PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

TEMA:

DESARROLLO DE PROTOTIPO DE UN SISTEMA WEB DE GESTIÓN DE CITAS MÉDICAS PARA EL CENTRO DE ESPECIALIDADES GALENUS

AUTOR:

RICHARD MOLINA

QUITO DM, JUNIO DE 2024

Índice

1 Capítulo I: Introducción 11

1.1 Marco de Referencia 11

1.1.1 Justificación 11
1.1.2 Planteamiento del Problema 11
1.1.3 Objetivo General 12
1.1.4 Objetivos Específicos 12
1.2 Antecedentes 12

1.3 Alcance 13

2 Capítulo II: Fundamentación Teórica 14

2.1 Marco Teórico 14

2.1.1 Metologogías Ágiles 14
2.1.2 Framework (Backend) 16
2.1.3 Framework (Frontend) 17
2.1.4 IDEs Backend 19
2.1.5 IDEs Frontend 20
2.1.6 Herramientas de Prueba 22
2.1.7 Herramientas de Contenedores 24
2.1.8 Base de Datos 25
2.1.9 Arquitectura 27
3 Capítulo III: Metodología 30

3.1 Extreme Programming (XP) 30

3.1.1 Características 30
3.1.2 Fases de la Metodología 30
4 Capítulo IV Levantamiento de Requerimientos 34

4.1 Levantamiento de requerimientos 34

4.1.1 Requerimientos Funcionales y No funcionales 35
5 Capítulo V: Desarrollo y Pruebas 36

5.1 Herramientas utilizadas en el desarrollo 36

5.1.1 Docker ..... **¡Error! Marcador no definido.**
5.1.2 Backend 37
5.1.3 Frontend 37
5.2 Iteración 1: Paciente 38

5.2.1 Planificación. 38
5.2.2 Diseño 40
5.2.3 Codificación 42
5.2.4 Pruebas 44
5.3 Iteración 2: Médico y Especialidad 46

5.3.1 Planificación. 46
5.3.2 Diseño 50
5.3.3 Codificación 55
5.3.4 Pruebas 58
5.4 Iteración 3: Reserva, Estado y Método de Pago 63

5.4.1 Planificación. 63
5.4.2 Diseño 68
5.4.3 Codificación 75
5.4.4 Pruebas 79
5.5 Iteración 4: Usuarios 86

5.5.1 Planificación. 86
5.5.2 Diseño 88
5.5.3 Codificación 92
5.5.4 Pruebas 94
5.6 Iteración 5: Recordatorios 97

5.6.1 Planificación. 97
5.6.2 Diseño 98
5.6.3 Codificación 99
5.6.4 Pruebas 99
6 Capítulo VI: Conclusiones y Recomendaciones 101

6.1 Conclusiones 101

6.2 Recomendaciones 102

7 Referencias Bibliográficas 103

8 Anexos 107

Índice de Figuras

Figura 1 Esquema de Diseño para Paciente 41

Figura 2 Esquema de Diseño con Modal en Paciente 42

Figura 3 Diseño Final de la pantalla de Pacientes 43

Figura 4 Diseño Final de la pantalla con modal de Pacientes 43

Figura 5 Prueba de Registro en Pacientes 44

Figura 6 Prueba de Edición en Pacientes 45

Figura 7 Prueba de Eliminación de Pacientes 45

Figura 8 Prueba de Consulta en Paciente 46

Figura 9 Esquema de Diseño para Médico 52

Figura 10 Esquema de Diseño con Modal en Médico 53

Figura 11 Esquema de Diseño para Especialidad 54

Figura 12 Esquema de Diseño con Modal en Especialidad 55

Figura 13 Diseño Final de la pantalla de Médicos 56

Figura 14 Diseño Final de la pantalla con modal de Médicos 56

Figura 15 Diseño Final de la pantalla de Especialidades 57

Figura 16 Diseño Final de la pantalla con modal de Especialidades 58

Figura 17 Prueba de Registro de Médicos 59

Figura 18 Prueba de Edición en Médicos 59

Figura 19 Prueba de Eliminación de Médicos 60

Figura 20 Prueba de Consulta en Médico 61

Figura 21 Prueba de Registro en Especialidades 61

Figura 22 Prueba de Edición en Especialidades 62

Figura 23 Prueba de Eliminación en Especialidades 63

Figura 24 Esquema de Diseño para Reserva 70

Figura 25 Esquema de Diseño con Modal en Reserva 71

Figura 26 Esquema de Diseño para Método de Pago 72

Figura 27 Esquema de Diseño con Modal en Método de Pago 73

Figura 28 Esquema de Diseño para Estado de Cita 74

Figura 29 Esquema de Diseño con Modal en Estado de Cita 75

Figura 30 Diseño Final de la pantalla de Reserva de citas 76

Figura 31 Diseño Final de la pantalla con modal de Reserva de citas 76

Figura 32 Diseño Final de la pantalla de Estado de la cita 77

Figura 33 Diseño Final de la pantalla con modal de Estados de cita 78

Figura 34 Diseño Final de la pantalla de Método de pago 78

Figura 35 Diseño Final de la pantalla con modal de Métodos de pago 79

Figura 36 Prueba de Registro en Reservas de citas 80

Figura 37 Prueba de Edición en Reservas de citas 80

Figura 38 Prueba de Eliminación en Reservas de citas 81

Figura 39 Prueba de Consulta en Reservas de citas 82

Figura 40 Prueba de Registro en Estados de Cita 82

Figura 41 Prueba de Edición en Estados de Cita 83

Figura 42 Prueba de Eliminación en Estados de Cita 84

Figura 43 Prueba de Registro en Métodos de Pago 84

Figura 44 Prueba de Edición en Métodos de Pago 85

Figura 45 Prueba de Eliminación en Métodos de Pago 86

Figura 46 Esquema de Diseño para Login 90

Figura 47 Esquema de Diseño para Usuario 91

Figura 48 Esquema de Diseño con Modal en Usuario 92

Figura 49 Diseño Final del Login 93

Figura 50 Diseño Final de la pantalla de Usuarios 93

Figura 51 Diseño Final de la pantalla con modal de Usuarios 94

Figura 52 Prueba de funcionamiento de Login 95

Figura 53 Prueba de Registro en Usuarios 96

Figura 54 Prueba de Edición en Usuario 96

Figura 55 Prueba de Eliminación en Usuario 97

Figura 56 Prueba de Recordatorio 99

Índice de Tablas

Tabla 1 Historia de Usuario Plantilla 30

Tabla 2 Plantilla de Release Plan 31

Tabla 3 Plantilla Tarjeta CRC 32

Tabla 4 Entrevista realizada al Dr. y Dueño de Galenus Patricio Pérez. 34

Tabla 5 Requerimientos Funcionales y No funcionales para el Sistema de Gestión de Citas Médicas 35

Tabla 6 Historia de Usuario Paciente - Registro 38

Tabla 7 Historia de Usuario Paciente – Edición 38

Tabla 8 Historia de Usuario Paciente – Eliminación 39

Tabla 9 Historia de Usuario Paciente – Consulta 39

Tabla 10 Tarjeta CRC Paciente 40

Tabla 11 Historia de Usuario Médico – Registro 47

Tabla 12 Historia de Usuario Médico – Edición 47

Tabla 13 Historia de Usuario Médico - Eliminación 47

Tabla 14 Historia de Usuario Médico - Consulta 48

Tabla 15 Historia de Usuario Especialidad – Registro 48

Tabla 16 Historia de Usuario Especialidad – Edición 49

Tabla 17 Historia de Usuario Especialidad – Eliminación 49

Tabla 18 Tarjeta CRC Médico 50

Tabla 19 Tarjeta CRC Especialidad 51

Tabla 20 Historia de Usuario Reserva - Registro 63

Tabla 21 Historia de Usuario Reserva - Edición 64

Tabla 22 Historia de Usuario Reserva - Eliminación 64

Tabla 23 Historia de Usuario Reserva - Consulta 65

Tabla 24 Historia de Usuario Estado - Registro 65

Tabla 25 Historia de Usuario Estado - Edición 66

Tabla 26 Historia de Usuario Estado - Eliminación 66

Tabla 27 Historia de Usuario Método de Pago - Registro 67

Tabla 28 Historia de Usuario Método de Pago - Edición 67

Tabla 29 Historia de Usuario Método de Pago - Eliminación 68

Tabla 30 Tarjeta CRC Reserva de Citas 68

Tabla 31 Tarjeta CRC Estado de las citas 69

Tabla 32 Tarjeta CRC Método de Pago para la cita 69

Tabla 33 Historia de Usuario Login 86

Tabla 34 Historia de Usuario, Usuario - Registro 87

Tabla 35 Historia de Usuario, Usuario - Edición 87

Tabla 36 Historia de Usuario, Usuario - Eliminación 88

Tabla 37 Tarjeta CRC Login 89

Tabla 38 Tarjeta CRC Usuarios 89

Tabla 42 Historia de Usuario Recordatorios 97

Tabla 43 Tarjeta CRC Recordatorios 98

Índice de Anexos

Anexo 1 Instancia de PostgreSQL en Docker 107

Anexo 2 Código de Backend – Paciente 108

Anexo 3 Código de Backend – Médico 108

Anexo 4 Código de Backend – Reserva 109

Anexo 5 Código de Frontend – Paciente 110

Anexo 6 Código de Frontend – Médico 110

Anexo 7 Código de Frontend – Reserva 111

1 Capítulo I: Introducción

1.1 Marco de Referencia

1.1.1 Justificación

El Centro de Especialidades Galenus hoy en día se encuentra en un proceso de innovación de sus procesos con el objetivo de garantizar la calidad y eficiencia de sus servicios, puesto que está buscando optimizar la experiencia tanto de los pacientes como de su personal médico y administrativo, a través del proceso de agendamiento de citas médicas dentro del establecimiento, el cual requiere una actualización tecnológica.

El desarrollo en conjunto con la implementación del sistema web para la gestión de citas médicas, adaptado a los requerimientos específicos de Galenus representa una gran iniciativa para responder de manera estratégica a los desafíos que presenta actualmente. Este sistema no solo simplificará y automatizará el proceso de las citas, sino que también permitirá una gestión más efectiva de la agenda médica de forma integral.

1.1.2 Planteamiento del Problema

El Centro de Especialidades Galenus, centro de atención médica en la ciudad del Tena, enfrenta desafíos significativos en la gestión médica dentro de sus instalaciones, lo que está afectando la eficiencia operativa y la calidad de la atención al paciente. Actualmente, Galenus maneja las citas médicas de manera manual, utilizando sistemas informáticos obsoletos, lo que aumenta la probabilidad de errores en la coordinación de citas entre los pacientes y el personal médico.

Por lo que se plantean las siguientes preguntas:

- ¿Cuáles son los requisitos necesarios para automatizar el proceso de gestión de citas médicas en el Centro de Especialidades Galenus?

- ¿Cuáles ventajas y beneficios se esperan para Galenus al implementar la automatización en la gestión de citas médicas?

1.1.3 Objetivo General

Desarrollar un prototipo de sistema web para la gestión de citas médicas en el Centro de Especialidades Galenus, con el objetivo de automatizar y optimizar el proceso, priorizando la eficiencia y mejorando significativamente la experiencia de los pacientes y de su personal médico.

1.1.4 Objetivos Específicos

1.1.4.1 Investigar y analizar las necesidades específicas del Centro de Especialidades Galenus en cuanto a la gestión de citas médicas.

1.1.4.2 Analizar el proceso actual de programación de citas e identificar áreas de mejora y posibles puntos de automatización.

1.1.4.3 Implementar recordatorios de las citas médicas dentro del prototipo del sistema web.

1.2 Antecedentes

El Centro de Especialidades Galenus comenzó su historia en el año 2012 donde el Sr. Patricio Pérez comenzó a desarrollar el sueño de su madre, el cual era tener un establecimiento médico. Comenta el Sr. Patricio que al principio era un sueño difícil de cumplir, ya que, en la ciudad del Tena, se dificultaba gestionar el negocio, puesto que no se había hecho un estudio de mercado detallado, y la ciudad donde iba a residir este establecimiento no era lo más adecuado para un centro privado por la situación económica y estilo de vida de la ciudad. Pero esta situación no fue impedimento para iniciar el proyecto Galenus, el centro médico comenzó con una sola planta donde contaba con una farmacia y con pocas especialidades a cargo del Dr. Patricio Pérez y su esposa la Dra. Beatriz Vera.

Con el pasar de los años este centro tomó el nombre de Centro de Especialidades Galenus, donde ganó gran reconocimiento a nivel territorial y regional, hoy en día cuanta con 12 especialidades con una amplia gama de profesionales especializados y equipamiento médico. Galenus continua en expansión al proyectar la implementación de un tercer piso para la unidad de cuidados intensivos (UCI).

1.3 Alcance

Este trabajo de titulación está orientado a levantar las necesidades del Centro de Especialidades Galenus, donde se desarrollará un sistema web que incluirá las funcionalidades de gestión de citas médicas, permitiendo al personal administrativo tener una mejor gestión de citas médicas y a los pacientes recibir recordatorios sobre el estado de su cita.

2 Capítulo II: Fundamentación Teórica

2.1 Marco Teórico

2.1.1 Metologogías Ágiles

2.1.1.1 SCRUM.

Según Ramírez et al. (2019), es una metodología ágil basada en buenas prácticas y trabajo en equipo, diseñada para obtener mejores resultados. Esta metodología tiene presentación de entregas parciales durante el desarrollo del proyecto y son validadas por el cliente.

2.1.1.1.1 Características:

- Se puede adelantar la planificación del proyecto, priorizando la gestión y la solución en base a costos, horarios y gestión de parámetros. (Ramírez et al., 2019)
- El proyecto se basará en los requerimientos del cliente. (Ramírez et al., 2019)
- El Equipo Scrum está conformado de seis a diez miembros, el ambiente de trabajo será para aprendizaje. (Ramírez et al., 2019)

2.1.1.2 Extreme Programming (XP).

Por otro lado, XP (Extreme Programming) como una metodología ágil busca el desarrollo rápido y funcional del proyecto, donde dan mayor valor a la colaboración con el cliente y el desarrollo incremental mediante iteraciones. (Penadés y Torres, 2006)

2.1.1.2.1 Características:

- Permite una rápida adaptación a los posibles cambios que se dan en las entregas de versiones. (Beck, 2002)

- Incluye buenas prácticas de desarrollo de software, que guían a los desarrolladores durante el proyecto. (Beck, 2002)
- Se realizan revisiones cada dos o tres semanas del desarrollo de los módulos espcificados para las entregas. (Beck, 2002)

2.1.1.3 Kanban.

Según Lendinez (2019), Kanban empezó su desarrollo en base a la técnica creada en Toyota para tener un mejor control del progreso en la cadena de suministro y esta ha sido adaptada al desarrollo de sofware donde busca gestionar proyectos mediante la visualización del trabajo y la mejora continua del mismo.

2.1.1.3.1 Características:

- Se usan tableros visuales para representar los elementos del flujo del proceso. (Lendinez, 2019)
- Limita el número total de tareas que se realizan simultáneamente para evitar retrasos dentro del desarrollo. (Lendinez, 2019)
- Se puede adaptar a cualquier proyecto porque no impone roles o reuniones cada cierta frecuencia. (Lendinez, 2019)

2.1.1.4 Metodología Seleccionada.

El motivo por el que se escogió la metodología XP es porque se tiene mayor experiencia en proyectos anteriormente realizados, tanto durante la carrera como en pasantías. Asimismo, al ser este proyecto a medida para la Clínica Galenus, se necesitará una comunicación frecuente con el cliente, tanto para el levantamiento de los requerimientos mediante las historias de usuario como las presentaciones de los módulos ya desarrollados.

2.1.2 Framework (Backend)

2.1.2.1 ASP.NET (C#).

Es un framework creado por Microsoft de código abierto basado en el lenguaje de programación C#, utilizado para la creación de servicios y aplicaciones web, incluyendo soporte para infraestructura de Modelo Vista Controlador (MVC) y Microservicios. (Arias, 2015)

2.1.2.1.1 Características:

- Ofrece seguridad avanzada mediante la integración de autenticaciones integradas en Windows. (Guérin, 2018)
- Tiene su propio ecosistema conectado a windows, lo que permite conectarse facilmente a herramientas como Visual Studio Code y Azure. (Guérin, 2018)
- Facilita el desarrollo, la implementación y el mantenimiento de aplicaciones web y los servicios web creados. (Guérin, 2018)

2.1.2.2 Django (Python).

Según Guilindro Moreira (2022), es un framework de desarrollo web basado en Python, que busca la simplicidad, la rapidez dentro de la creación de aplicaciones, dentro del mismo se pueden usar herramientas, lo que permite acelerar el desarrollo y la simplificación de tareas comunes.

2.1.2.2.1 Características:

- Tiene un ORM que se encarga de la comunicación con la Base de datos. (Guilindro Moreira, 2022)
- Es capaz de proteger nuestra aplicación de ataques de SQL injection o XSS (cross site scripting). (Guilindro Moreira, 2022)

- Incluye elementos middleware que dan protección como la normalización de URLs. (Guilindro Moreira, 2022)

2.1.2.3 Spring Boot (JAVA).

Según Haro (2019), Spring boot también es un framework para el desarrollo de aplicaciones, con la diferencia que el lenguaje usado es JAVA, ofrece un ecosistema de Spring Framework, donde podemos acceder a configuraciones y posibles simplificaciones al momento de desarrollar.

2.1.2.3.1 Características:

- Al tener servidores de aplicaciones embebidos como Tomcat permite la creación de aplicaciones de Spring independientes. (Haro, 2019)
- Reduce la complejidad de configuración de librerías de Spring y externas. (Haro, 2019)
- Se puede realizar empaquetamiento a través de Maven. (Haro, 2019)

2.1.2.4 Framework (Backend) Seleccionado.

La decisión de escoger Spring Boot en conjunto con su lenguaje JAVA, fue tomada por conocimientos académicos y profesionales, y también por la popularidad actual en el mundo de la programación.

2.1.3 Framework (Frontend)

2.1.3.1 Angular.

Es un framework que permite el desarrollo de aplicaciones web utilizando HTML y JavaScript, también está orientado a objetos, al trabajar con clases. Es popular en la comunidad de desarrolladores. (Oriols y Gutiérrez, 2020)

2.1.3.1.1 Características:

- Tiene mantenimiento por parte de Google. (Oriols y Gutiérrez, 2020)
- Su utilización para aplicaciones en dispositivos móviles está actualizada. (Oriols y Gutiérrez, 2020)
- Es fácil la encapsulación de la funcionalidad gracias al uso de los componentes. (Oriols y Gutiérrez, 2020)

2.1.3.2 React.js.

Según Anchundia Medrano (2022), es una biblioteca de JavaScript, que se enfoca en el desarrollo y diseño de interfaces de usuario, cuyo objetivo es facilitar el desarrollo de aplicaciones en una sola página. Le da mantenimiento Facebook y la comunidad de software libre.

2.1.3.2.1 Características:

- Resulta fácil aprender este framework, gracias a la simplicidad en su sintaxis. (Anchundia Medrano, 2022)
- Durante el diseño de las aplicaciones resalta el gran nivel de flexibilidad y máxima responsividad. (Anchundia Medrano, 2022)
- La migración entre versiones es muy sencilla y fácil de manejar. (Anchundia Medrano, 2022)

2.1.3.3 Svelte.

Según Bhardwaz y Godha (2023), este framework relativamente nuevo, al igual que los anteriores también está basado en JavaScript, el cual se centra en desarrollar aplicaciones más eficientes y ligeras, gracias a que el trabajo se ejecuta en conjunto con el tiempo de compilación y no en el de ejecución con el navegador.

2.1.3.3.1 Características:

- Permite la creación fácil de interfaces interactivas y responsivas. (Bhardwaz y Godha, 2023)
- Resulta un framework simple para dar mantenimiento a largo plazo. (Bhardwaz y Godha, 2023)
- Permite la creación de componentes que encapsulan la lógica y la presentación. (Bhardwaz y Godha, 2023)

2.1.3.4 Framework (Frontend) Seleccionado.

La elección de escoger Angular fue por motivos de experiencia académica y laboral sobre este framework. Además, Angular al ser un framework popular tiene mayor soporte en internet.

2.1.4 IDEs Backend para JAVA

2.1.4.1 IntelliJ IDEA.

Según Magaly et al. (2020), está desarrollado por JetBrains, actualmente es uno de los IDEs más inteligentes dentro del mercado, este IDE permite la participación de varios integrantes durante el desarrollo de manera eficiente.

2.1.4.1.1 Características:

- Contiene un autocomplaetado más sofisticado y sensitivo. (Magaly et al., 2020)
- Tiene soporte para Maven, que es una herramienta open-source que tiene como objetivo de simplificar procesos de compilación. (Magaly et al., 2020)
- Es compatible conel lenguaje de programación JAVA. (Magaly et al., 2020)

2.1.4.2 Eclipse.

Es un software que ya tiene tiempo en el mercado, este IDE está pensado para el lenguaje JAVA, aunque también tiene la posibilidad de agregar plugins para poder trabajar con otros lenguajes como: C++, Python, PHP, entre otros. (Soler-Adillon, 2017)

2.1.4.2.1 Características:

- El acceso a los archivos del proyecto es muy eficiente. (Soler-Adillon, 2017)
- Contiene el autocompletar al momento de desarrollar la aplicación. (Soler-Adillon, 2017)
- Tiene una gran cantidad de plugins para trabajar durante el desarrollo. (Soler-Adillon, 2017)

2.1.4.3 IDE Backend Seleccionado.

La elección del IDE para desarrollar el backend en Intellij IDEA, es porque al trabajar con el lenguaje JAVA se resulta más sencillo gracias a las ventajas que ofrece este IDE, con el autocompletado de código y su sencilles a la hora de desarrollo.

2.1.5 IDEs Frontend para Angular

2.1.5.1 Visual Studio Code.

Según Studio (2013), es un editor de código que mediante sus extensiones permite su uso como un IDE para desarrollar frontend, está mantenido por Microsoft y tiene compatibilidad con muchos frameworks, al trabajar con Angular, este editor de texto incluye extensiones para este framework.

2.1.5.1.1 Características:

- Tiene disponibilidad para Windows, macOS y Linux. (Studio, 2013)

- Es personalizable, al contener una extensa colección de extensiones disponibles. (Studio, 2013)
- Ofrece un gran conjunto de características como editor de texto, donde nos permite ver los errores de manera clara y a su vez posibles correciones o recomendaciones a la hora del desarrollo. (Studio, 2013)

2.1.5.2 WebStorm.

Es un IDE desarrollado y mantenido por Jetbrains enfocado al desarrollo web y frontend. Este IDE ofrece un conjunto completo de herramientas para desarrollar y depurar código HTML, CSS y JavaScript. (Montalvo Cano, 2022)

2.1.5.2.1 Características:

- Tiene compatibilidad con la mayoria de frameworks frontend como Angular, React, Vue.js, entre otros. (Montalvo Cano, 2022)
- Permite la integración sencilla con Git, GitHub, Mercurial y Subversion, permitiendo un control más sencillo de versionamiento. (Montalvo Cano, 2022)
- Ofrece un soporte integrado para pruebas unitarias y de integración. (Montalvo Cano, 2022)

2.1.5.3 Sublime Text.

Sublime Text es un editor de texto versátil y potente que ofrece una experiencia de edición eficiente y adaptable para una amplia gama de tareas de programación y desarrollo. (Kinder, 2013)

2.1.5.3.1 Características:

- Ofrece una interfaz de usuario minimalista y amplia gama de extensiones disponibles con complementos. (Kinder, 2013)

- Ofrece una alta velocidad, simplicidad y potencia. (Kinder, 2013)
- Permite la organización y gestión de proyectos de desarrollo de manera eficiente. (Kinder, 2013)

2.1.5.4 IDE Frontend Seleccionado.

El IDE escogido para desarrollar el front de nuestra aplicación es Visual Studio Code, por motivos de popularidad y las características que ofrece tanto como para personalizar nuestro ambiente de desarrollo como la extensa gama de extensiones que ofrece.

2.1.6 Herramientas de Prueba de Peticiones Frontend - Backend

2.1.6.1 Postman.

Según Hyams y Pilko (2022), postman tiene dos versiones, como cliente y en aplicación web desarrollada para crear, probar, documentar y compartir APIs de manera eficiente mediante el conjunto de herramientas que ofrece.

2.1.6.1.1 Características:

- Permite el envío de solicites HTTP como GET, POST, PUT, DELETE, etc. (Hyams y Pilko, 2022)
- Permite la definición de variables de entorno y variables globales para su uso en las solicitudes. (Hyams y Pilko, 2022)
- Permite la creación y ejecución de scripts mediante JavaScript. (Hyams y Pilko, 2022)

2.1.6.2 Thunder.

Según Tosic (2023), es una extensión/herramienta que se la puede encontrar en Visual Studio Code, la cual permite conectarnos a nuestro backend y realizar peticiones HTTP para probar el funcionamiento de nuestro programa.

2.1.6.2.1 Características:

- Permite a los desarrolladores una fácil creación, manejo y ejecución de solicitudes API. (Tosic, 2023)
- La herramienta permite el envio de solicitudes y la recepción de los datos de nuestra API. (Tosic, 2023)
- Puede modificar el formato de respuesta, ya sea JSON o CSV. (Tosic, 2023)

2.1.6.3 Selenium.

Selenium es un conjunto de utilidades que facilita la obtención de pruebas para aplicaciones web, está herramienta es multiplataforma, lo que permite su instalación en cualquier sistema operativo. Selenium permite grabar, editar y depurar casos de prueba. (Morales et al., 2017)

2.1.6.3.1 Características:

- Permite la ejecución automática e iterativa de las pruebas. (Morales et al., 2017)
- Facilidad de registro y ejecución de las pruebas. (Morales et al., 2017)
- Permite la creación de breakpoints dentro de la ejecución de las pruebas. (Morales et al., 2017)

2.1.6.4 Herramienta de Prueba Seleccionado

La herramienta escogida es Thunder por el motivo de que al trabajar con Visual Studio Code, solo tendríamos que agregar la extensión de Thunder para trabajar y probar el funcionamiento correcto de nuestras APIs desarrolladas en nuestro backend.

2.1.7 Herramientas de Contenedores

2.1.7.1 Docker.

Según Tortosa (2019), es una plataforma que permite crear instancias de varias aplicaciones en contenedores, donde se las puede gestionar. Incluye varios componentes y herramientas que permiten desarrollar y ejecutar aplicaciones de manera eficiente.

2.1.7.1.1 Características:

- Permite la creación de infraestructuras virtuales y la gestión de estas. (Tortosa, 2019)
- Se puede encapsular todo el entorno de trabajo, lo que permite trabajar en un servidor local. (Tortosa, 2019)
- Permite adquirir de manera rápida conocimiento sobre la gestión de contenedores. (Tortosa, 2019)

2.1.7.2 Kubernetes.

Según García (2018), desarrollada por Google, Kubernetes es una plataforma de código abierto que tiene como objetivo gestionar contenedores automatizando la implementación y el escalado. Actualmente está mantenido por Cloud Native Computing Foundation (CNCF)

2.1.7.2.1 Características:

- Los contenedores se distribuyen en pods, que a su vez estos pueden estar en varios nodos. (García, 2018)
- Gestiona de manera automática los contenedores creados. (García, 2018)
- Permite el escalado según los recursos que tenga nuestro computador. (García, 2018)

2.1.7.3 Podman.

Es un motor creado por Red Hat que es utilizado para la gestión y ejecución de contenedores. A diferencia de otras herramientas de gestión de contenedores, Podman puede ser ejecutado mediante comandos desde la raíz del sistema Linux o desde usuarios sin privilegios. (Gualotuña Llumiquinga, 2022)

2.1.7.3.1 Características:

- Es compatible con las especificaciones de contenedores abiertas (OCI). (Gualotuña Llumiquinga, 2022)
- Gestiona el almacenamiento de los contenedores mediante capas y soporta varias opciones de redes. (Gualotuña Llumiquinga, 2022)
- Permite la migración de otras herramientas de manejo de contenedores mediante la compatibilidad con imágenes y archivos de contenedores. (Gualotuña Llumiquinga, 2022)

2.1.7.4 Herramienta de Contenedores Seleccionada.

La herramienta para gestionar el contenedor que va a tener la Base de datos será Docker por las herramientas que ofrece para gestionar contenedores y la simplicidad que ofrece para gestionar dicho contenedor.

2.1.8 Base de Datos

2.1.8.1 SQL Server.

Según Vershinin y Mustafina (2021), desarrollado por Microsoft, SQL Server es un sistema de gestión de base de datos relacional diseñado para manejar una gran cantidad de datos en entornos corporativos, incluyendo transacciones de alta demanda y procesamiento analítico.

2.1.8.1.1 Características:

- Ofrece escalabilidad, estabilidad y seguridad. (Vershinin y Mustafina, 2021)
- Permite la creación de roles y usuarios. (Vershinin y Mustafina, 2021)
- Permite usar comandos DDL y DML de forma gráfica. (Vershinin y Mustafina, 2021)

2.1.8.2 MySQL.

Es un sistema gestor de base de datos relacional que se caracteriza por su popularidad al ser de código abierto y ser apoyado por Oracle Corporation. Además, tiene una comunidad activa que brinda documentación para garantizar un soporte continuo. (López Herrera, 2016)

2.1.8.2.1 Características:

- Es conocido por su rapidez, fiabilidad y facilidad de uso. (López Herrera, 2016)
- Es compatible con Windows, Linux y macOS.
- Ofrece mayor eficiencia en el desarrollo de aplicaciones web.

2.1.8.3 PostgreSQL.

Según López Herrera (2016), es un sistema de gestión de base de datos objeto-relacional basado en Postgres, es conocido por su robustez y conformidad con estándares SQL hasta su flexibilidad y extensibilidad, también se caracteriza por su avanzada seguridad, como la autenticación de columna y su sistema robusto de roles y permisos.

2.1.8.3.1 Características:

- Es compatible con ACID (atomicidad, consistencia, aislamiento, durabilidad). (López Herrera, 2016)
- Es compatible con Windows, Linux y macOS. (López Herrera, 2016)

- Ofrece soporte para tipos de datos avanzados como JSON, XML, HSTORE (para pares clave-valor). (López Herrera, 2016)

2.1.8.4 Base de Datos Seleccionada.

Se escoge el sistema de gestión de base de datos PostgreSQL por el motivo de la robustez que ofrece dentro de la seguridad, tanto para roles y permisos. También, la ventaja que se nos ofrece es la optimización de consultas de manera automática.

2.1.9 Arquitectura

2.1.9.1 Modelo Vista Controlador (MVC).

Según Bascón Pantoja (2004), MVC es un patrón de diseño de software que divide nuestra aplicación web en tres módulos: el modelo, que es el conjunto de clases que representan la información que el sistema procesará; las vistas, que muestran al usuario la información extraída del modelo y el controlador, objeto que engloba el giro de negocio de nuestra aplicación.

2.1.9.1.1 Características:

- La aplicación está implementada mediante módulos. (Bascón Pantoja, 2004)
- Permite realizar pruebas sencillas al tener módulos independientes. (Bascón Pantoja, 2004)
- Ofrece soporte y documentación para la mayoría de los lenguajes de programación populares. (Bascón Pantoja, 2004)

2.1.9.2 Microservicios.

Según Surianarayanan et al. (2019), es una arquitectura que organiza una aplicación como un conjunto de servicios pequeños y autónomos, permitiendo la división en módulos independientes. Esta estructura reduce significativamente la posibilidad de errores que

puedan deshabilitar la aplicación en su totalidad, ya que cada módulo opera de manera autónoma.

2.1.9.2.1 Características:

- Los servicios pueden ser desplegados de manera independiente. (Surianarayanan et al., 2019)
- Los microservicios se comunican entre sí utilizando protocolos como HTTP/HTTPS con JSON. (Surianarayanan et al., 2019)
- Ofrece la capacidad de aislar los errores al tener servicios individuales. (Surianarayanan et al., 2019)

2.1.9.3 Cliente - Servidor.

Esta arquitectura es un modelo donde hay sistemas distribuidos donde varios clientes realizan n números de solicitudes en múltiples servidores que procesan y responden cada una de estas. (Bircher, 2017)

2.1.9.3.1 Características:

- Permite la escalabilidad ya que se puede aumentar la capacidad de clientes y servidores por separado. (Bircher, 2017)
- Al tener únicamente dos componentes permite el fácil mantenimiento. (Bircher, 2017)
- Las solicitudes tienen un tiempo limitado, por el hecho de que una vez resuetas se eliminan por completo. (Bircher, 2017)

2.1.9.4 Arquitectura Seleccionada.

Se escogió la arquitectura de Microservicios por la ventaja que ofrece de crear módulos para gestionar nuestra aplicación, lo cual nos permite encontrar y corregir los errores

con mayor facilidad. Además, nos permite un fácil manejo de los datos al ser de tipo JSON la recepción y envío de datos a nuestra aplicación.

3 Capítulo III: Metodología

3.1 Extreme Programming (XP)

Según Letelier (2006), XP (Extreme Programming) es una metodología ágil en el desarrollo de software que contiene un conjunto de técnicas que otorgan dinamismo y flexibilidad al desarrollo de proyectos, esta metodología se centra en entregar un producto basado en los requerimientos exactos del cliente.

3.1.1 Características

Algunas características que se pueden definir de esta metodología son:

- La comunicación con el cliente es constante. (Letelier, 2006)
- La planificación es abierta con un cronograma flexible de las actividades. (Letelier, 2006)
- Los requerimientos al ser exactos, el desarrollo tendrá menos dificultad al ser a medida. (Letelier, 2006)

3.1.2 Fases de la Metodología

3.1.2.1 Planificación.

Una vez realizada la identificación de las historias de usuario, estas se descomponen en pequeñas versiones, las mismas que se irán revisando cada semana por el cliente, donde se debe obtener un software útil, funcional y listo para probar. (Beck, 2002)

Tabla 1

Historia de Usuario Plantilla

HISTORIA DE USUARIO	
Número:	**Nombre:**
Usuario:	

Modificación de Historia Número:	**Iteración Asignada:**
Prioridad en Negocio: (Alta / Media / Baja)	**Riesgo en desarrollo:** (Alta / Media / Baja)
Descripción:	
Observaciones:	

Nota. En este cuadro se representa la plantilla para desarrollar historias de usuario.

Según Beck (2002), estas historias de usuario se pueden ver reflejadas en el "Release plan", que tiene como objetivo indicar que historia de usuario se presentará en cada versión del programa en conjunto con su fecha de revisión.

Tabla 2

Plantilla de Release Plan

Módulo	**Nro. Iteración**	**Historia de Usuario**				**Fecha de revisión**
			Semanas Estimadas	**Días Estimados**	**Horas Estimadas**	
Estimación de tiempo:						

Nota. En este cuadro se representa la plantilla para desarrollar el Release Plan donde se incluye posibles fechas para los entregables de los módulos identificados.

3.1.2.2 Diseño

Según Beck (2002), en esta fase, la metodología XP sugiere que los diseños deben ser simples y sencillos, reduciendo la complejidad para conseguir un diseño fácilmente entendible e implementable que a largo plazo costará menos tiempo y esfuerzo desarrollar. En conjunto con la creación de diseños simples también se estandariza los nombres de las variables de nuestras clases, lo que ayudará en posteriores versiones del prototipo.

También se hace uso de tarjetas CRC (Class Responsibility and Collaboration), una herramienta valiosa en el desarrollo orientado a objetos. Estas tarjetas facilitan al desarrollador la tarea de conceptualizar y organizar las clases dentro de un sistema, enfocándose en tres aspectos clave: las clases (Class), sus responsabilidades (Responsibilities) y sus colaboraciones (Collaborations). (Beck, 2002)

Tabla 3

Plantilla Tarjeta CRC

TARJETA CRC		
Número:	**Escenario:**	
Nombre CRC:		
Responsabilidades	**Colaboradores**	**Métodos**
Observaciones:		

Nota. Este cuadro representa la plantilla de las Tarjetas CRC.

3.1.2.3 Codificación

Los desarrolladores en esta fase desarrollan desde la perspectiva de funcionalidad en base a cada historia de usuario, considerando los detalles comunicados por el cliente, el desarrollo de cada historia de usuario tendrá su respectiva revisión con el cliente para aclarar posibles dudas o correcciones dentro del módulo desarrollado. (Beck, 2002)

3.1.2.4 Pruebas

Después de completar cada historia de usuario se realizan pruebas que verifiquen que el módulo desarrollado esté funcionando correctamente, las pruebas dentro del desarrollo de

software es un paso crucial, ya que permiten identificar y corregir cualquier problema dentro del desarrollo de cada módulo. (Beck, 2002)

4 Capítulo IV Levantamiento de Requerimientos

4.1 Levantamiento de requerimientos

Durante las primeras reuniones con el personal y dueño del establecimiento se pudo reconocer algunos requisitos iniciales, donde se describía que el manejo actual de la gestión de citas médicas se realizaba de manera manual, como primer paso en el proceso era la atención normal al cliente donde se anotaba en un papel, los datos del paciente en caso de ser primera vez, para después anotar la especialidad, el médico y la hora de la cita. Finalmente, el proceso terminaba ingresando toda esta información de la hoja en un archivo Excel donde se encontraban todas las citas registradas de ese día.

Luego de las reuniones iniciales con el personal administrativo, se tuvo una reunión únicamente con el Dr. Patricio Pérez donde se realizó una pequeña entrevista que detallaba algunas descripciones y funcionalidades que desearía que se implementara dentro del sistema a desarrollar.

Tabla 4

Entrevista realizada al Dr. y Dueño de Galenus Patricio Pérez.

Preguntas	Respuestas
¿Cuáles son los principales desafíos que enfrenta su clínica en la gestión de citas médicas?	El Dr. Patricio Pérez resaltó principalmente el tiempo extenso en el registro de la cita médica, la falta de un sistema para gestionar las citas.
¿Qué características específicas le gustaría que incluyera el sistema de gestión de citas médicas?	El Dr. Patricio Pérez menciona que el sistema incluyera la gestión de pacientes, médicos, las especialidades de la clínica, la reserva de las citas médicas y que se implementen recordatorios sobre la cita un día antes de la fecha preestablecida.
¿Cuántos médicos y especialidades dispone su clínica?	El Dr. Patricio Pérez comenta que actualmente la clínica maneja 13 especialidades con un médico para cada

	especialidad, al igual con una cantidad de 13 médicos.
¿Qué tan importante es para usted la capacidad de enviar recordatorios de citas a los pacientes?	El Dr. Patricio Pérez comenta que es de gran importancia esta capacidad ya que algunos pacientes se les olvida la fecha y hora de la cita,
¿Hay algún requisito específico de seguridad que deba cumplir el sistema?	El Dr. Patricio Pérez menciona que como seguridad le gustaría tener un inicio de sesión donde se controle el acceso a la gestión de la información.
En base a la pregunta anterior: ¿Qué usuarios tendrán acceso al sistema?	El Dr. Patricio Pérez comenta que por ahora le gustaría comenzar con el personal de recepción y el usuario administrador.
¿Tiene alguna preferencia en cuanto a la plataforma en la que se desarrolle el sistema (web, móvil)?	El Dr. Patricio Pérez recomienda que el sistema le gustaría que fuera desarrollado de manera web.

Nota. El cuadro describe toda la entrevista realizada al Dr. Patricio Pérez, donde se describe algunas descripciones, características y recomendaciones que tendrá el sistema para la gestión de citas médicas.

4.1.1 Requerimientos Funcionales y No funcionales

Tabla 5

Requerimientos Funcionales y No funcionales para el Sistema de Gestión de Citas Médicas

Requerimientos Funcionales	Requerimientos No Funcionales
El sistema permita la gestión de Pacientes y Médicos proporcionando información personal como cédula, nombre, dirección, celular, dirección, correo, sexo, entre otros.	El sistema debe incluir un control de acceso y permisos adecuados.
El sistema permita la gestión de Especialidades de la Clínica.	El sistema debe dar una respuesta rápida, especialmente en la búsqueda y programación de citas
El sistema permita el inicio de sesión para los dos usuarios mencionados.	El sistema debe soportar el crecimiento de pacientes y médicos.
El sistema permita la gestión de citas con los médicos y horarios disponibles.	El sistema debe ser compatible con diferentes navegadores.
El sistema permita el envío de notificaciones a los pacientes un día antes de la cita.	

Nota. En esta tabla se registran los requerimientos funcionales y no funcionales del sistema a desarrollar

Después de que la encuesta fue realizada se tuvo una última reunión con el Dr. Patricio Pérez donde se trataron temas como el orden del desarrollo de cada uno de los módulos y sus respectivas fechas de entrega y revisión. A continuación, se presentará en orden cada una de las iteraciones con todas las fases de la Metodología XP.

5 Capítulo V: Desarrollo y Pruebas

5.1 Herramientas utilizadas en el desarrollo

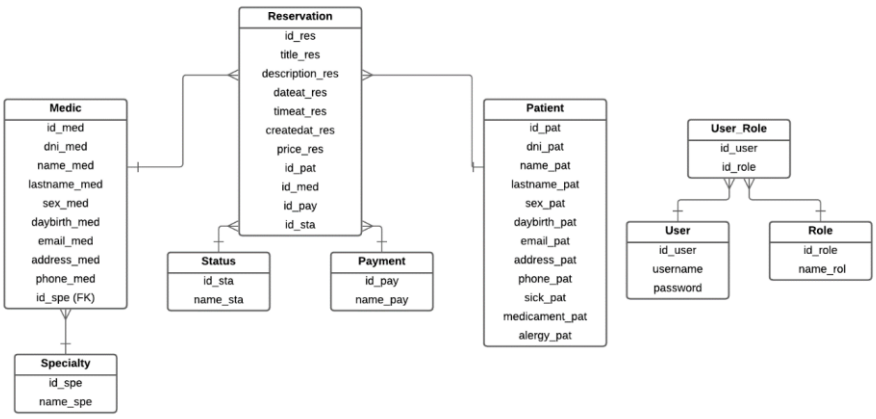
5.1.1 Diseño de Base de Datos

En la herramienta Docker, que actúa como gestor de contenedores, se establecerá una instancia de PostgreSQL. Esta instancia será utilizada para generar automáticamente la estructura de la base de datos requerida para la gestión de citas médicas, abarcando cada una de las clases identificadas en el proceso de levantamiento de requerimientos.

Por lo que el modelo físico de la base de datos quedaría de la siguiente forma:

Ilustración 1

Modelo Físico de BDD



Nota. En esta figura se muestra el modelo físico de la base de datos para nuestro sistema.

5.1.2 Backend

En el desarrollo del backend se va a utilizar la librería Lombok, la cual es una herramienta que funciona en conjunto con Spring Boot. Esta biblioteca proporciona una extensa variedad de utilidades para la codificación. Entre sus características destacadas se encuentra la simplificación de la creación de constructores, getters y setters para los atributos de nuestras clases en desarrollo. Además, Lombok ofrece configuración automática para las capas de repositorio, servicio y controladores, agilizando considerablemente el proceso de desarrollo.

5.1.3 Frontend

Para el diseño del frontend, se tomó la decisión de adquirir una plantilla de Angular Material denominada Trinta. Esta plantilla se destacó por su diseño moderno y atractivo, lo cual fue aceptado por el Dr. Patricio Pérez. Considerando su opinión positiva, se procedió a

personalizar y adaptar la plantilla Trinta para que se ajustara a las necesidades específicas del sistema web que se estaba desarrollando.

5.2 Iteración 1: Paciente

5.2.1 Planificación.

Luego de la encuesta, se dividió en módulos el sistema, para gestionar por partes cada elemento de este. El Dr. Patricio sugirió comenzar con la gestión del paciente, donde se pueda registrar pacientes, editar información en caso de estar errónea y eliminar pacientes.

Por lo que se plantea la primera historia de usuario de la siguiente forma:

Tabla 6

Historia de Usuario Paciente - Registro

HISTORIA DE USUARIO	
Número: 1	**Nombre:** Registro de paciente
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Primera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se coloquen los datos del paciente.	
Observaciones: Dentro de los campos debe haber tres que se especifique si el paciente tiene o no lo siguiente: enfermedad, alergia y si toma medicamentos.	

Nota. En esta historia de usuario se detalla descripciones para el registro de la información del paciente.

Tabla 7

Historia de Usuario Paciente – Edición

HISTORIA DE USUARIO

Número: 2	**Nombre:** Edición de paciente
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Primera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se carguen los datos del paciente para luego editar el registro de este.	
Observaciones:	

Nota. En esta historia de usuario se detalla descripciones para la edición de la información del paciente.

Tabla 8

Historia de Usuario Paciente – Eliminación

HISTORIA DE USUARIO	
Número: 3	**Nombre:** Eliminación de paciente
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Primera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un botón eliminar de manera definitiva al paciente y sus datos.	
Observaciones:	

Nota. En esta historia de usuario se detalla descripciones para la eliminación del paciente.

Tabla 9

Historia de Usuario Paciente – Consulta

HISTORIA DE USUARIO	
Número: 4	**Nombre:** Consulta de pacientes
Usuario: Administrador y Recepcionista	

Modificación de Historia Número: NA	**Iteración Asignada:** Primera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un cuadro de búsqueda, consultar o filtrar por cierto dato incluido en cada paciente.	
Observaciones:	

Nota. En esta historia de usuario se describe cómo será el método de búsqueda o un filtro en la información del paciente.

5.2.2 Diseño

Según lo definido en la fase anterior se puede crear un borrador sobre cómo será visualmente este módulo del paciente. Y también añadir la documentación de esta fase que sería la tarjeta CRC.

Tabla 10

Tarjeta CRC Paciente

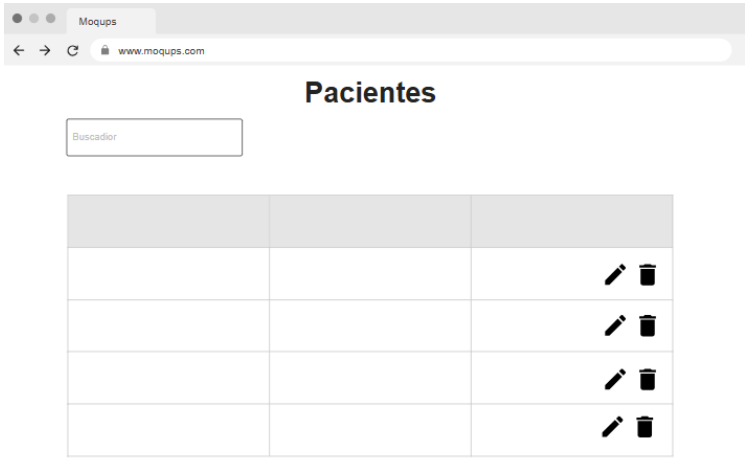
TARJETA CRC		
Número: 1	**Escenario:** Registro, Edición y Eliminación de Pacientes	
Nombre CRC: Pacientes		
Responsabilidades: • Almacenar y gestionar información personal del paciente. • Proporcionar información del paciente para consultas médicas. • Actualizar información del paciente. • Eliminar información del paciente.	**Colaboradores:**	**Métodos:** • crearPaciente() • consultarPaciente() • editarPaciente() • eliminarPaciente()
Observaciones:		

Nota. Dentro de esta tabla podemos observar los métodos y acciones que tendrá la clase de Pacientes.

5.2.2.1 Diseño de referencia:

Figura 1

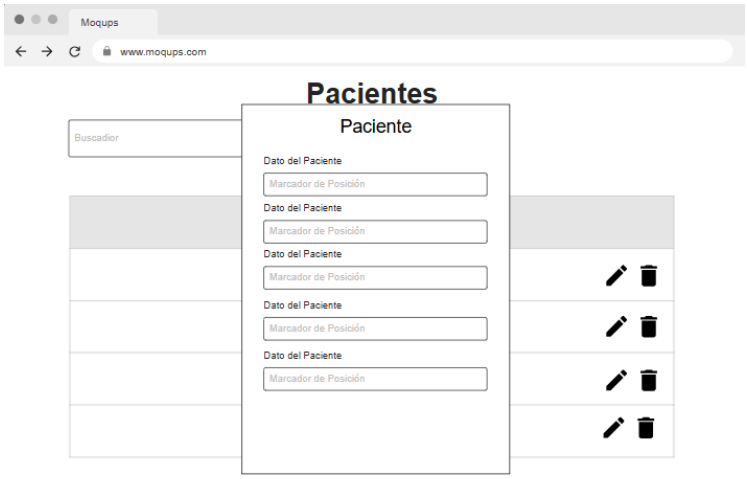
Esquema de Diseño para Paciente



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla del Paciente.

Figura 2

Esquema de Diseño con Modal en Paciente



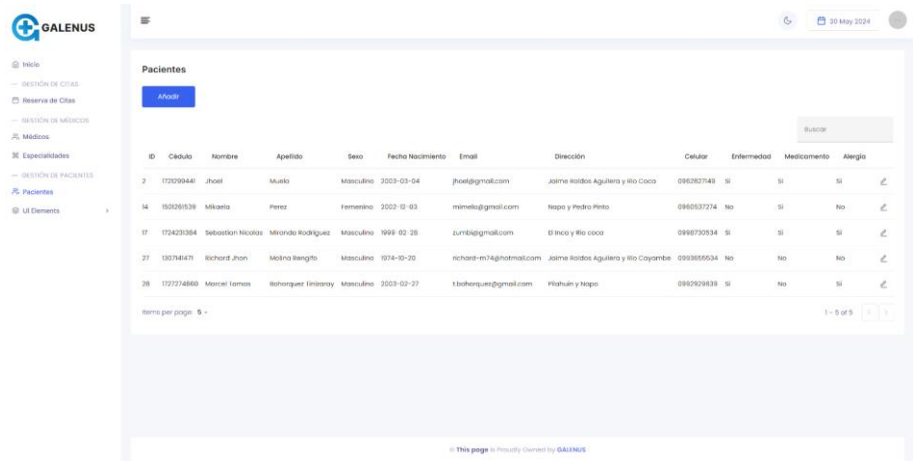
Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del paciente

5.2.3 Codificación

Ahora se procederá al desarrollo del módulo de la gestión del paciente, con una visualización de lo que sería la interfaz de usuario:

Figura 3

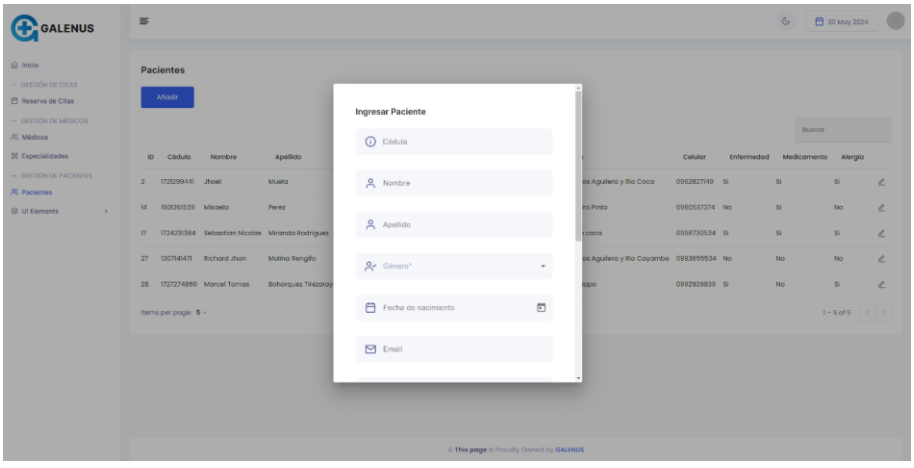
Diseño Final de la pantalla de Pacientes



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Pacientes donde se observa el listado de los pacientes registrados, con botones de añadir, editar y eliminar.

Figura 4

Diseño Final de la pantalla con modal de Pacientes



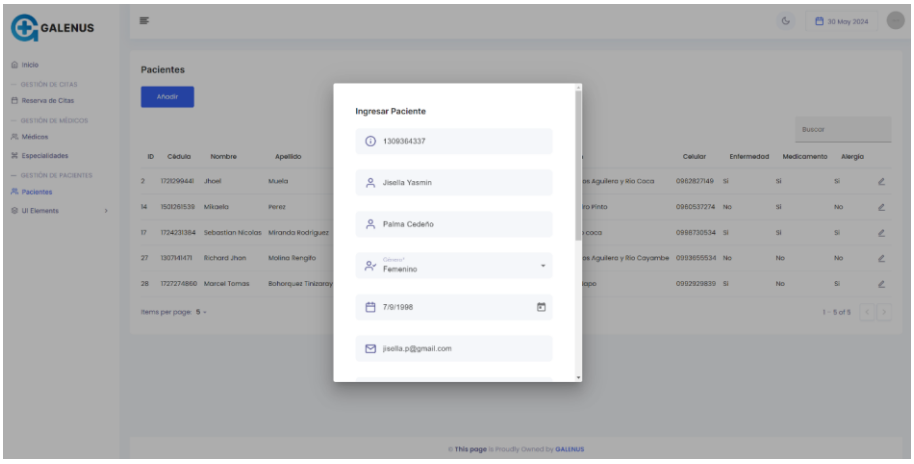
Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Pacientes.

5.2.4 Pruebas

Una vez ya finalizado el módulo del paciente se procederá con las pruebas con el Dr. Patricio quien probará cada funcionalidad de este módulo.

Figura 5

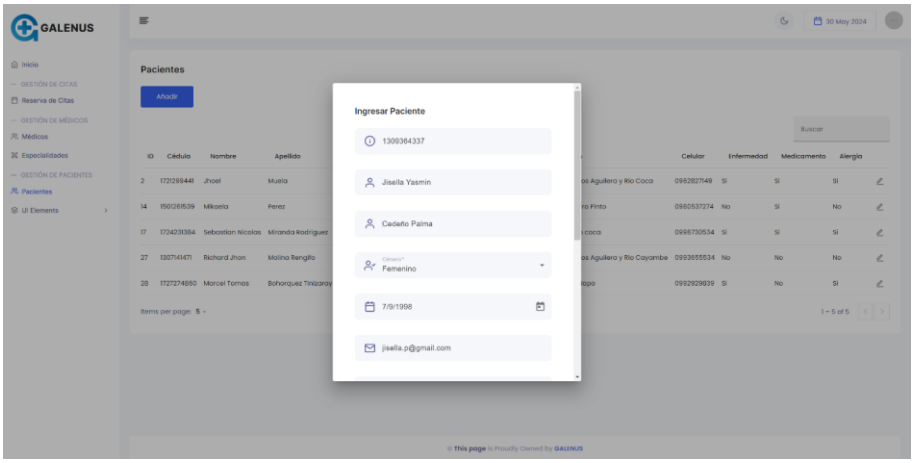
Prueba de Registro en Pacientes



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Pacientes.

Figura 6

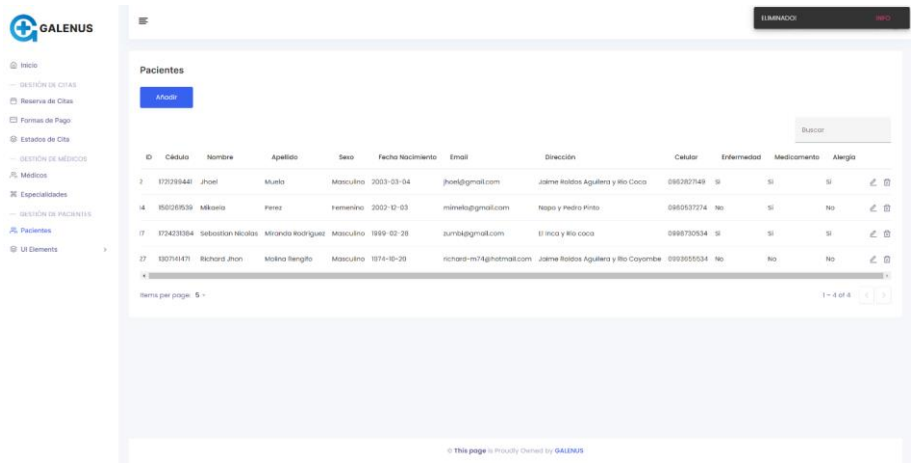
Prueba de Edición en Pacientes



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de un Paciente.

Figura 7

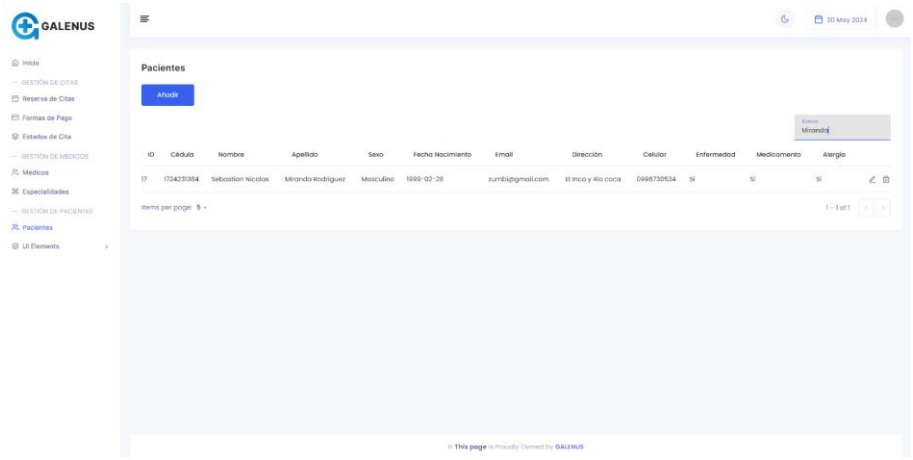
Prueba de Eliminación de Pacientes



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de un Paciente.

Figura 8

Prueba de Consulta en Paciente



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez filtrando los Pacientes.

5.3 Iteración 2: Médico y Especialidad

5.3.1 Planificación.

Como segundo módulo el médico fue escogido para desarrollar, en este módulo se debe agregar su especialidad, y Galenus al contar con una cantidad considerable de especialidades se ha considerado aumentar un tercer módulo para la gestión de especialidades.

Por lo que se plantea las historias de usuario de la siguiente forma:

Tabla 11

Historia de Usuario Médico – Registro

HISTORIA DE USUARIO	
Número: 5	**Nombre:** Registro de médico
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se coloquen los datos del médico.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para el registro de médicos.

Tabla 12

Historia de Usuario Médico – Edición

HISTORIA DE USUARIO	
Número: 6	**Nombre:** Edición de médico
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se carguen los datos del médico para luego editar el registro de este.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición de médicos.

Tabla 13

Historia de Usuario Médico - Eliminación

HISTORIA DE USUARIO

Número: 7	**Nombre:** Eliminación de médico
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un botón eliminar de manera definitiva al médico y sus datos.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la eliminación de médicos.

Tabla 14

Historia de Usuario Médico - Consulta

HISTORIA DE USUARIO	
Número: 8	**Nombre:** Consulta de médicos
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un cuadro de búsqueda, consultar o filtrar por cierto dato incluido en cada médico.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la consulta en médicos.

Tabla 15

Historia de Usuario Especialidad – Registro

HISTORIA DE USUARIO	
Número: 9	**Nombre:** Registro de especialidad
Usuario: Administrador	

Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se coloque el nombre de la especialidad.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para el registro de especialidades.

Tabla 16

Historia de Usuario Especialidad – Edición

HISTORIA DE USUARIO	
Número: 10	**Nombre:** Edición de especialidad
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se cargue el nombre de la especialidad para luego editar el registro de este.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición de especialidades.

Tabla 17

Historia de Usuario Especialidad – Eliminación

HISTORIA DE USUARIO	
Número: 11	**Nombre:** Eliminación de especialidad
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Segunda
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)

Descripción: Mediante un botón eliminar de manera definitiva la especialidad.
Observaciones:

Nota. En esta tabla se presenta la historia de usuario para la eliminación de especialidades.

5.3.2 Diseño

Según lo definido en la fase anterior se procederá con la creación del borrador sobre el diseño de las pantallas de especialidad y médico. Y también se añade la documentación de esta fase que sería la tarjeta CRC.

Tabla 18

Tarjeta CRC Médico

TARJETA CRC		
Número: 2	**Escenario:** Registro, Edición y Eliminación de Médicos	
Nombre CRC: Médicos		
Responsabilidades: • Almacenar y gestionar información personal del médico. • Proporcionar información del médico para consultas médicas. • Actualizar información del médico. • Eliminar información del médico.	**Colaboradores:** • Especialidad	**Métodos:** • crearMedico() • consultarMedico () • editarMedico () • eliminar Medico () • consultarEspecialidad()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de médicos.

Tabla 19

Tarjeta CRC Especialidad

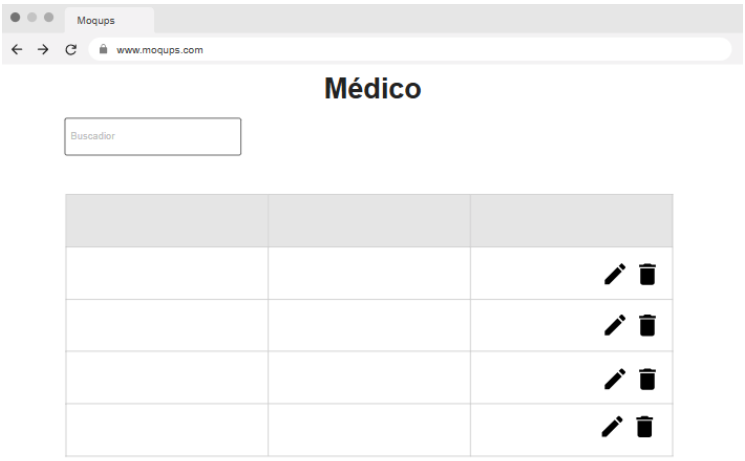
<table>
<tr><th colspan="3">TARJETA CRC</th></tr>
<tr><td>Número: 3</td><td colspan="2">Escenario: Registro, Edición y Eliminación de Especialidades</td></tr>
<tr><td colspan="3">Nombre CRC: Especialidades</td></tr>
<tr><td>Responsabilidades:
Almacenar y gestionar la especialidad.
Proporcionar información de la especialidad.
Actualizar nombre de la especialidad.
Eliminar la especialidad.</td><td>Colaboradores:</td><td>Métodos:
crearEspecialidad()
consultarEspecialidad()
editarEspecialidad()
eliminarEspecialidad()</td></tr>
<tr><td colspan="3">Observaciones:</td></tr>
</table>

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de especialidades.

5.3.2.1 Diseño de referencia:

Figura 9

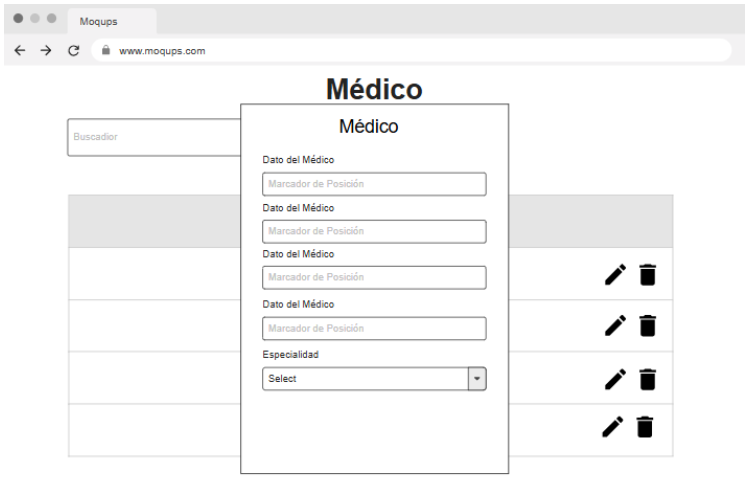
Esquema de Diseño para Médico



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla del Médico.

Figura 10

Esquema de Diseño con Modal en Médico



Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del médico.

Figura 11

Esquema de Diseño para Especialidad



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla de la Especialidad .

Figura 12

Esquema de Diseño con Modal en Especialidad



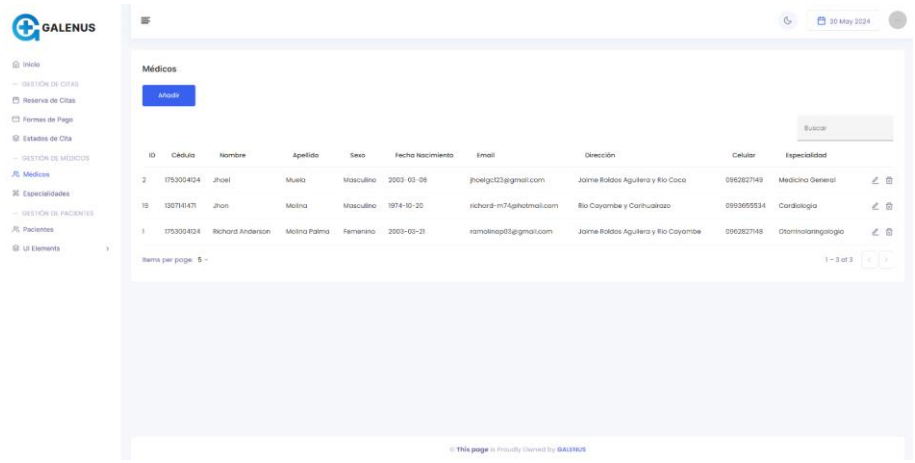
Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del paciente

5.3.3 Codificación

Ahora se desarrollará el módulo para la gestión del médico y su especialidad, ocupando de referencia el borrador realizado anteriormente, por lo que la interfaz de usuario quedaría:

Figura 13

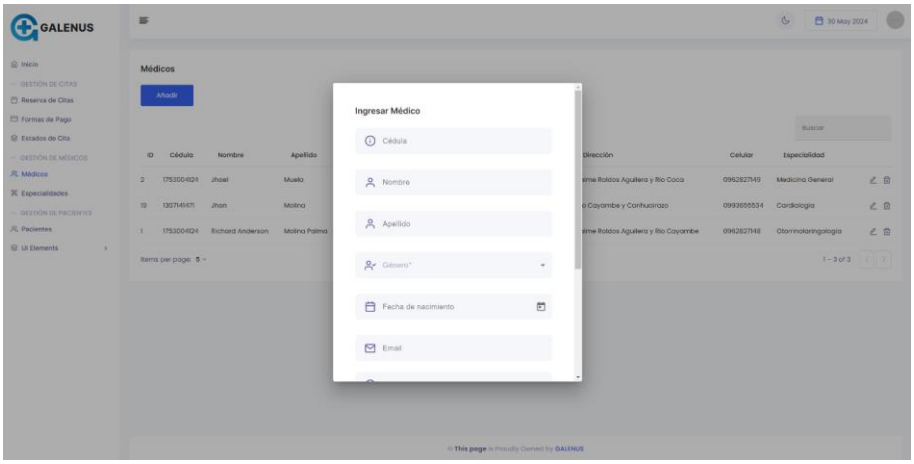
Diseño Final de la pantalla de Médicos



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Médicos donde se observa el listado de los médicos registrados, con botones de añadir, editar y eliminar.

Figura 14

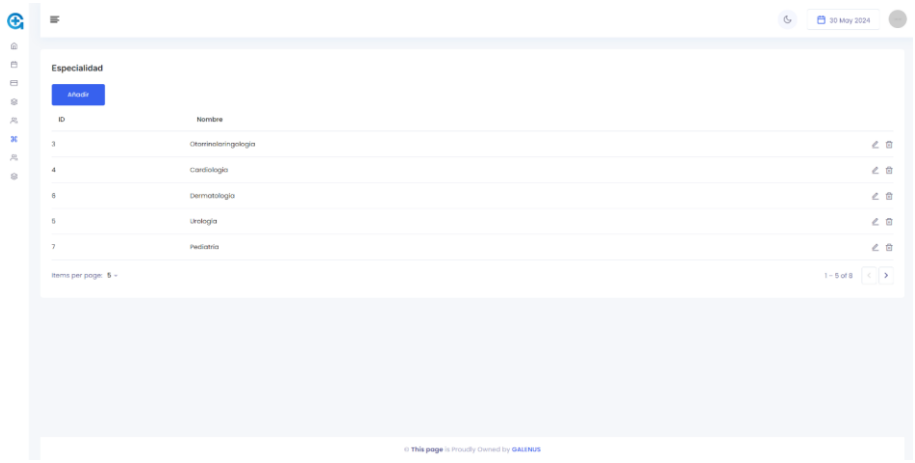
Diseño Final de la pantalla con modal de Médicos



Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Médicos.

Figura 15

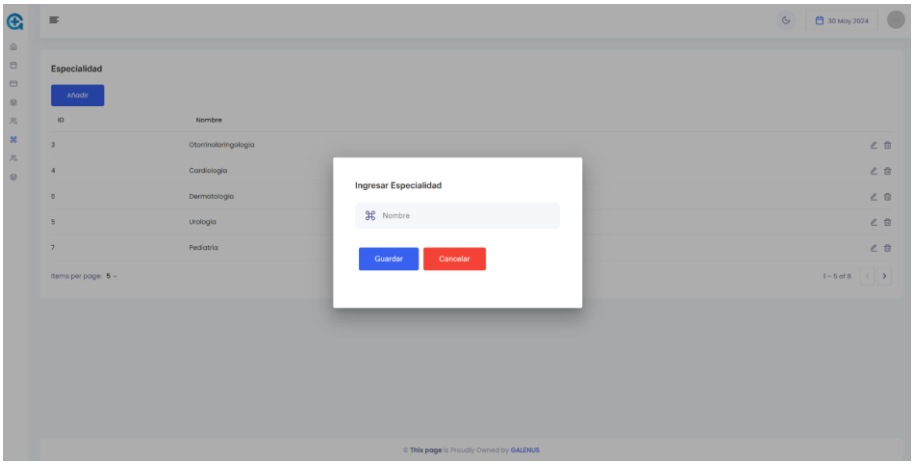
Diseño Final de la pantalla de Especialidades



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Especialidades donde se observa el listado de los especialidades registrados, con botones de añadir, editar y eliminar.

Figura 16

Diseño Final de la pantalla con modal de Especialidades



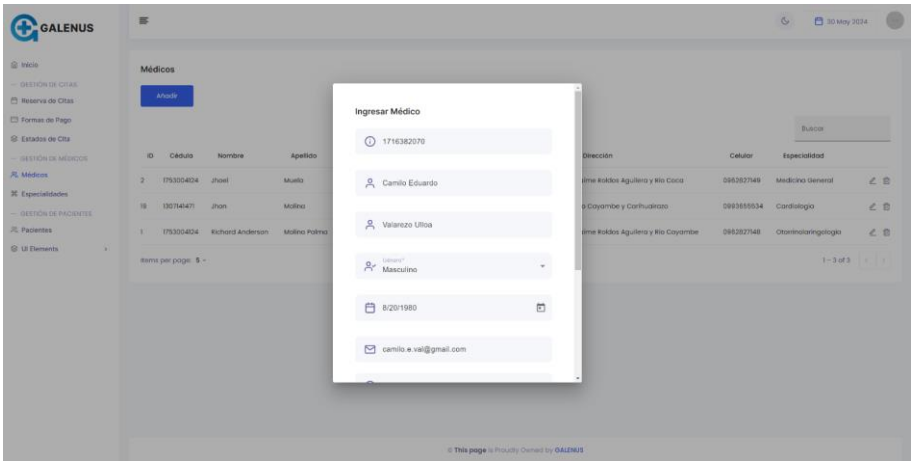
Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Especialidades.

5.3.4 Pruebas

Después de completar el módulo del médico y su especialidad, se realizará las pruebas con el Dr. Patricio.

Figura 17

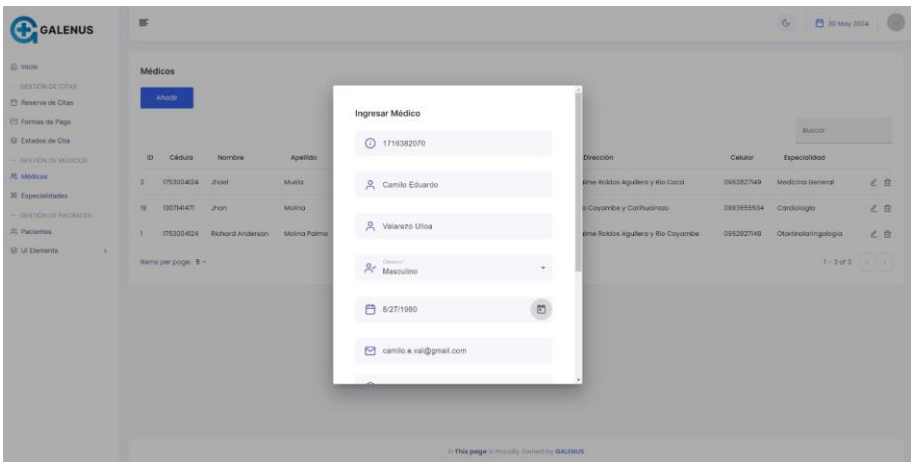
Prueba de Registro de Médicos



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Pacientes.

Figura 18

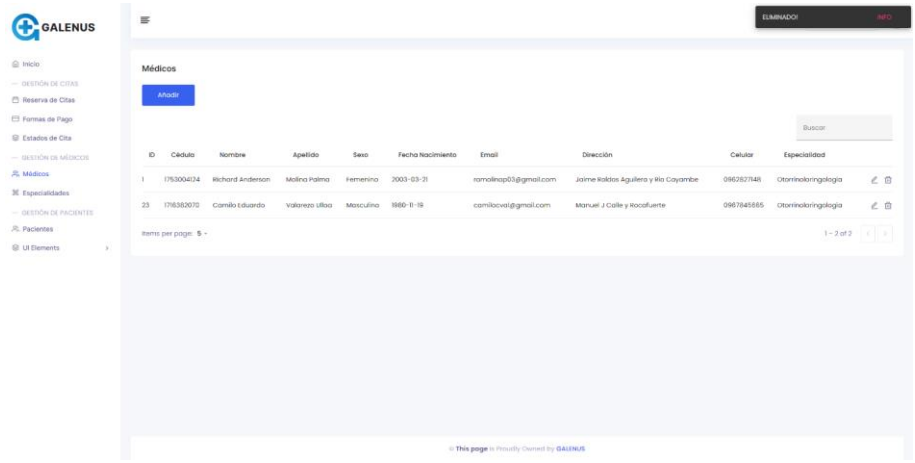
Prueba de Edición en Médicos



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de un Médico.

Figura 19

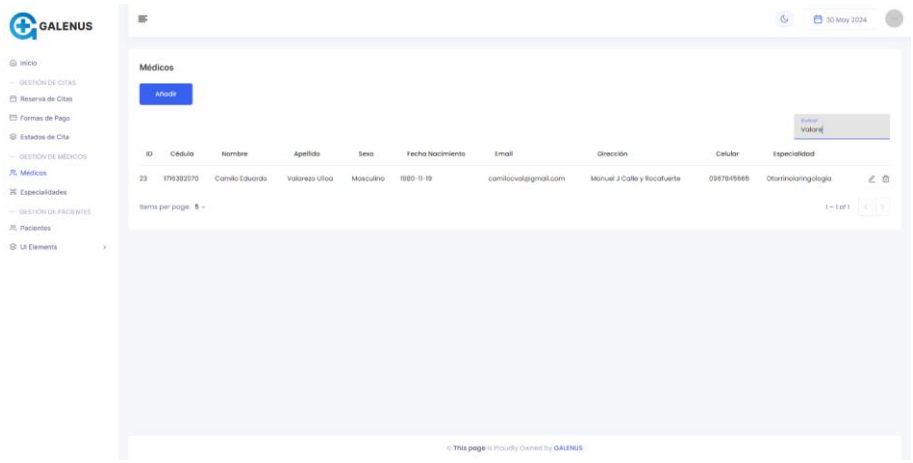
Prueba de Eliminación de Médicos



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de un Médico.

Figura 20

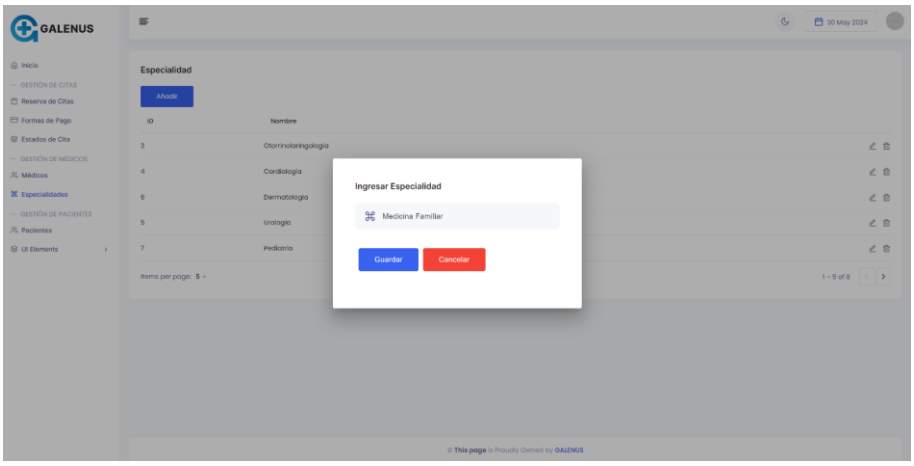
Prueba de Consulta en Médico



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez filtrando los Médicos.

Figura 21

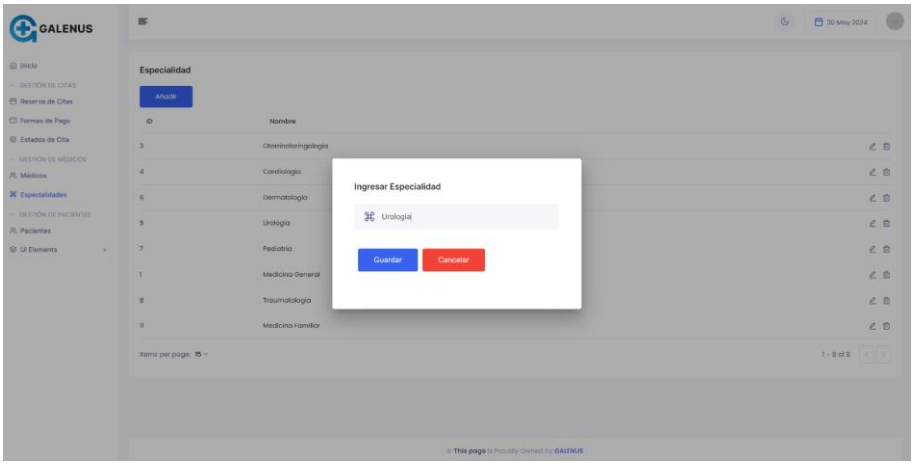
Prueba de Registro en Especialidades



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Especialidades.

Figura 22

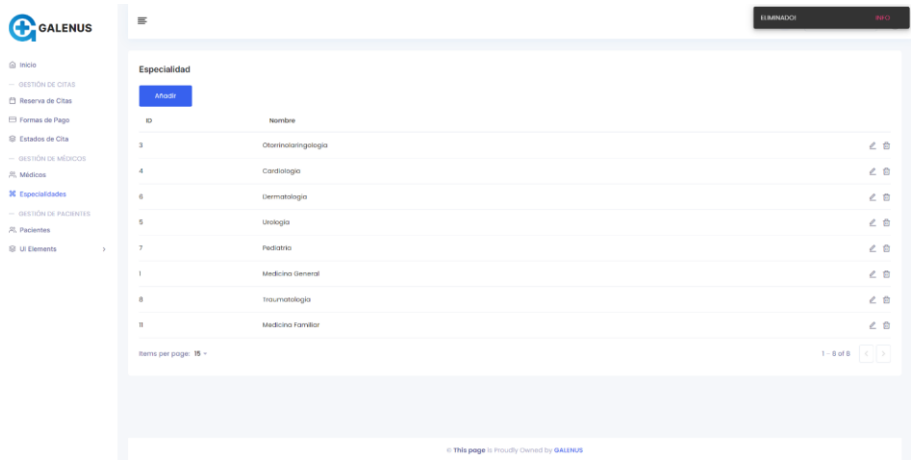
Prueba de Edición en Especialidades



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de Especialidades.

Figura 23

Prueba de Eliminación en Especialidades



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de una Especialidad.

5.4 Iteración 3: Reserva, Estado y Método de Pago

5.4.1 Planificación.

Como tercer módulo desarrollado será la reserva junto a la creación de los módulos encargados para gestionar el estado de la reserva y el método de pago de la misma, en donde reserva captura el estado de la reserva y el método de pago.

Por lo que se plantea las siguientes historias de usuario:

Tabla 20

Historia de Usuario Reserva - Registro

HISTORIA DE USUARIO

Número: 12	**Nombre:** Registro de Reserva de la cita
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se coloquen los datos clave de la reserva de la cita médica.	
Observaciones: Dentro de los datos de la reserva incluir el estado de la cita médica y el método de pago que realizó el paciente a la hora de la reserva.	

Nota. En esta tabla se presenta la historia de usuario para el registro de reserva de citas.

Tabla 21

Historia de Usuario Reserva - Edición

HISTORIA DE USUARIO	
Número: 13	**Nombre:** Edición de Reserva de la cita
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se carguen los datos de la reserva de la cita para luego editar algún dato de esta.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición de reserva de citas.

Tabla 22

Historia de Usuario Reserva - Eliminación

HISTORIA DE USUARIO	
Número: 14	**Nombre:** Eliminación de Reserva de la cita
Usuario: Administrador y Recepcionista	

Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un botón eliminar de manera definitiva la reserva de la cita y sus datos.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la eliminación de reserva de citas.

Tabla 23

Historia de Usuario Reserva - Consulta

HISTORIA DE USUARIO	
Número: 15	**Nombre:** Consulta de Reservas
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un cuadro de búsqueda, consultar o filtrar por cierto dato incluido en cada Reserva.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la consulta en reserva de citas.

Tabla 24

Historia de Usuario Estado - Registro

HISTORIA DE USUARIO	
Número: 16	**Nombre:** Registro de Estado
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta	**Riesgo en desarrollo:** Bajo

(Alta / Media / Baja)	(Alta / Media / Baja)
Descripción: Generar un formulario donde se coloque el nombre del estado de la cita.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para el registro de estados de la cita.

Tabla 25

Historia de Usuario Estado - Edición

HISTORIA DE USUARIO	
Número: 17	**Nombre:** Edición de Estado
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se cargue el nombre del estado de la cita para luego editar el nombre de este.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición de estados de la cita.

Tabla 26

Historia de Usuario Estado - Eliminación

HISTORIA DE USUARIO	
Número: 18	**Nombre:** Eliminación de Estado
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un botón eliminar de manera definitiva el estado de la cita.	

Observaciones:

Nota. En esta tabla se presenta la historia de usuario para la eliminación de estados de la cita.

Tabla 27

Historia de Usuario Método de Pago - Registro

HISTORIA DE USUARIO	
Número: 19	**Nombre:** Registro de Método de pago
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se coloque el nombre del método de pago.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para el registro de métodos de pago para la cita.

Tabla 28

Historia de Usuario Método de Pago - Edición

HISTORIA DE USUARIO	
Número: 20	**Nombre:** Edición de Método de pago
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Generar un formulario donde se cargue el nombre del método de pago de la cita para luego editar el nombre de este.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición de métodos de pago para la cita.

Tabla 29

Historia de Usuario Método de Pago - Eliminación

HISTORIA DE USUARIO	
Número: 21	**Nombre:** Eliminación de Método de pago
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Tercera
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: Mediante un botón eliminar de manera definitiva el método de pago de la cita.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la eliminación de métodos de pago para la cita.

5.4.2 Diseño

Según lo definido en las historias de usuario de la reserva, estado y método de pago se procederá con la creación del borrador sobre el diseño de las pantallas de reserva, estado y método de pago. Y también se añadirá la documentación de esta fase que sería la tarjeta CRC.

Tabla 30

Tarjeta CRC Reserva de Citas

TARJETA CRC		
Número: 4	**Escenario:** Registro, Edición y Eliminación de Reserva de Citas	
Nombre CRC: Reserva de Citas		
Responsabilidades:	**Colaboradores:**	**Métodos:**

• Almacenar y gestionar la reserva. • Proporcionar información de la reserva. • Actualizar información de la reserva. • Eliminar la reserva.	• Estado • Método de Pago	• crearReserva() • consultarReserva() • editarReserva() • eliminarReserva() • consultarMetodoP() • consultarEstado()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de reserva de citas.

Tabla 31

Tarjeta CRC Estado de las citas

TARJETA CRC		
Número: 5	**Escenario:** Registro, Edición y Eliminación de Estado	
Nombre CRC: Estados de Cita		
Responsabilidades: • Almacenar y gestionar el estado. • Proporcionar información del estado. • Actualizar nombre del estado. • Eliminar el estado.	**Colaboradores:**	**Métodos:** • crearEstado() • consultarEstado() • editarEstado() • eliminarEstado()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de estados de la cita.

Tabla 32

Tarjeta CRC Método de Pago para la cita

TARJETA CRC	
Número: 6	**Escenario:** Registro, Edición y Eliminación de Método de Pago
Nombre CRC: Método de Pago	

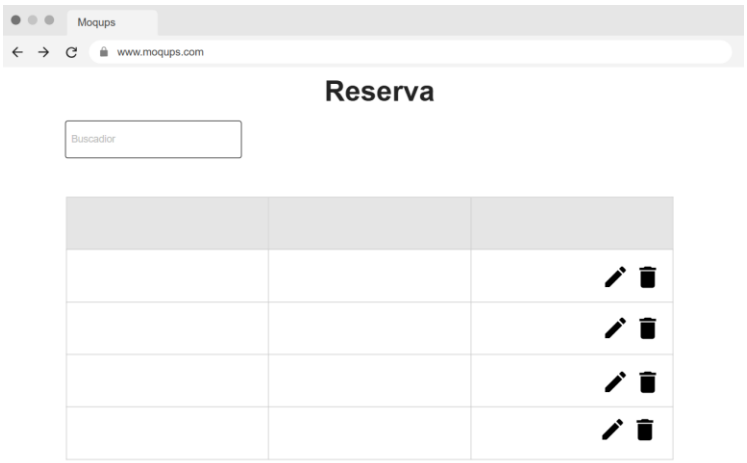
Responsabilidades:	**Colaboradores:**	**Métodos:**
• Almacenar y gestionar el método de pago. • Proporcionar información del método de pago. • Actualizar nombre del método de pago. • Eliminar el método de pago.		• crearMetodoP() • consultarMetodoP() • editarMetodoP() • eliminarMetodoP()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de Métodos de pago.

5.4.2.1 Diseños de referencia:

Figura 24

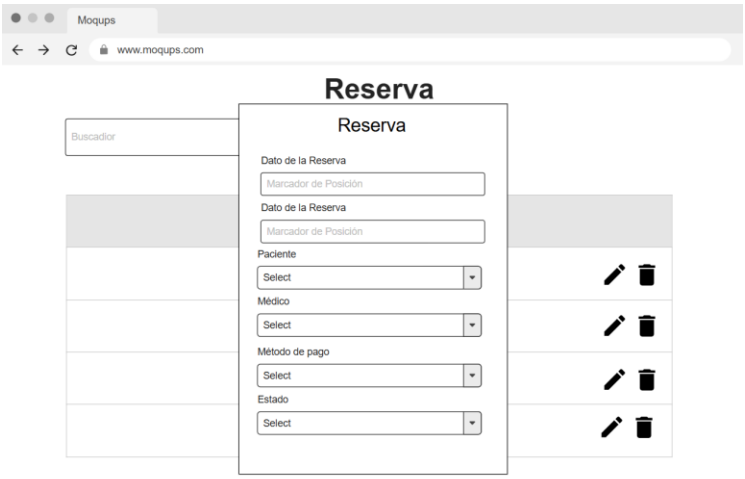
Esquema de Diseño para Reserva



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla de la Reserva de la cita.

Figura 25

Esquema de Diseño con Modal en Reserva



Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información de la reserva de la cita.

Figura 26

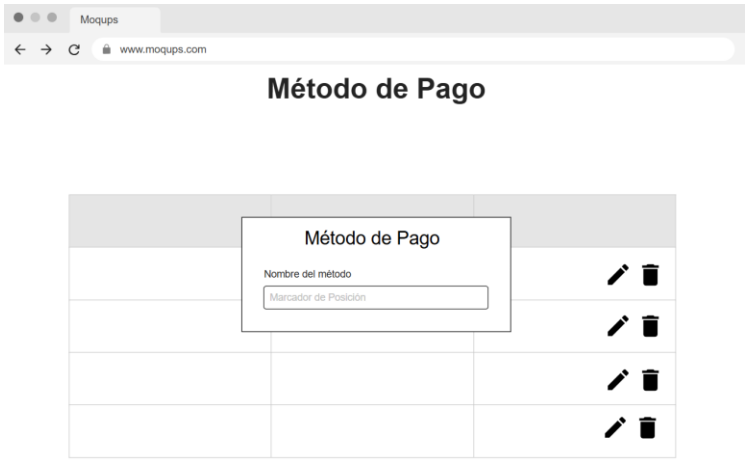
Esquema de Diseño para Método de Pago



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla del Método de pago.

Figura 27

Esquema de Diseño con Modal en Método de Pago



Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del método de pago.

Figura 28

Esquema de Diseño para Estado de Cita



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla del estado de la cita.

Figura 29

Esquema de Diseño con Modal en Estado de Cita



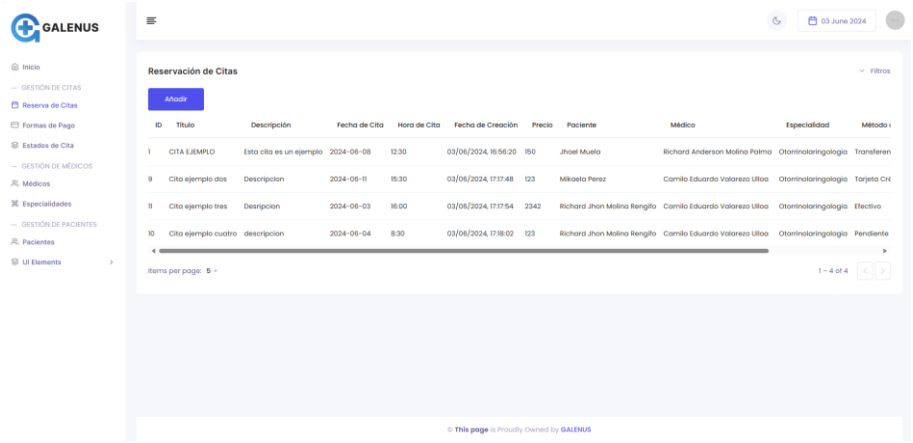
Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del estado de la cita.

5.4.3 Codificación

Ahora se desarrollará el módulo para la gestión de la reserva, el estado y su método de pago, ocupando de referencia el borrador realizado anteriormente, por lo que la interfaz de usuario quedaría:

Figura 30

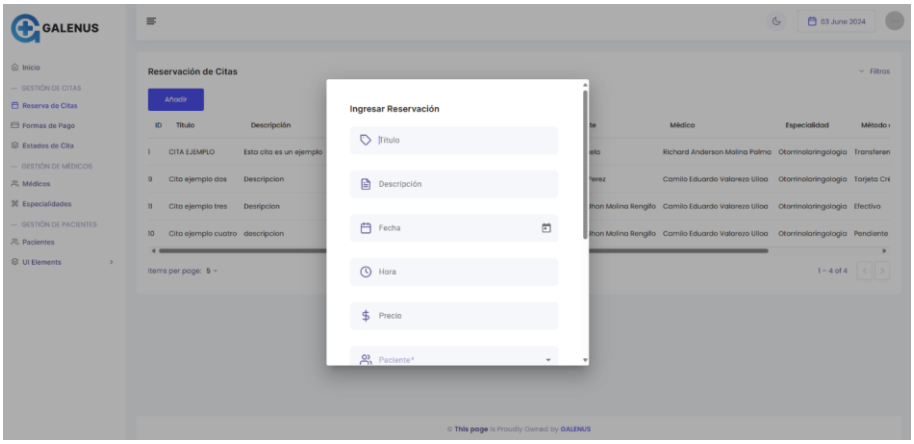
Diseño Final de la pantalla de Reserva de citas



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Reservas donde se observa el listado de las reservas registradas, con botones de añadir, editar y eliminar.

Figura 31

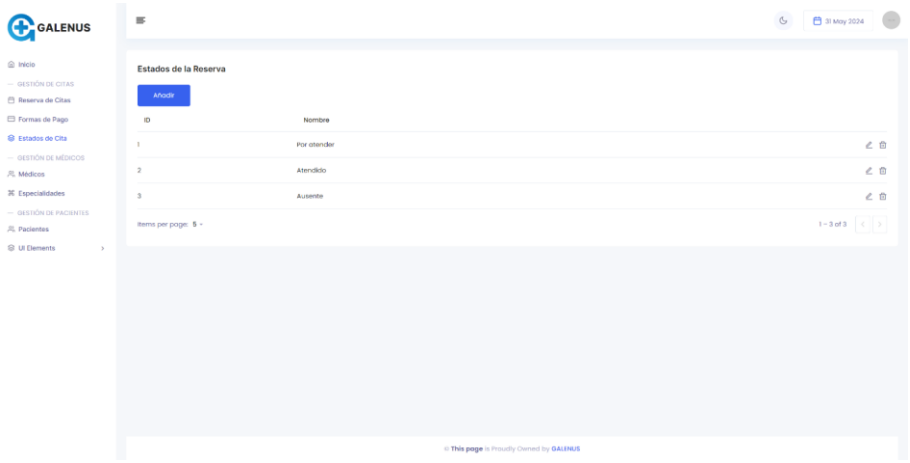
Diseño Final de la pantalla con modal de Reserva de citas



Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Reserva de citas.

Figura 32

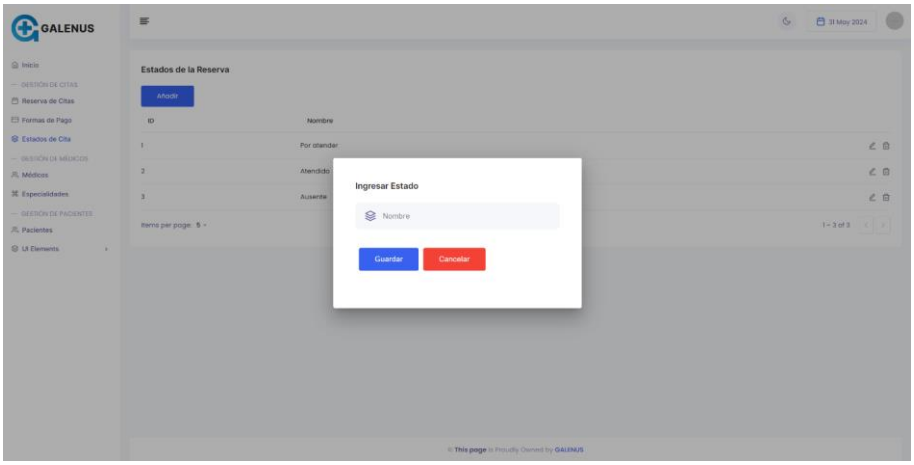
Diseño Final de la pantalla de Estado de la cita



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Estado de citas donde se observa el listado de los estados de la cita registrados, con botones de añadir, editar y eliminar.

Figura 33

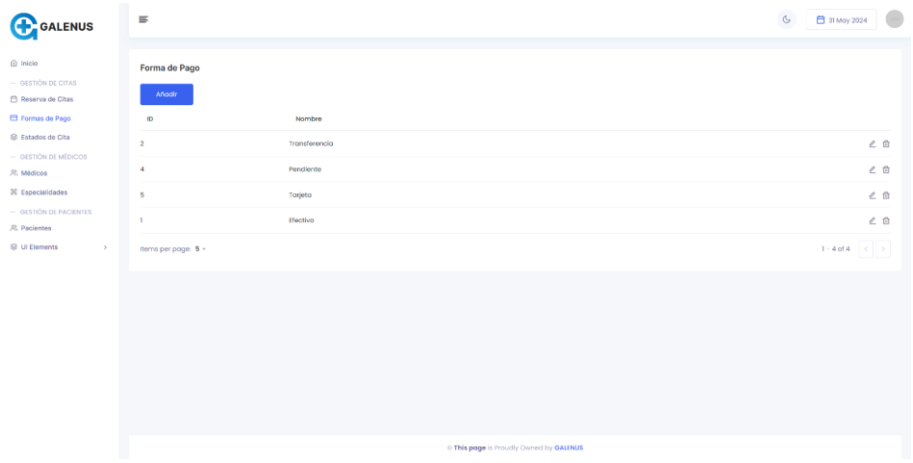
Diseño Final de la pantalla con modal de Estados de cita



Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Estados de cita.

Figura 34

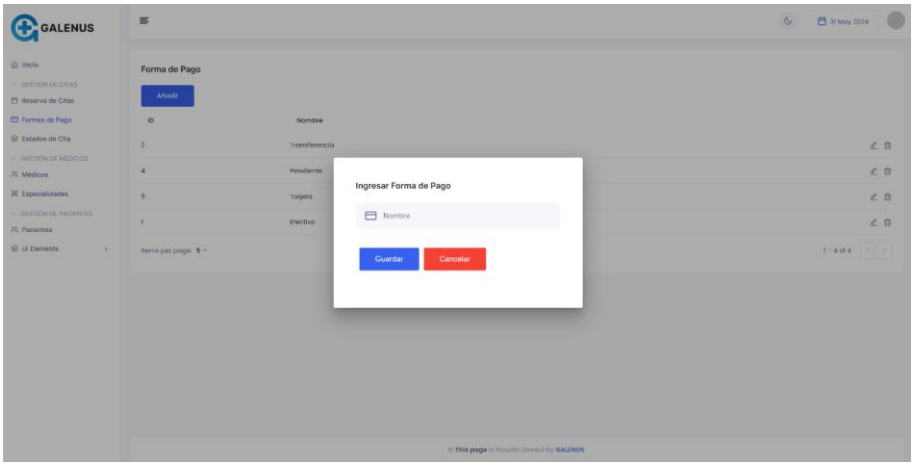
Diseño Final de la pantalla de Método de pago



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Métodos de pago donde se observa el listado de los métodos de pago registrados, con botones de añadir, editar y eliminar.

Figura 35

Diseño Final de la pantalla con modal de Métodos de pago



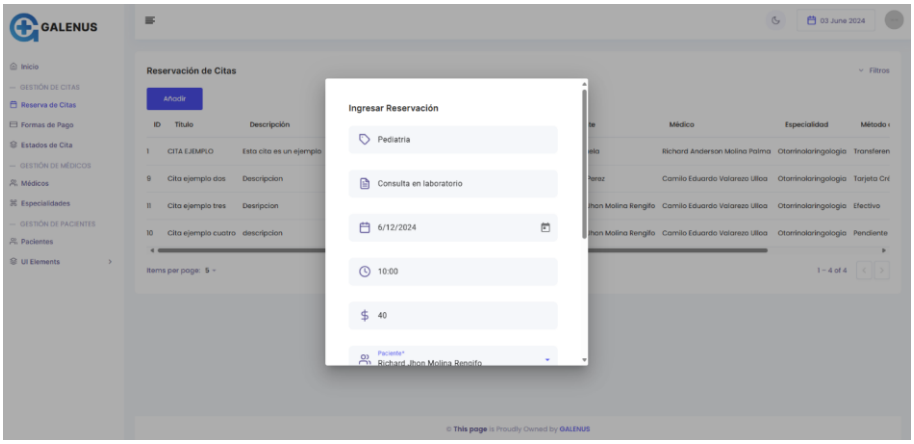
Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Métodos de pago.

5.4.4 Pruebas

Después de completar el módulo de la reserva, el estado y su método de pago, se realizará las pruebas con el Dr. Patricio.

Figura 36

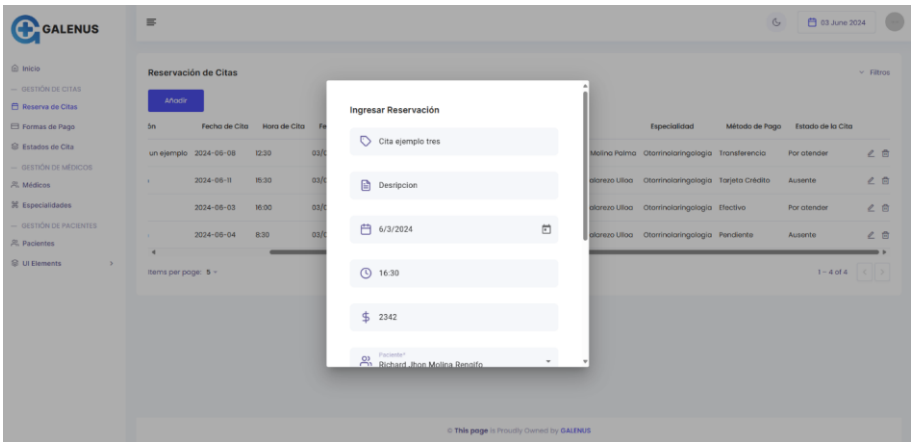
Prueba de Registro en Reservas de citas



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Reservas de citas.

Figura 37

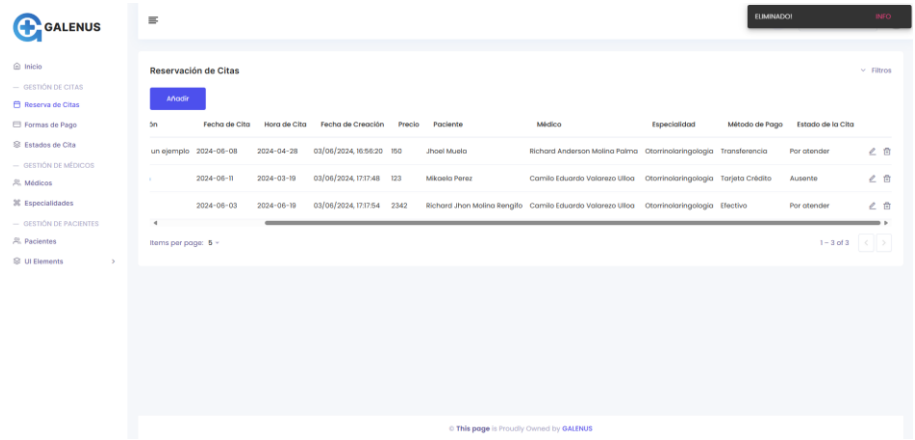
Prueba de Edición en Reservas de citas



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de Reservas de citas.

Figura 38

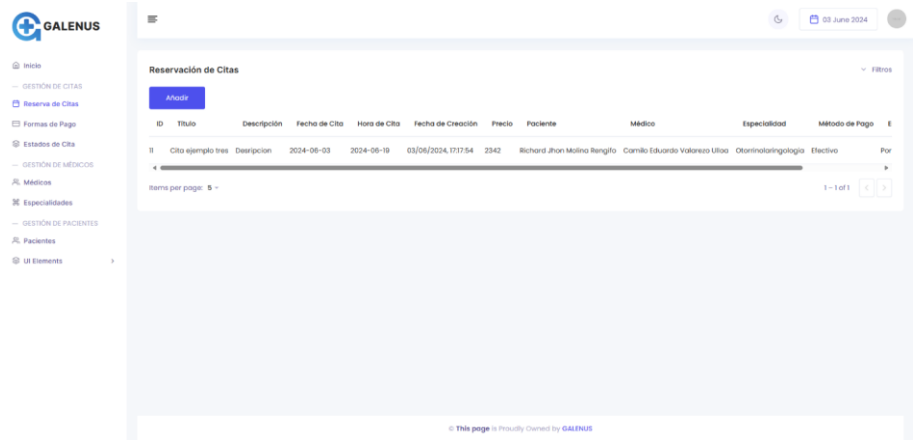
Prueba de Eliminación en Reservas de citas



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de Reservas de citas.

Figura 39

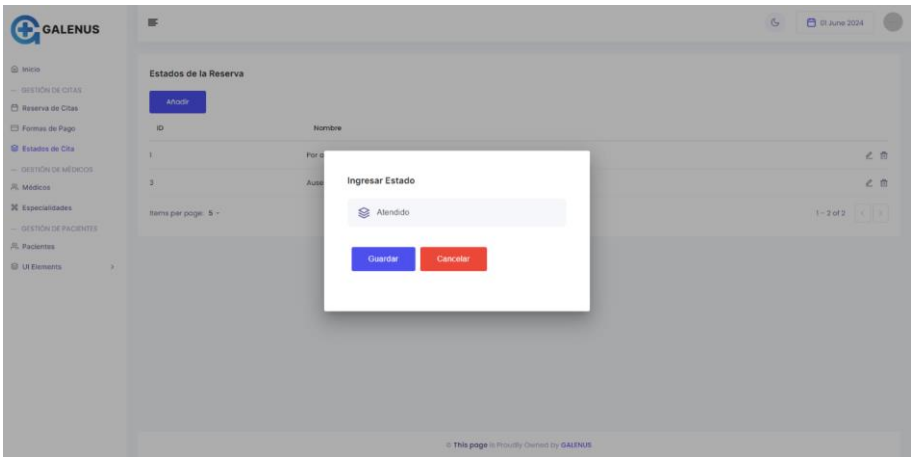
Prueba de Consulta en Reservas de citas



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el filtro de las citas de hoy.

Figura 40

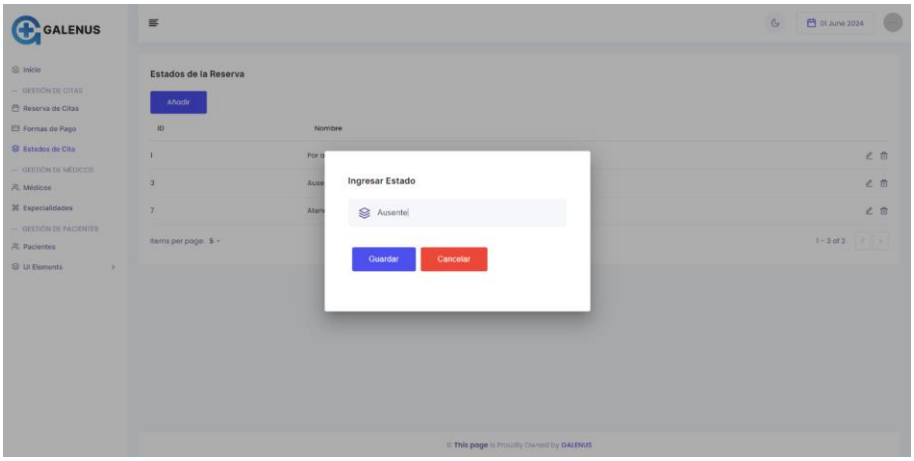
Prueba de Registro en Estados de Cita



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Estados de citas.

Figura 41

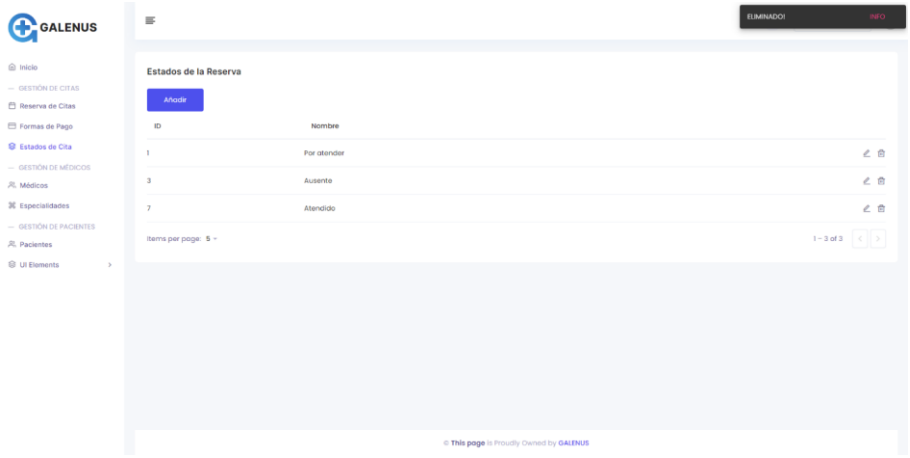
Prueba de Edición en Estados de Cita



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de Estados de citas.

Figura 42

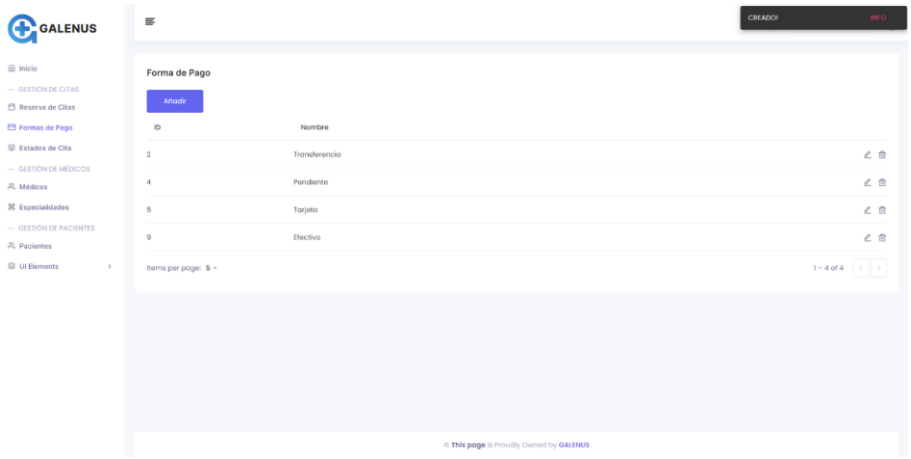
Prueba de Eliminación en Estados de Cita



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de Estados de citas.

Figura 43

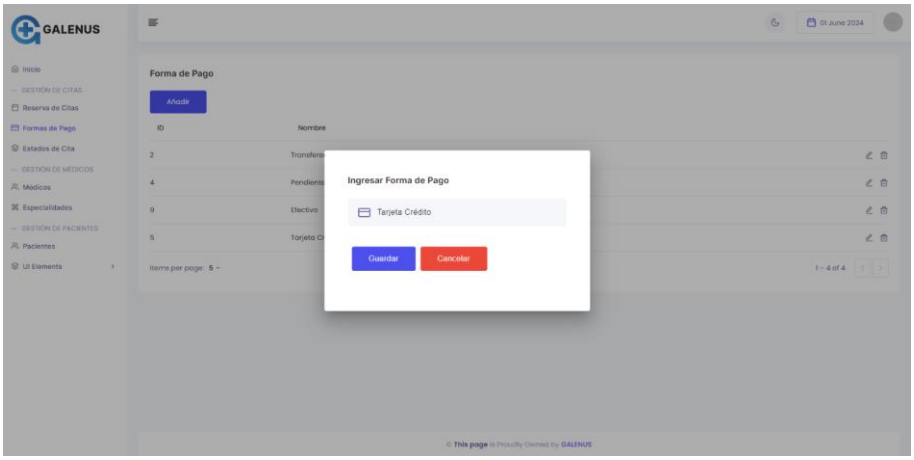
Prueba de Registro en Métodos de Pago



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Métodos de pago.

Figura 44

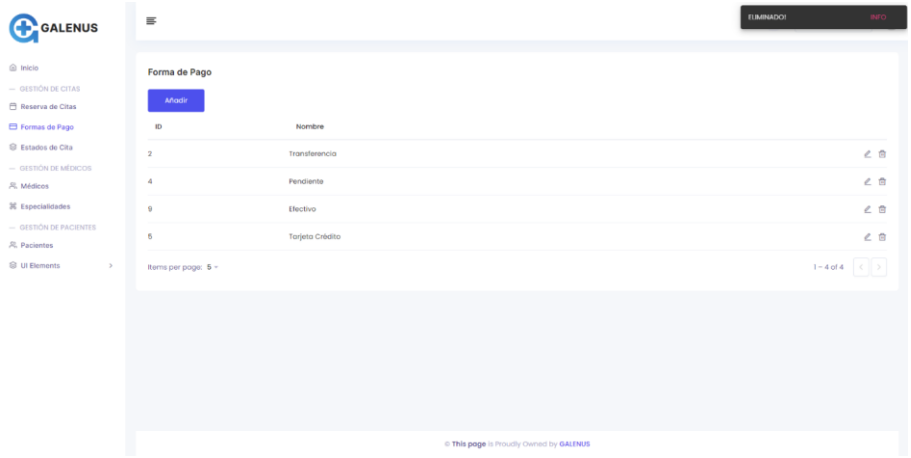
Prueba de Edición en Métodos de Pago



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de Métodos de pago.

Figura 45

Prueba de Eliminación en Métodos de Pago



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de Métodos de pago.

5.5 Iteración 4: Usuarios

5.5.1 Planificación.

En este módulo se va a desarrollar la gestión de usuarios, durante la entrevista el Dr. Patricio Pérez nos comenta que inicialmente desea que el login del sistema esté enfocado a dos usuarios, para el personal administrativo en la recepción y un usuario administrador. Por lo que en base a esta información las historias de usuario serán desarrolladas de la siguiente forma:

Tabla 33

Historia de Usuario Login

HISTORIA DE USUARIO

Número: 22	**Nombre:** Login
Usuario: Administrador y Recepcionista	
Modificación de Historia Número: NA	**Iteración Asignada:** Cuarta
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: En este módulo se permita inicio de sesión de los usuarios, con su username y contraseña.	
Observaciones: El sistema solo debe contar con dos usuarios: Administrador y Recepción.	

Nota. En esta tabla se presenta la historia de usuario para el inicio de sesión del sistema.

Tabla 34

Historia de Usuario, Usuario - Registro

HISTORIA DE USUARIO	
Número: 23	**Nombre:** Registro de Usuarios
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Cuarta
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: En este módulo se permita el registro de usuarios, con su username y contraseña.	
Observaciones: El sistema solo debe contar con dos usuarios: Administrador y Recepción.	

Nota. En esta tabla se presenta la historia de usuario para el registro de usuarios.

Tabla 35

Historia de Usuario, Usuario - Edición

HISTORIA DE USUARIO	
Número: 24	**Nombre:** Edición de Usuarios
Usuario: Administrador	

Modificación de Historia Número: NA	**Iteración Asignada:** Cuarta
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: En este módulo se permita la edición del username o la contraseña del usuario.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la edición del username o contraseña del usuario.

Tabla 36

Historia de Usuario, Usuario - Eliminación

HISTORIA DE USUARIO	
Número: 25	**Nombre:** Eliminación de Usuarios
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Cuarta
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: En este módulo se permita la eliminación de un usuario.	
Observaciones:	

Nota. En esta tabla se presenta la historia de usuario para la eliminación de usuarios.

5.5.2 Diseño

Según lo definido en las historias de usuario de usuarios se procederá con la creación del borrador sobre el diseño de la pantalla para la gestión de usuarios. Y también se añadirá la documentación de esta fase que sería la tarjeta CRC.

Tabla 37

Tarjeta CRC Login

TARJETA CRC		
Número: 7	**Escenario:** Inicio de sesión	
Nombre CRC: Login		
Responsabilidades: • Permitir el acceso a usuarios. • Denegar acceso a usuarios no autorizados.	**Colaboradores:** • Usuarios	**Métodos:** • validarLogin()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para el funcionamiento del inicio de sesión.

Tabla 38

Tarjeta CRC Usuarios

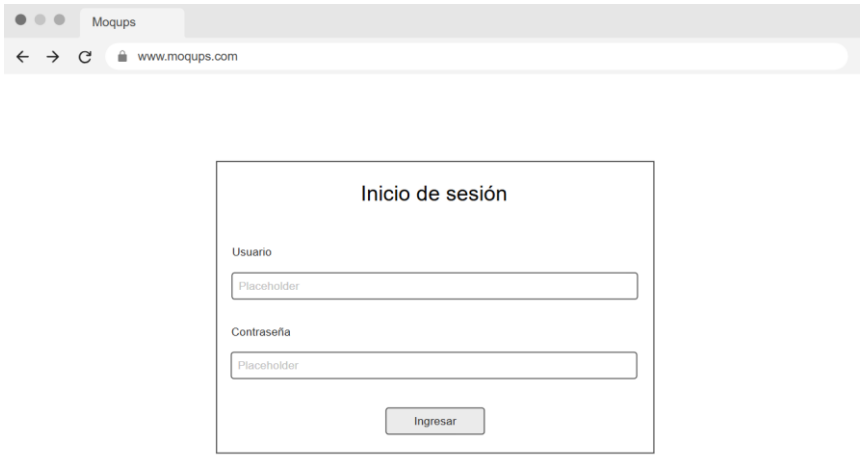
TARJETA CRC		
Número: 8	**Escenario:** Registro, Edición y Eliminación de Usuarios	
Nombre CRC: Usuarios		
Responsabilidades: • Almacenar y gestionar usuarios. • Proporcionar información de usuarios. • Actualizar información del usuario. • Eliminar el usuario.	**Colaboradores:**	**Métodos:** • crearUsuario() • consultarUsuario() • editarUsuario() • eliminarUsuario()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para la gestión de usuarios.

5.5.2.1 Diseños de referencia:

Figura 46

Esquema de Diseño para Login



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla del login.

Figura 47

Esquema de Diseño para Usuario



Nota. En esta figura se presenta el esquema a seguir para el diseño de la pantalla de Usuario.

Figura 48

Esquema de Diseño con Modal en Usuario



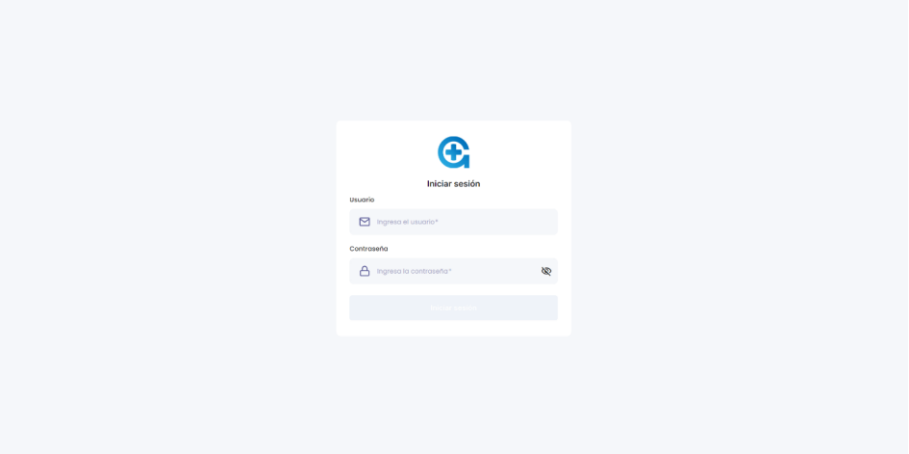
Nota. En esta figura se presenta el esquema para el diseño del modal para el ingreso y edición de la información del usuario.

5.5.3 Codificación

Ahora se desarrollará el módulo para la gestión de los usuarios, ocupando de referencia el borrador realizado anteriormente, por lo que la interfaz de usuario quedaría:

Figura 49

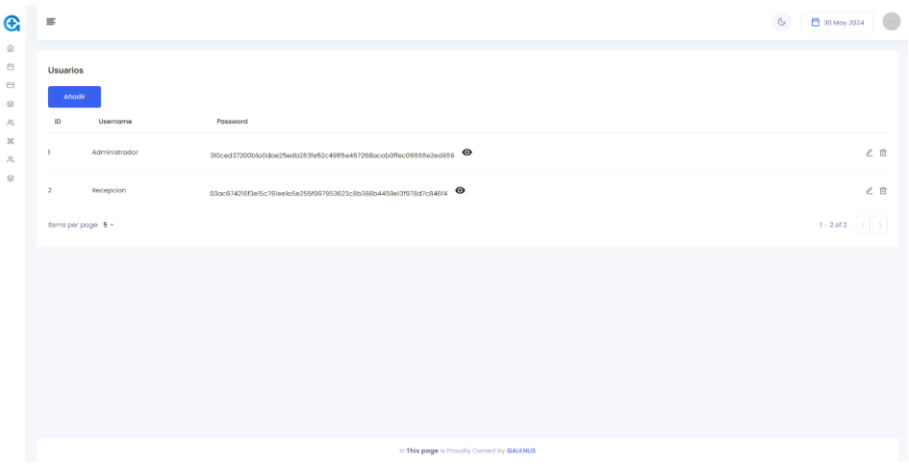
Diseño Final del Login



Nota. En esta figura se presenta el diseño final de la pantalla del login donde se puede observar los inputs para el ingreso del usuario y la contraseña.

Figura 50

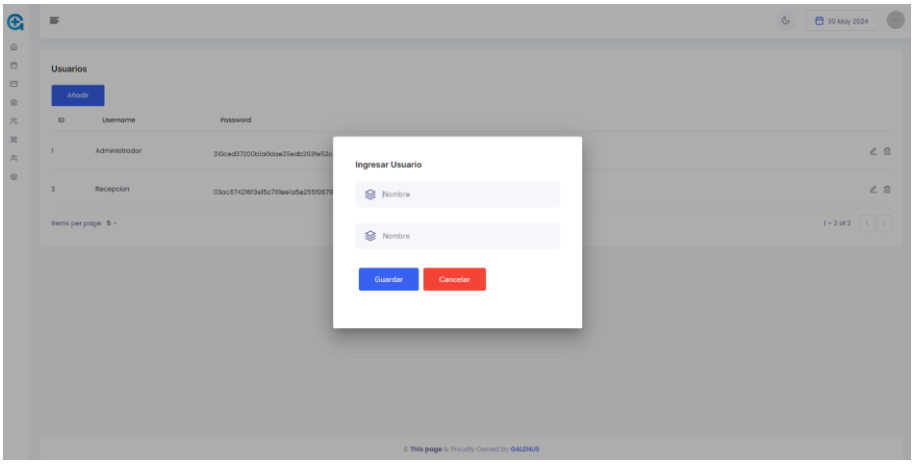
Diseño Final de la pantalla de Usuarios



Nota. En esta figura se observa el diseño final e implementado de la pantalla de Usuarios donde se observa el listado de los usuarios registrados, con botones de añadir, editar y eliminar.

Figura 51

Diseño Final de la pantalla con modal de Usuarios



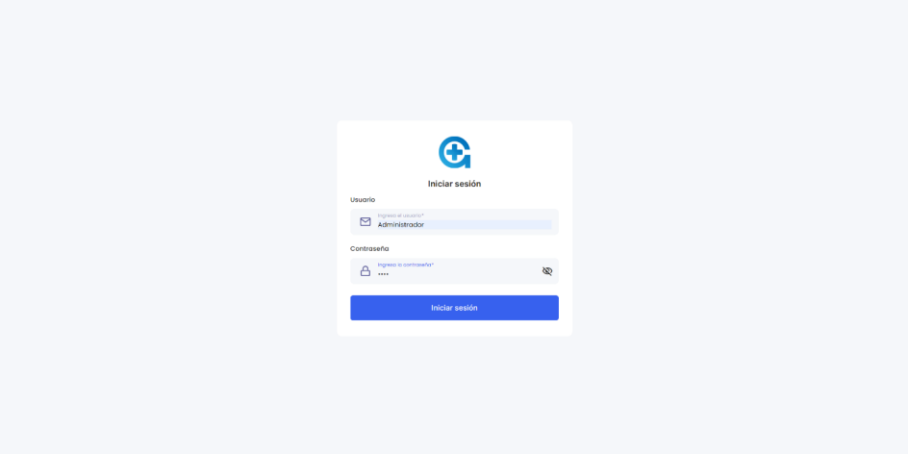
Nota. En esta figura se observa el diseño final e implementado de la pantalla con modal de Usuarios.

5.5.4 Pruebas

Después de completar el módulo del login y los usuarios, se realizará las pruebas con el Dr. Patricio.

Figura 52

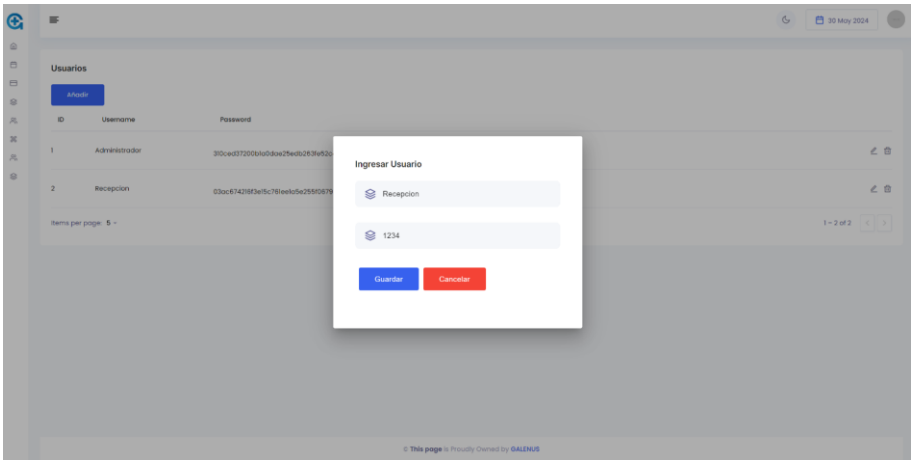
Prueba de funcionamiento de Login



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el inicio de sesión.

Figura 53

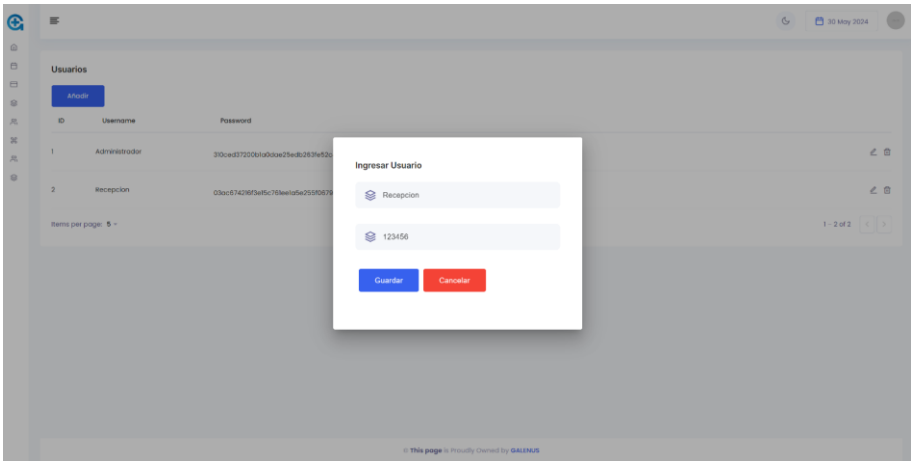
Prueba de Registro en Usuarios



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en el registro de Usuarios.

Figura 54

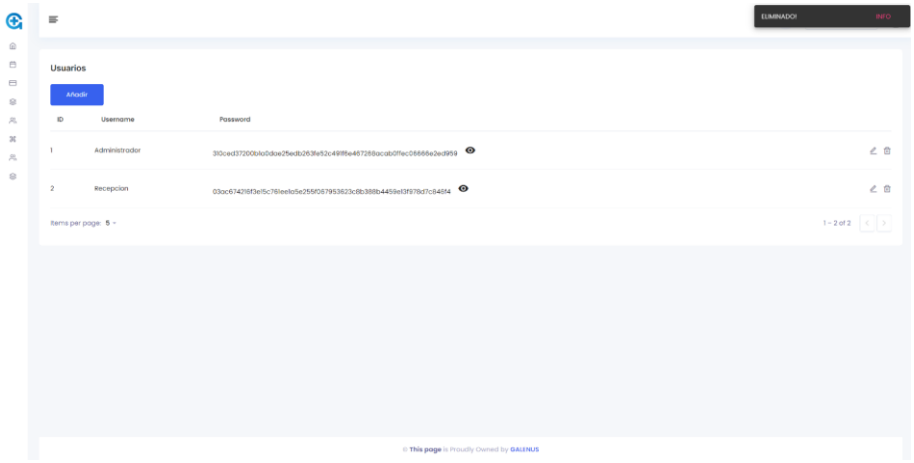
Prueba de Edición en Usuario



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la edición de un Usuario.

Figura 55

Prueba de Eliminación en Usuario



Nota. En esta figura se aprecia la prueba realizada por el Dr. Patricio Pérez en la eliminación de un Usuario.

5.6 Iteración 5: Recordatorios

5.6.1 Planificación.

Finalmente, en este módulo se nos menciona que se desearía que se envíen recordatorios anticipados a los pacientes sobre su cita. Por lo que, la historia de usuario quedaría de la siguiente forma:

Tabla 39

Historia de Usuario Recordatorios

HISTORIA DE USUARIO

Número: 26	**Nombre:** Recordatorios
Usuario: Administrador	
Modificación de Historia Número: NA	**Iteración Asignada:** Quinta
Prioridad en Negocio: Alta (Alta / Media / Baja)	**Riesgo en desarrollo:** Bajo (Alta / Media / Baja)
Descripción: En este módulo se permita la notificación a los pacientes sobre su cita.	
Observaciones: Enviar el recordatorio un día antes de la fecha establecida para la cita.	

Nota. En esta tabla se presenta la historia de usuario para él envió de recordatorios a los pacientes.

5.6.2 Diseño

Según lo definido en las historias de usuario de usuarios se procederá con la creación del borrador sobre el diseño de la pantalla para la gestión de usuarios. Y también se añadirá la documentación de esta fase que sería la tarjeta CRC.

Tabla 40

Tarjeta CRC Recordatorios

TARJETA CRC		
Número: 8	**Escenario:** Gestión de Recordatorios	
Nombre CRC: Recordatorios		
Responsabilidades: • Notificar mediante correo un recordatorio de su cita.	**Colaboradores:** • Reserva	**Métodos:** • crearRecordatorio()
Observaciones:		

Nota. En esta tabla se presenta la tarjeta CRC para el envío de recordatorios a los pacientes de sus citas médicas.

5.6.2.1 Diseño de referencia:

Dentro del diseño de referencia para el correo, el Dr. Patricio recomienda colocar primeramente un saludo de parte del Centro Médico, el recordatorio con la fecha programada para la cita.

5.6.3 Codificación

Comentado [MP1]: Falta esto

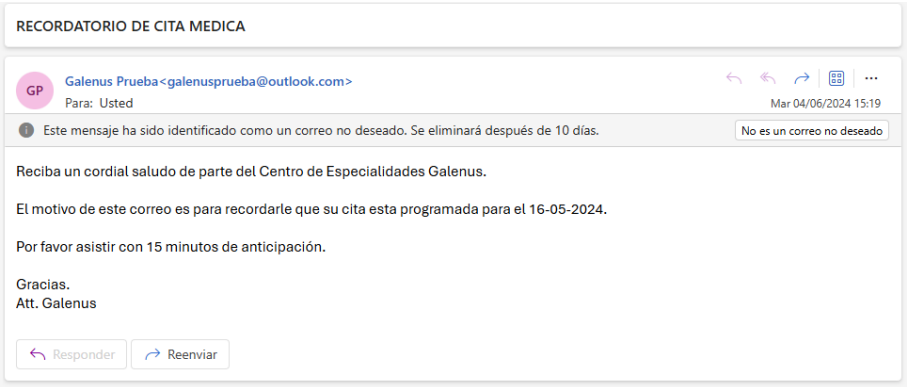
Para el desarrollo del envio de recordatorios, se va a utilizar RapidAPI, la cual es una herramienta que se encarga de gestionar APIs dentro de nuestro backend y a su vez tiene otras funcionalidades como es la de generar y enviar correos electrónicos.

5.6.4 Pruebas

Una vez desarrollado el módulo de recordatorios se realizaron las pertinentes pruebas con el Dr. Patricio donde se validó el funcionamiento del envío de recordatorios al correo electrónico del paciente.

Figura 56

Prueba de Recordatorio



Nota. En esta figura se aprecia la recepción del recordatorio mediante el correo de prueba de Galenus.

5.7 Release Plan del proyecto

Tabla 41

Release Plan del Sistema

Módulo	Nro Iteración.	Historia de Usuario				Fecha de revisión
			Semanas Estimadas	Días Estimados	Horas Estimadas	
Pacientes	1	1,2,3,4	1	4	12	09/03/2024
Médicos y Especialidades	2	5,6,7,8,9,10,11	1,5	7	21	23/03/2024
Reserva, Estado y Método de Pago	3	12,13,14,15,16,17,18,19,20,21	2	10	30	13/04/2024
Login y Usuarios	4	22,23,24,25	1	7	21	27/04/2024
Recordatorios	5	26	1	5	15	04/05/2024
Estimación de tiempo:			6,5	33	99	18/05/2024

Nota. En esta tabla se puede observar los módulos identificados en conjunto con su itreración, tiempo estimado de desarrollo y la fecha de revisión de cada módulo.

Finalmente, con todas las pruebas de integración y funcionalidad realizadas y revisadas por el Dr. Patricio Pérez llevó a un correcto funcionamiento del prototipo planteado y desarrollado en este trabajo de titulación.

6 Capítulo VI: Conclusiones y Recomendaciones

6.1 Conclusiones

- El desarrollo del prototipo del sistema web de gestión de citas médicas para el Centro de Especialidades Galenus ha permitido optimizar significativamente la administración de citas médicas, reduciendo tiempos de espera del paciente y facilitando el proceso realizado por el recepcionista.
- La automatización de los procesos de la gestión de reserva de citas ha resultado en una notable mejora en la eficiencia operativa del centro médico. Por lo que se ha disminuido la carga de trabajo manual del personal administrativo, permitiendo un enfoque más centrado en la atención al paciente y mejorando la precisión en la gestión de datos.
- La integración de recordatorios automatizados apoyará a una notable mejora a la comunicación entre el centro médico y los pacientes. Los recordatorios enviados un día antes de la cita han contribuido a una reducción en el número de citas perdidas, lo que a su vez ha optimizado el uso de los recursos del centro y mejorado la satisfacción del paciente.
- La adopción de metodologías ágiles, específicamente Extreme Programming (XP), ha permitido una mayor flexibilidad y adaptabilidad durante el desarrollo del proyecto. La constante retroalimentación del cliente y la iteración continua han asegurado que el producto final cumpla con las expectativas y necesidades del centro de especialidades Galenus.

6.2 Recomendaciones

- Se recomienda establecer un plan de mejora continua del sistema, basado en la retroalimentación de los usuarios y en el análisis de datos de uso. Esto permitirá identificar áreas de mejora y añadir nuevas funcionalidades que respondan a las necesidades cambiantes del centro médico y de los pacientes.
- Para maximizar los beneficios de la automatización, es esencial capacitar al personal administrativo y médico en el uso eficiente del sistema. Programas de formación periódicos ayudarán a asegurar que todo el personal esté al tanto de las funcionalidades y buenas prácticas para el uso del sistema.
- Se sugiere la adopción de la arquitectura de microservicios para próximos proyectos que demanden una estructura modular. Esta arquitectura posibilita la fragmentación del sistema en componentes más pequeños y manejables, los cuales pueden ser desarrollados, implementados y ampliados de forma independiente.
- Ampliar el sistema de notificaciones para incluir opciones de confirmación de citas por parte de los pacientes. Estas funcionalidades adicionales pueden proporcionar datos valiosos para mejorar la atención al paciente y optimizar la gestión del centro.

7 Referencias Bibliográficas

Anchundia Medrano, L. A. (2022). Análisis comparativo de tecnologías Front End

Angular Js Vs React Js, en el modelo de procesos para el desarrollo de aplicaciones web (Bachelor's thesis, Babahoyo: UTB-FAFI. 2022).

Arias, Á. (2015). Aprende a Programar ASP. NET y C#: 2ª Edición. IT campus academy.

Bascón Pantoja, E. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. Acta Nova, 2(4), 493-507.

Bec, K. (2002). Extreme Programming. Addison Wesley.

Bhardwaz, S., & Godha, R. (2023, March). Svelte. js: The most loved framework today. In 2023 2nd International Conference for Innovation in Technology (INOCON) (pp. 1-7). IEEE.

Bircher, C. (2017). Arquitectura Cliente-Servidor/Cliente-Servidor en Aplicaciones de Misión Crítica.

Castellano Lendínez, L. (2019). KANBAN. METODOLOGÍA PARA AUMENTAR LA EFICIENCIA DE LOS PROCESOS. 3C Tecnologia, 8(1).

García, J. N. (2018). Orquestación de contenedores con Kubernetes (Doctoral dissertation, BS thesis).

Guilindro Moreira, Y. W. (2022). Análisis comparativo de tecnologías Django Js Vs Laravel, en el modelo de procesos para el desarrollo de aplicaciones web (Bachelor's thesis, Babahoyo: UTB-FAFI. 2022).

Haro, E., Guarda, T., Peñaherrera, A. O. Z., & Quiña, G. N. (2019). Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot. Revista Ibérica de Sistemas e Tecnologias de Informação, (E17), 309-321.

Hyams, R., & Pilko, T. (2022). “You could use the API!”: A Crash Course in Working with the Alma APIs using Postman. Code4Lib Journal, (54).

Kinder, K. (2013). Sublime text: one editor to rule them all?. Linux Journal, 2013(232), 2.

Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).

Lendínez, L. C. (2019). Kanban. Metodología para aumentar la eficiencia de los procesos. 3c Tecnología: glosas de innovación aplicadas a la pyme, 8(1), 30-41.

López Herrera, P. (2016). Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL.

Magaly, L. M. N., Rolando, M. R. J., Fausto, L. M., & Erick, C. V. (2020, August). INTELLIJ IDEA 2018.3. 1 y GITHUB como herramientas para el control de cambios en proyectos de equipos de desarrollo distribuidos: INTELLIJ IDEA 2018.3. 1 and GITHUB as tools for the control of changes in projects of distributed development teams. In Conference Proceedings (Machala) (Vol. 4, No. 1, pp. 122-143).

Morales, E. C., Rivera, M. E. R., & Lizama, E. R. (2017). Desarrollo de un modelo de pruebas funcionales de software basado en la herramienta SELENIUM. Industrial data, 20(1), 139-147.

Oriols, D. M. B., & Gutiérrez, J. G. (2020). El gran libro de Angular. Marcombo.

Penadés, M. C., & Torres, P. O. L. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Técnica administrativa, 5(26), 1.

Ramírez, M. R., Soto, M. D. C. S., Moreno, H. B. R., Rojas, E. M., Millán, N. D. C. O., & Cisneros, R. F. R. (2019). Metodología SCRUM y desarrollo de Repositorio Digital. Revista Ibérica De Sistemas e Tecnologias De Informação, (E17), 1062-1072.

Ramírez Echeverri, D. Gengular: hacia la automatización de aplicaciones empresariales bajo el paradigma de arquitectura SPA y el enfoque MDE.

Soler-Adillon, J. (2017). Tres IDEs para programar: Processing, Eclipse y Visual Studio. Mosaic, (149), 4.

Studio, V. (2013). Microsoft Visual Studio. URL: https://visualstudio. microsoft. com/ru/(дата звернення 25.05. 2022).

Surianarayanan, C., Ganapathy, G., & Pethuru, R. (2019). Essentials of microservices architecture: Paradigms, applications, and techniques. Taylor & Francis.

Tortosa Calabuig, M. (2019). Diseñando una fábrica del Futuro con la Internet de las Cosas. Una experiencia práctica (Doctoral dissertation, Universitat Politècnica de València).

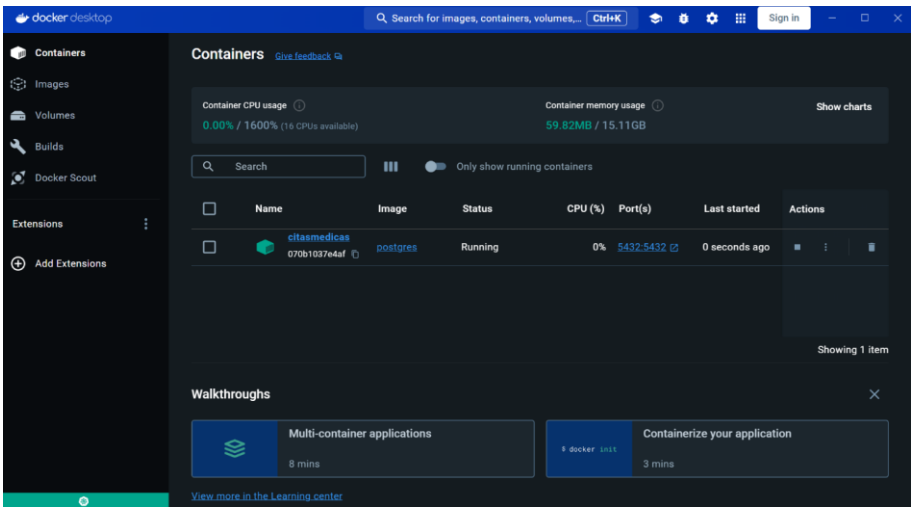
Tosic, D. (2023). Artificial Intelligence-driven web development and agile project management using OpenAI API and GPT technology: A detailed report on technical integration and implementation of GPT models in CMS with API and agile web development for quality user-centered AI chat service experience.

Vershinin, I. S., & Mustafina, A. R. (2021, September). Performance analysis of PostgreSQL, MySQL, Microsoft SQL server systems based on TPC-H tests. In 2021 International Russian Automation Conference (RusAutoCon) (pp. 683-687). IEEE.

8 Anexos

Anexo 1

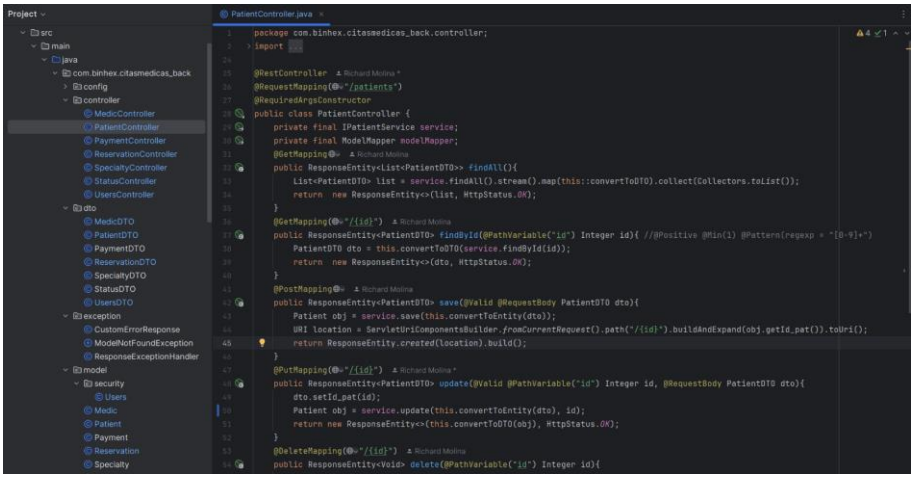
Instancia de PostgreSQL en Docker



Nota. En este anexo se presenta la instancia configurada de nuestra base de datos PostgreSQL.

Anexo 2

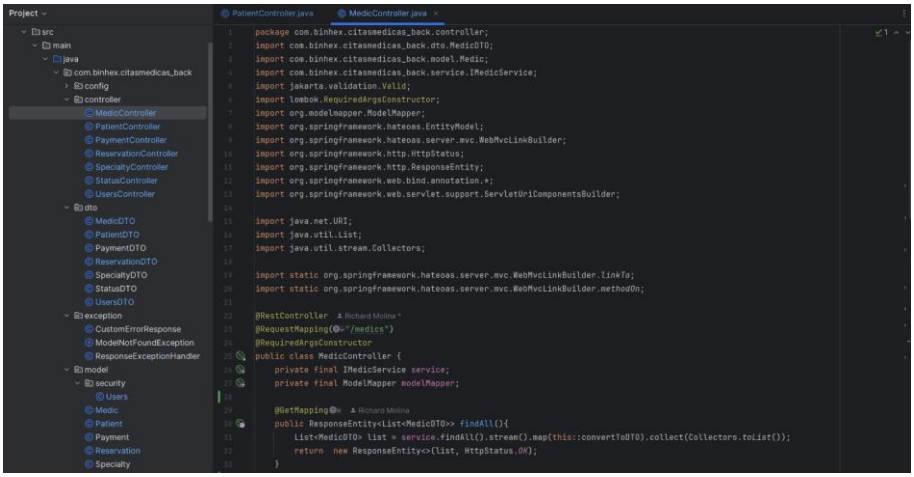
Código de Backend – Paciente



Nota. En este Anexo se presenta una parte del código del controlador del módulo de Paciente en el Backend.

Anexo 3

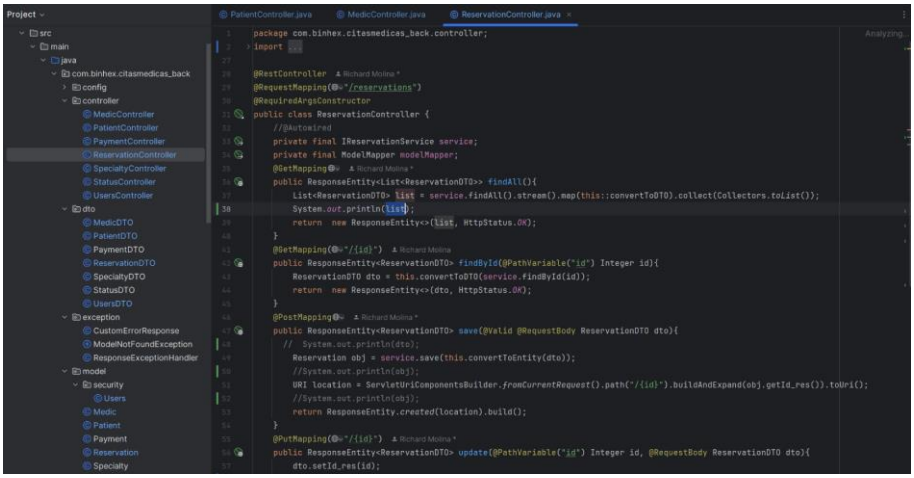
Código de Backend – Médico



Nota. En este Anexo se presenta una parte del código del controlador del módulo de Médico en el Backend.

Anexo 4

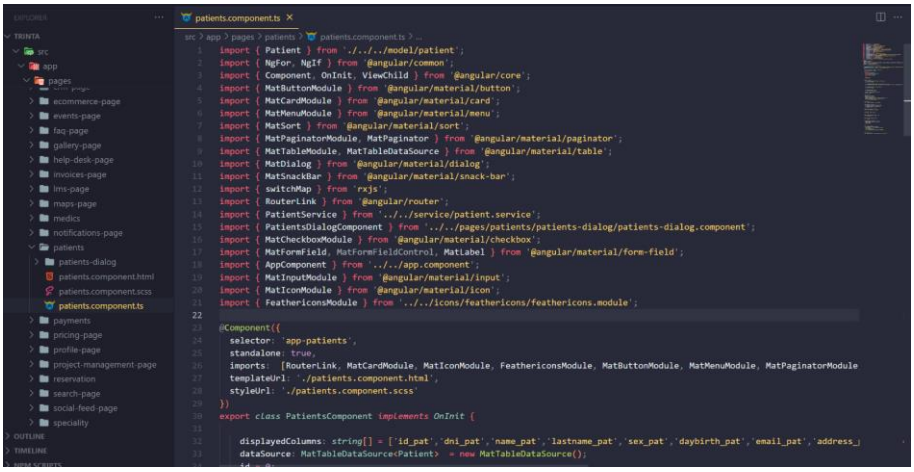
Código de Backend – Reserva



Nota. En este Anexo se presenta una parte del código del controlador del módulo de Reserva en el Backend.

Anexo 5

Código de Frontend – Paciente



Nota. En este Anexo se presenta una parte del código del typescript del módulo de Paciente en el Frontend.

Anexo 6

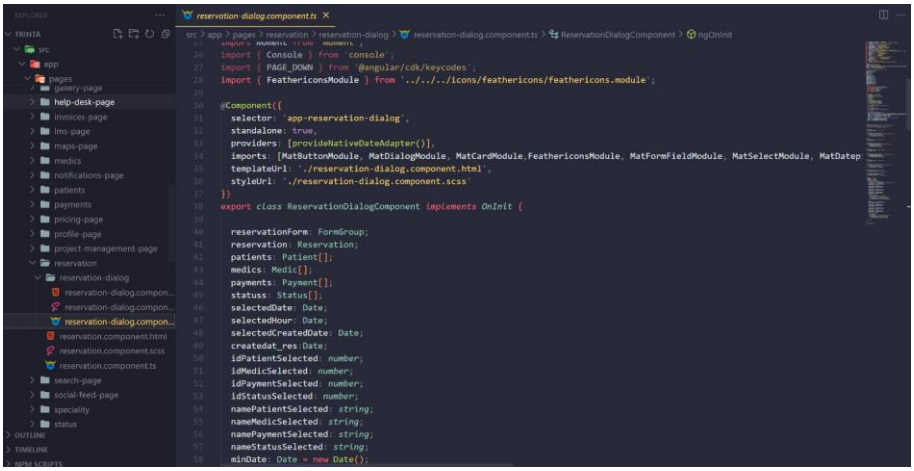
Código de Frontend – Médico



Nota. En este Anexo se presenta una parte del código del typescript del módulo de Médico en el Frontend.

Anexo 7

Código de Frontend – Reserva



Nota. En este Anexo se presenta una parte del código del typescript del módulo de Reserva en el Frontend.