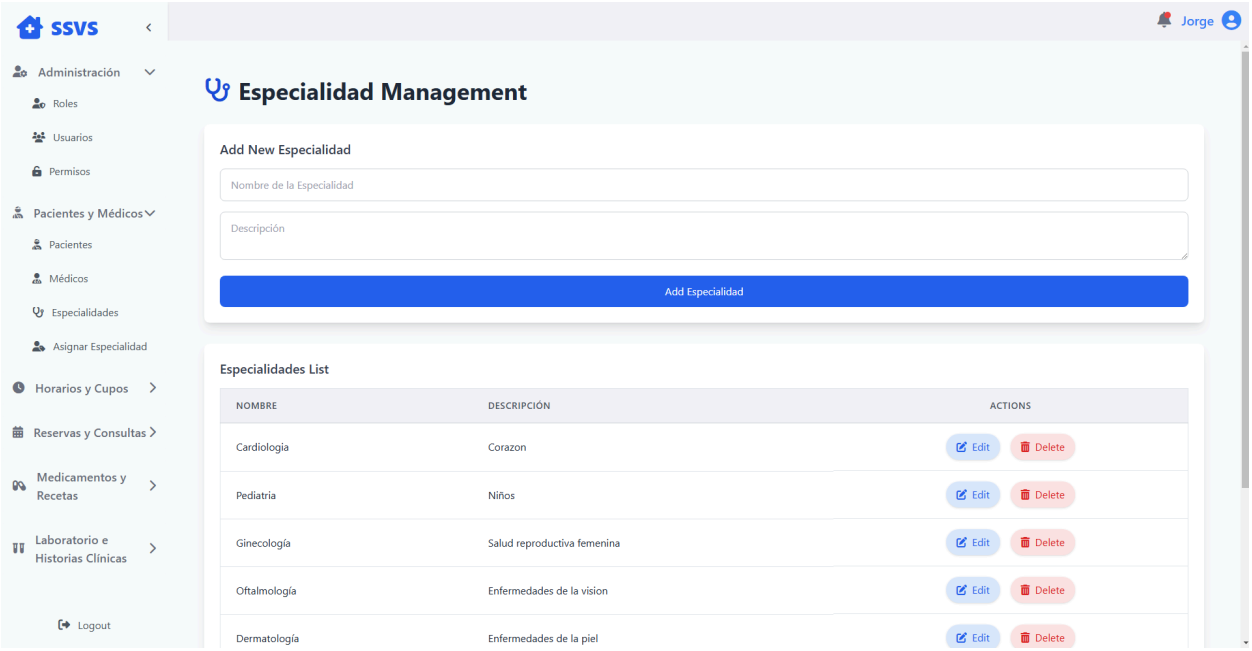
Al hacer clic en "Paciente", se abrirá la pestaña destinada a la gestión de pacientes, permitiendo crear un nuevo paciente o modificar uno existente.



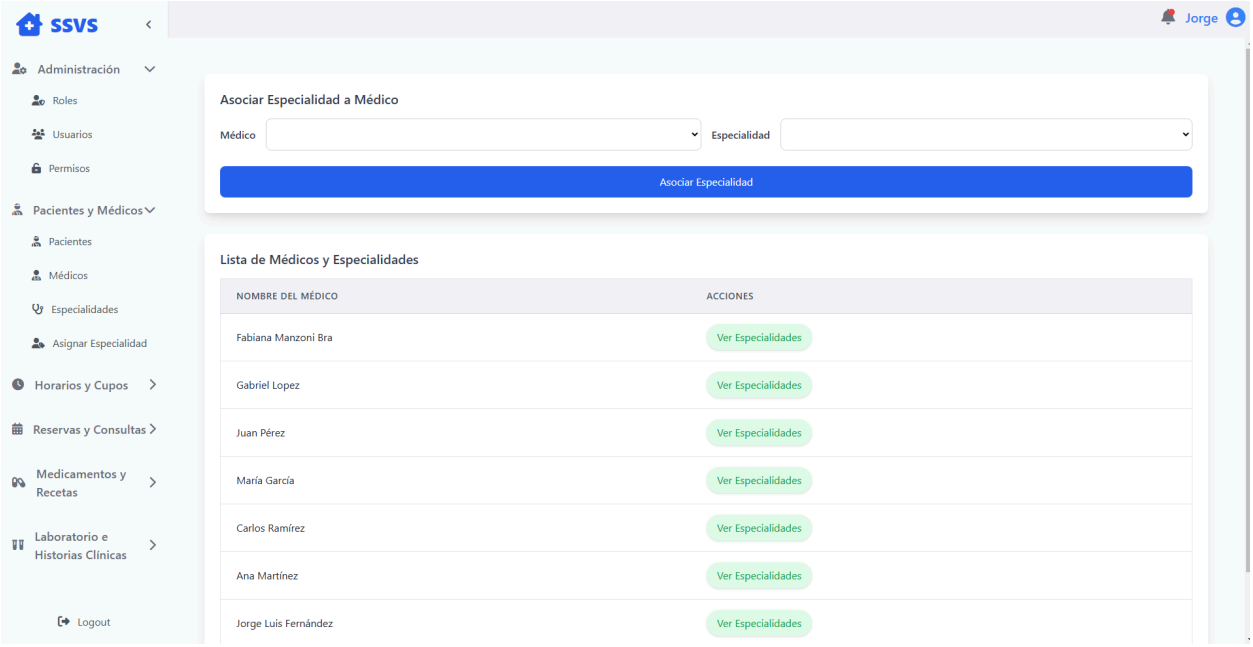
Al hacer clic en "Médico", se abrirá la pestaña destinada a la gestión de médicos, permitiendo crear un nuevo médico o eliminar uno existente.



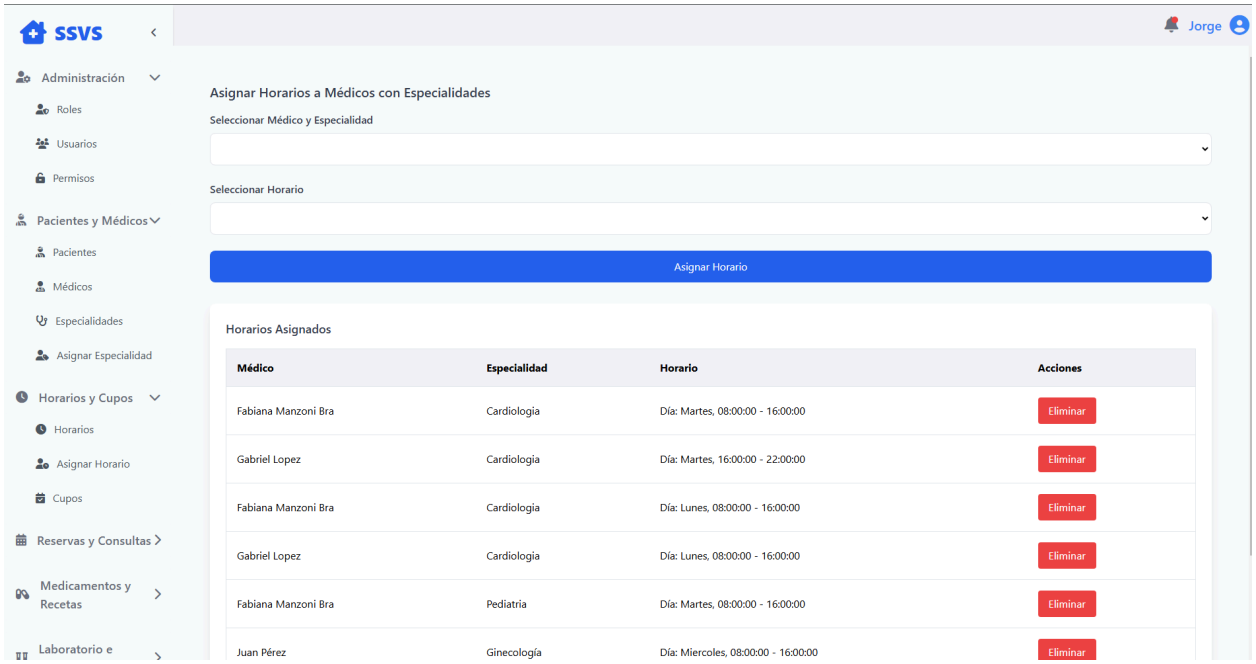
Al hacer clic en "Especialidad", se abrirá la pestaña destinada a la gestión de especialidades, permitiendo crear una nueva especialidad o eliminar una existente.



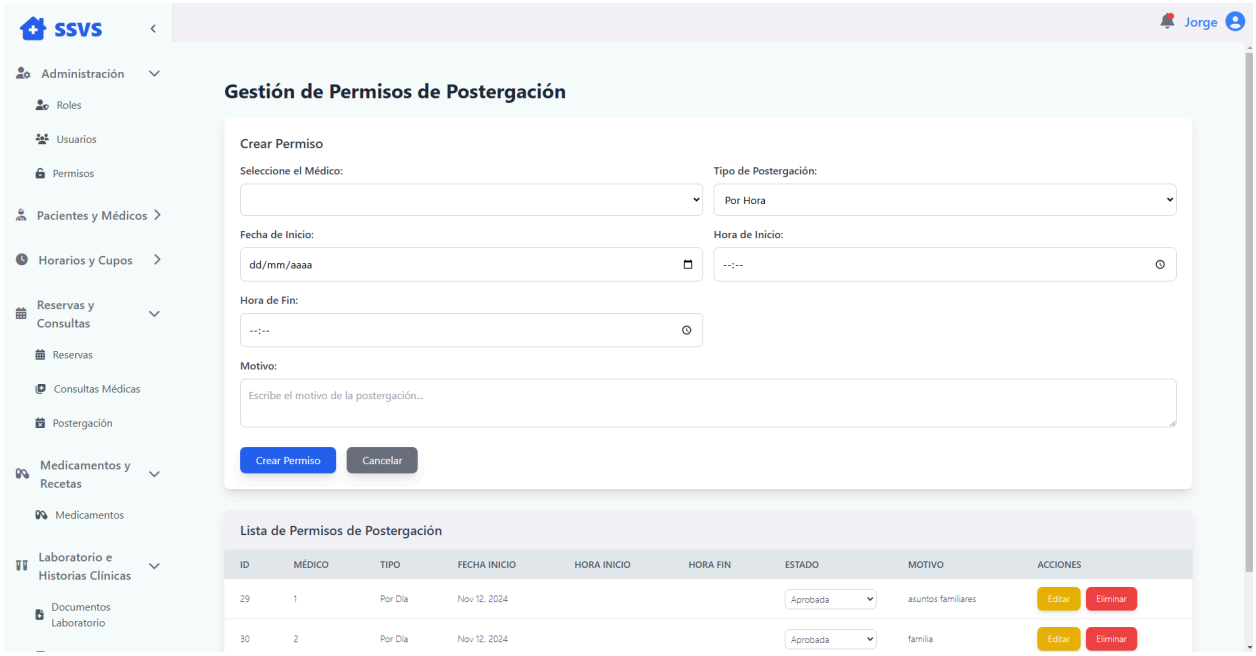
Al hacer clic en "Asociar Especialidad", se abrirá la pestaña destinada a asignar médicos a una especialidad, permitiendo asociar un nuevo médico a una especialidad o desvincular uno ya asignado.



El módulo permite asignar horarios a médicos en función de sus especialidades. Para ello, se selecciona el médico y su especialidad correspondiente, además del horario que se desea asignar. Una vez completada la selección, se hace clic en el botón 'Asignar Horario' para guardar la asignación. Las asignaciones realizadas se muestran en la tabla inferior, donde es posible visualizarlas y eliminarlas si es necesario.

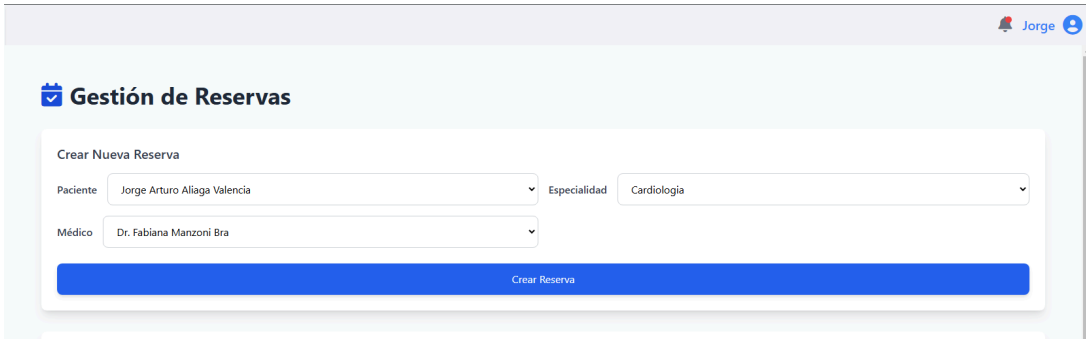


Selecciona el permiso de la lista, verifica los detalles, selecciona "Aprobar" o “Rechazar” para gestionar el permiso creado.

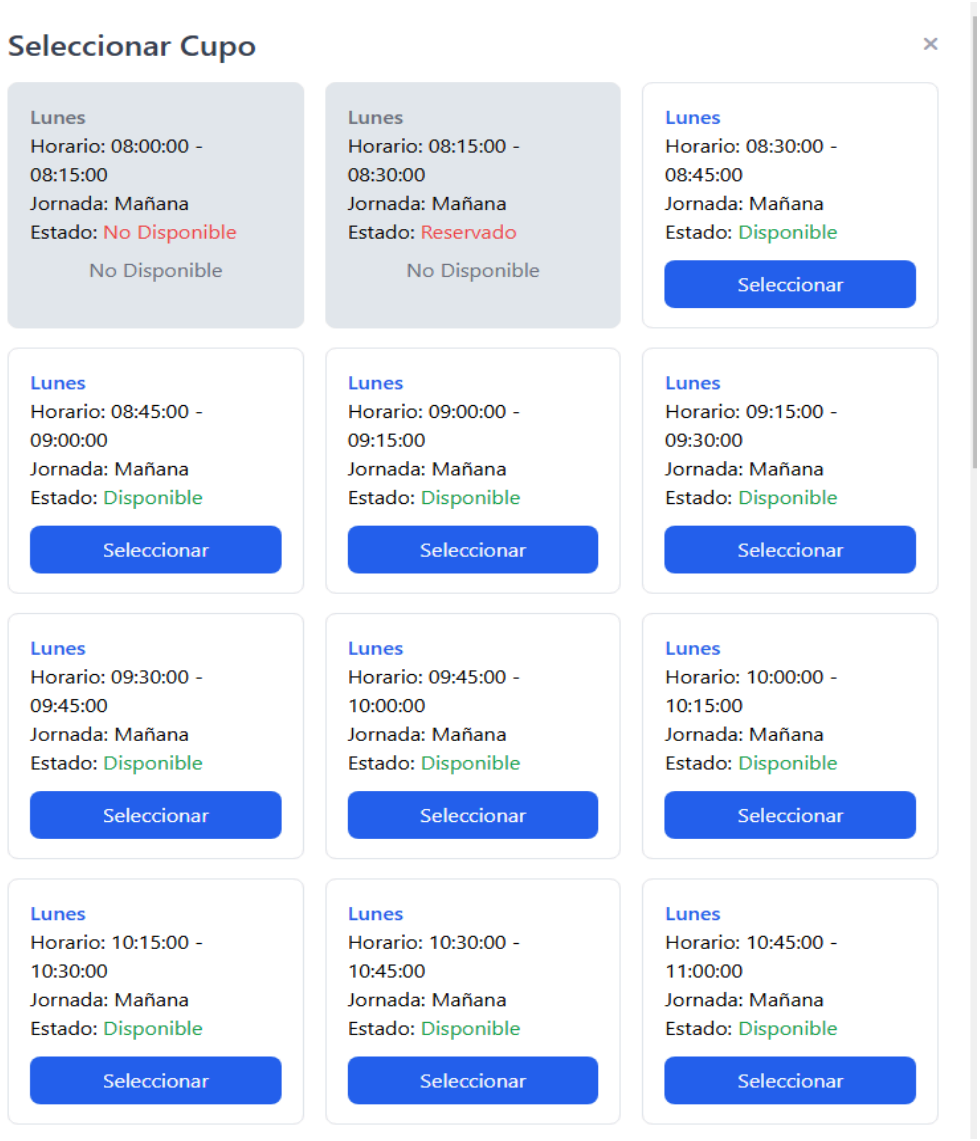


Paciente

Para crear una nueva reserva, selecciona al paciente, médico, especialidad y horario disponible, luego haz clic en "Crear Reserva".

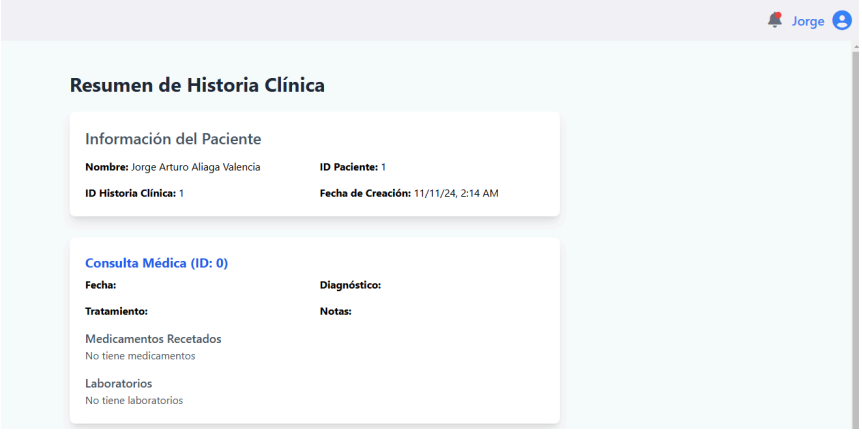


Para seleccionar un cupo, identifica un horario disponible en el selector y haz clic en "Seleccionar". Los horarios reservados estarán deshabilitados y no podrán ser seleccionados.



En la sección **Resumen de Historia Clínica**, se presenta la información del paciente y los detalles de sus consultas médicas. A continuación, las instrucciones para su uso:

1. **Información del Paciente**:
    - Revisa los datos básicos del paciente, como el nombre, ID del paciente, ID de la historia clínica y la fecha de creación.
2. **Consulta Médica**:
    - Visualiza el resumen de la consulta médica seleccionada, incluyendo la fecha, diagnóstico, tratamiento, notas, medicamentos recetados y laboratorios asociados.
    - Si no hay registros disponibles, se indicará con mensajes como "No tiene medicamentos" o "No tiene laboratorios".



Medico

Selecciona una reserva del desplegable y completa el formulario con el ID de la historia, diagnóstico, tratamiento y notas.

En la sección de Consultas Médicas Registradas, puedes realizar las siguientes acciones:

1. Visualiza la consulta médica registrada junto con el ID de la consulta, ID de la reserva, nombre del paciente, médico y fecha de consulta.
2. Haz clic en los botones de acciones:
    - Ver: Para consultar los detalles completos de la consulta.
    - Receta: Para generar o visualizar la receta asociada.
    - Laboratorio: Para gestionar los análisis de laboratorio relacionados.
    - Ver Lab: Para visualizar los resultados de laboratorio.

Crear Consulta Médica

Seleccionar Reserva:

Reserva #64 - Paciente: Andrés

Detalles de la Reserva Seleccionada

Fecha de Reserva	Nov 12, 2024
Estado	Cancelada
Médico	Gabriel Lopez
Especialidad	Cardiologia
Hora de Inicio	17:15:00
Hora de Fin	17:30:00

Formulario de Consulta

Historia ID:

6

Diagnóstico:

Gripe

Tratamiento:

Tomar los sgtes medicamentos

Notas:

Ver Recetas

Guardar Consulta

Consultas Médicas Registradas

CONSULTA ID	RESERVA ID	PACIENTE	MÉDICO	FECHA DE CONSULTA	ACCIONES
28	62	Claudia Rodríguez	Gabriel Lopez	11/12/24, 12:15 PM	Ver Receta Ver Receta Laboratorio Ver Lab

Crear Receta

Instrucciones:

Guardar Receta

Detalles de las Recetas

Receta #1
Fecha de Emisión: 11/12/24, 12:21 PM
Instrucciones: eada
Diagnóstico: mal estado
Tratamiento: a
Notas: aaa

Detalles del Paciente
Nombre: Claudia Rodríguez

Detalles del Médico
Nombre: Gabriel Lopez
Especialidad: Cardiologia

Detalles de la Consulta
Día de Atención: Martes
Hora de Inicio: 16:45
Hora de Fin: 17:00

Medicamentos Recetados

Medicamento	Dosis	Frecuencia	Duración
Omeprazol	11	1	2 días

Cerrar

Programar Laboratorio

Tipo de Análisis:
Ingrese el tipo de análisis

Descripción:

Fecha y Hora de Programación:
18/11/2024 09:27

Guardar Cancelar

Detalles del Laboratorio

Tipo de Análisis: Hemograma
Descripción: l
Fecha Programada: 11/18/24, 9:24 AM

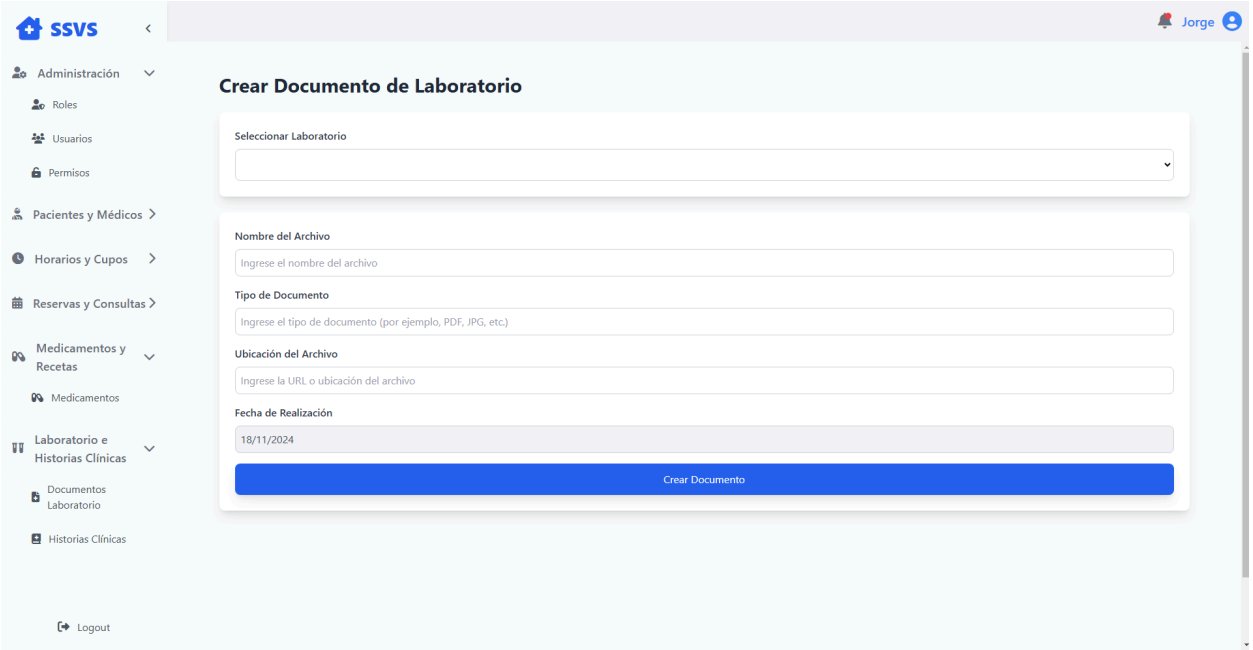
Cerrar

Para crear un documento de laboratorio, sigue estas instrucciones:

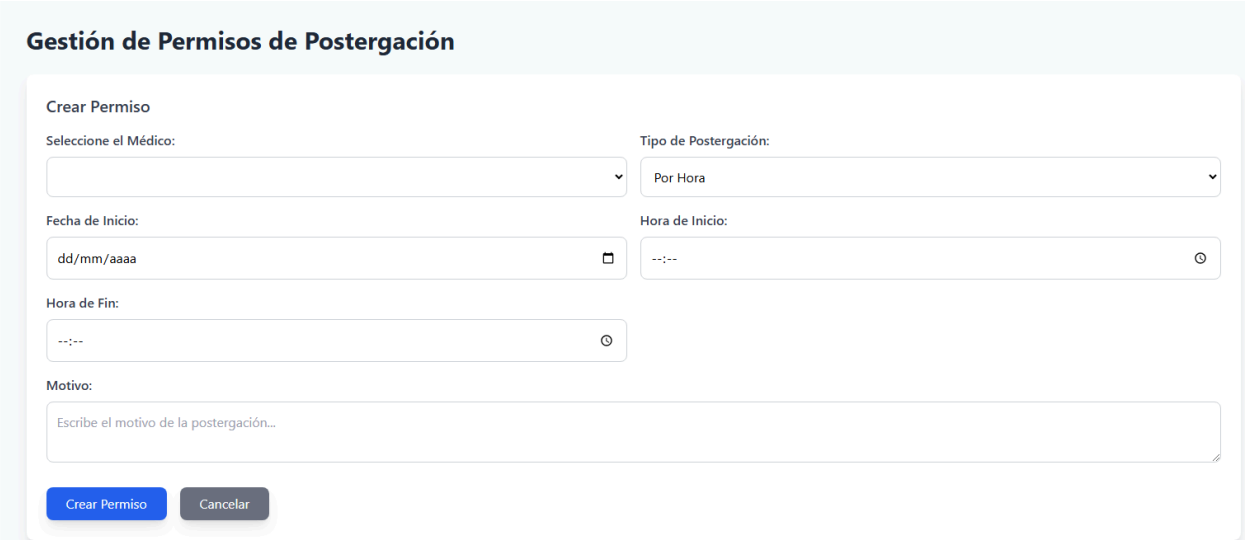
1. **Seleccionar Laboratorio**: Escoge el laboratorio relacionado desde el desplegable.
2. **Completar el formulario**:
   - Ingresa el **nombre del archivo**.
   - Especifica el **tipo de documento** (por ejemplo, PDF, JPG, etc.).

- Proporciona la **ubicación del archivo** (URL o ruta de almacenamiento).
- Selecciona o verifica la **fecha de realización**.
  Todas estas especificaciones vienen automáticas al subir el archivo

3. Haz clic en el botón **"Crear Documento"** para guardar el registro.



Define el tipo de postergación, especifica la fecha, hora de inicio y fin, ingresa el motivo, y haz clic en "Crear Permiso" para registrar la solicitud.
El médico se escoge automáticamente sacando datos del token al iniciar sesión.



ANEXOS

https://github.com/Aliaga23/ssvs_frontend

https://github.com/Aliaga23/ssvs_backend

https://ssvsfrontend-production.up.railway.app

Usuarios:

- Administrador:
  Usuario: jaav2801@gmail.com
  Contraseña: 1234
- Medico:
  Usuario: jaav28011@gmail.com
  Contraseña: 1234
- Asegurado:
  Usuario: cju228063@gmail.com
  Contraseña:123