PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
FACULTAD DE INGENIERIA
ESCUELA DE SISTEMAS



DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DE PROCESOS ASISTENCIALES DE UN CENTRO MÉDICO, CASO DE ESTUDIO: CENTRO MÉDICO ANEZ

AUTOR:
THALÍA BELÉN ZAPATA TOAPANTA

DIRECTOR:

ING. ALFREDO CALDERÓN

QUITO, 2021

RESUMEN

El presente trabajo de disertación se orienta a la implementación de un sistema web para la automatización de procesos asistenciales del centro médico ANEZ, desarrollado empleando el motor de bases de datos relacionales PostgreSQL, la librería de React.js y el framework de Express.js.

Utilizando la metodología ágil de desarrollo Extreme Programming (XP), se inicia con un análisis de los requerimientos, las cuales lograron ser completamente recabados, gracias a la activa participación y comunicación con el cliente.

Posteriormente, se realiza el diseño de la base de datos, así como las interfaces lo que permite tener una visión completamente clara acerca del sistema web, el cual una vez desarrollado, fue sometido a pruebas unitarias y de integración con el propósito de cerciorarse que todas las funcionalidades se desempeñen correctamente.

Finalmente, como resultado del presente trabajo, se entrega un sistema web que permite la gestión de citas, pacientes e historias clínicas, el mismo que se encuentra alojado en un servidor en la nube, accesible para el personal del centro médico.

DEDICATORIA

El presente trabajo de disertación está dedicado a:

Mis padres Carlos y Susana quienes con su amor, sabiduría y palabras de aliento me han sabido guiar y ayudar a alcanzar mis metas, demostrándome que las mejores recompensas únicamente se las obtiene si se ha luchado y trabajado por ello.

Mis hermanas y cuñado Estefanía, Carla y Luis quienes con cariño y paciencia desde la distancia o junto a mi me han apoyado incondicionalmente en cada paso que he dado.

AGRADECIMIENTOS

Quiero empezar agradeciendo a Dios por ser mi guía y bendecirme en cada etapa de mi vida.

A mi familia por estar siempre junto a mi brindándome su apoyo, sus consejos y motivándome para ser cada día mejor.

A la Pontificia Universidad Católica del Ecuador, a la Facultad de Ingeniería, a todos mis docentes y de manera especial a mi director el ingeniero Alfredo Calderón quien, con paciencia, dedicación y transmitiendo sus conocimientos de forma desinteresada me ha sabido encaminar en el proceso para el desarrollo del presente trabajo de disertación.

Al centro médico ANEZ por permitirme desarrollar mi trabajo de disertación utilizando su nombre e instalaciones y por proveerme la información necesaria para la implementación del sistema web.

A mis compañeros y amigos por su compañía durante el transcurso de la carrera, ya sea con sus consejos, enseñanzas o risas.

Y a todas aquellas personas que hicieron posible la culminación del presente trabajo.

Tabla de contenidos

Resumen ..... *I*

Dedicatoria ..... *II*

Agradecimientos ..... *III*

1. Aspectos Introductorios *1*

1.1 Datos Del Centro Médico **1**
1.1.1 Misión 1
1.1.2 Visión 1
1.1.3 Mapa De Procesos Asistenciales 1

1.2 Justificación **6**

1.3 Planteamiento Del Problema **7**

1.4 Alcance **8**

1.5 Objetivos **9**
1.5.1 Objetivo General 9
1.5.2 Objetivos Específicos 9

2. Fundamento Teórico *10*

2.1 Sistemas Web **10**

2.2 Arquitectura Web **10**
2.2.1 Arquitectura Monolítica 11
2.2.2 Arquitectura Cliente – Servidor 12
2.2.3 Arquitectura En Capas 13
2.2.4 Arquitectura Basada En Eventos 14
2.2.5 Arquitectura Orientada Al Servicio 15
2.2.6 Arquitectura De Microservicios 16

2.3 Servicios Web **17**
2.3.1 Servicio Web Soap 17
2.3.2 Servicio Web Restful 18
2.3.3 Graphql 19

2.4 Base De Datos **19**

2.5 Tecnologías **21**
2.5.1 Back-End 21
2.5.2 Front-End 22

2.6 Herramientas **22**
2.6.1 Base De Datos 22
2.6.1.1 Postgresql 22
2.6.2 Backend 23
2.6.2.1 Node.Js 23
2.6.2.2 Express.Js 23
2.6.2.3 Sequelize 23
2.6.3 Frontend 24

2.6.3.1 React.Js ... 24
2.6.3.2 React Suite ... 24
2.6.3.3 Semantic Ui React ... 24
2.6.4 Pruebas ... 25
2.6.4.1 Jest ... 25
2.6.5 Despliegue ... 25
2.6.5.1 Amazon Web Services (Aws) ... 25
2.6.5.2 Vercel ... 26
2.7 Metodología ... 27
2.8 Metodología De Desarrollo ... 28
2.8.1 Metodologías Ágiles ... 28
2.8.2 Extreme Programming ... 30
2.9 Justificación De Las Herramientas y Metodologías Seleccionadas ... 34
3. Desarrollo Del Sistema Web Para La Automatización De Procesos Asistenciales ... 36
3.1 Fase De Exploración ... 36
3.1.1 Historias De Usuario ... 36
3.1.2 Formalización De Requerimientos ... 41
3.1.2.1 Requerimientos Funcionales Y No Funcionales ... 41
3.1.2.2 Casos De Uso ... 44
3.2 Fase De Mantenimiento ... 50
3.2.1 Fase De Planificación ... 50
3.2.1.1 Definición De Entregables ... 50
3.2.2 Fase De Iteraciones ... 51
3.2.2.1 Diseño ... 52
a. Arquitectura ... 52
b. Base de datos ... 53
3.2.2.2 Iteración 1 ... 56
a. Historia de usuario 1 ... 56
b. Historia de usuario 2 ... 58
3.2.2.3 Iteración 2 ... 62
a. Historia de usuario 3 ... 62
b. Historia de usuario 4 ... 67
3.2.2.4 Iteración 3 ... 70
a. Historia de usuario 5 ... 71
b. Historia de usuario 6 ... 76
3.2.2.5 Iteración 4 ... 81
a. Historia de usuario 7 ... 82
b. Historia de usuario 8 ... 85
3.2.3 Fase De Producción ... 90
3.2.3.1 Despliegue ... 90
3.2.3.2 Aceptación ... 90
4. Conclusiones Y Recomendaciones ... 91
4.1 Conclusiones ... 91

4.2 Recomendaciones 93

Referencias 94

Anexos 98

Manual De Usuario 1

Índice de figuras

Figura 1 Clasificación de procesos asistenciales. 2
Figura 2. Diagrama de procesos, Gestión de citas. 3
Figura 3. Diagrama de procesos, Recepción del paciente. 4
Figura 4. Diagrama de procesos, Consultas médicas. 5
Figura 5 Tipos de Arquitectura. 11
Figura 6 Arquitectura monolítica 12
Figura 7 Arquitectura cliente – servidor. 12
Figura 8 Arquitectura en capas. 13
Figura 9 Arquitectura orientada a eventos EDA. 14
Figura 10 Arquitectura Orientada a Servicios. 15
Figura 11 Arquitectura de microservicios. 16
Figura 12. Proceso cualitativo. 27
Figura 13 Fases de XP. 32
Figura 14. Distribución de las herramientas seleccionadas. 36
Figura 15 Formato de historias de usuario. 37
Figura 16 Historia de usuario No. 1 38
Figura 17 Historia de usuario No. 2 38
Figura 18 Historia de usuario No. 3 39
Figura 19 Historia de usuario No. 4 39
Figura 20 Historia de usuario No. 5 40
Figura 21 Historia de usuario No. 6 40
Figura 22 Historia de usuario No. 7 41
Figura 23 Historia de usuario No. 8 41
Figura 24 Nivel general. 45
Figura 25 RF1. Ingreso al sistema. 46
Figura 26 RF2. Administración de Pacientes. 46
Figura 27 RF3. Agendamiento de Citas. 47
Figura 28 RF4. Administración de Historias Clínicas. 47
Figura 29 RF5. Administración de Evoluciones. 48
Figura 30 RF6. Administración de Usuarios. 48
Figura 31 RF7. Administración de Médicos. 49
Figura 32 RF8. Administración de Consultorios. 49
Figura 33 Planificación inicial del proyecto. 50
Figura 34 Definición de entregables. 51
Figura 35. Arquitectura del sistema. 53

Figura 36. Modelo conceptual. 54
Figura 37. Modelo físico....55
Figura 38. Historia de usuario 1, Login. 56
Figura 39. Historia de usuario 1, Restablecer contraseña. 57
Figura 40. Historia de usuario 1, Pruebas unitarias. 57
Figura 41. Historia de usuario 2, Crear paciente. 58
Figura 42. Historia de usuario 2, Editar paciente. 59
Figura *43*. Historia de usuario 2, Eliminar paciente. 59
Figura 44. Historia de usuario 2, Listado de pacientes. 60
Figura *45*. Historia de usuario 2, Detalle de paciente por id. 60
Figura 46. Historia de usuario 2, Buscar pacientes. 61
Figura *47*. Historia de usuario 2, Pruebas unitarias. 61
Figura *48*. Historia de usuario 3, Crear cita médica. 63
Figura 49. Historia de usuario 3, Editar cita médica. 64
Figura 50. Historia de usuario 3, Eliminar cita médica. 64
Figura 51. Historia de usuario 3, Detalle de cita por id. 65
Figura 52. Historia de usuario 3, Calendario de citas. 65
Figura 53. Historia de usuario 3, Buscar citas. 65
Figura 54. Historia de usuario 3, Enviar recordatorios de citas. 66
Figura 55. Historia de usuario 3, Pruebas unitarias. 66
Figura 56. Historia de usuario 4, Crear historia clínica. 67
Figura 57. Historia de usuario 4, Editar historia clínica. 68
Figura 58. Historia de usuario 4, Eliminar historia clínica. 68
Figura 59. Historia de usuario 4, Detalle de historia clínica por id. 69
Figura 60. Historia de usuario 4, Buscar historias clínicas. 69
Figura 61. Historia de usuario 4, Listado de historias clínicas. 69
Figura 62. Historia de usuario 4, Pruebas unitarias. 70
Figura 63. Historia de usuario 5, Crear evolución. 72
Figura 64. Historia de usuario 5, Editar evolución. 72
Figura 65. Historia de usuario 5, Eliminar evolución. 73
Figura 66. Historia de usuario 5, Detalle de evolución por id. 73
Figura 67. Historia de usuario 5, Listado de evoluciones. 74
Figura 68. Historia de usuario 5, Buscar evoluciones. 74
Figura 69. Historia de usuario 5, Generar certificado médico. 75
Figura 70. Historia de usuario 5, Generar certificado médico. 75
Figura 71. Historia de usuario 5, Pruebas unitarias. 76
Figura 72. Historia de usuario 6, Crear usuario. 77
Figura 73. Historia de usuario 6, Editar usuario. 78
Figura 74. Historia de usuario 6, Eliminar usuario. 78
Figura 75. Historia de usuario 6, Detalle de usuario por id. 79
Figura 76. Historia de usuario 6, Listado de usuarios. 79
Figura 77. Historia de usuario 6, Buscar usuarios. 79
Figura 78. Historia de usuario 6, Recuperar contraseña. 80
Figura 79. Historia de usuario 6, Actualizar datos. 80
Figura 80. Historia de usuario 6, Cambiar contraseña. 80

Figura 81. Historia de usuario 6, Pruebas unitarias. 81
Figura 82. Historia de usuario 7, Crear médico. 82
Figura 83. Historia de usuario 7, Editar médico. 83
Figura 84. Historia de usuario 7, Eliminar médico. 83
Figura 85. Historia de usuario 7, Detalle de médico por id. 84
Figura 86. Historia de usuario 7, Listado de médicos. 84
Figura 87. Historia de usuario 7, Buscar médicos. 84
Figura 88. Historia de usuario 7, Pruebas unitarias. 85
Figura 89. Historia de usuario 8, Crear consultorio. 86
Figura 90. Historia de usuario 8, Editar consultorio. 86
Figura 91. Historia de usuario 8, Eliminar consultorio. 87
Figura 92. Historia de usuario 8, Detalle de consultorio por id. 87
Figura 93. Historia de usuario 8, Listado de consultorios. 88
Figura 94. Historia de usuario 8, Buscar consultorios. 88
Figura 95. Historia de usuario 8, Pruebas unitarias. 89

Índice de tablas

Tabla 1 Clasificación de las bases de datos. 20
Tabla 2 Requerimientos funcionales. 42
Tabla 3 Requerimientos no funcionales. 43
Tabla 4 Requerimientos mínimos. 43
Tabla 5. Historia de usuario 1, Endpoints 56
Tabla 6. Historia de usuario 2, Endpoints 58
Tabla 7. Historia de usuario 3, Endpoints 62
Tabla 8. Historia de usuario 4, Endpoints 67
Tabla 9. Historia de usuario 5, Endpoints 71
Tabla 10. Historia de usuario 6, Endpoints 76
Tabla 11. Historia de usuario 7, Endpoints 82
Tabla 12. Historia de usuario 8, Endpoints 85

1. ASPECTOS INTRODUCTORIOS

1.1 DATOS DEL CENTRO MÉDICO

El centro médico ANEZ, ubicado en la ciudad Quito, parroquia Benalcázar, se establece como una entidad médica privada en formación que brinda servicios de salud con calidad mediante consultas en medicina general, procedimientos quirúrgicos menores y tratamientos estéticos no invasivos a nivel facial y corporal, manteniendo siempre la integridad de sus pacientes.

A continuación, se presenta la misión y visión, ANEZ (2021):

1.1.1 MISIÓN

Brindar servicios de salud integral mediante la aplicación de técnicas y tecnología en tratamientos médicos de salud facial y corporal, para el mejoramiento la calidad de vida.

1.1.2 VISIÓN

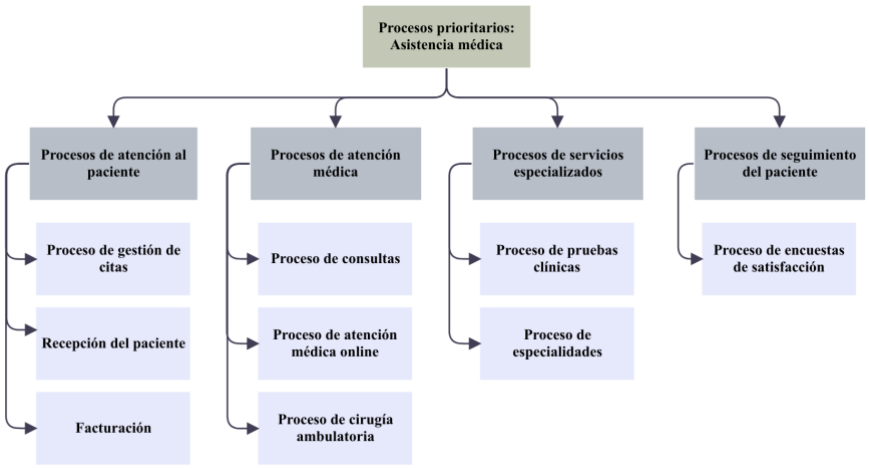
Ser un centro de salud integral, capaz de entregar una atención personalizada con calidad y calidez para el mejoramiento del estilo de vida de las personas.

1.1.3 MAPA DE PROCESOS ASISTENCIALES

Los procesos asistenciales o de asistencia médica comprenden todas aquellas actividades clínico – asistenciales cuyo principal enfoque radica en aumentar la satisfacción del paciente y facilitar las tareas de los profesionales (Goldberg et al., 2006).

Figura 1

Clasificación de procesos asistenciales



Nota. Adaptada de Alcaraz, 2020.

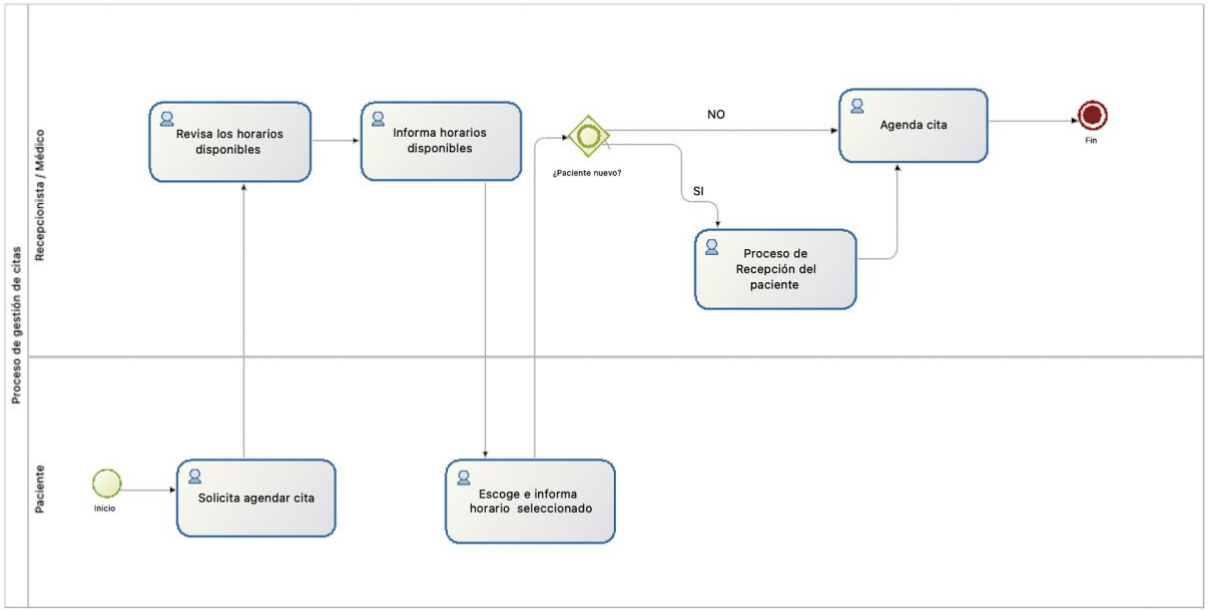
Los principales procesos que se manejan e implementan en el centro médico a través del sistema web son los siguientes:

1. Proceso de Gestión de citas

Figura 2.

Diagrama de procesos, Gestión de citas.

MACROPROCESO:	PROCESOS PRIORITARIOS: ASISTENCIA MÉDICA	**CÓDIGO:**	ANEZ_ACM_1
PROCESO:	PROCESOS DE ATENCIÓN AL PACIENTE	**VERSIÓN:**	1.0
SUBPROCESO:	PROCESO DE GESTIÓN DE CITAS	**FECHA:**	1/3/21
PROCEDIMIENTO		**PAGINA 1 DE 1**	



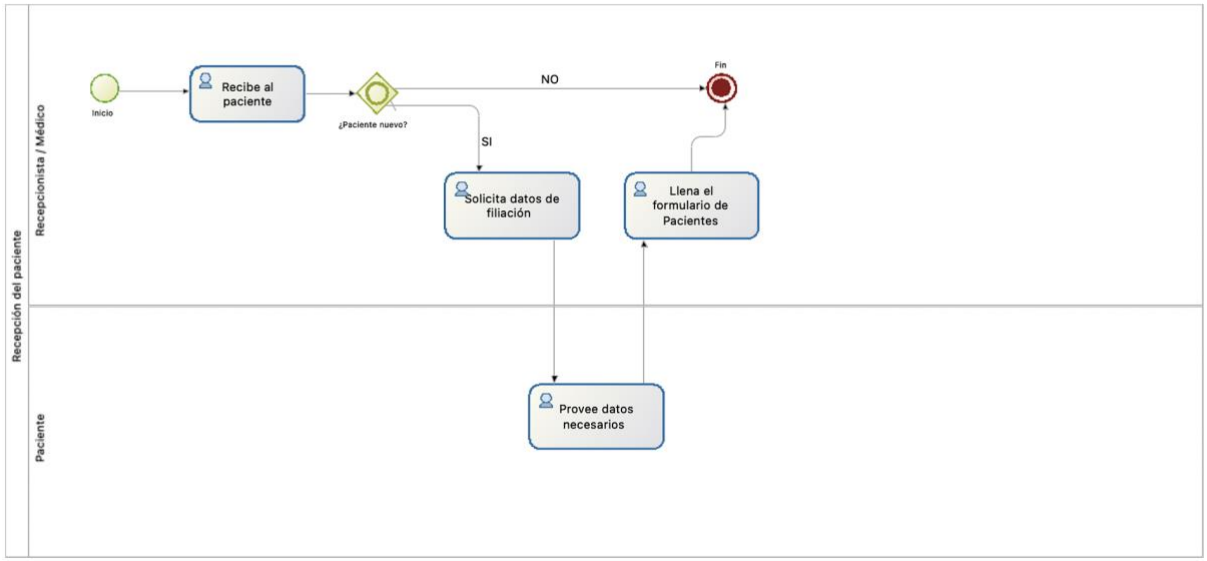
Nota. E. Zapata, Gerente general del centro médico ANEZ (comunicación personal, 22 de febrero, 2021)

2. Proceso de Recepción del paciente.

Figura 3.

Diagrama de procesos, Recepción del paciente.

MACROPROCESO:	PROCESOS PRIORITARIOS: ASISTENCIA MÉDICA	**CÓDIGO:**	ANEZ_RP_1
PROCESO:	PROCESOS DE ATENCIÓN AL PACIENTE	**VERSIÓN:**	1.0
SUBPROCESO:	RECEPCIÓN DEL PACIENTE	**FECHA:**	1/3/21
		PAGINA 1 DE 1	



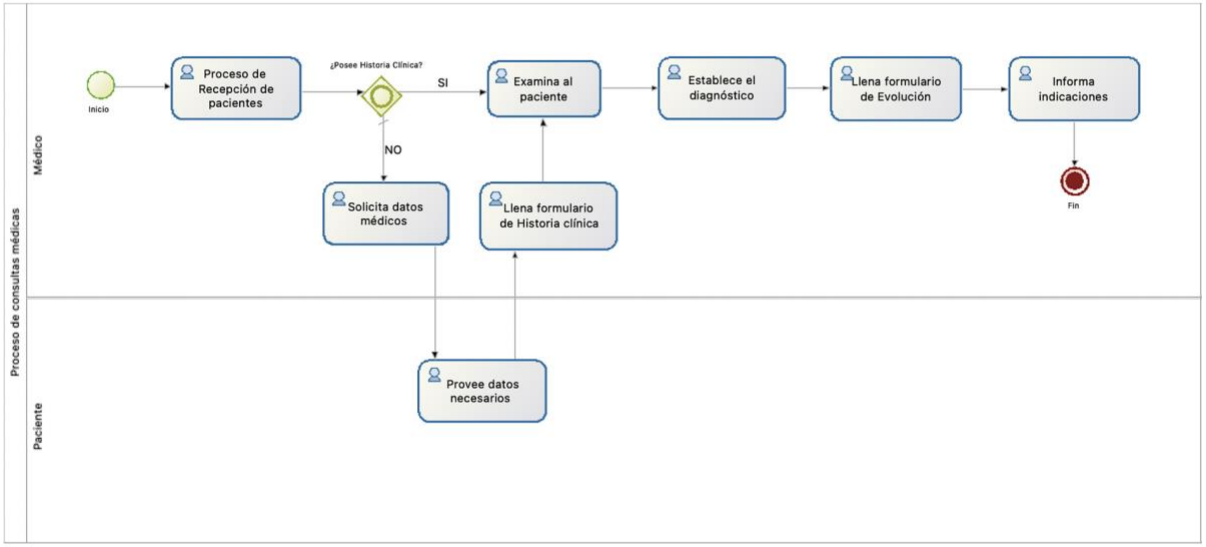
Nota. E. Zapata, Gerente general del centro médico ANEZ (comunicación personal, 22 de febrero, 2021)

3. Proceso de consultas médicas

Figura 4.

Diagrama de procesos, Consultas médicas.

MACROPROCESO:	PROCESOS PRIORITARIOS: ASISTENCIA MÉDICA	**CÓDIGO:**	ANEZ_CM_1
PROCESO:	PROCESOS DE ATENCIÓN MÉDICA	**VERSIÓN:**	1.0
SUBPROCESO:	PROCESO DE CONSULTAS MÉDICAS	**FECHA:**	1/3/21
PROCEDIMIENTO		**PAGINA 1 DE 1**	



Nota. E. Zapata, Gerente general del centro médico ANEZ (comunicación personal, 22 de febrero, 2021)

1.2 JUSTIFICACIÓN

La correcta implementación de los sistemas de información en el área de salud como historias clínicas electrónicas y sistemas de registros médicos digitales permiten mejorar considerablemente la gestión y procesos clínicos, el control eficiente de los recursos hospitalarios, así como el manejo de riesgos en cuanto a la divulgación de contenido no autorizado o alteración de la información (Yoinne et al., 2016). De ahí que, el presente proyecto se enfoca en desarrollar un sistema web eficiente e integrado para el soporte de procesos del centro médico ANEZ, éste permitirá viabilizar una adecuada gestión de citas, historias clínicas y consultas.

Para el desarrollo de este proyecto, se hará uso de la metodología XP, la cual se basa en distintos valores, principios y técnicas para construir un producto que se alinee con los requerimientos del cliente mediante una comunicación y retroalimentación constante. Asimismo, el ciclo de vida con el que cuenta esta metodología: Exploración, Planificación, Iteraciones y Puesta en producción resulta eficiente ya que su tasa de errores es mínima y su respuesta ante cambios inesperados es rápida, manteniendo el bienestar entre los desarrolladores y los clientes y, por consiguiente, asegurando el éxito del proyecto.

La importancia del presente trabajo de titulación radica en integrar los sistemas informáticos con la medicina; por lo que, el sistema web impulsará el uso de herramientas tecnológicas en el ámbito de la salud, constituyéndose como un antecedente para futuras investigaciones en las que se pretenda diseñar, desarrollar e

implementar un sistema para un establecimiento médico ya sea público o privado y que se identifique con las características del centro médico ANEZ.

1.3 PLANTEAMIENTO DEL PROBLEMA

Los sistemas informáticos se han convertido en una pieza clave para la administración de todos aquellos procesos en los que es fundamental contar con accesibilidad, confidencialidad e integridad de datos que se manejan, puesto que una gestión deficiente de los datos e información limitan la toma de decisiones, perjudicando considerablemente la calidad del servicio y la satisfacción del usuario (Lezama López & Machuca Abanto, 2017).

En este contexto, con el presente proyecto se pretende desarrollar un sistema web para el centro médico ANEZ, una entidad médica que brinda servicios de salud con calidad mediante consultas en medicina general, procedimientos quirúrgicos menores y tratamientos estéticos no invasivos a nivel facial y corporal, manteniendo siempre la integridad de sus pacientes.

Los principales procesos asistenciales que manejará son:

- Agendamiento de citas.
- Administración de pacientes.
- Administración de historias clínicas.

Al realizarse estos procesos de forma manual existe un uso ineficiente de recursos y la probabilidad de pérdida de información aumenta, lo cual se traduce en una prestación de servicios con un nivel de calidad inapropiado.

En función a lo expuesto, se identifica la siguiente pregunta principal del proyecto de titulación:

- ¿Cómo influye el desarrollo de un sistema web para la automatización de procesos asistenciales en el centro médico ANEZ?

Y las siguientes preguntas secundarias:

- ¿Diseñando el modelo respectivo de la base de datos se conseguirá un manejo eficiente de citas, datos de pacientes e historias clínicas?
- ¿Se encuentra la información sobre las citas, datos de pacientes e historias clínicas centralizada?
- ¿Es posible reducir la probabilidad de pérdida de información sobre citas, datos de pacientes e historias clínicas?

1.4 ALCANCE

El presente trabajo de disertación pretende desarrollar un sistema web para la automatización de procesos asistenciales del centro médico ANEZ, por lo tanto, inicia con el análisis de requerimientos de dicho centro médico, seguido por la definición del diseño de base de datos, el desarrollo del sistema web y finaliza con la implementación del sistema.

El sistema web contará con los siguientes módulos:

- Agendamiento de citas.
- Administración de pacientes.

- Administración de historias clínicas.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

- Diseñar, desarrollar e implementar un sistema web para la automatización de procesos asistenciales del centro médico ANEZ.

1.5.2 OBJETIVOS ESPECÍFICOS

- Analizar los requerimientos del centro médico para el diseño y desarrollo del sistema.
- Diseñar una base de datos acorde a los requerimientos del Centro médico ANEZ.
- Desarrollar en sistema para la gestión de citas, datos del paciente e historias clínicas.
- Implementar el sistema web en el Centro médico ANEZ.

2. FUNDAMENTO TEÓRICO

2.1 SISTEMAS WEB

Actualmente, el internet es uno de los más importantes medios de comunicación, razón por cual cada vez la demanda de sistemas web incrementa y por ende el surgimiento de nuevas e innovadoras aplicaciones web que ofrezcan diversos tipos de servicios.

Molina Rios et al. (2017) establecen que las aplicaciones web son "herramientas que permiten realizar operaciones desde un ordenador a través de la utilización del Internet" (p. 58) y utilizan el "formato estándar HTML (HyperText Markup Language o Lenguaje de Hipertextos) para efectuar las peticiones que el usuario desea" (p. 58).

Otro concepto muy similar lo presenta Sataloff et al. (2018) quienes definen a un sistema web como "un programa informático o sitio Web que se ejecuta en el Internet sin necesidad de una instalación en el ordenador, tan solo con el uso de un navegador, ya que se programa en lenguaje HTML" (p. 33).

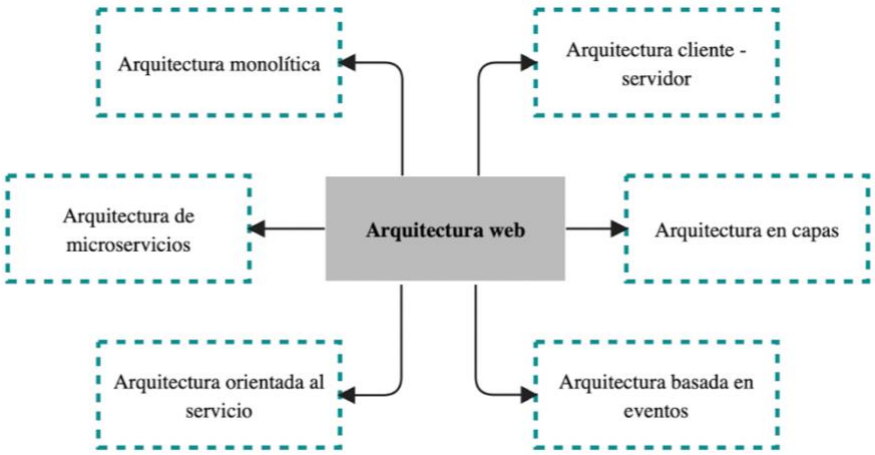
2.2 ARQUITECTURA WEB

Una arquitectura de aplicaciones proporciona un plan y prácticas recomendadas para el diseño y desarrollo de sistemas, mediante la descripción de patrones y técnicas que aseguran la obtención de una aplicación correctamente estructurada (Red Hat Inc., 2021).

Los principales tipos de arquitectura son:

Figura 5

Tipos de Arquitectura



Nota. Adaptada de (Red Hat Inc., 2021).

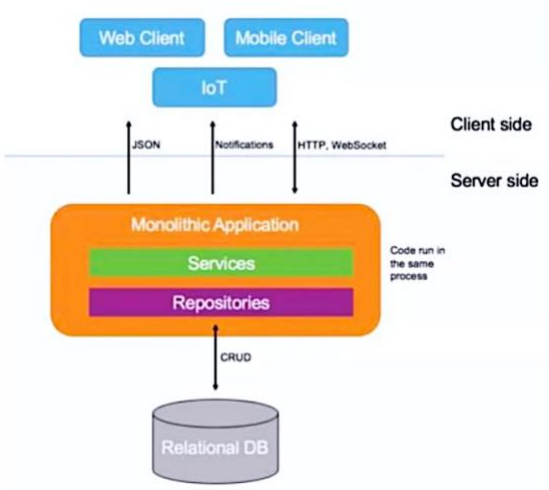
2.2.1 ARQUITECTURA MONOLÍTICA

Tipo de arquitectura generalmente asociado con los sistemas heredados, se caracteriza por contener todas las funciones dentro de una misma aplicación.

Debido a que todos los aspectos funcionales del sistema están acoplados y dependen de un mismo programa, como se presenta en la Figura 6, la escalabilidad o actualización de sistemas monolíticos implica una gran repercusión en la aplicación y su infraestructura subyacente (Red Hat Inc., 2021).

Figura 6

Arquitectura monolítica.



Nota. Tomada de Rodríguez et al., 2019.

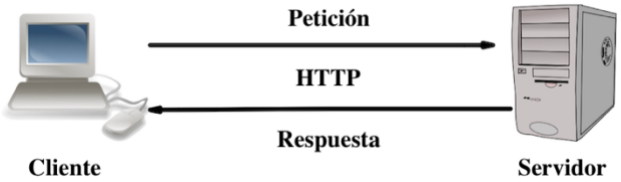
2.2.2 ARQUITECTURA CLIENTE – SERVIDOR

Arquitectura tradicional, consiste básicamente en realizar peticiones que viajan normalmente por TCP (Protocolo de Control de Transmisión) desde el lado del cliente al servidor y devolver respuestas desde el lado del servidor al cliente.

Las peticiones y respuestas pueden ser enviadas en forma de texto HTML, imágenes u otros (Duarte Vega, 2016).

Figura 7

Arquitectura cliente – servidor.



Nota. Adaptada de Rodríguez et al., 2019.

2.2.3 ARQUITECTURA EN CAPAS

Se diferencia del modelo cliente – servidor por la distribución de capas para el procesamiento de peticiones.

El desarrollo de aplicaciones se puede manejar en varios niveles; sin embargo, como se muestra en la Figura 8, el diseño en tres capas es el más utilizado debido a que distribuye jerárquicamente las responsabilidades lo cual facilita la escalabilidad y mantenimiento del sistema (Duarte Vega, 2016).

Figura 8

Arquitectura en capas.



Nota. Adaptada de Rodríguez et al., 2019.

Capa de presentación: también conocida como capa de usuario, presenta la interfaz del sistema al usuario, esta capa únicamente se comunica con la capa de negocio.

Capa de negocio: contempla la lógica del negocio, se comunica con la capa de presentación para receptar solicitudes y presentar resultados, y a su vez con la capa de datos para solicitar el almacenamiento o recuperación datos.

Capa de datos: formada por uno o más gestores de base de datos, se encarga de almacenar y acceder a los datos.

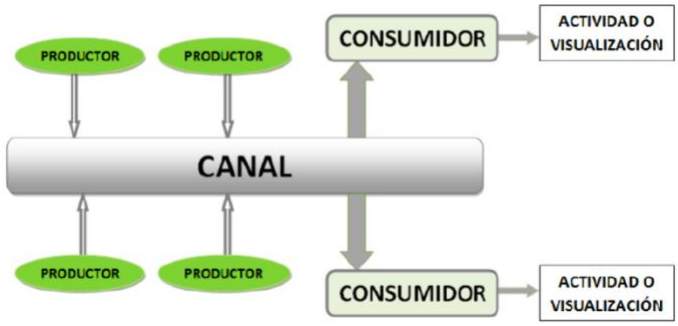
2.2.4 ARQUITECTURA BASADA EN EVENTOS

EDA (Event Driven Architecture) es un tipo de arquitectura orientada a objetos cuya estructura central de la solución se basa en la captura, comunicación, procesamiento y permanencia de eventos.

Se caracteriza por permitir un acoplamiento mínimo, siendo ideal para las arquitecturas de aplicaciones distribuidas y modernas (Red Hat Inc., 2021).

Figura 9

Arquitectura orientada a eventos EDA.



Nota. Tomada de (Moreno Saiz, 2015).

Como se aprecia en la Figura 9, la arquitectura consta de cuatro elementos productores, canal, consumidores y actividad.

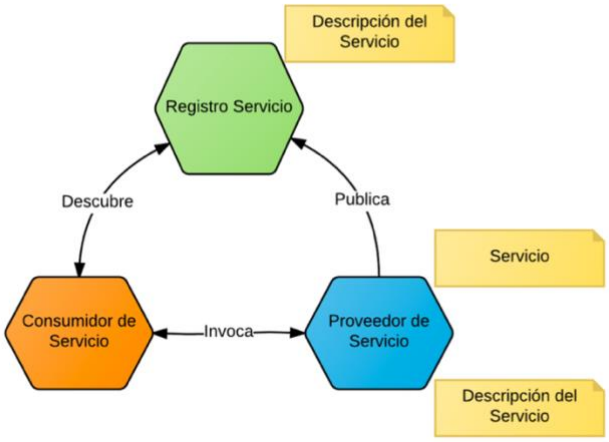
De acuerdo con Duarte Vega (2016) los eventos son generados por un productor los cuales son transmitidos por un canal hasta llegar al consumidor el mismo que, es el encargado de llevar a cabo determinada actividad.

2.2.5 ARQUITECTURA ORIENTADA AL SERVICIO

También conocida por sus siglas en inglés SOA (Service Oriented Architecture) es una arquitectura que "orquesta el funcionamiento del software a partir de un servidor central (Registro Servicio) quien es el que reparte los servicios entre diferentes servidores y estos a su vez lo llevan hacia sus clientes respectivos" (Duarte Vega, 2016, p. 8).

Figura 10

Arquitectura Orientada a Servicios.



Nota. Tomada de Duarte Vega, 2016.(Moreno Saiz, 2015)(Moreno Saiz, 2015)(Moreno Saiz,

En la Figura 10, se muestra cómo los servicios independientes y reutilizables se comunican entre sí mediante un bus de servicios empresariales (ESB).

La base de esta arquitectura al igual que el modelo cliente – servidor son las conexiones realizadas mediante petición – respuesta (Duarte Vega, 2016).

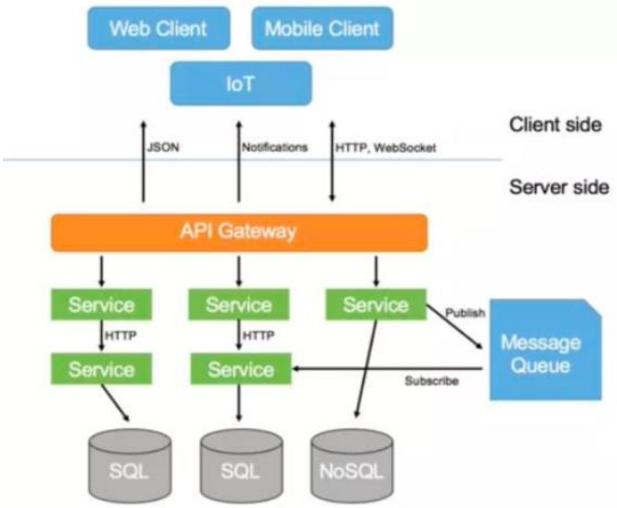
2.2.6 ARQUITECTURA DE MICROSERVICIOS

La principal característica de este tipo de arquitectura radica en que los servicios pueden ser acoplados de forma flexible debido a que, cada uno de los servicios cuenta con un código base pequeño e independiente.

La Figura 11 presenta la estructura de un sistema distribuido que ejecuta varios servicios en distintos servidores o hosts, habitualmente se utilizan dos protocolos de comunicación, la mensajería asincrónica ligera y la petición – respuesta HTTP por medio de APIs (Application Programming Interfaces) (Rodríguez et al., 2019).

Figura 11

Arquitectura de microservicios.



Nota. Tomada de Rodríguez et al., 2019.

2.3 SERVICIOS WEB

Los servicios web surgen ante la necesidad de permitir a las computadoras conectadas a internet comunicarse entre sí y con sistemas heterogéneos dentro y fuera de una organización (Severance, 2020).

Existen diversas definiciones de Servicios web, lo cual demuestra la complejidad de los servicios e implicaciones asociadas a este término, así tenemos:

IBM, (2020) define a los servicios web como "aplicaciones autónomas modulares que se pueden describir, publicar, localizar e invocar a través de una red" (párr. 1).

Por su parte, Lamarca M. (2013) señala que los Servicios web se describen como un "conjunto de aplicaciones o tecnologías con capacidad para inter operar en la Web" (párr. 1) y capaces de intercambiar "datos entre ellas con el fin de ofrecer unos servicios" (párr. 1).

Por lo anteriormente expuesto y citando al Dept. Ciencia de la Computación e IA de la Universidad de Alicante (2014), un servicio web se define como un componente de software capaz de comunicarse con otras aplicaciones mediante protocolos web estandarizados.

Los tipos de servicios web se describen a continuación:

2.3.1 SERVICIO WEB SOAP

SOAP (Simple Object Access Protocol) se define como un "formato de mensaje XML (Extensible Markup Language) utilizado en interacciones de servicios web" (párr. 1), generalmente los mensajes

SOAP son enviados a través de HTTP o JMS, sin embargo, puede utilizar otros protocolos (IBM, 2019).

La empleabilidad de SOAP se rige por un conjunto de protocolos ligeros y estándares de comunicación basados en XML, SOAP para la transferencia de datos y WSDL (Web Services Description Language) para la descripción de las funcionalidades del servicio web (Da Silva, 2013; Sayago et al., 2019).

2.3.2 SERVICIO WEB RESTFUL

Como lo plantea Sayago et al. (2019), el servicio web RESTful se identifica por ser un diseño basado en la arquitectura REST (REpresentational State Transfer), la misma que se enfoca en construir aplicaciones empleando el protocolo HTTP de manera explícita, lo cual favorece a la optimización de recursos y por ende mejora el trabajo en torno a aplicaciones y sitios web.

Las principales características de RESTful son:

- Establecer una comunicación sin mantener estados, lo cual significa una optimización de almacenamiento en memoria caché.
- Transferir datos en formatos XML o JSON.
- Exponer los métodos estandarizados GET, POST, PUT y DELETE, así como la reutilización de mecanismos de autenticación, cifrado y autorización por HTTP.

2.3.3 GRAPHQL

Sayago et al. (2019) describe a GraphQL como un lenguaje de consulta desarrollado por Facebook en 2012 con el objetivo de reducir la sobrecarga en términos de cantidad de datos innecesarios y número de consultas necesarias para la transferencia de datos y a su vez, disminuir los errores causados por consultas no válidas, principales inconvenientes suscitados con los servicios web existentes hasta entonces.

GraphQL surgió como una alternativa a REST y SOAP; esto significa que combina varias características de ambos servicios web y lo convierte en una tecnología mucho más ligera y capaz de reducir la carga útil de la red.

2.4 BASE DE DATOS

Las bases de datos surgen a partir de la necesidad de almacenar grandes cantidades de información o datos, desde la aparición de las primeras computadoras, el concepto de base de datos siempre ha estado estrechamente ligado a la informática y se consideran un elemento fundamental en los sistemas de información debido a la capacidad de recuperación ante fallos, gestión centralizada y muchas otras funcionalidades que garantizan la seguridad, consistencia y accesibilidad a los datos.

Básicamente, una base de datos se define como “una colección de información organizada de modo que sea fácilmente accesible, gestionada y actualizada” (Pulido et al., 2019, p.18).

Pulido et al. (2019) destacan las siguientes características de las bases de datos:

- Facilitar la independencia de los datos.
- Reducir la redundancia, lo cual permite un mayor aprovechamiento del espacio y evita inconsistencias entre los datos.
- Proporcionar seguridad a los datos frente a usuarios no autorizados.
- Mantener la integridad de los datos mediante restricciones de entidad, referenciales, de dominio y aquellas definidas por el usuario con la finalidad de evitar que los datos se corrompan.
- Mantener la coherencia entre todos los datos almacenados en la base de datos.

De acuerdo con el modelo de la base de datos, se clasifican en:

Tabla 1

Clasificación de las bases de datos.

Base de datos jerárquica	Los datos se organizan en forma de árbol invertido, en el que cada nodo padre o nodo raíz puede tener uno o varios hijos también denominados hojas.
Base de datos de red	La principal diferencia con el modelo jerárquico radica en la modificación del concepto de nodo dado que un mismo nodo provenga de varios padres, concepto no admitido en el modelo jerárquico. Provee una solución eficiente al problema de redundancia de datos.
Base de datos transaccionales	Modelo poco común, generalmente utilizado en el entorno de análisis de calidad y datos de producción. Su principal objetivo es el envío y recepción de datos a la mayor velocidad posible
Base de datos relacionales	Se caracteriza por ser el modelo más utilizado para representar problemas reales y administrar datos de forma dinámica, se fundamenta en el uso de “relaciones” las

	mismas que, son consideradas en forma lógica como conjuntos de datos.
Base de datos multidimensionales	Bases de datos bastante similares a las relacionales, la principal diferencia reside a nivel conceptual debido a que en las bases de datos multidimensionales los campos de una tabla pueden ser de dos tipos o representar dimensiones de la tabla.
Base de datos orienta a objetos	El propósito del modelo es el almacenamiento de objetos completos, es decir, su estado y comportamiento.
Base de datos documentales	Usualmente utilizadas para el almacenamiento de grandes volúmenes de información como registros históricos, permite la indexación a texto completo y la realización de búsquedas potentes.
Base de datos deductivas	También denominadas bases de datos lógicas permiten realizar deducciones a través de inferencias basadas en reglas y hechos almacenados en la base de datos.

Nota. Adaptada de Pulido et al., 2019.

2.5 TECNOLOGÍAS

2.5.1 BACK-END

Dentro del contexto de las aplicaciones web, el back-end se identifica como la parte lógica de la aplicación, es decir, es el encargado de ejecutar aquellos procesos necesarios para que la página o sistema web funcionen de manera adecuada, abarca todas las actividades realizadas del lado del servidor, incluyendo las tareas de base de datos y los servidores de aplicaciones.

Los lenguajes de programación comúnmente utilizados son PHP, Java, Ruby, .NET, Python, entre otros. (Pinto et al., 2016; Jovaldiv, 2016).

2.5.2 FRONT-END

Se define como la interfaz gráfica de usuario (GUI), se caracteriza por ser la parte de la aplicación web que interactúa con el usuario, razón por la cual debe cumplir con estándares de usabilidad y contar con un diseño adecuado. Generalmente estas tecnologías son desarrolladas en lenguajes como HTML, CSS, JavaScript.

En el ámbito del desarrollo de aplicaciones web, es posible implementar el front-end de una aplicación sin contar con la parte del back-end, sin embargo, comúnmente es el responsable de enviar peticiones y consumir los recursos provistos por el back-end (Jovaldiv, 2016).

2.6 HERRAMIENTAS

2.6.1 BASE DE DATOS

2.6.1.1 POSTGRESQL

PostgreSQL se define como un avanzado sistema de bases de datos relaciones Open Source que incluye características de la orientación a objetos como la herencia, tipos de datos, funciones, restricciones, triggers, integridad transaccional, full ACID, entre otros.

Se caracteriza por usar el modelo cliente/servidor y ser un sistema estable, de alto rendimiento, gran flexibilidad y compatibilidad (Aliaga & Miani, 2008; Denzer, 2002)

2.6.2 BACKEND

2.6.2.1 NODE.JS

Se define como un "entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome" (Node.js, 2021, párr. 1), creado por Ryan Dahl a principios de 2009, permite construir rápidas y escalables aplicaciones de red mediante la integración de un conjunto de librerías básicas o módulos nativos.

Utiliza el modelo de entrada y salida (E/S) sin bloqueo y completamente controlado por eventos; esto lo hace ligero y eficiente, ideal para aplicaciones en tiempo real (Muñoz, 2013).

2.6.2.2 EXPRESS.JS

Proyecto de la Fundación OpenJS, lanzado inicialmente en 2010, se define como un framework rápido, minimalista y flexible para Node.js, proporciona un conjunto robusto de características para el desarrollo de aplicaciones web y móviles (Express.js, 2021 & Rick, 2016).

2.6.2.3 SEQUELIZE

ORM basado en promesas para Node.js, es compatible con los dialectos PostgreSQL, MySQL, MariaDB, SQLite y Microsoft SQL Server. Además cuenta con un sólido soporte para la ejecución de transacciones, relaciones, replicación de datos, entre otros (Sequelize, 2021).

2.6.3 FRONTEND

2.6.3.1 REACT.JS

De acuerdo con Veps (2017), React.js se precisa como una librería para JavaScript de software libre creada por Facebook en 2013, principalmente enfocada al desarrollo de interfaces de usuario mediante el uso de componentes fácilmente reutilizables, es ideal para todo tipo de aplicaciones web y Single Page Application (SPA).

Se destaca por trabajar con un Virtual DOM (Document Object Model) lo cual le provee mayor velocidad y gran performance para el renderizado de las vistas.

Permite su integración con frameworks como React, Angular, Meteor y Ember. (Semantic-Org, 2018)

2.6.3.2 REACT SUITE

Desarrollado en 2016 por el equipo de HYPERS y UX, se establece como una librería de componentes para React.js utilizada para el desarrollo de sistemas empresariales.

Cuenta con una amplia variedad de componentes que incluyen tablas, formularios, selectores de fecha, calendarios, portales, entre otros (HYPERS, 2016).

2.6.3.3 SEMANTIC UI REACT

Framework orientado al diseño de interfaces, permite a los desarrolladores una ágil creación de agradables sitios web mediante el

uso de HTML conciso, JavaScript intuitivo y una depuración simplificada (Semantic-Org, 2018).

2.6.4 PRUEBAS

2.6.4.1 JEST

Framework de JavaScript diseñado para crear, ejecutar y estructurar pruebas, es ampliamente utilizado para testear código de JavaScript incluyendo aplicaciones desarrolladas en React.

Se destaca por proporcionar una experiencia integrada de configuración cero, brinda un entorno rápido y aislado de pruebas y permite la fácil creación de informes de cobertura de código integrados (Pérez et al., n.d.; Facebook Inc, 2017).

2.6.5 DESPLIEGUE

2.6.5.1 AMAZON WEB SERVICES (AWS)

Plataforma en la nube que surgió en 2006, ofrece más de 200 servicios integrales entre los cuales se encuentran potencia de computación, opciones de almacenamiento, redes y bases de datos que se suministran como una utilidad.

Proporciona una infraestructura altamente fiable, disponible, escalable y de bajo costo (Amazon Web Services Inc, 2018).

Los servicios de AWS que se utilizarán en el presente proyecto son:

- *RDS:* Amazon Relational Database Service se establece como un servicio de base de datos relacional, sencillo de configurar,

utilizar y escalar, incluye los siguiente motores de bases de datos: Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database y SQL Server.

Se caracteriza por su alto rendimiento, disponibilidad, seguridad y compatibilidad (Amazon Web Services Inc, 2021b).

- *EC2:* Amazon Elastic Compute Cloud es un "servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable" (Amazon Web Services Inc, 2021ª, párr. 1), está diseñado para simplificar a los desarrolladores el uso de la informática en la nube a escala web, además, proporciona un control completo sobre los recursos informáticos a nivel de procesador, almacenamiento, red y sistema operativo (Amazon Web Services Inc, 2021a).

2.6.5.2 VERCEL

Plataforma en la nube lanzada en 2016 con el objetivo de facilitar a los desarrolladores el despliegue de sus aplicaciones.

Se caracteriza por ser una plataforma de despliegue y colaboración para desarrolladores Frontend que permite el alojamiento tanto de sitios web como de servicios web de forma instantánea y sin configuración.

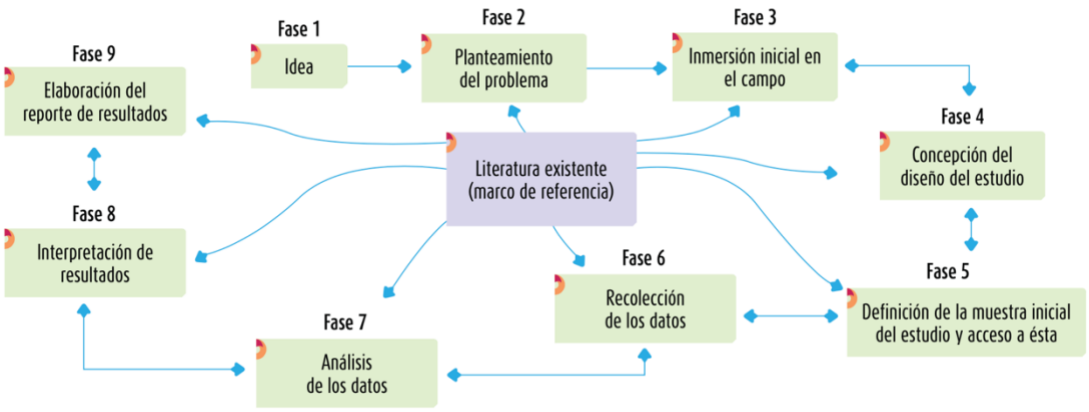
Vercel se integra directamente con sistemas para el control de versiones como GitHub, GitLab o Bitbucket lo cual facilita el despliegue de aplicaciones (Vercel Inc, 2021).

2.7 METODOLOGÍA

La metodología cualitativa representa un proceso circular, que carece de un patrón predecible y estructurado, por lo que es posible retroceder, eludir o ejecutar varias fases de manera simultánea. No obstante, esto varía dependiendo del estudio.

Su objetivo es reconstruir la realidad mediante el conocimiento adquirido al introducirse en la experiencias de los participantes y entender la perspectiva de los actores de un sistema social previamente definido (Hernández et al., 2014).

Figura 12.
Proceso cualitativo.



Nota. Tomada de Hernández et al., 2014.

De acuerdo con Hernández et al., 2014, el proceso cualitativo (Figura 12) es complejo y flexible, por lo que es necesario destacar los siguientes aspectos:

- Existe una revisión inicial de la literatura, sin embargo, esta puede complementarse desde cualquier fase de la investigación.

- La fase 3 aborda todas aquellas actividades enfocadas en conocer el ambiente o entorno en el cual se desarrollará el estudio, permite verificar la factibilidad del mismo.
- La investigación cualitativa admite el retorno a fases previas por lo que, a partir de la fase 3 hasta la fase 9 se visualizan flechas en doble sentido.
- Las fases relacionadas con la muestra, recolección y análisis pueden realizarse de manera simultánea.

2.8 METODOLOGÍA DE DESARROLLO

2.8.1 METODOLOGÍAS ÁGILES

Las metodologías ágiles surgen en 2001 como resultado de una reunión celebrada en Utah, Estados Unidos, donde un grupo de expertos desarrolladores de software manifestaron la importancia de que un equipo de desarrollo responda de manera oportuna a cambios o cualquier tipo de inconveniente que se presente a lo largo de la ejecución de un proyecto de software (López Menéndez de Jiménez, 2015).

A consecuencia de lo anteriormente expuesto, se desarrolló un manifiesto que contempla todos aquellos principios y valores necesarios para el desarrollo de software ágil.

De acuerdo con Bahit (2012), los doce principios establecidos en el manifiesto son:

1. Satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo.
3. Entregar software funcional en el periodo de tiempo más corto posible.
4. Los desarrolladores y el cliente trabajan de forma cotidiana durante todo el desarrollo del proyecto.
5. Desarrollar proyectos en torno a individuos motivados.
6. Conversar de cara a cara con el cliente y con el equipo de desarrolladores.
7. Medir el progreso del proyecto mediante la entrega de software completamente funcional.
8. Mantener un ritmo constante de trabajo, esto aplica tanto a promotores, desarrolladores como a usuarios.
9. Verificar continuamente la excelencia técnica y al buen diseño.
10. Maximizar la cantidad de trabajo no realizado en cada ciclo.
11. Construir un equipo de trabajo auto organizado.
12. Maximizar la calidad del trabajo mediante una retroalimentación continua.

Asimismo, Bahit (2012) lista las principales metodologías ágiles, entre ellas se encuentran:

- SCRUM
- Extreme Programming (XP)
- Crystal Clear

Cada una de estas metodología permite la gestión de proyectos adaptativos para el desarrollo de software; no obstante, para el desarrollo del sistema web se ha seleccionado Extremme Programming, misma que es descrita a continuación.

2.8.2 EXTREME PROGRAMMING

Más conocida por sus sigla en inglés, XP es considerada según Laínez Fuentes (2015) como una metodología leve de desarrollo de software, que surgió a partir de las ideas de Kent Beck y Ward Cunningham en 1996 con la finalidad de resolver los problemas de entrega de software de calidad en el menor tiempo posible y cumplir con todas las necesidades de negocio que se mantienen en constante cambio.

Extreme Programming destaca por su simplicidad, comunicación constante con todos los involucrados en el proyecto y sobretodo la retroalimentación permanente en cada fase, XP tiene como objetivos satisfacer al cliente, potenciar el trabajo en equipo y minimizar el riesgo en cuanto a costos, tiempo, calidad y alcance.

Generalmente se utiliza en proyectos con equipos pequeños o medianos y en los cuales los requerimientos cambien con un nivel de frecuencia alto (Laínez Fuentes, 2015; Montero Molina et al., 2018).

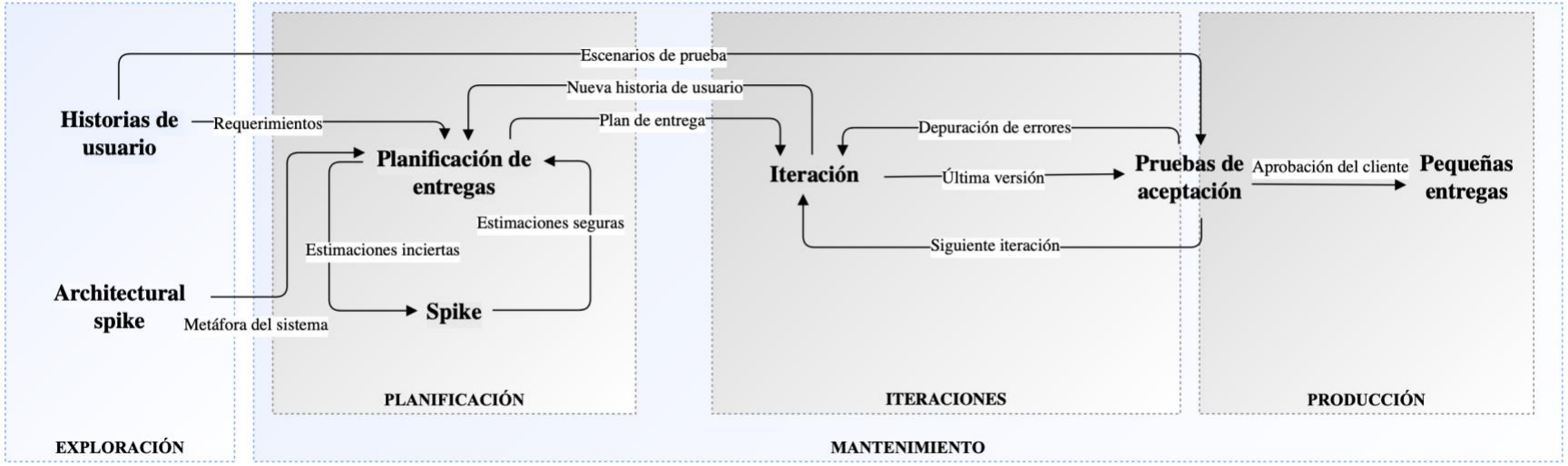
Bahit (2012) menciona que XP se apoya en cinco valores que enfatizan la esencia colaborativa que caracteriza a la metodología, estos valores son:

- *Comunicación:* mantiene una constante comunicación entre el cliente y el equipo de desarrollo a lo largo de todo el proyecto.
- *Simplicidad:* se enfoca en el desarrollo de únicamente lo necesario sin perder tiempo en detalles que no sean requeridos.
- *Retroalimentación:* sostiene una permanente retroalimentación con el cliente a fin de conocer los cambios o nuevos requerimientos e implementarlos.
- *Respeto:* tanto el equipo de desarrolladores como el cliente respeta la idoneidad de cada uno.
- *Coraje:* el equipo debe mostrar la realidad sobre el avance del proyecto sin buscar excusas.

El ciclo de vida de XP, como se presenta en la Figura 13, gira en torno a 4 fases.

Figura 13

Ciclo de vida de XP.



Nota. Adaptada de Ambler, 2002.

Ambler (2002) como se muestra en la Figura 13 divide al ciclo de vida de XP en dos grandes fases Exploración y Mantenimiento, esta última a su vez se abarca tres fases, las cuales son:

- *Fase de exploración:* incluye el desarrollo inicial de las historias de usuario, así como la definición del alcance del proyecto a partir de los requerimientos del cliente.
- *Fase de mantenimiento:* abarca tanto las fases de planificación, iteraciones y producción como las actividades enfocadas a la operación y soporte del sistema.
    - *Fase de planificación:* su propósito radica en definir el tiempo necesario y las fechas de entrega para la implementación de las historias de usuario.
    - *Fase de iteraciones:* constituye el mayor esfuerzo de desarrollo dado que contempla el modelamiento, programación, testeo e integración de cada una de las historias de usuario definidas.
    - *Fase de puesta en producción:* se enfoca en certificar que el sistema está listo para desplegarse en un ambiente de producción, durante esta fase se realiza la documentación del sistema, documentación de operaciones, documentación de soporte y documentación para el usuario.

2.9 JUSTIFICACIÓN DE LAS HERRAMIENTAS Y METODOLOGÍAS SELECCIONADAS

Para el desarrollo del presente trabajo de disertación se seleccionó la metodología cualitativa debido a que ésta es utilizada en investigaciones enfocadas a evaluar ciertas características de una población o situaciones en específico. Su principal objetivo es la observación sistémica del objeto de estudio a fin de catalogar la información que se observa para que posteriormente pueda ser utilizada y replicada por terceros (Yanez, 2019).

Por su parte para el desarrollo del sistema web se optó por la metodología XP, misma que resalta por su simplicidad, comunicación constante con todos los involucrados en el proyecto y sobretodo la retroalimentación permanente en cada fase, XP tiene como objetivos satisfacer al cliente, potenciar el trabajo en equipo y minimizar el riesgo en cuanto a costos, tiempo, calidad y alcance.

Asimismo, en cuanto a las herramientas para la implementación del sistema web se decidió utilizar las siguientes:

- El motor de bases de datos relacionales PostgreSQL debido a que es Open Source, ofrece un gran rendimiento y entorno de alta disponibilidad, soporta alta concurrencia, permite mantener la integridad referencial y la seguridad de los datos.
- La librería de React.js por ser una librería de código abierto, brindar un mayor rendimiento a la aplicación y garantizar la estabilidad del código.

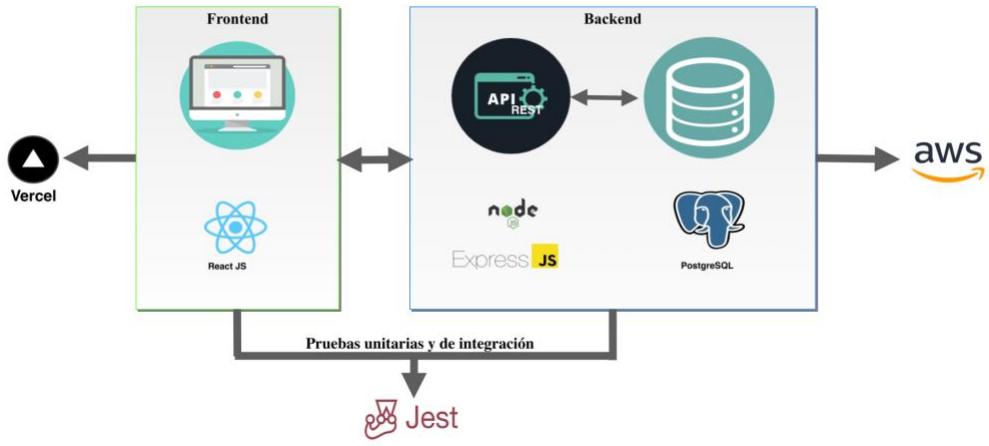
- El framework de desarrollo Backend Express.js por su flexibilidad para el uso de una amplia variedad de bibliotecas, adaptabilidad y versatilidad para todo tipo de aplicaciones del lado del servidor.
- El entorno de ejecución Node.js. puesto que facilita la creación de aplicaciones escalables, cuenta con un alto nivel de rendimiento y es compatible con cientos de paquetes para el desarrollo de proyectos de manera ágil.

En cuanto a la realización de pruebas del sistema, se eligió el framework Jest dado que brinda una retroalimentación inmediata, cuenta con una gran compatibilidad con React.js y permite generar informes sin necesidad de configuraciones adicionales.

En relación con el despliegue del sistema web, se consideró a AWS por su flexibilidad, seguridad, escalabilidad y alto desempeño al momento de hospedar una aplicación y Vercel por su ágil integración con gestores de versiones, despliegue instantáneo y gran rendimiento para el usuario final.

A continuación, en la Figura 14 se presenta la estructura general del sistema web junto con las herramientas previamente mencionadas.

Figura 14.
Distribución de las herramientas seleccionadas.



3. DESARROLLO DEL SISTEMA WEB PARA LA AUTOMATIZACIÓN DE PROCESOS ASISTENCIALES

3.1 FASE DE EXPLORACIÓN

Fase en la que se define lo que el cliente necesita mediante la redacción de historias de usuario, puede durar un par de semanas, depende del tamaño del proyecto y del conocimiento que tengan los desarrolladores en el tema central de la aplicación y en las herramientas a utilizarse. El resultado de esta fase es una visión general del sistema y el tiempo total estimado para el desarrollo del sistema.

3.1.1 HISTORIAS DE USUARIO

Las historias de usuario describen las funcionalidades que se requieren incorporar en la aplicación, no existe un formato obligatorio para la redacción de las historias de usuario, sin embargo, el más conocido es en forma de prosa (Amaya, 2021), como se muestra a continuación en la Figura

15.

Figura 15

Formato de historias de usuario.



Nota. Adaptada de Amaya, 2021.

De acuerdo con Amaya (2021) una historia de usuario debe contar con las propiedades INVEST:

- **Independent (Independiente)**
- **Negotiable (Negociable):** no todas las historias de usuario se hacen a la vez, se van priorizando.
- **Valuable (Valiosa):** deben tener asociado el valor esperado.
- **Estimable:** debe ser posible evaluar su complejidad.
- **Small (Pequeña):** debe describir lo fundamental de cada funcionalidad.
- **Testable (Comprobable):** deriva a los criterios de aceptación, los cuales especifican los escenarios de prueba para comprobar que se está construyendo el producto esperado.

Para el desarrollo del sistema web para la automatización de procesos asistenciales, se establecieron las siguientes historias de usuario:

Figura 16

Historia de usuario No. 1

Historia de usuario 1
Como cliente quiero que el personal del centro médico pueda ingresar al sistema, con accesos restringidos para poder agendar citas, crear y acceder a pacientes, historias clínicas y evoluciones.
Criterios de aceptación
1. En caso de que se requiera ingresar al sistema cuando un usuario inicie sesión, el sistema deberá permitir acceder únicamente a las funcionalidades asignada para cada rol.

Figura 17

Historia de usuario No. 2

Historia de usuario 2
Como médico - recepcionista quiero ingresar datos de los pacientes, y realizar consultas para poder administrar a los pacientes que acuden al centro médico.
Criterios de aceptación
1. En caso de que se requiera ingresar los datos de un nuevo paciente, el sistema deberá presentar un formulario para el ingreso de pacientes que incluya los siguientes datos de filiación: Nombres, Apellidos, Cédula, Fecha de nacimiento, Estado civil, Lugar de nacimiento, Correo electrónico, Dirección, Teléfono, Etnia, Tipo de sangre, Nivel de instrucción, Contacto de emergencia 2. En caso de que se requiera consultar pacientes, el sistema deberá permitir la búsqueda por apellido y número de cédula.

Figura 18

Historia de usuario No. 3

Historia de usuario 3
Como médico - recepcionista quiero agendar y consultar citas para poder administrar correctamente las citas médicas, siguiendo un horario establecido.
Criterios de aceptación
1. En caso de que se requiera agendar la cita de un paciente, bajo la condición de que se haya registrado previamente a dicho paciente, el sistema deberá presentar un formulario que incluya fecha, hora y el motivo de la cita. 2. En caso de que se requiera consultar citas, el sistema deberá permitir la búsqueda por fecha.

Figura 19

Historia de usuario No. 4

Historia de usuario 4
Como médico quiero ingresar y consultar historias clínicas para poder gestionar adecuadamente la historia clínica de cada uno de los pacientes.
Criterios de aceptación
1. En caso de que se requiera ingresar la historia clínica de un paciente, bajo la condición de que se haya registrado previamente a dicho paciente, el sistema deberá presentar un formulario que incluya los siguientes datos: Antecedentes patológicos personales, Antecedentes quirúrgicos, Alergias, Gestas, Partos, Cesáreas, Abortos, Método anticonceptivo. 2. En caso de que se requiera consultar historias clínicas, el sistema deberá permitir la búsqueda por apellido y número de cédula del paciente.

Figura 20

Historia de usuario No. 5

Historia de usuario 5
Como médico quiero ingresar y consultar evoluciones para poder gestionar adecuadamente cada una de las consultas o evoluciones.
Criterios de aceptación
1. En caso de que se requiera ingresar la evolución de un paciente, bajo la condición de que se haya creado previamente la historia clínica de dicho paciente, el sistema deberá presentar un formulario que incluya los siguientes datos: Fecha, Motivo de consulta, Fecha de última menstruación, Procedimiento, Diagnóstico, Medicación, Indicación, Fotos, Próximo control 2. En caso de que se requiera consultar evoluciones, el sistema deberá permitir la búsqueda por fecha.

Figura 21

Historia de usuario No. 6

Historia de usuario 6
Como administrador quiero crear usuarios con diferentes tipos de roles para los funcionarios del centro médico para controlar el acceso al sistema.
Criterios de aceptación
1. En caso de que se requiera crear un nuevo usuario, el sistema deberá permitir registrar usuarios con diferentes tipos de roles.

Figura 22

Historia de usuario No. 7

Historia de usuario 7
Como administrador quiero ingresar, modificar y eliminar médicos para administrar a los médicos que laboran.
Criterios de aceptación
1. En caso de que se requiera crear un nuevo médico, el sistema deberá permitir registrar médicos con sus respectivas especialidades.

Figura 23

Historia de usuario No. 8

Historia de usuario 8
Como administrador quiero ingresar datos de los consultorios para administrar la información asociada a cada consultorio.
Criterios de aceptación
1. En caso de que se requiera ingresar los datos de un consultorio, el sistema deberá presentar un formulario para el registro del consultorio.

3.1.2 FORMALIZACIÓN DE REQUERIMIENTOS

3.1.2.1 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

De acuerdo con Berzal (s.f.) los requerimientos o requisitos especifican los servicios que ofrecerá el sistema, así como las restricciones asociadas a su funcionamiento, éstos se clasifican en:

- *Requerimientos funcionales*: define qué debe hacer un sistema, es decir, su funcionamiento e interacción con su entorno.
- *Requerimientos no funcionales*: describen cómo debe ser el sistema, en concreto, definen las cualidades del sistema.

A continuación, en la Tabla 2, Tabla 3 y Tabla 4 se establecen los requerimientos funcionales, no funcionales y requerimientos mínimos del sistema web.

Tabla 2

Requerimientos funcionales

Requerimientos funcionales	Requerimientos de diseño
Módulo de autenticación y autorización	Ingreso al sistema
Módulo de gestión de pacientes.	Administración de pacientes.
Módulo de gestión de citas.	Agendamiento de citas.
Módulo de gestión de historias clínicas.	Administración de historias clínicas.
Módulo de gestión de evoluciones.	Administración de evoluciones.
Módulo de gestión de usuarios.	Administración de usuarios.
Módulo de gestión de médicos.	Administración de médicos.
Módulo de gestión de consultorios.	Administración de consultorios.

Tabla 3

Requerimientos no funcionales

Requerimientos no funcionales	Detalle
Desempeño	• El sistema debe responder adecuadamente a todas las solicitudes que recibe.
Pertinencia	• El sistema debe validar la integridad de los datos ingresados por el usuario. • El sistema debe presentar información confiable.
Seguridad	• El sistema debe asegurar que los datos estén protegidos ante accesos no autorizados. • El sistema debe mantener la integridad de los datos.
Usabilidad	• El sistema debe presentar mensajes de error e informativos al usuario. • El sistema debe presentar interfaces intuitivas.

Tabla 4

Requerimientos mínimos

Requerimientos mínimos	Detalle
Hardware	No aplica.
Software	Navegador web con soporte para JavaScript.

3.1.2.2 CASOS DE USO

De las Pueblas Encinas (2019) sostiene que los diagramas de casos de uso proporcionan una visión general del sistema, son utilizados durante el proceso de análisis y diseño del sistema dado que esquematizan los requisitos funcionales y brindan una "clara fotografía de los principales actores involucrados en el sistema y de los principales procesos existentes en él" (p. 41).

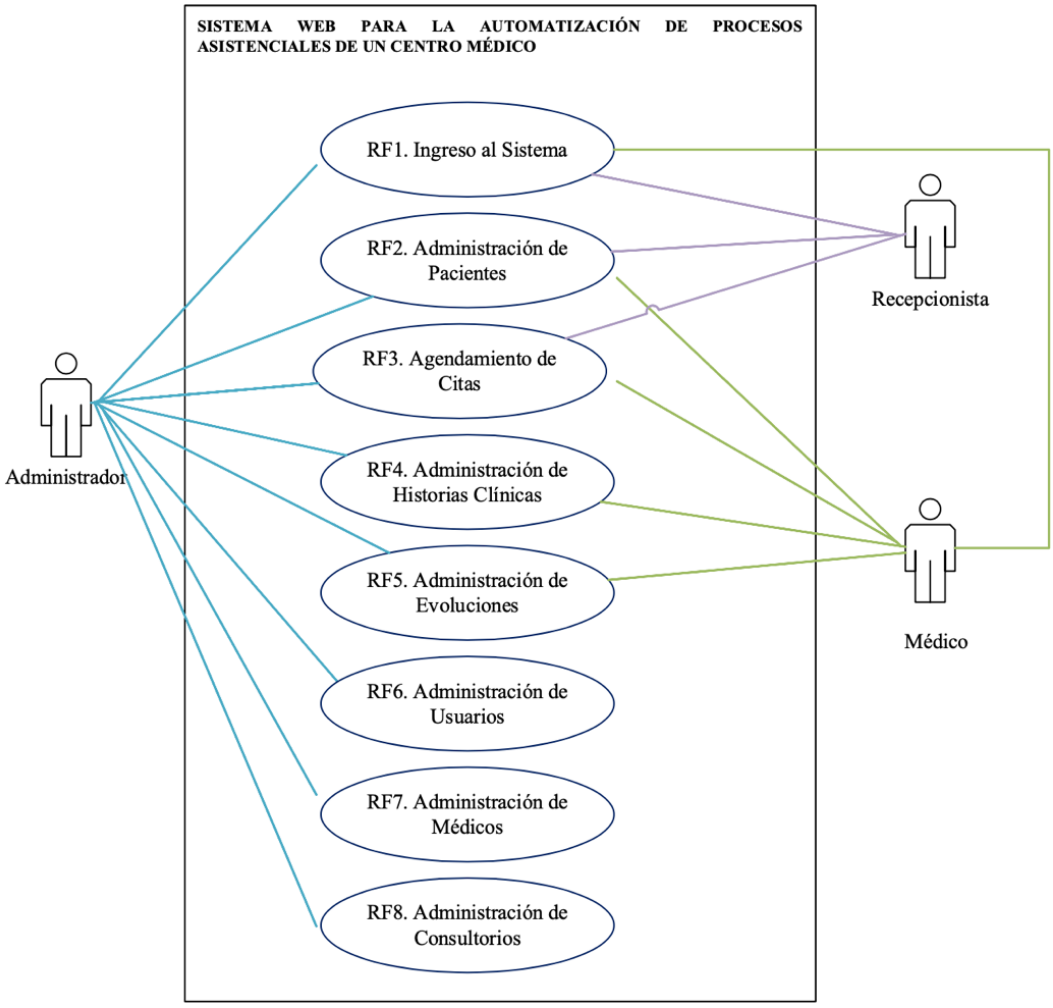
De acuerdo con los requerimientos funcionales descritos en el Tabla 2, los diagramas de casos de uso del sistema web son:

a. NIVEL GENERAL

Caso de uso a nivel general describe todas las funcionalidades del sistema y presenta los actores que intervendrán en él.

Figura 24

Nivel general.



b. SIGUIENTE NIVEL

Presenta a detalle cada una de las funcionalidades expuestas en el nivel general y especifica las interacciones de los usuario en función su rol.

Figura 25

RF1. Ingreso al sistema

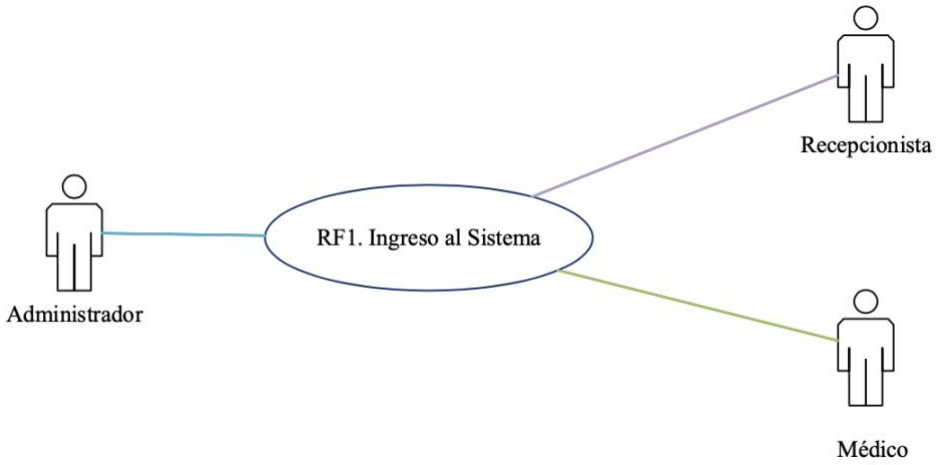


Figura 26

RF2. Administración de Pacientes.

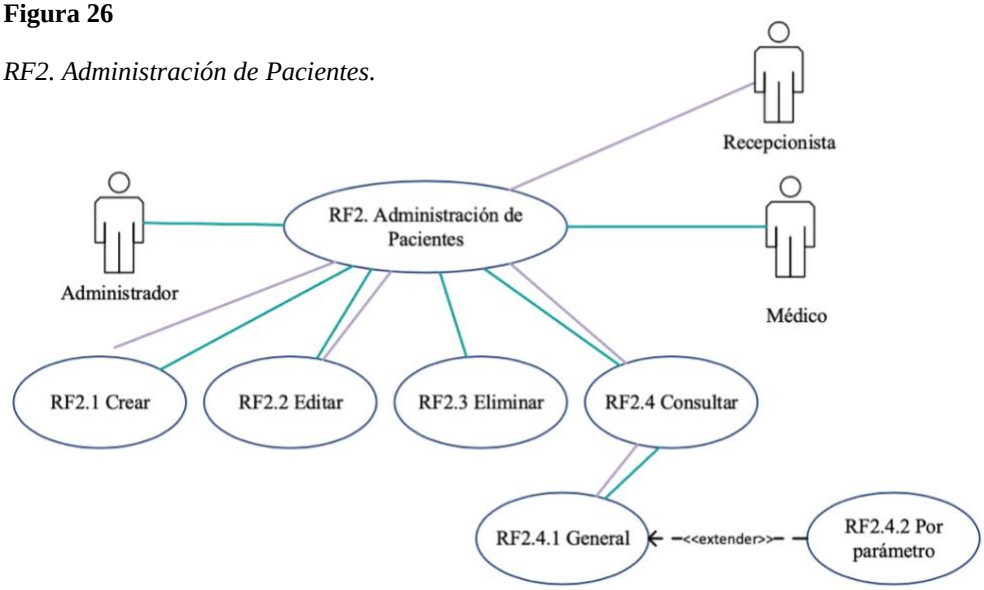


Figura 27

RF3. Agendamiento de Citas.

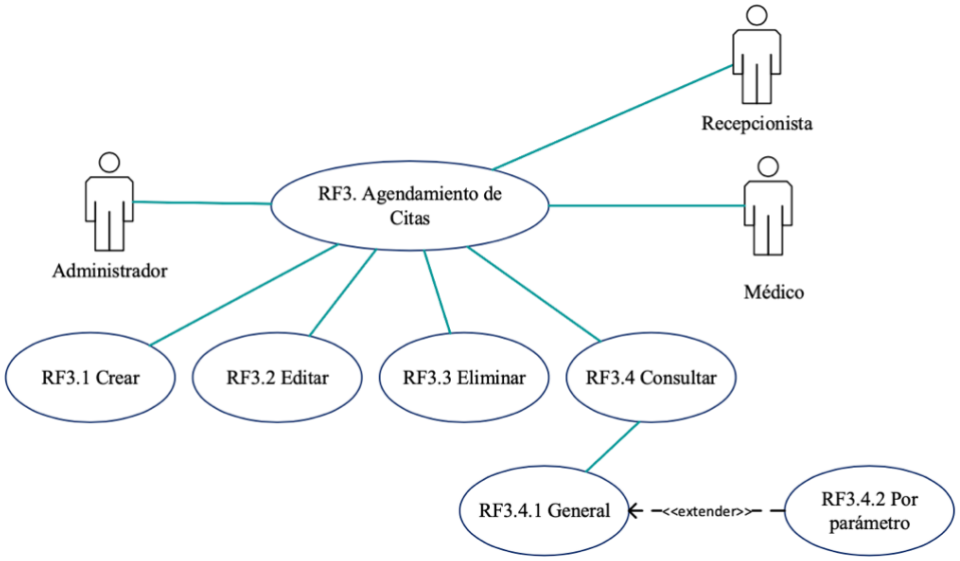


Figura 28

RF4. Administración de Historias Clínicas.

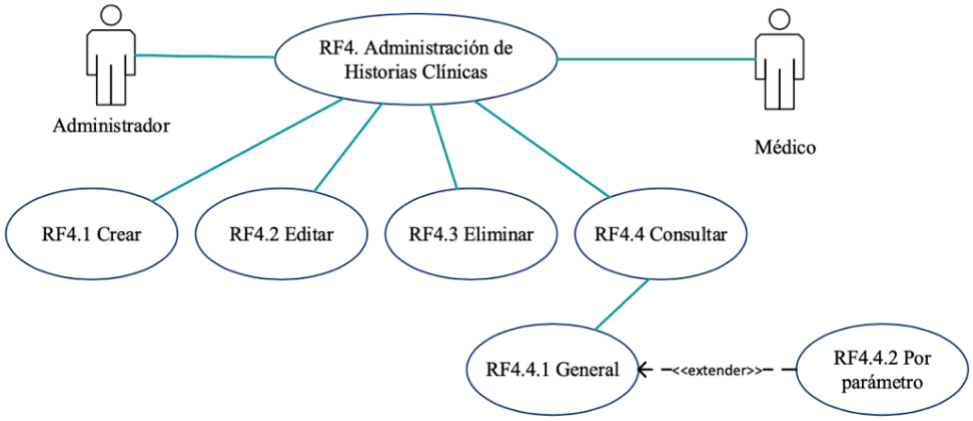


Figura 29

RF5. Administración de Evoluciones.

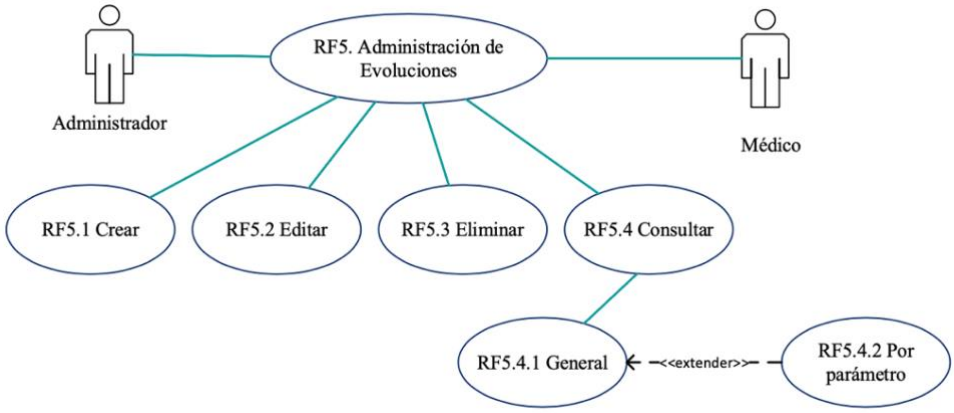


Figura 30

RF6. Administración de Usuarios.

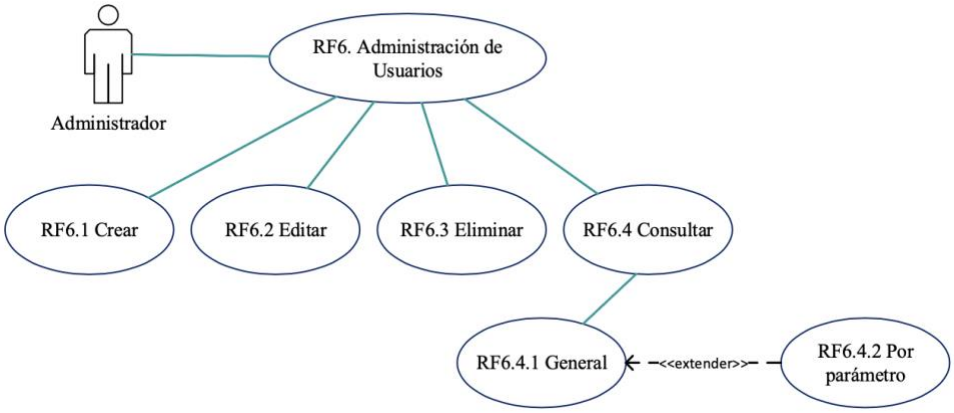


Figura 31

RF7. Administración de Médicos.

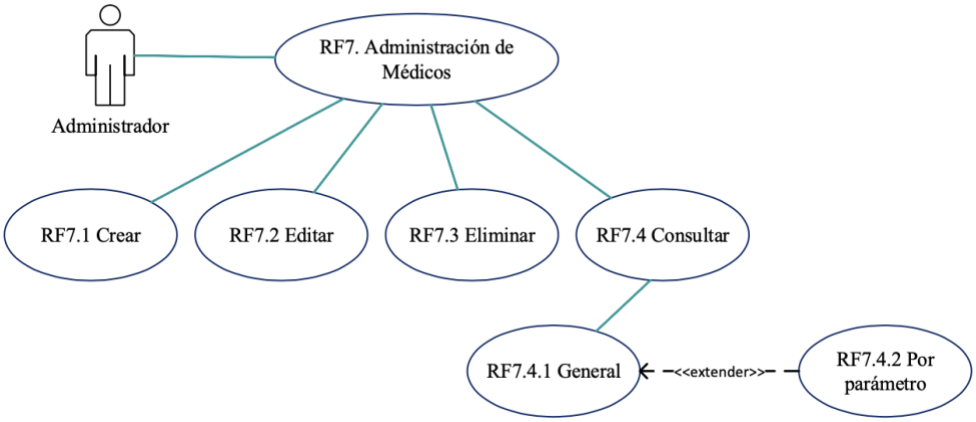


Figura 32

RF8. Administración de Consultorios.



3.2 FASE DE MANTENIMIENTO

Incluye las fases de planificación, iteraciones y producción, constituye el estado normal de los proyectos puesto que éstos pueden evolucionar con el tiempo (Ambler, 2002).

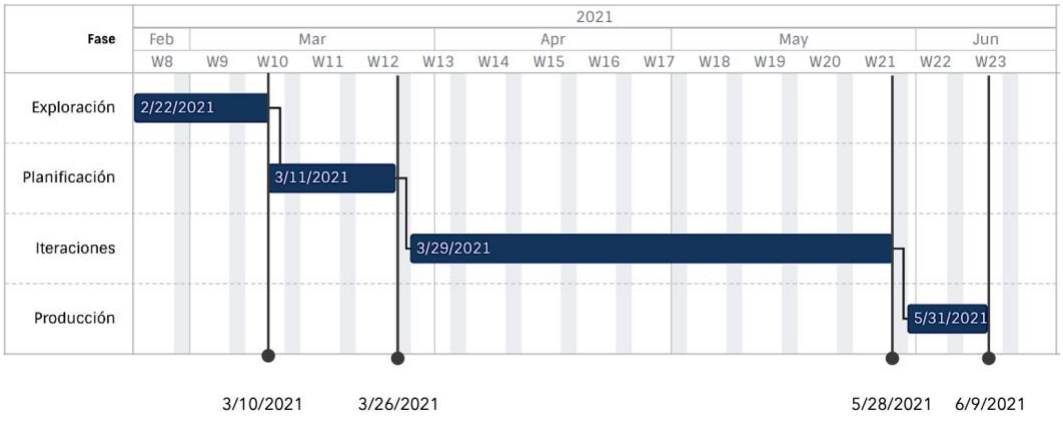
3.2.1 FASE DE PLANIFICACIÓN

Fase corta donde todos los involucrados en el proyecto definen el orden en que se implementarán los requerimientos, así como las fechas de entrega (Joskowicz, 2008).

A continuación, en la Figura 31, se presenta la planificación inicial y general del proyecto.

Figura 33

Planificación inicial del proyecto.



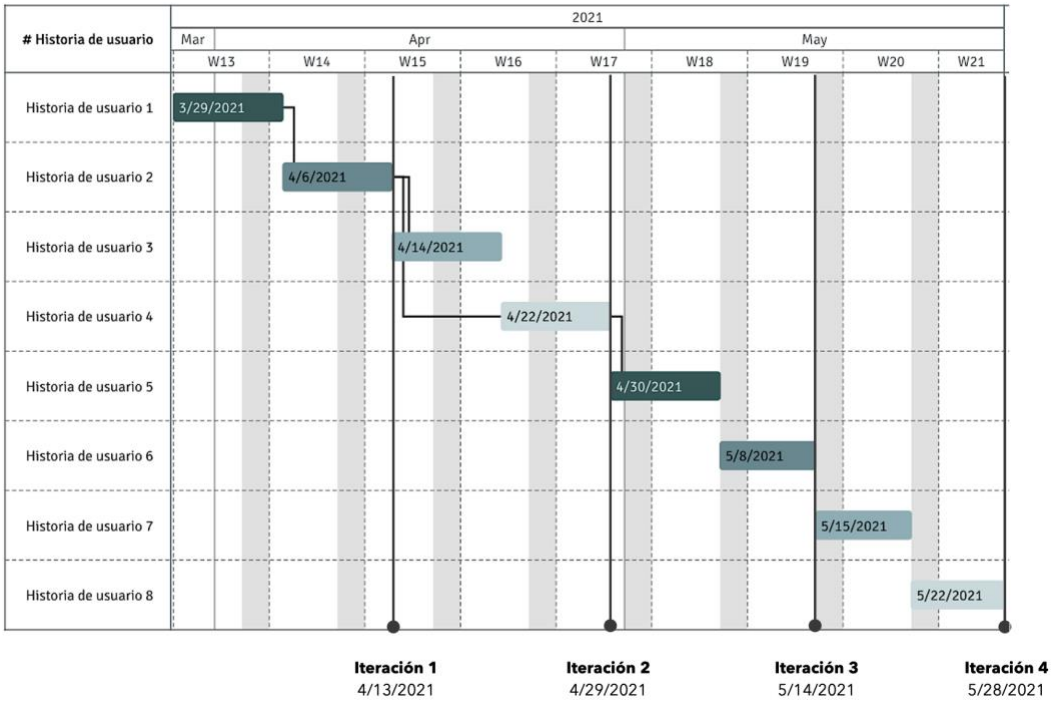
3.2.1.1 DEFINICIÓN DE ENTREGABLES

En las metodologías ágiles el desarrollo del sistema es realizado en el transcurso de iteraciones sucesivas de acuerdo con las historias de usuario planteadas en la fase de Exploración (Encalada, 2018), por

consiguiente, en la Figura 32 se muestra la planificación para la fase de Iteraciones.

Figura 34

Definición de entregables.



3.2.2 FASE DE ITERACIONES

Fase principal en el ciclo de desarrollo de XP dado que en ésta se desarrollan las funcionalidades descritas en las historias de usuario y se genera un entregable funcional sin defectos, por lo que la detección y corrección de errores, así como aplicación de pruebas unitarias y de aceptación es fundamental.

Cada iteración provee un porcentaje de avance para medir el progreso del proyecto (Joskowicz, 2008).

3.2.2.1 DISEÑO

La metodología XP enfatiza en el desarrollo de diseños simples y claros debido a que un diseño simple se implementa mucho más rápido que uno complejo. Asimismo, sugiere el uso de metáforas como una forma sencilla de explicar el propósito del proyecto e incentiva a la refactorización del código de manera que se mantenga el código lo más simple y entendible posible (Collazo Garcia & Labrador, 2017)

a. ARQUITECTURA

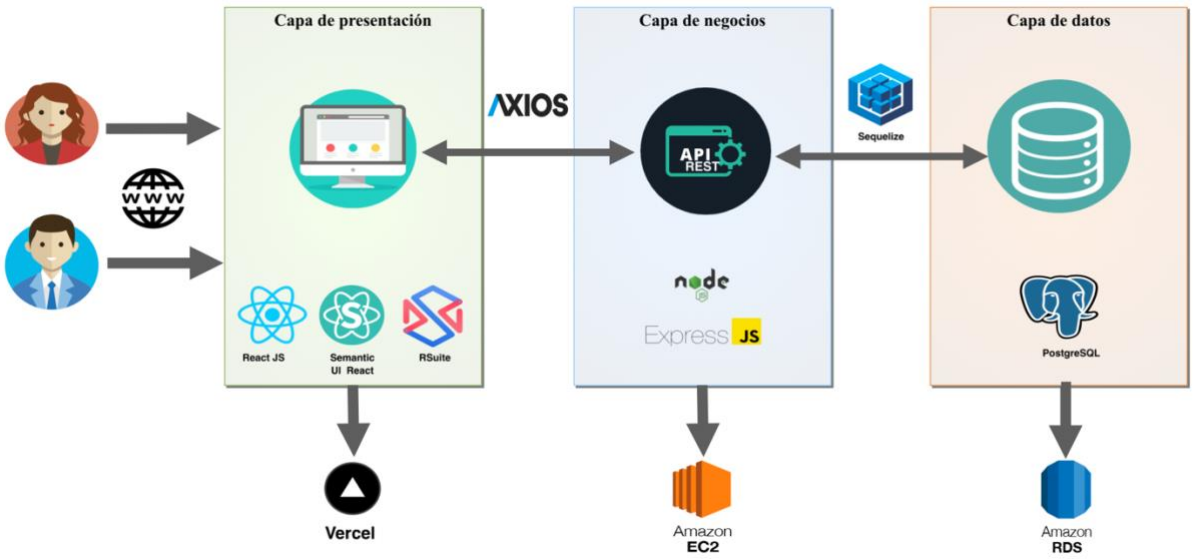
La arquitectura establecida para el desarrollo del sistema web es por capas, la misma que estará compuesta por la capa de presentación (interfaz gráfica), capa de negocios (lógica del negocio) y la capa de datos (acceso a datos).

Para la división de las capas anteriormente mencionadas, se manejarán dos aplicaciones, una destinada para el Backend y otra para el Frontend, la comunicación de ambas aplicaciones será a través de la librería Axios, un cliente HTTP que se encarga de transformar la información de peticiones y respuestas del servidor.

En la Figura 35 se presenta la arquitectura de la aplicación junto con las herramientas utilizadas para su implementación.

Figura 35.

Arquitectura del sistema.

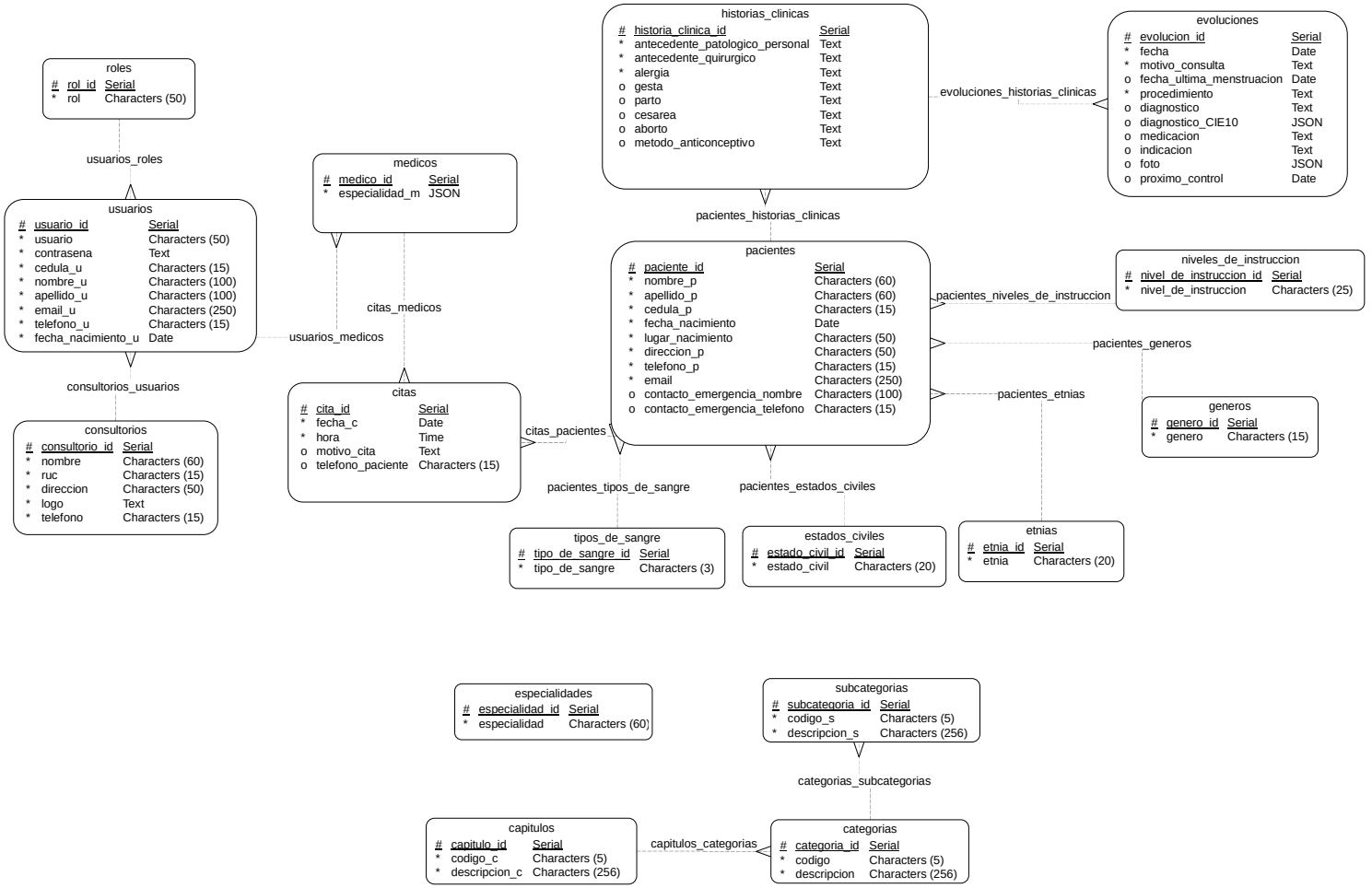


b. BASE DE DATOS

En base a los requerimientos descritos en las historias de usuario y analizando la problemática planteada, el modelo conceptual y físico del sistema son los siguientes:

Figura 36.

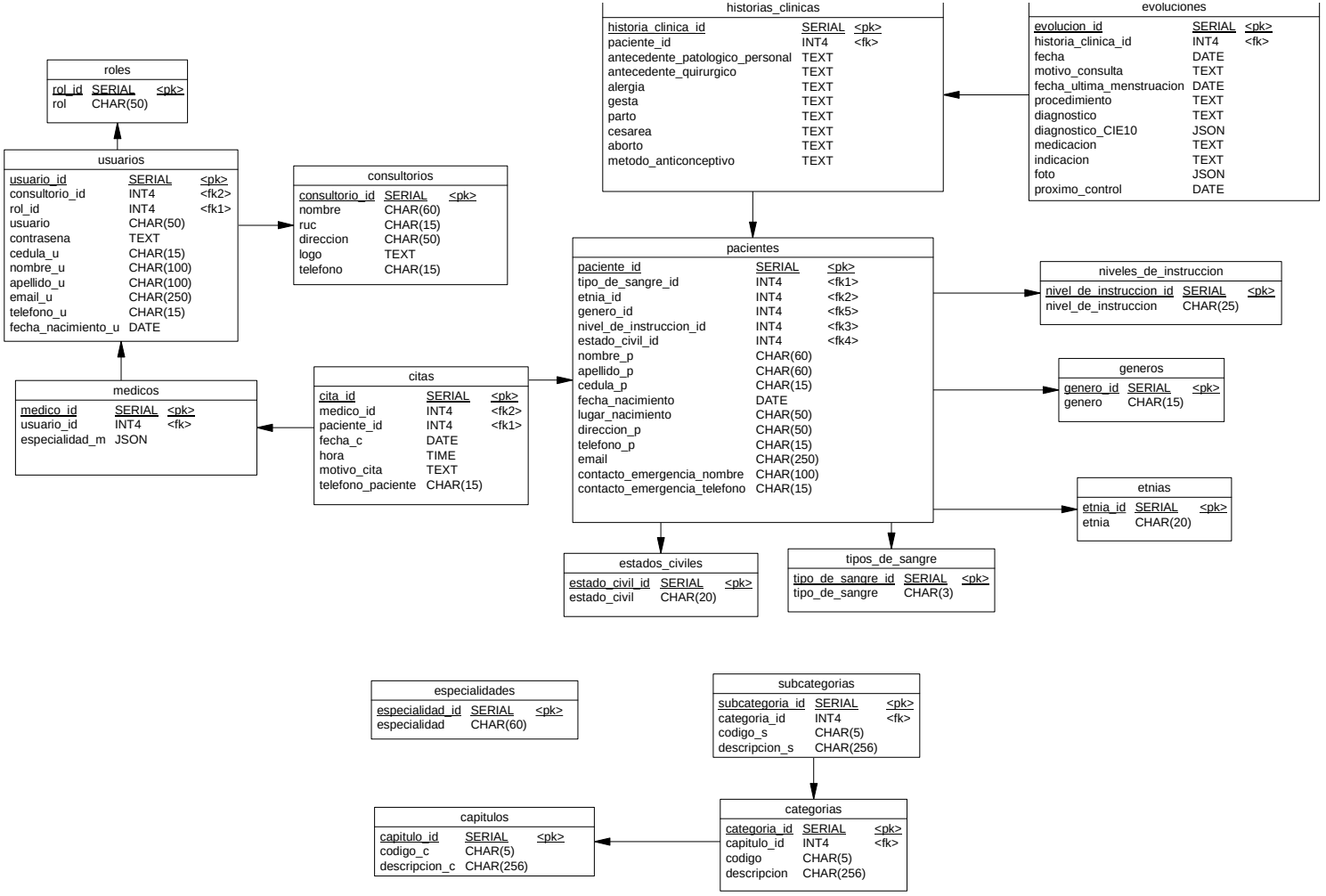
Modelo conceptual.



Nota. El modelo conceptual identifica las entidades sistema y sus relaciones a nivel superior.

Figura 37.

Modelo físico.



Nota. El modelo físico presenta de forma específica las relaciones entre las tablas, incluyendo claves primarias y foráneas.

3.2.2.2 ITERACIÓN 1

En base a la planificación realizada, para la primera iteración se implementaron las historias de usuario 1 y 2, es decir:

- Módulo de autenticación y autorización
- Módulo de gestión de pacientes.

a. Historia de usuario 1

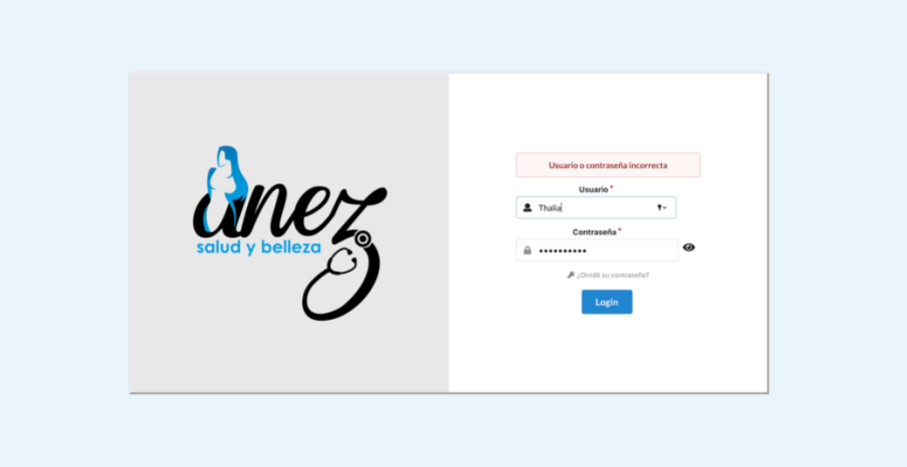
i. Backend

Tabla 5.
Historia de usuario 1, Endpoints.

Método	URI	Descripción
POST	/confirm_user	Autenticación de usuarios
POST	/recuperar_pass	Restablecer contraseña mediante el usuario.

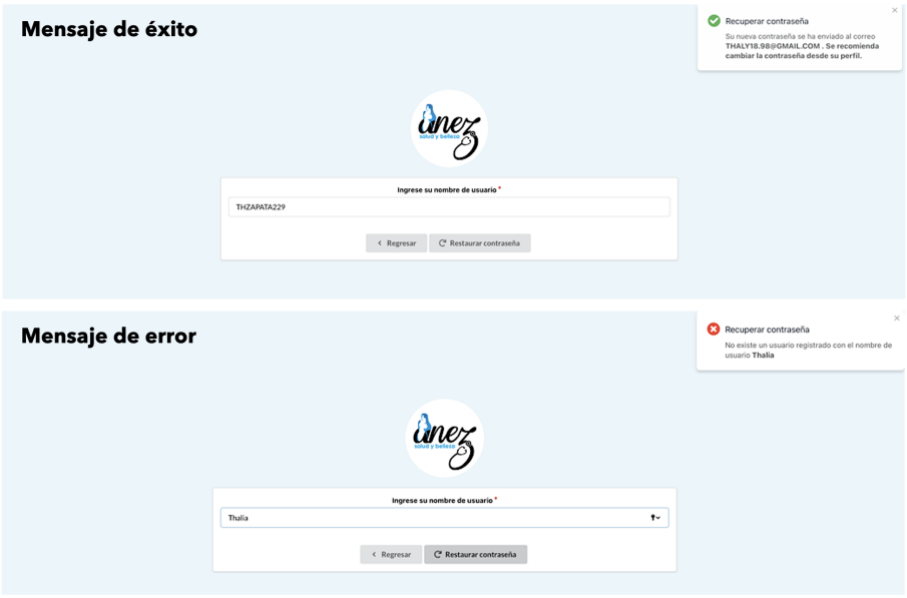
ii. Frontend

Figura 38.
Historia de usuario 1, Iniciar sesión.



Nota. Página iniciar sesión: el usuario debe ingresar su usuario y contraseña, en caso de ser alguno de estos campos incorrecto presentará el mensaje "*Usuario o contraseña incorrecta*".

Figura 39.
Historia de usuario 1, Restablecer contraseña.



Nota. Página restablecer contraseña: el usuario debe ingresar su nombre de usuario, en caso de existir el usuario muestra un mensaje informando el correo electrónico al cual se le envío su nueva contraseña o de lo contrario presenta un mensaje de error.

iii. Pruebas

Figura 40.
Historia de usuario 1, Pruebas unitarias.

Backend			2.047s
Login	/confirm_user	passed	0.074s
Login	/recuperar_pass	passed	0.079s
Frontend			4.216s
Login	Render del componente Login	passed	0.005s
Login	Render del componente Recuperar contraseña	passed	0s
Login	Comprobar el botón de Login	passed	0.064s
Login	Comprobar el botón de recuperar contraseña	passed	0.008s

Nota. Utilizando el framework Jest se realizaron pruebas unitarias y pruebas de intregración a las rutas, los componentes y eventos asociados a los botones para ingresar al sistema y restablecer la contraseña.

b. Historia de usuario 2

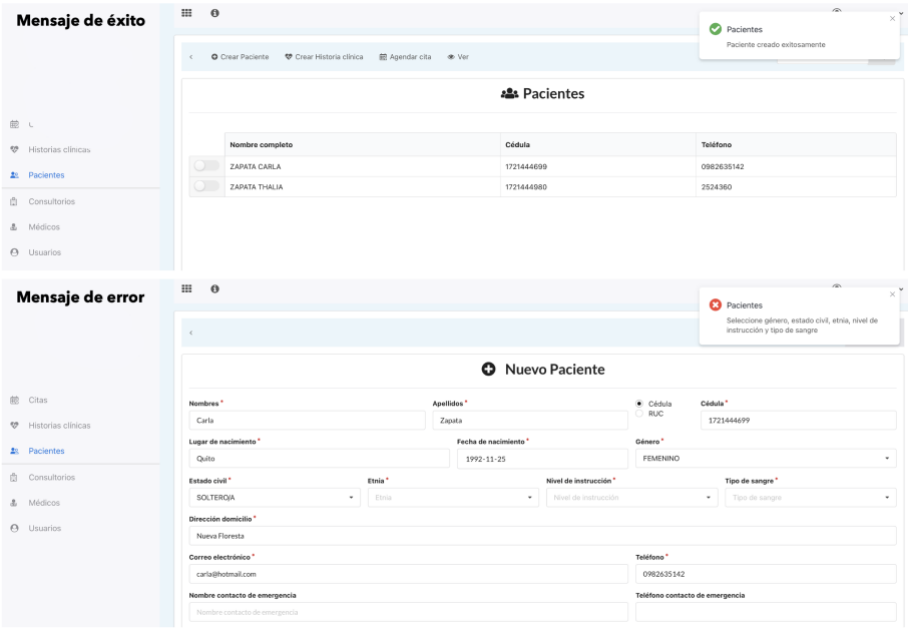
i. Backend

Tabla 6.
Historia de usuario 2, Endpoints.

Método	URI	Descripción
POST	/paciente	Crear paciente.
GET	/ pacientes	Obtener todos los pacientes.
GET	/paciente:id	Obtener paciente por id.
GET	/pacientes_buscar/:value	Obtener pacientes por el apellido, nombre o cédula.
PUT	/paciente/:id	Editar un paciente.
DELETE	/paciente/:id	Eliminar paciente.

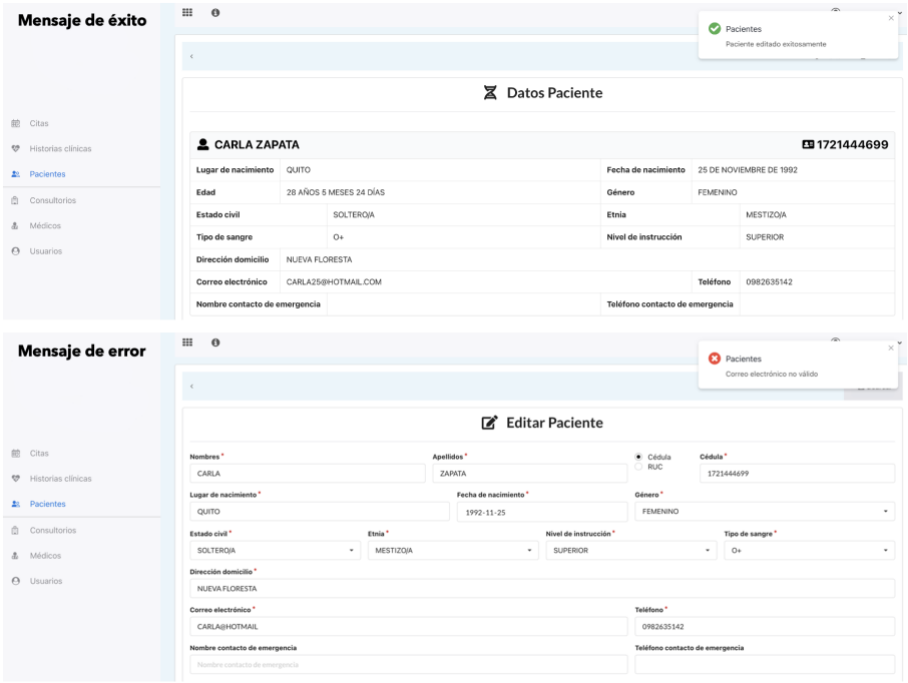
ii. Frontend

Figura 41.
Historia de usuario 2, Crear paciente.



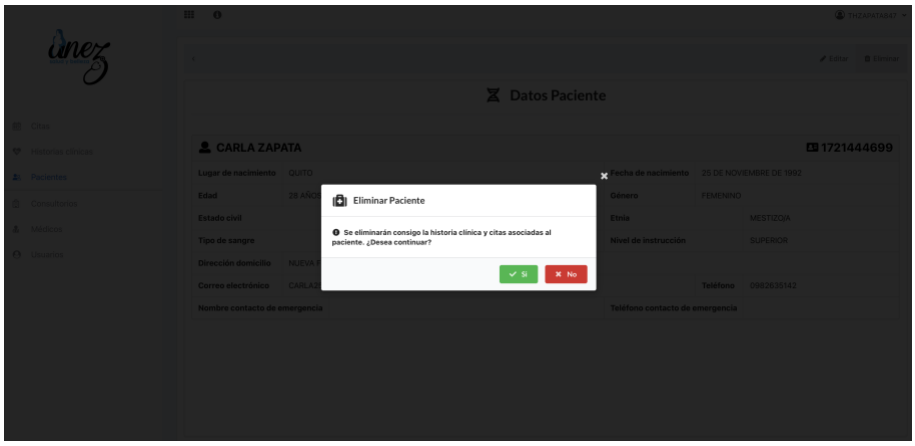
Nota. Página crear paciente: permite al usuario ingresar y/o seleccionar todos los campos requeridos, ingresar correctamente el correo electrónico y no ingresar un número cédula ya existente, de lo contrario se mostrará un mensaje de error.

Figura 42.
Historia de usuario 2, Editar paciente.



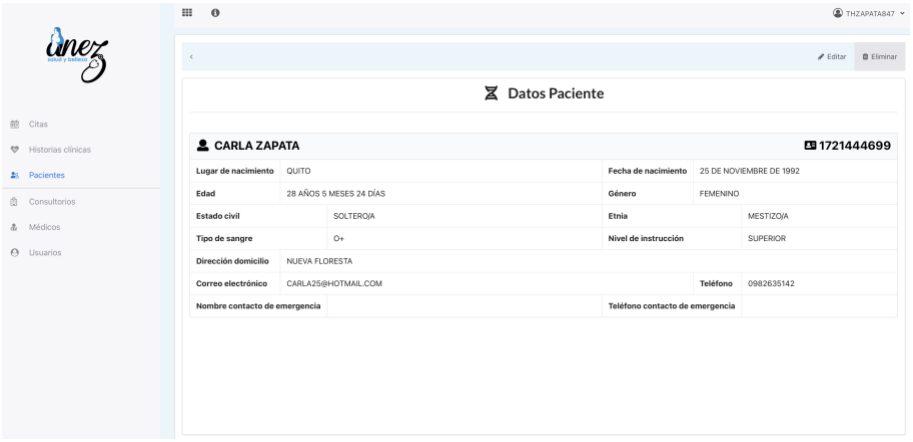
Nota. Página editar pacientes: para una editado exitoso se requiere cumplir con las mismas validaciones señaladas en el ingreso de pacientes.

Figura *43*.
Historia de usuario 2, Eliminar paciente.



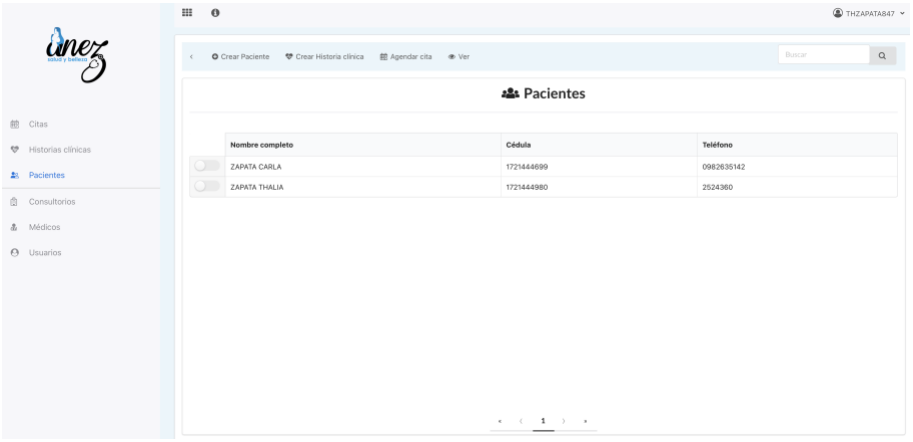
Nota. Modal eliminar paciente: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura *45*.
Historia de usuario 2, Detalle de paciente por id.



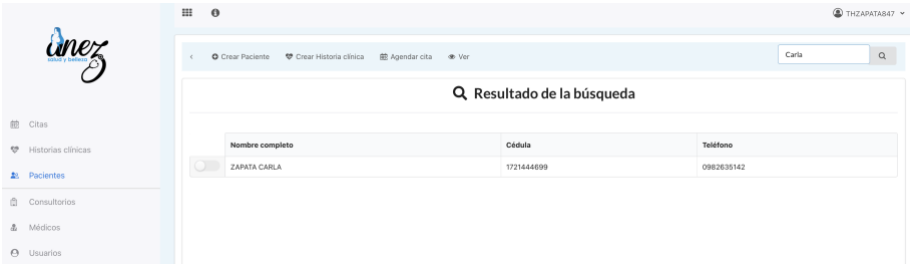
Nota. Página detalle de paciente: muestra los datos del paciente y las opciones para editar o eliminar.

Figura 44.
Historia de usuario 2, Listado de pacientes.



Nota. Página listado de pacientes: presenta todos los pacientes registrados.

Figura 46.
Historia de usuario 2, Buscar pacientes.



Nota. Página buscar pacientes: el usuario puede ingresar el nombre, apellido o número de cédula del paciente.

iii. Pruebas

Figura *47*.
Historia de usuario 2, Pruebas unitarias.

Backend			3.317s
Pacientes	/paciente	passed	0.038s
Pacientes	/pacientes	passed	0.014s
Pacientes	/paciente/:id	passed	0.016s
Pacientes	/pacientes_buscar/:value	passed	0.009s
Pacientes	/paciente/:id	passed	0.008s
Pacientes	/paciente/:id	passed	0.016s
Frontend			3.971s
Pacientes	Render del componente Listado de pacientes - Buscar	passed	0.005s
Pacientes	Render del componente Detalle de paciente por id	passed	0s
Pacientes	Render del componente Agregar - Editar	passed	0.001s
Pacientes	Comprobar el botón de crear nuevo paciente	passed	0.115s
Pacientes	Comprobar el botón de editar paciente	passed	0.042s
Pacientes	Comprobar el botón de eliminar paciente	passed	0.016s
Pacientes	Comprobar el botón de buscar paciente	passed	0.046s

Nota. Pruebas unitarias y pruebas de intregración de los endpoints requeridos para la gestión de pacientes así como de los componentes y botones para la ejecución de cada una de las opciones presentadas.

3.2.2.3 ITERACIÓN 2

En esta iteración se implementaron las historias de usuario 3 y 4, mismas que corresponden a:

- Módulo de gestión de citas.
- Módulo de gestión de historias clínicas.

a. Historia de usuario 3

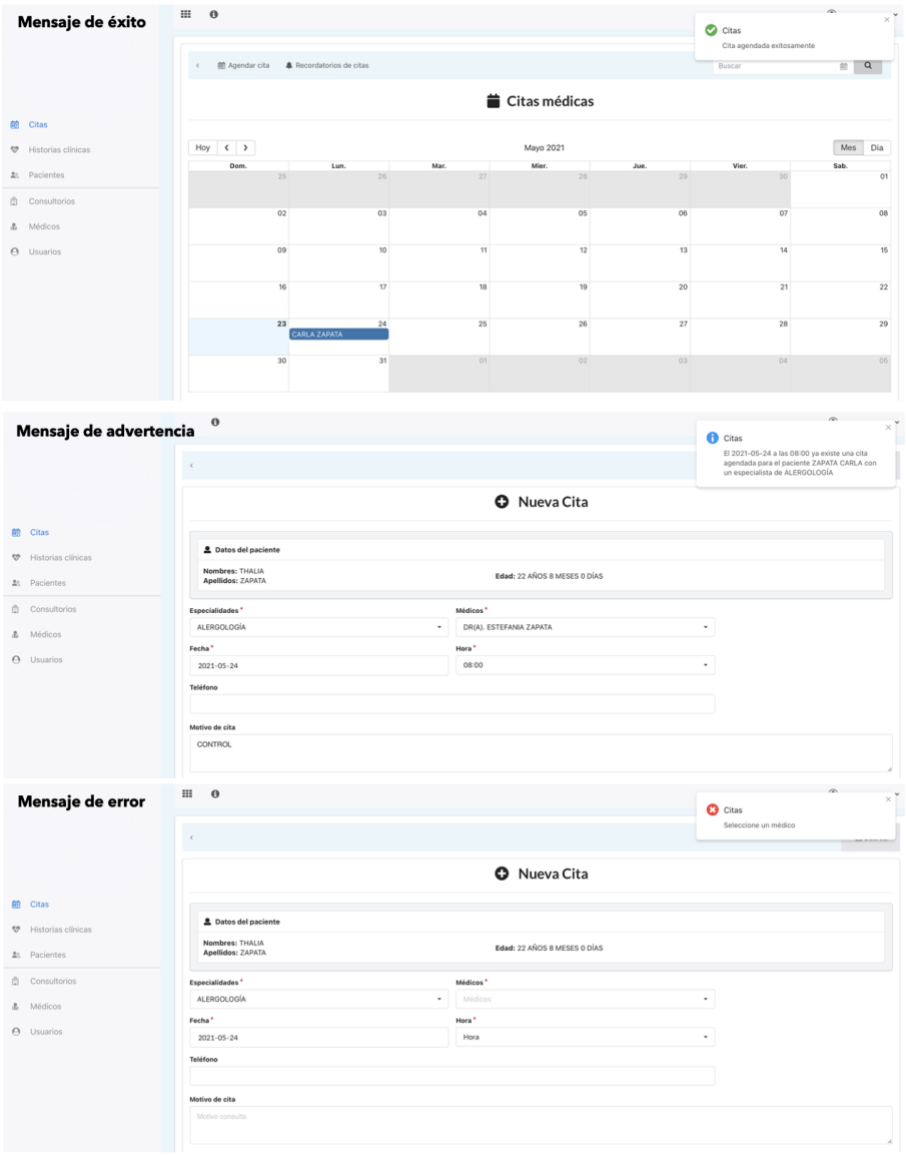
i. Backend

Tabla 7.
Historia de usuario 3, Endpoints.

Método	URI	Descripción
POST	/cita	Crear cita.
GET	/citas	Obtener todas las citas.
GET	/cita/:id	Obtener cita por id.
GET	/citas_fecha/:fecha1 /:fecha2	Obtener citas dentro de un rango de tiempo.
GET	/citas_fecha_med /:fecha1/:fecha2/:id	Obtener citas dentro de un rango de tiempo de acuerdo con el id del médico.
PUT	/cita/:id	Editar una cita.
DELETE	/cita/:id	Eliminar cita.

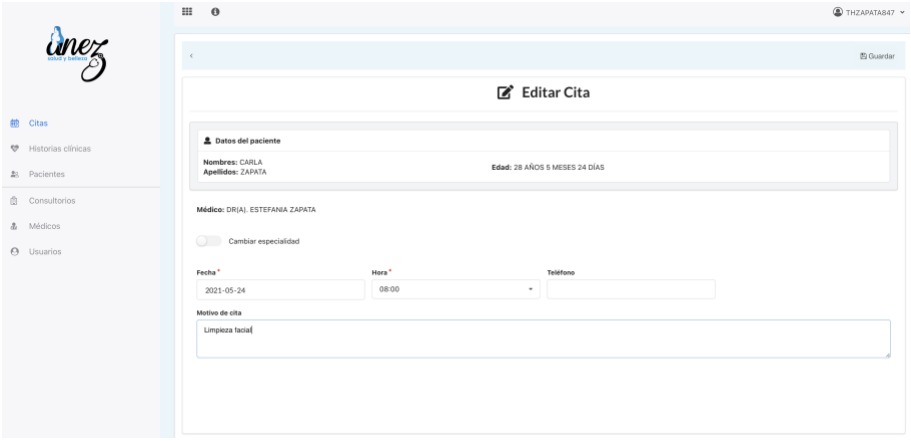
ii. Frontend

Figura *48*.
Historia de usuario 3, Crear cita médica.



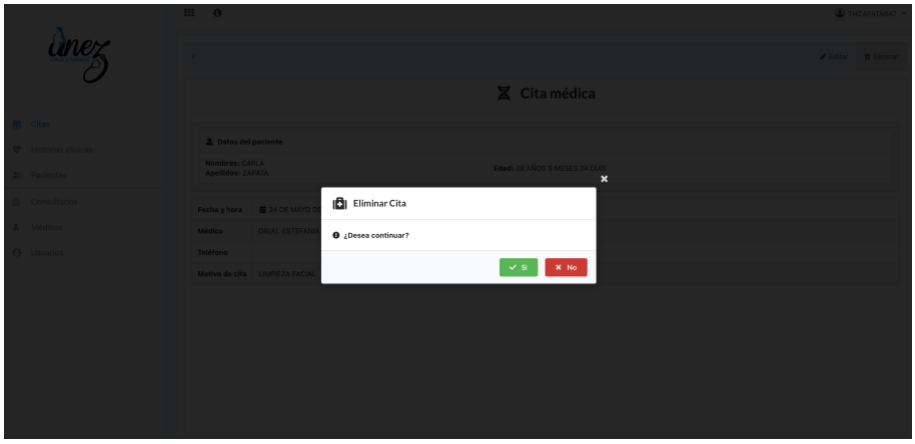
Nota. Página crear cita: se requiere que el usuario seleccione la especialidad y médico asociada a ésta, en caso de no escoger un médico o fecha u hora de la cita mostrará un mensaje de error, asimismo en caso de existir otra cita a la misma fecha u hora se presentará un mensaje de advertencia.

Figura 49.
Historia de usuario 3, Editar cita médica.



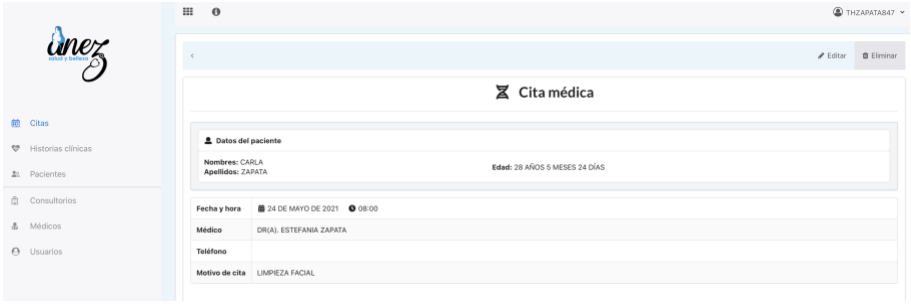
Nota. Página editar cita: presenta la opción de cambiar de especialidad y/o médico, tiene las mismas validaciones que la página de crear citas.

Figura 50.
Historia de usuario 3, Eliminar cita médica.



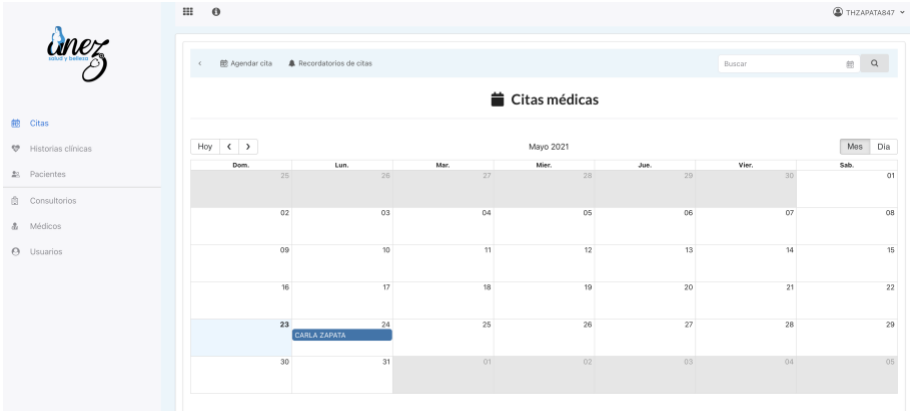
Nota. Modal eliminar cita: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 51.
Historia de usuario 3, Detalle de cita por id.



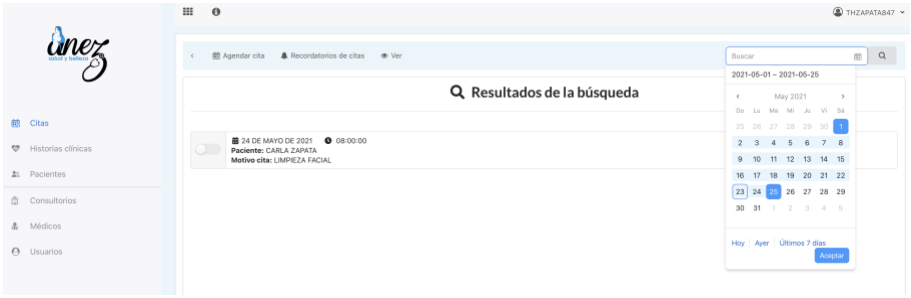
Nota. Página detalle de cita: presenta los datos del paciente así como los datos de la cita y las opciones para editar o eliminar.

Figura 52.
Historia de usuario 3, Calendario de citas.



Nota. Página calendario de citas: muestra todas las citas por mes.

Figura 53.
Historia de usuario 3, Buscar citas.



Nota. Página buscar citas: el usuario debe seleccionar un rango de tiempo para realizar la búsqueda.

Figura 54.
Historia de usuario 3, Enviar recordatorios de citas.



Nota. Página recordatorios de citas: el usuario puede seleccionar una o varias citas para enviar por correo electrónico los recordatorios a los pacientes, una vez enviados los correos se presentará un mensaje de éxito.

iii. Pruebas

Figura 55.
Historia de usuario 3, Pruebas unitarias.

Backend			3.019s
Citas	/cita	passed	0.044s
Citas	/notificaciones	passed	0.007s
Citas	/citas	passed	0.011s
Citas	/cita/:id	passed	0.009s
Citas	/citas_fecha/:fecha1/:fecha2	passed	0.019s
Citas	/citas_fechas_med/:fecha1/:fecha2/:id	passed	0.005s
Citas	/cita/:id	passed	0.011s
Citas	/cita/:id	passed	0.003s
Frontend			4.074s
Citas	Render del componente Calendario de citas	passed	0.005s
Citas	Render del componente Detalle de cita por id	passed	0s
Citas	Render del componente Agregar	passed	0.001s
Citas	Render del componente Editar	passed	0.001s
Citas	Render del componente Buscar	passed	0s
Citas	Render del componente Recordatorio de citas	passed	0.001s
Citas	Comprobar el botón de agendar cita	passed	0.124s
Citas	Comprobar el botón de editar cita	passed	0.058s
Citas	Comprobar el botón de eliminar cita	passed	0.016s
Citas	Comprobar el botón de enviar recordatorios	passed	0.022s
Citas	Comprobar el botón de buscar cita	passed	0.036s

Nota. Pruebas unitarias y pruebas de intregración de las rutas, componentes y eventos de los botones asociados a la historia de usuario 3.

b. Historia de usuario 4

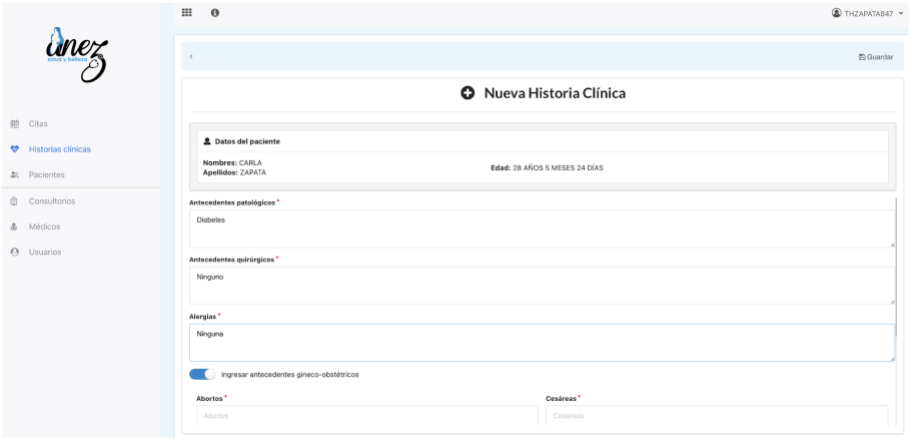
i. Backend

Tabla 8.
Historia de usuario 4, Endpoints.

Método	URI	Descripción
POST	/historia_clinica	Crear historia clínica.
GET	/historias_clinicas	Obtener todas las historias clínicas.
GET	/historia_clinica/:id	Obtener historia clínica por id.
GET	/hc_buscar/:value	Obtener historias clínicas por el apellido, nombre o cédula del paciente
GET	/paciente_historia/:id	Obtener historia clínica por el id del paciente
PUT	/historia_clinica/:id	Editar una historia clínica.
DELETE	/historia_clinica/:id	Eliminar historia clínica.

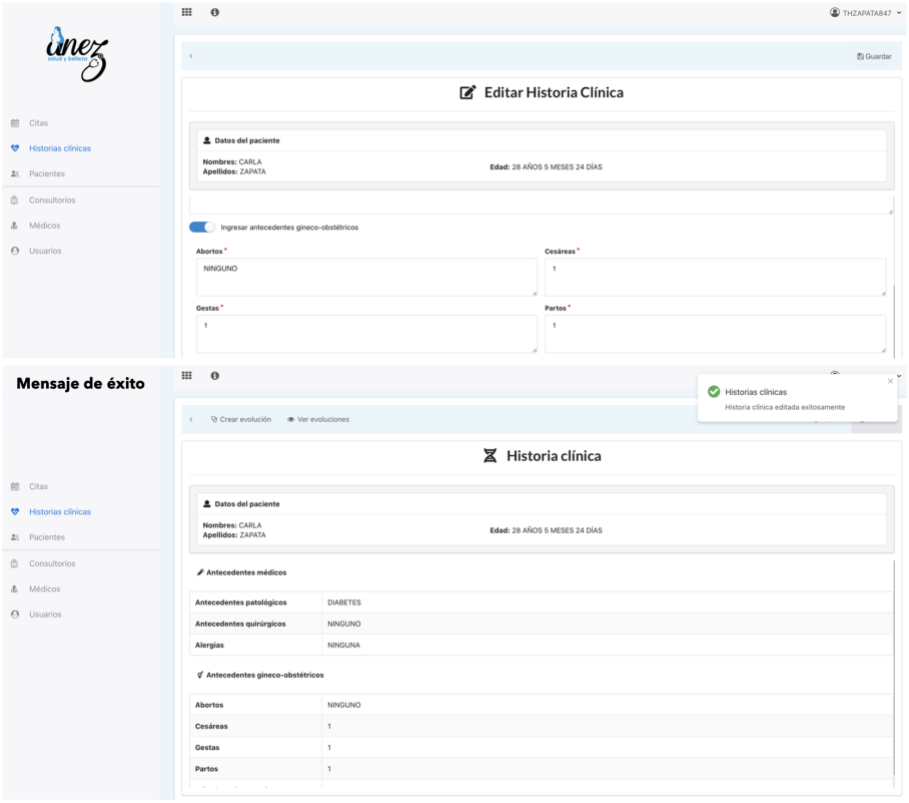
ii. Frontend

Figura 56.
Historia de usuario 4, Crear historia clínica.



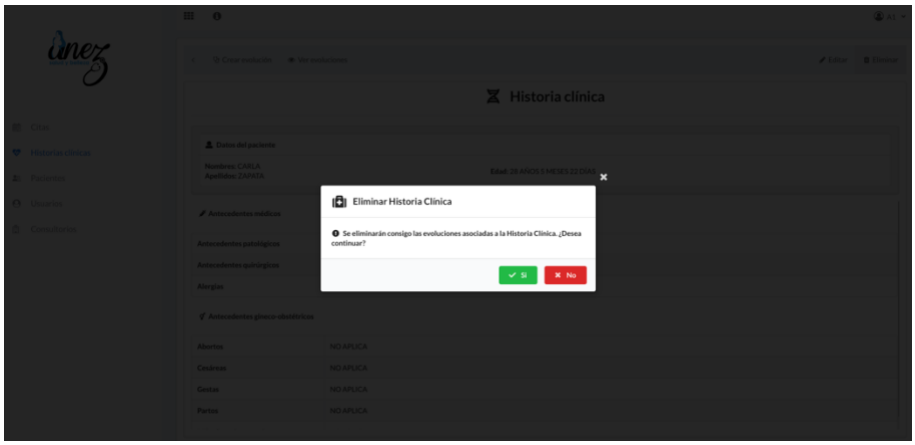
Nota. Página crear historias clínicas: presenta los datos del paciente y un formulario con campos requeridos y campos referentes a los antecedentes gineco-obstréticos los cuales son opcionales.

Figura 57.
.Historia de usuario 4, Editar historia clínica.



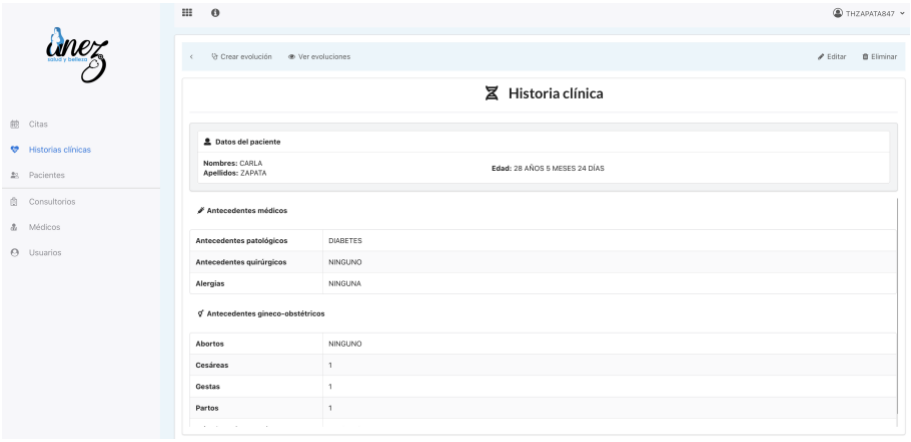
Nota. Página editar historia clínica: presenta los datos del paciente y los datos previamente ingresados, en caso de realizarse correctamente la edición muestra un mensaje de éxito.

Figura 58.
Historia de usuario 4, Eliminar historia clínica.



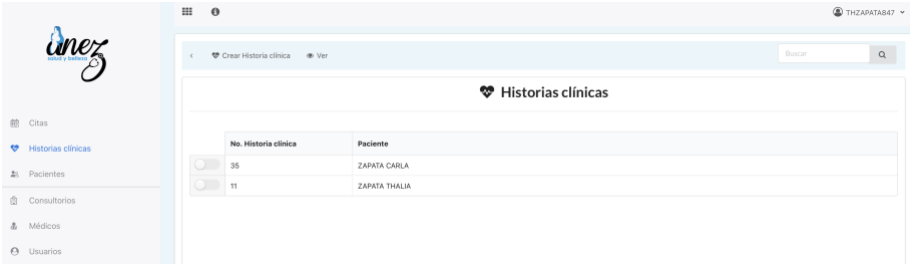
Nota. Modal eliminar historia clínica: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 59.
Historia de usuario 4, Detalle de historia clínica por id.



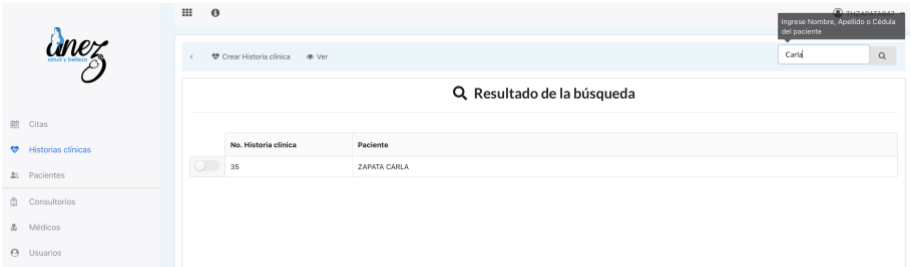
Nota. Página detalle de historia clínica: presenta los datos del paciente así como los antecedentes médicos y gineco-obstétricos del paciente, además de las opciones para editar o eliminar.

Figura 61.
Historia de usuario 4, Listado de historias clínicas.



Nota. Página listado de historias clínicas: presenta todas las historias creadas.

Figura 60.
Historia de usuario 4, Buscar historias clínicas.



Nota. Página buscar historias clínicas: el usuario puede ingresar el nombre, apellido o número de cédula del paciente.

iii. Pruebas

Figura 62.
Historia de usuario 4, Pruebas unitarias.

Backend			2.862s
Historias clínicas	/historia_clinica	passed	0.016s
Historias clínicas	/historias_clinicas	passed	0.007s
Historias clínicas	/historia_clinica/:id	passed	0.009s
Historias clínicas	/hc_buscar/:value	passed	0.036s
Historias clínicas	/paciente_historia/:id	passed	0.006s
Historias clínicas	/historia_clinica/:id	passed	0.011s
Historias clínicas	/historia_clinica/:id	passed	0.004s
Frontend			4.578s
Historias clínicas	Render del componente Listado de historias clínicas - Buscar	passed	0.005s
Historias clínicas	Render del componente Detalle de historia clínica por id	passed	0.001s
Historias clínicas	Render del componente Agregar - Editar	passed	0s
Historias clínicas	Comprobar el botón de crear nueva historia clínica	passed	0.098s
Historias clínicas	Comprobar el botón de editar historia clínica	passed	0.048s
Historias clínicas	Comprobar el botón de eliminar historia clínica	passed	0.022s
Historias clínicas	Comprobar el botón de buscar historia clínica	passed	0.053s

Nota. Pruebas unitarias y pruebas de intregración de los endpoints, componentes y eventos utilizados para la implementación de la historia de usuario 4.

3.2.2.4 ITERACIÓN 3

Las historias de usuario 5 y 6 implementadas en la iteración 3 corresponden a:

- Módulo de gestión de evoluciones.
- Módulo de gestión de usuarios.

a. Historia de usuario 5

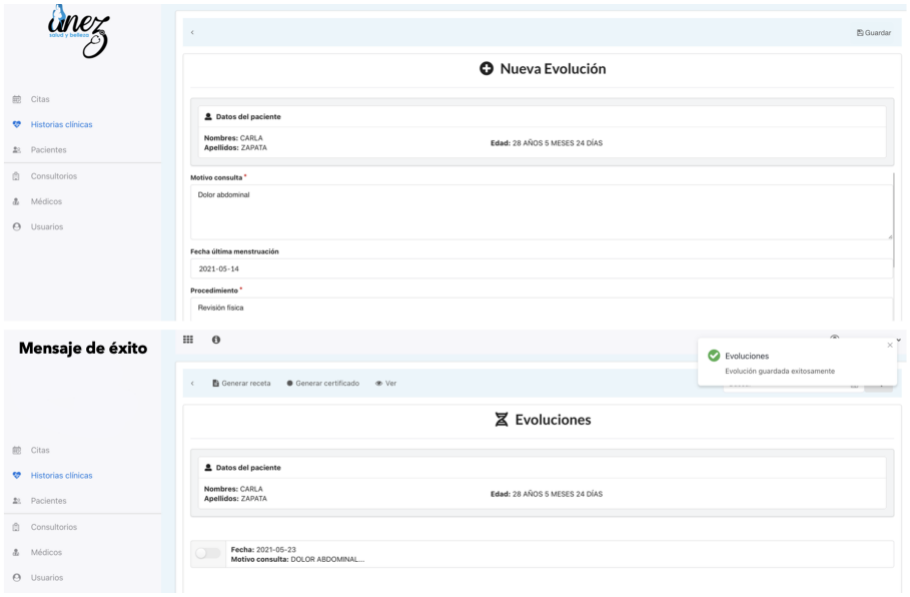
i. Backend

Tabla 9.
Historia de usuario 5, Endpoints.

Método	URI	Descripción
POST	/evolucion	Crear evolución.
GET	/evoluciones	Obtener todas las evoluciones.
GET	/evolucion/:id	Obtener evolución por id.
GET	/evoluciones_fecha /:id/:fecha1/:fecha2	Obtener evoluciones dentro de un rango de tiempo.
GET	/cie10_sub/:value	Obtener las subcategorías de CIE10 por el código o descripción.
PUT	/evolucion/:id	Editar una evolución.
DELETE	/evolucion/:id	Eliminar evolución.

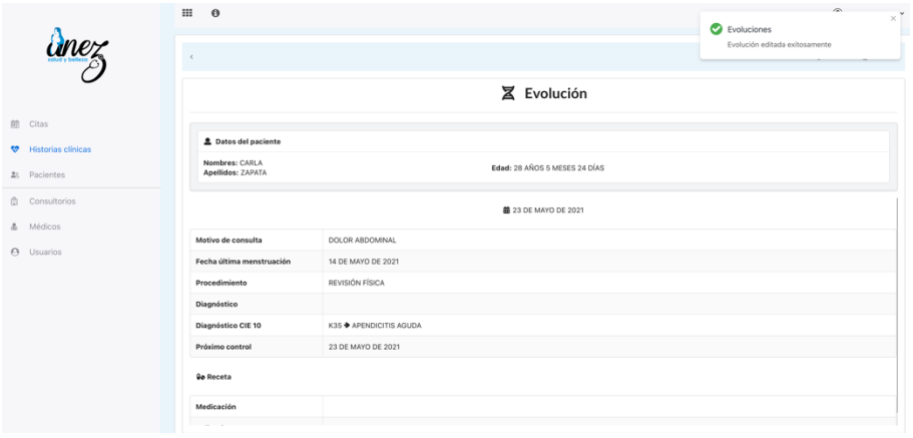
ii. Frontend

Figura 63.
Historia de usuario 5, Crear evolución.



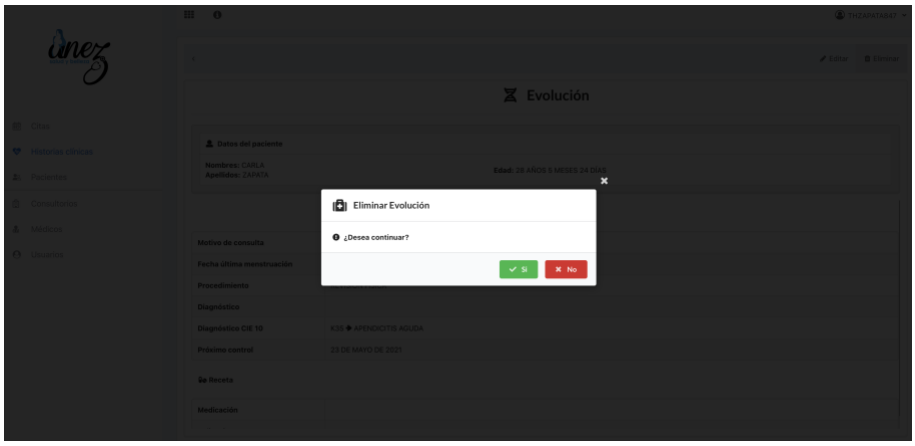
Nota. Página crear evolución: se presenta un formulario para ingresar los detalles de la consulta, adicionalmente se muestra la opción de ingresar imágenes mediante su url. Una vez ingresados los campos requeridos se presentará un mensaje de éxito.

Figura 64.
Historia de usuario 5, Editar evolución.



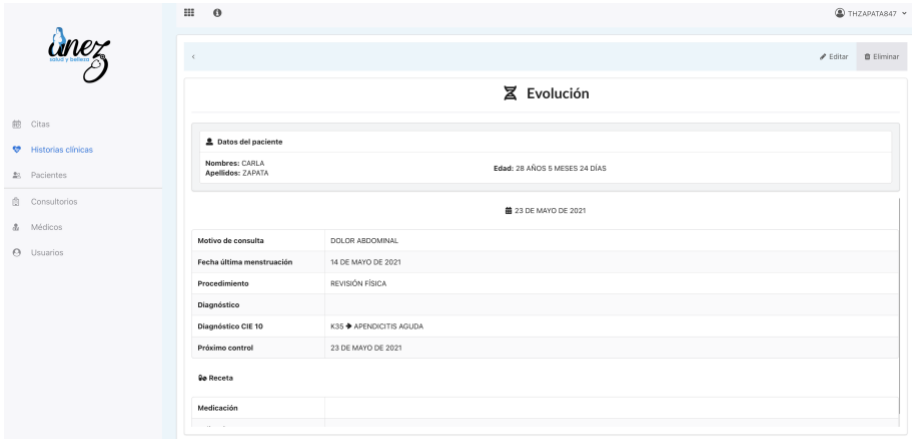
Nota. Página editar evolución: permite al usuario cambiar alguno de los datos previamente ingresados, al ingresar todos los campos requeridos se presentará un mensaje de éxito.

Figura 65.
Historia de usuario 5, Eliminar evolución.



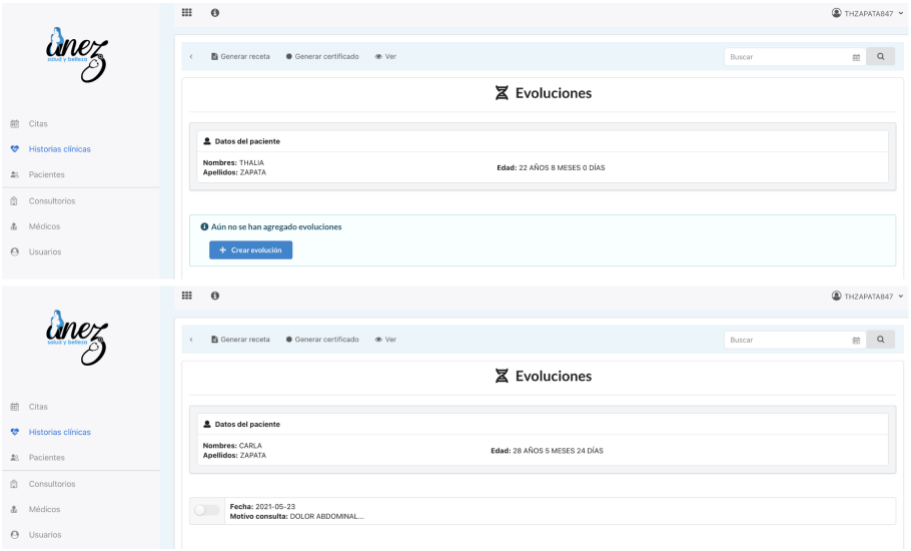
Nota. Modal eliminar evolución: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 66.
Historia de usuario 5, Detalle de evolución por id.



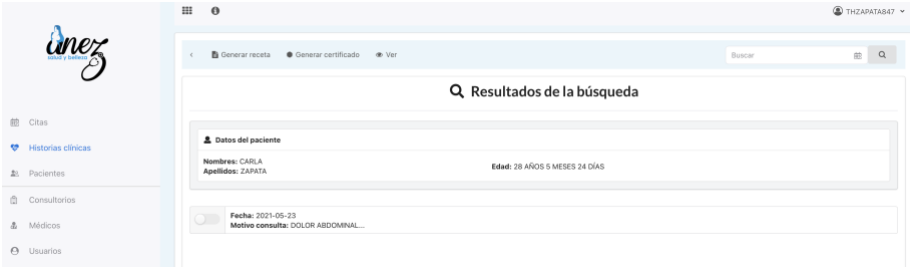
Nota. Página detalle de evolución: se presentan todos los datos previamente ingresados junto con las opciones de Editar y Eliminar.

Figura 67.
Historia de usuario 5, Listado de evoluciones.



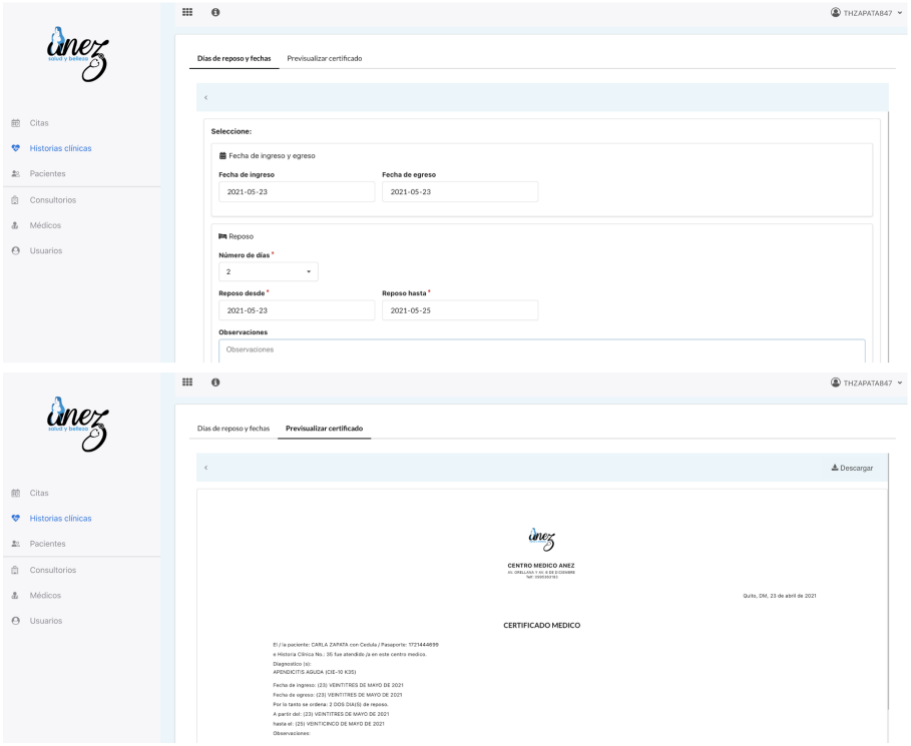
Nota. Página lisatdo de evoluciones: al no existir evoluciones muestra un mensaje junto con la opción de crear una evolución, caso contrario presenta todas las evoluciones creadas.
En esta página se muestras las opciones para Generar receta y Generar certificado médico.

Figura 68.
Historia de usuario 5, Buscar evoluciones.



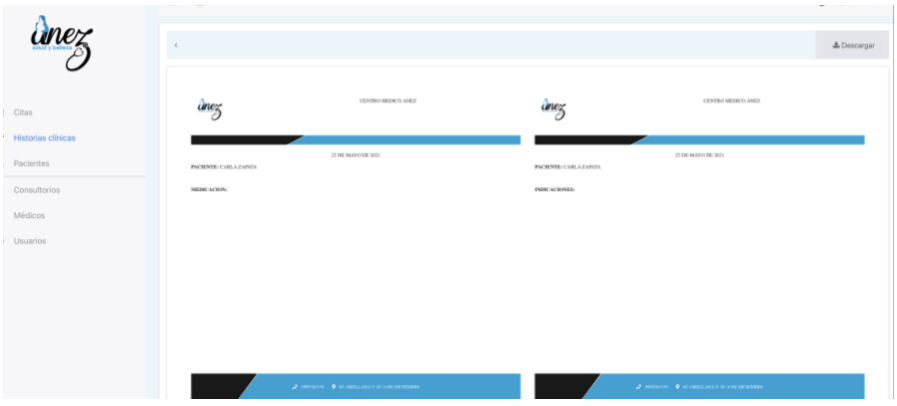
Nota. Página buscar evoluciones: permite la búsqueda entre un rango de fechas.

Figura 69.
Historia de usuario 5, Generar certificado médico.



Nota. Página generar certificado: presenta dos componentes, el primero para seleccionar la fecha de ingreso y egreso, tiempo de reposo y observaciones, y la segunda para previsualizar el certificado junto con la opción para descargar.

Figura 70.
Historia de usuario 5, Generar certificado médico.



Nota. Página generar receta: muestra la previsualización de la receta y la opción para descargarla en formato PDF.

iii. Pruebas

Figura 71.
Historia de usuario 5, Pruebas unitarias.

Backend			2.766s
Evoluciones	/evolucion	passed	0.019s
Evoluciones	/evoluciones	passed	0.008s
Evoluciones	/evolucion/:id	passed	0.006s
Evoluciones	/evoluciones_fecha/:id/:fecha1/:fecha2	passed	0.019s
Evoluciones	/cie10_sub/:value	passed	0.008s
Evoluciones	/evolucion/:id	passed	0.024s
Evoluciones	/evolucion/:id	passed	0.017s
Frontend			4.219s
Evoluciones	Render del componente Listado de evoluciones	passed	0.005s
Evoluciones	Render del componente Buscar evoluciones	passed	0.002s
Evoluciones	Render del componente Detalle de evolución por id	passed	0.001s
Evoluciones	Render del componente Agregar - Editar	passed	0.001s
Evoluciones	Render del componente Receta	passed	0s
Evoluciones	Render del componente Certificado médico	passed	0.001s
Evoluciones	Comprobar el botón de crear nueva evolución	passed	0.093s
Evoluciones	Comprobar el botón de editar evolución	passed	0.025s
Evoluciones	Comprobar el botón de eliminar evolución	passed	0.016s
Evoluciones	Comprobar el botón de buscar evolución	passed	0.048s

Nota. Pruebas unitarias y pruebas de intregración de los recursos utilizados para la implementación de la historia de usuario 5.

b. Historia de usuario 6

i. Backend

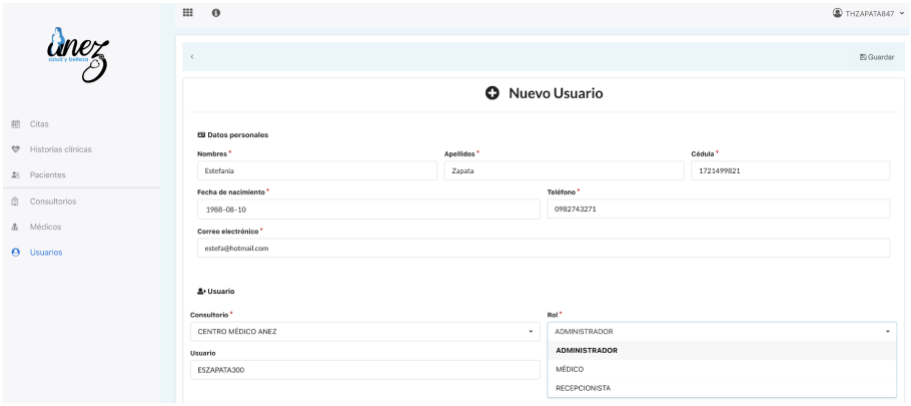
Tabla 10.
Historia de usuario 6, Endpoints.

Método	URI	Descripción
POST	/usuario	Crear usuario.
GET	/usuarios	Obtener todos los usuarios.
GET	/usuario/:id	Obtener usuario por id.

GET	/usuarios_buscar/:value	Obtener usuarios por el apellido, nombre o cédula.
PATCH	/usuario/:id	Editar un usuario.
PATCH	/usuario_update_pass /:value	Cambiar la contraseña desde el perfil
DELETE	/usuario/:id	Eliminar usuario.

ii. Frontend

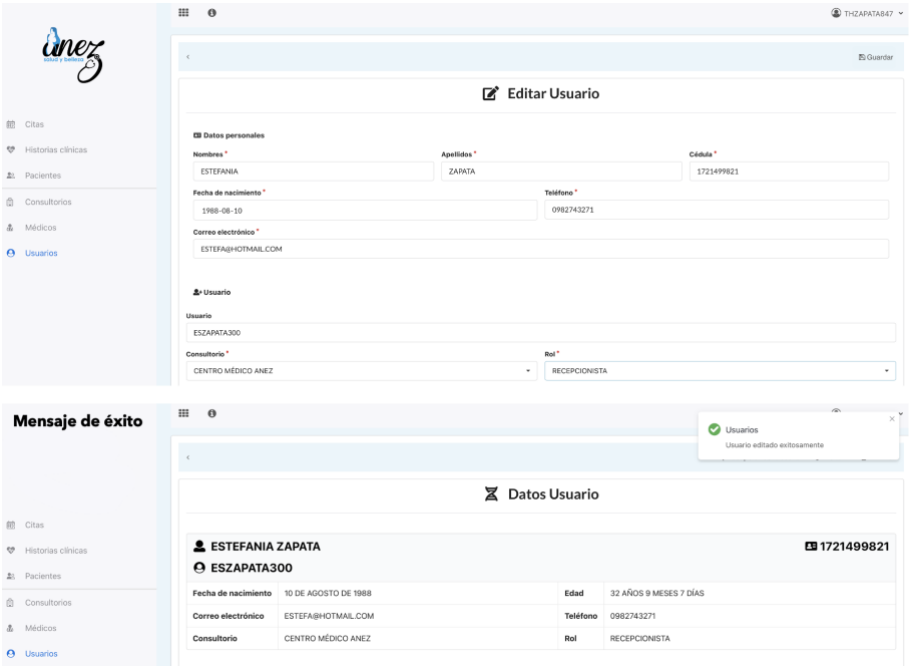
Figura 72.
Historia de usuario 6, Crear usuario.



Nota. Página crear usuario: permite al administrador ingresar los datos personales del usuario y seleccionar un rol.

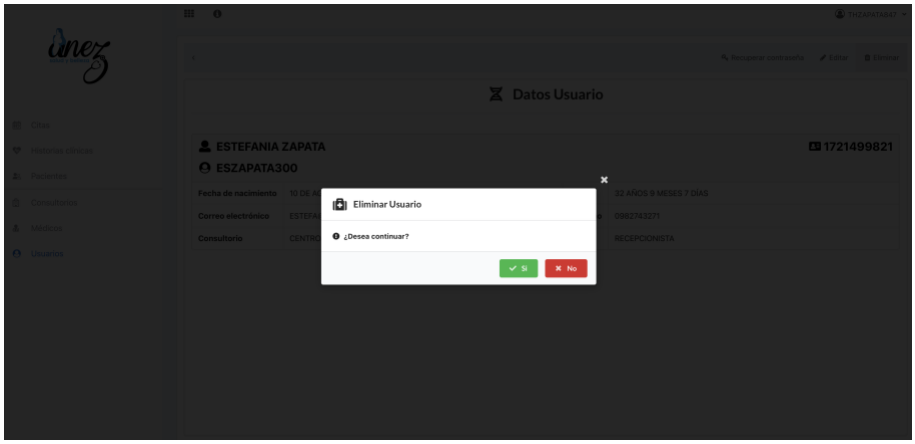
Automáticamente genera un nombre de su usuario y contraseña los mismos que, son enviados al correo proporcionado.

Figura 73.
Historia de usuario 6, Editar usuario.



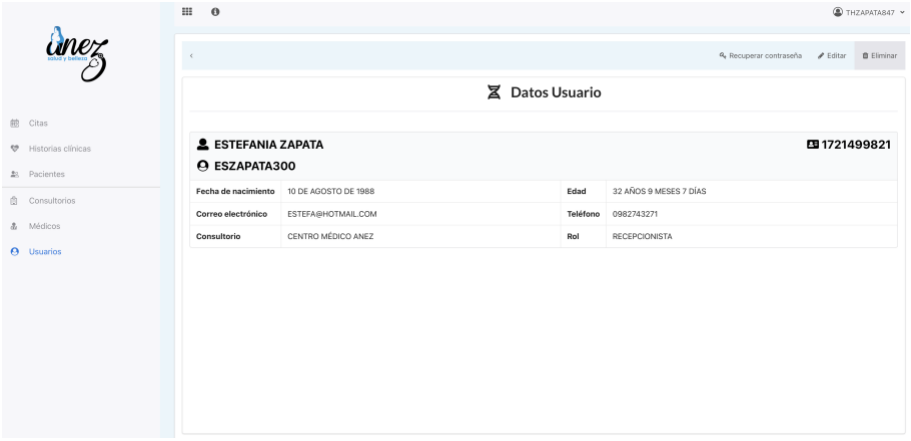
Nota. Página editar usuario: el administrador puede editar todos los datos del usuario a excepción de su nombre de usuario y contraseña. Si se llena adecuadamente el formulario presentará un mensaje de éxito

Figura 74.
Historia de usuario 6, Eliminar usuario.



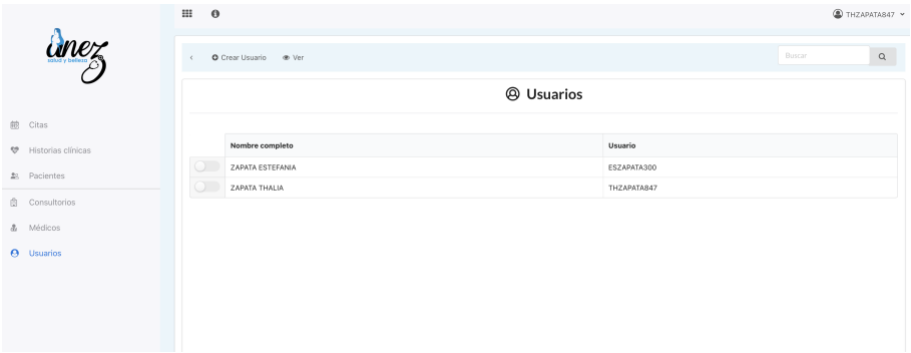
Nota. Modal eliminar usuario: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 75.
Historia de usuario 6, Detalle de usuario por id.



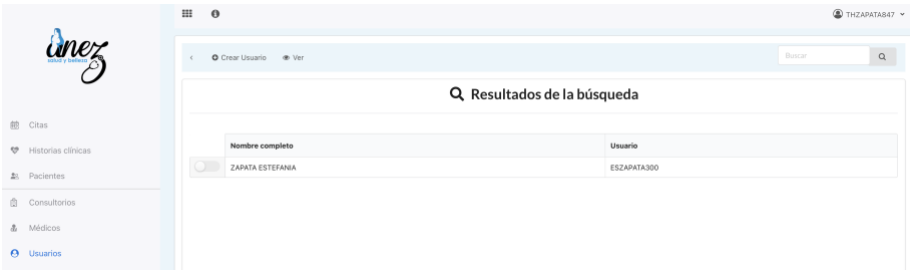
Nota. Página detalle de usuario: se muestran los datos del usuario y las opciones de Recuperar contraseña, Editar y Eliminar.

Figura 76.
Historia de usuario 6, Listado de usuarios.



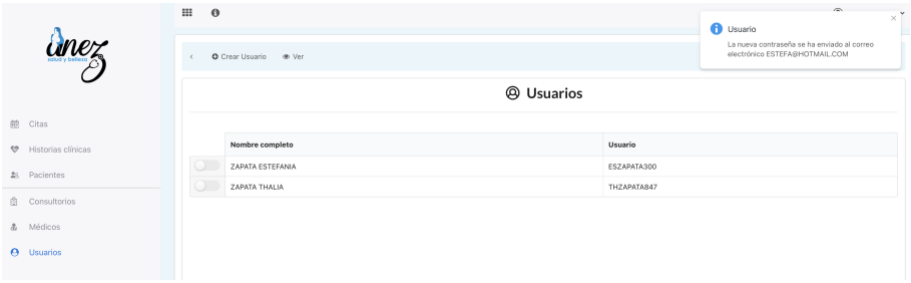
Nota. Página listado de usuarios: se presentan todos los usuarios registrados.

Figura 77.
Historia de usuario 6, Buscar usuarios.



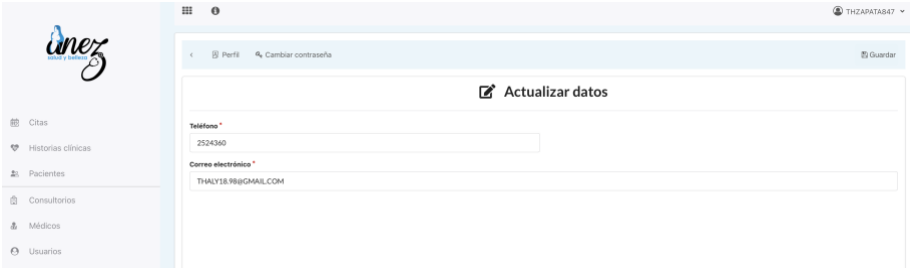
Nota. Página buscar usuarios: se puede hacer la búsqueda por el nombre de usuario, apellido o nombre del usuario.

Figura 78.
Historia de usuario 6, Recuperar contraseña.



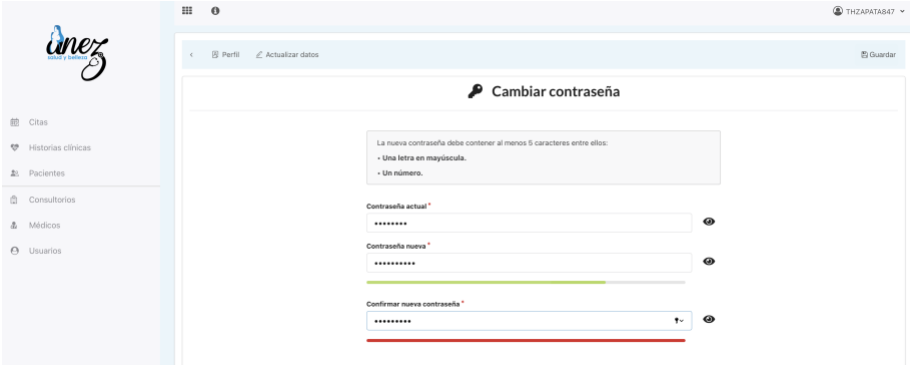
Nota. Página recuperar contraseña: la nueva contraseña es enviada al correo electrónico registrado, si se ha realizado correctamente se presenta un mensaje informando el correo al cual se ha enviado la contraseña.

Figura 79.
Historia de usuario 6, Actualizar datos.



Nota. Página actualizar datos: desde la opción Perfil se puede actualizar únicamente el teléfono y/o correo electrónico del usuario.

Figura 80.
Historia de usuario 6, Cambiar contraseña.



Nota. Página cambiar contraseña: desde la opción Perfil es posible cambiar la contraseña, para lo cual el usuario debe ingresar su contraseña actual y una nueva que cumpla con los requisitos propuestos.

iii. Pruebas

Figura 81.
Historia de usuario 6, Pruebas unitarias.

Backend			2.869s
Usuarios	/usuario	passed	0.099s
Usuarios	/usuarios	passed	0.006s
Usuarios	/usuario/:id	passed	0.006s
Usuarios	/usuarios_buscar/:value	passed	0.004s
Usuarios	/usuario/:id	passed	0.009s
Usuarios	/usuario_update_pass/:id/:value	passed	0.069s
Usuarios	/usuario/:id	passed	0.005s
Frontend			4.833s
Usuarios	Render del componente Listado de usuarios - Buscar	passed	0.005s
Usuarios	Render del componente Detalle de usuario por id	passed	0s
Usuarios	Render del componente Agregar - Editar	passed	0.001s
Usuarios	Comprobar el botón de crear nuevo usuario	passed	0.094s
Usuarios	Comprobar el botón de editar usuario	passed	0.026s
Usuarios	Comprobar el botón de eliminar usuario	passed	0.017s
Usuarios	Comprobar el botón de buscar usuario	passed	0.042s
Frontend			3.843s
Perfil	Render del componente Perfil	passed	0.005s
Perfil	Render del componente Actualizar datos	passed	0.001s
Perfil	Render del componente Cambiar contraseña	passed	0s
Perfil	Comprobar el botón de actualizar datos	passed	0.07s
Perfil	Comprobar el botón de cambiar contraseña	passed	0.019s

Nota. Pruebas unitarias y pruebas de intregración de las rutas, componentes y eventos utilizados para la gestión de usuarios.

3.2.2.5 ITERACIÓN 4

En la última iteración se implementaron las historias de usuario 7 y 8 correspondientes a los requerimientos:

- Módulo de gestión de médicos.
- Módulo de gestión de consultorios.

a. Historia de usuario 7

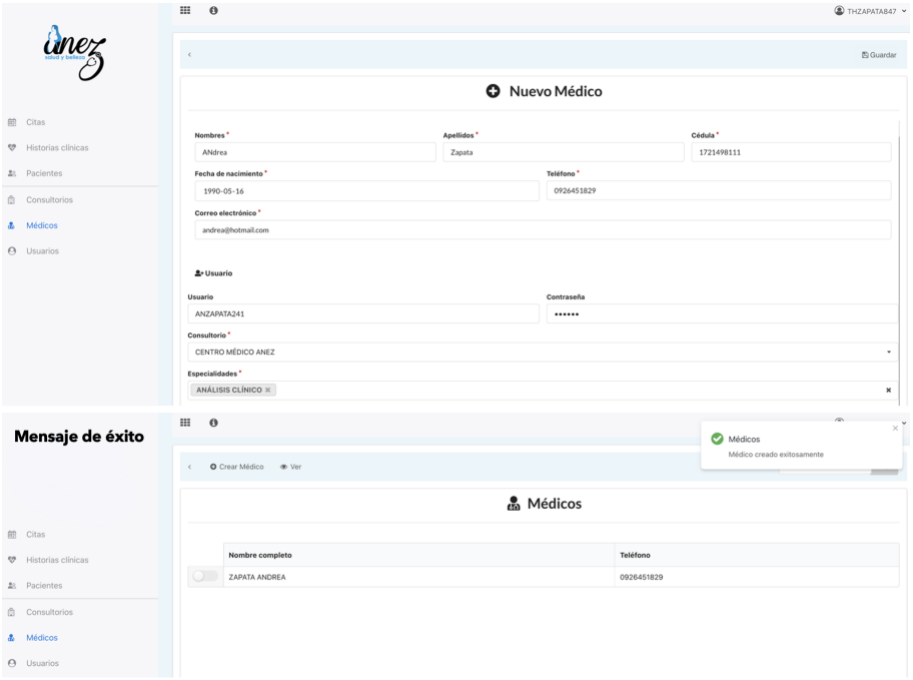
i. Backend

Tabla 11.
Historia de usuario 7, Endpoints.

Método	URI	Descripción
POST	/medico	Crear médico.
GET	/medicos	Obtener todos los médicos.
GET	/medico/:id	Obtener médico por id.
GET	/medicos_buscar/:value	Obtener todos los médicos.
PUT	/medico/:id	Editar un médico.
DELETE	/medico/:id	Eliminar médico.

ii. Frontend

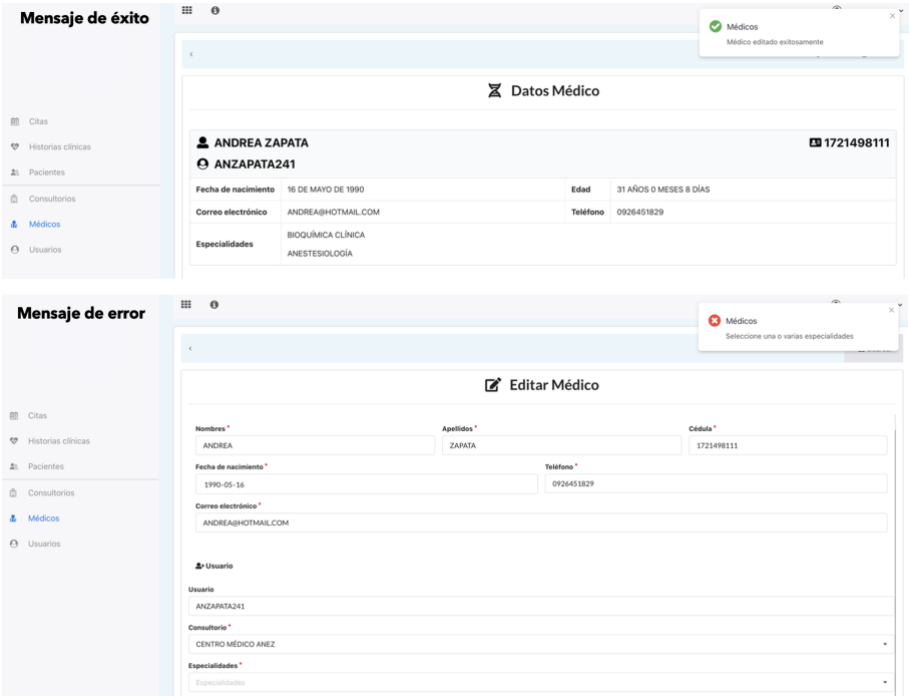
Figura 82.
Historia de usuario 7, Crear médico.



Nota. Página crear médico: permite el ingreso de los datos personales del médico así como sus especialidades médicas.

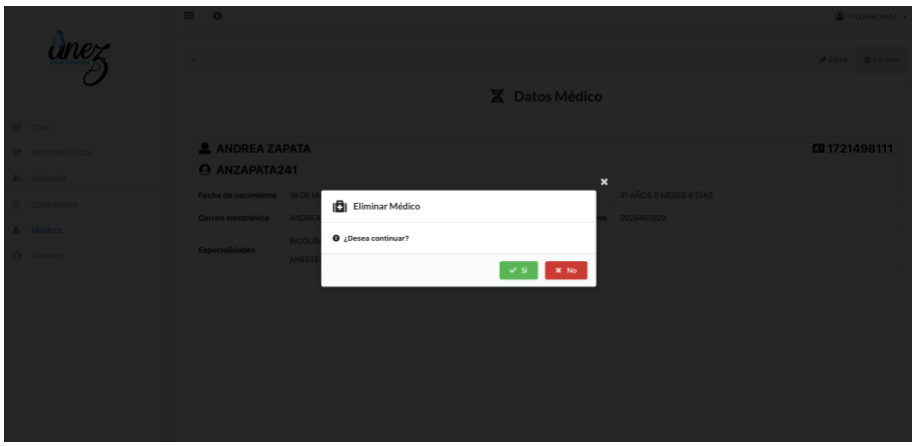
Se presenta como alternativa a la creación de usuarios con la diferencia de que no es posible escoger un rol, genera automáticamente el usuario y contraseña, mismos que son enviados al correo electrónico registrado.

Figura 83.
Historia de usuario 7, Editar médico.



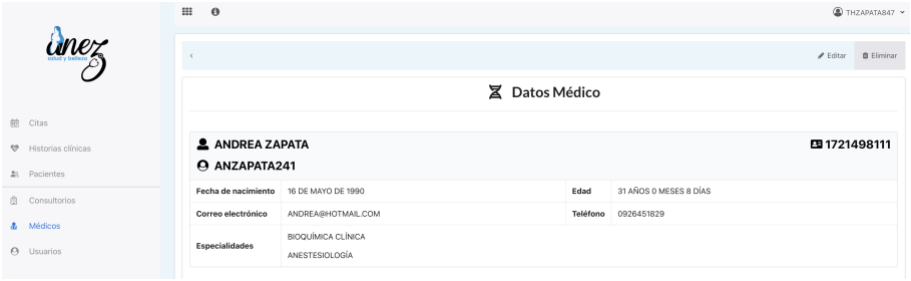
Nota. Página editar médico: es posible editar únicamente ciertos datos, en caso de no seleccionar ninguna especialidad mostrará un mensaje de error.

Figura 84.
Historia de usuario 7, Eliminar médico.



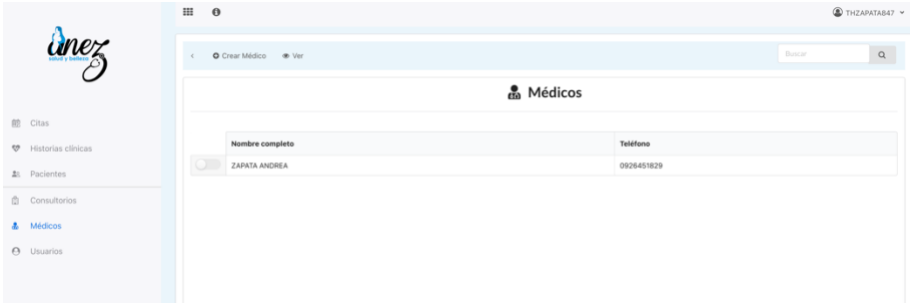
Nota. Modal eliminar médico: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 85.
Historia de usuario 7, Detalle de médico por id.



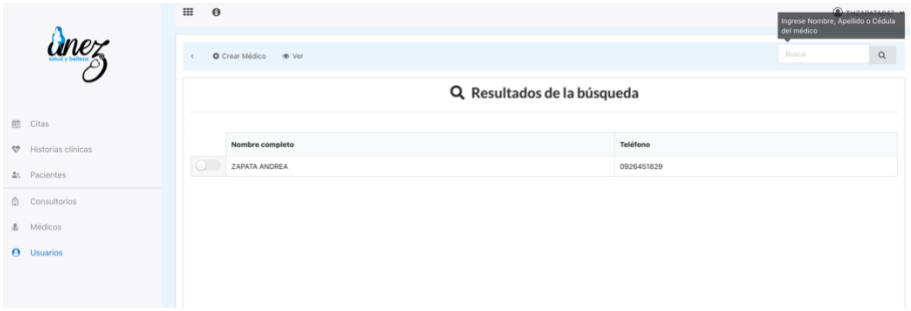
Nota. Página detalle de médico: presenta los datos del médico y las opciones para Editar o Eliminar.

Figura 86.
Historia de usuario 7, Listado de médicos.



Nota. Página listado de médicos: presenta todos los médicos registrados.

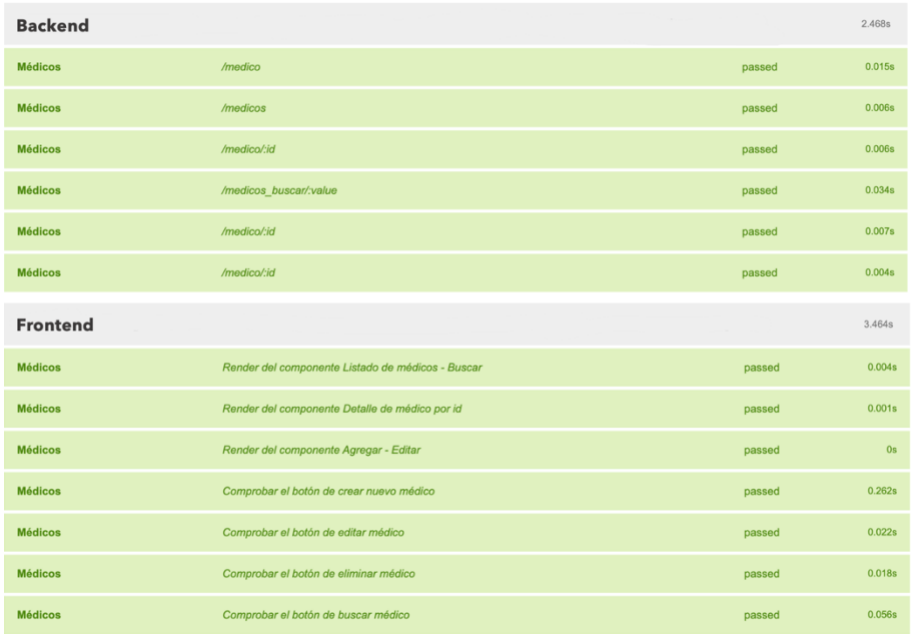
Figura 87.
Historia de usuario 7, Buscar médicos.



Nota. Página buscar médicos: el usuario puede buscar médicos por su nombre, apellido o cédula.

iii. Pruebas

Figura 88.
Historia de usuario 7, Pruebas unitarias.



Backend			2.468s
Médicos	/medico	passed	0.015s
Médicos	/medicos	passed	0.006s
Médicos	/medico/:id	passed	0.006s
Médicos	/medicos_buscar/:value	passed	0.034s
Médicos	/medico/:id	passed	0.007s
Médicos	/medico/:id	passed	0.004s
Frontend			3.464s
Médicos	Render del componente Listado de médicos - Buscar	passed	0.004s
Médicos	Render del componente Detalle de médico por id	passed	0.001s
Médicos	Render del componente Agregar - Editar	passed	0s
Médicos	Comprobar el botón de crear nuevo médico	passed	0.262s
Médicos	Comprobar el botón de editar médico	passed	0.022s
Médicos	Comprobar el botón de eliminar médico	passed	0.018s
Médicos	Comprobar el botón de buscar médico	passed	0.056s

Nota. Pruebas unitarias y pruebas de intregración de los recursos utilizados para la implementación de la historia de usuario 7.

b. Historia de usuario 8

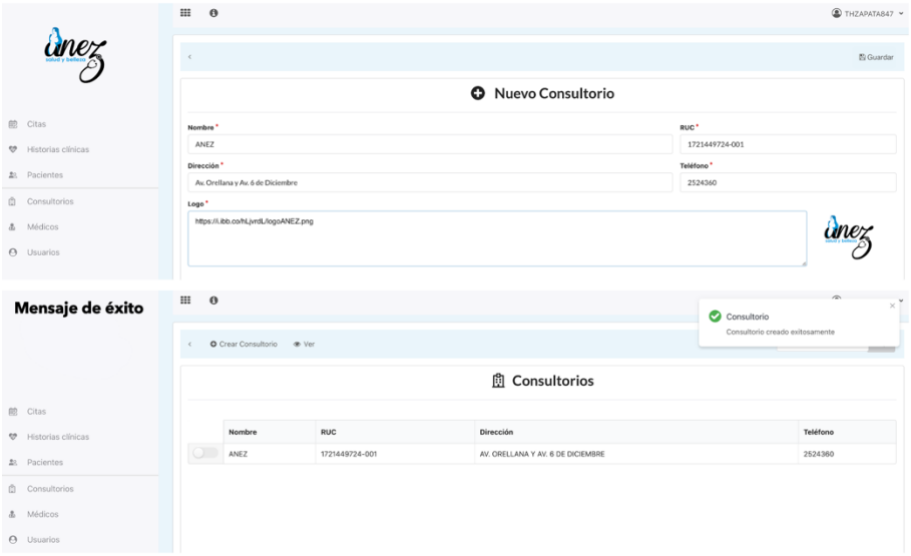
i. Backend

Tabla 12.
Historia de usuario 8, Endpoints.

Método	URI	Descripción
POST	/consultorio	Crear consultorio.
GET	/consultorios	Obtener todos los consultorios.
GET	/consultorio/:id	Obtener consultorio por id.
GET	/consultorio_buscar /:value	Obtener todos los consultorios.
PUT	/consultorio/:id	Editar un consultorio.
DELETE	/consultorio/:id	Eliminar consultorio.

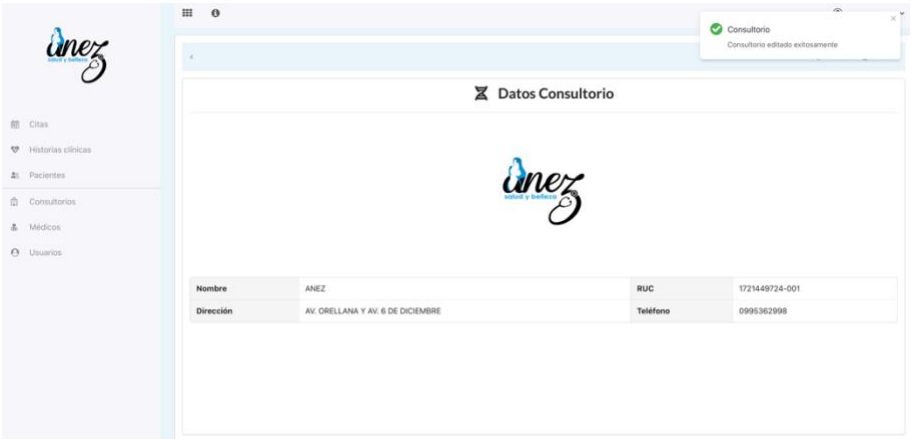
ii. Frontend

Figura 89.
Historia de usuario 8, Crear consultorio.



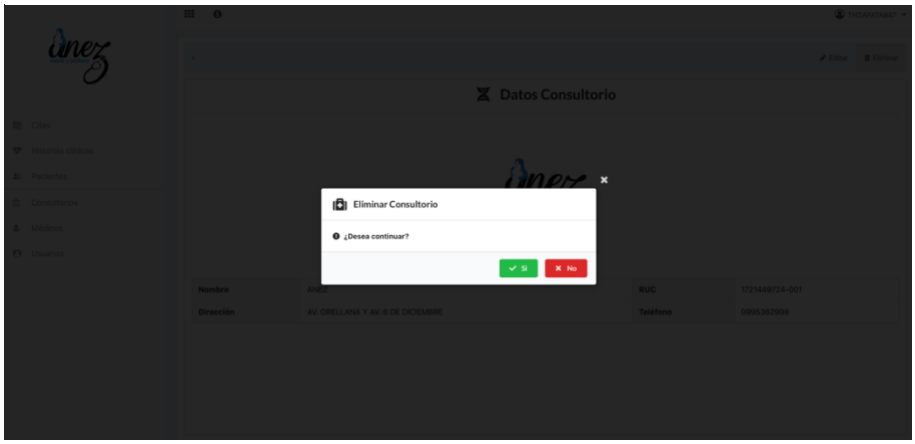
Nota. Página crear consultorio: admite el ingreso de datos principales sobre el centro médico.

Figura 90.
Historia de usuario 8, Editar consultorio.



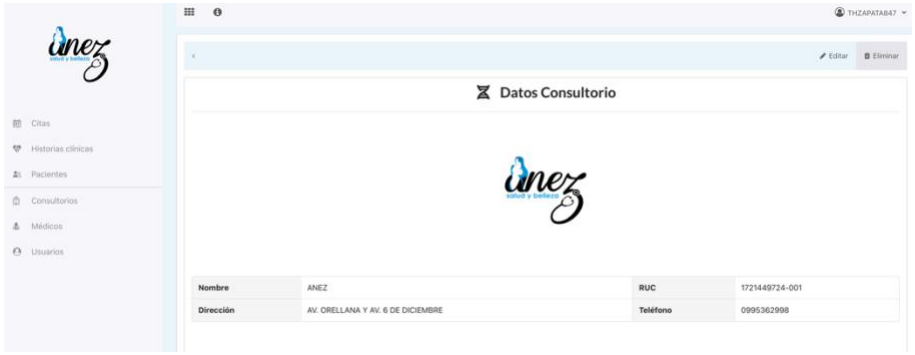
Nota. Página editar consultorio: una vez llenados todos los campos requeridos se presentará un mensaje de éxito.

Figura 91.
Historia de usuario 8, Eliminar consultorio.



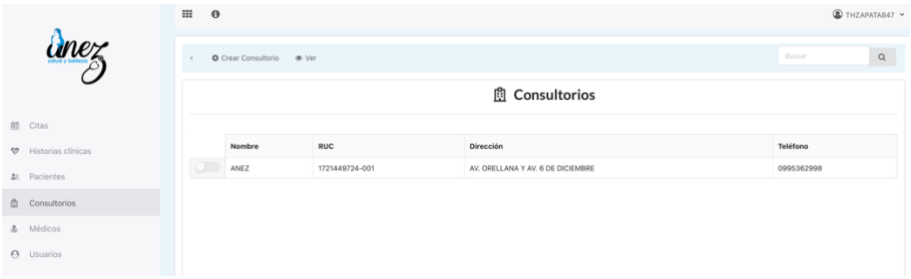
Nota. Modal eliminar consultorio: se muestra un mensaje de confirmación para proceder a la eliminación.

Figura 92.
Historia de usuario 8, Detalle de consultorio por id.



Nota. Página detalle de consultorio: presenta los datos del centro médico y las opciones para Editar y Eliminar.

Figura 93.
Historia de usuario 8, Listado de consultorios.



Nota. Página listado de consultorios: muestra todos los consultorios ingresados.

Figura 94.
Historia de usuario 8, Buscar consultorios.



Nota. Página buscar consultorios: permite la búsqueda por nombre del centro médico o RUC.

iii. Pruebas

Figura 95.
Historia de usuario 8, Pruebas unitarias.

Backend			2.711s
Consultorios	/consultorio	passed	0.02s
Consultorios	/consultorios	passed	0.009s
Consultorios	/consultorio/:id	passed	0.007s
Consultorios	/consultorios_buscar/:value	passed	0.004s
Consultorios	/consultorio/:id	passed	0.012s
Consultorios	/consultorio/:id	passed	0.009s
Frontend			4.849s
Consultorios	Render del componente Listado de consultorios - Buscar	passed	0.004s
Consultorios	Render del componente Detalle de consultorio por id	passed	0s
Consultorios	Render del componente Agregar - Editar	passed	0.001s
Consultorios	Comprobar el botón de crear nuevo consultorio	passed	0.086s
Consultorios	Comprobar el botón de editar consultorio	passed	0.016s
Consultorios	Comprobar el botón de eliminar consultorio	passed	0.022s
Consultorios	Comprobar el botón de buscar consultorio	passed	0.053s

Nota. Pruebas unitarias y pruebas de intregración de las rutas y componentes asociados a la historia de usuario 8.

3.2.3 FASE DE PRODUCCIÓN

De acuerdo con Joskowicz (2008) no se realizan más desarrollos funcionales puesto que para poner el sistema en producción se requiere tener desarrolladas todas las funcionalidades definidas en la fase de exploración y por ende en esta fase únicamente se realizan tareas de ajuste en base a los resultados obtenidos en la fase de iteraciones.

3.2.3.1 DESPLIEGUE

Tras haber culminado el desarrollo de cada uno de los módulos indicados en las historias de usuario y aprobado las pruebas unitarias y de aceptación, se procedió a realizar el despliegue del sistema web en un servidor en la nube, mismo al que se puede acceder mediante un navegador web, ingresando al siguiente enlace: https://www.anez-salud.com

3.2.3.2 ACEPTACIÓN

Una vez desplegado el sistema web y realizadas las pruebas correspondientes por parte del usuario final, se estableció un documento de aceptación del proyecto; éste corresponde a la versión 1.0, con el propósito de certificar que el sistema cumple con todas las funcionalidades propuestas y que las pruebas realizadas brindaron resultados satisfactorios.

El documento mencionado, se encuentra en la sección Anexos.

4. CONCLUSIONES Y RECOMENDACIONES

A continuación, se presentan las conclusiones y recomendaciones obtenidas una vez implementado el sistema web y el fundamento teórico del presente trabajo de disertación:

4.1 CONCLUSIONES

- La aplicación de la metodología ágil Extreme Programming mediante su esquema para el análisis de requerimientos, planificación, codificación, testeo y despliegue del sistema web permitió alcanzar cada uno de los objetivos planteados y por ende implementar un sistema óptimo para la gestión de citas, pacientes e historias clínicas.
- El diseño de una base de datos acorde a los requerimientos establecidos por el cliente garantiza que se cumpla con los factores de confidencialidad, seguridad, accesibilidad e integridad de los datos.
- El desarrollo de un sistema web diseñado en base a los requerimientos del centro médico ANEZ, facilita la gestión de citas, pacientes e historias clínicas y a su vez evita la pérdida de datos e información, puesto que éstos se encuentran centralizados y accesibles desde un navegador.
- La implementación de un sistema web que cumpla con todos los requerimientos establecidos por el cliente y cuyo rendimiento sea el adecuado, permitió que la automatización de procesos asistenciales para el centro médico tenga una gran incidencia para el funcionamiento del mismo.

- Concluido el diseño y desarrollo del sistema en base a los requerimientos establecidos con el cliente, realizado las respectivas pruebas unitarias y de integración, así como la implementación de este, se establece que el sistema web fue aceptado a satisfacción por el cliente, lográndose de esta manera cumplir exitosamente cada uno de los objetivos planteados.
- Finalmente, con el objetivo de aprovechar cada una de las funcionalidades implementadas en el sistema web, se procede a realizar una capacitación al usuario final sobre el manejo y uso de la aplicación.

4.2 RECOMENDACIONES

- Utilizar la metodología XP para proyectos donde exista una constante comunicación entre todos los involucrados del proyecto, los requerimientos evolucionen conforme se desarrolla el sistema; se requiera una retroalimentación inmediata y una planificación abierta que se ajuste a un cronograma de actividades flexible y dinámico.
- Realizar periódicamente el mantenimiento del sistema web, con el propósito de conservar el valor del sistema a lo largo del tiempo mediante la ampliación de sus funcionalidades, eliminación de funciones obsoletas o aplicación de nueva tecnología.
- Capacitar a los nuevos funcionarios del centro médico en el manejo y uso del sistema web.
- Ampliar las funcionalidades del sistema, considerando para futuras versiones la implementación de un módulo para gestión de inventario y un módulo de facturación, mismos que apoyarían a la automatización de procesos asistenciales.
- En cuanto a la facultad de Ingeniería, escuela de Sistemas y Computación se recomienda impulsar el desarrollo de proyectos interinstitucionales enfocados en las principales asignaturas que se estudia a lo largo de la carrera ; de esta forma se complementarán y reforzarán los conocimientos adquiridos y a la vez se adquirirá experiencia para una acertada incorporación en el mundo laboral.

REFERENCIAS

Aliaga, A., & Miani, M. (2008). *PostgreSQL* (Vol. 1).

Amaya, V. (2021). *Requerimientos & historias de usuario*.

Amazon Web Services Inc. (2018). *Información general sobre Amazon Web Services*.

Amazon Web Services Inc. (2021a). *Amazon EC2*. https://aws.amazon.com/es/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc

Amazon Web Services Inc. (2021b). *Amazon Relational Database Service (RDS)*. https://aws.amazon.com/es/rds/

Ambler, S. (2002). Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. In *John WileySons*. http://www.amazon.fr/exec/obidos/ASIN/0471202827/citeulike04-21%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Agile+Modeling+:#0

Bahit, E. (2012). *Scrum & Extreme Programming Para Programadores*. 162. http://www.cursosdeprogramacionadistancia.com/static/pdf/material-sin-personalizar-agile.pdf

Casas, M. (2014). *Servicios Web*. https://www.ibm.com/support/knowledgecenter/es/SSAW57_9.0.5/com.ibm.websphere.nd.multiplatform.doc/ae/cwbs_wbs2.html

Collazo Garcia, A., & Labrador, D. (2017). *La Programación Extrema Autores : Marycarmen Díaz Labrador*. *November 2013*. https://doi.org/10.13140/RG.2.2.29359.43687

De las Pueblas Encinas, G. M. (2019). *Definición de Requisitos Funcionales bajo Especificación IEEE para un*. 1–133.

Denzer, P. (2002). 51. PostgreSQL. *Profesores.Elo.Utfsm.Cl*, 21. http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf

Dept. Ciencia de la Computación e IA, U. de A. (2014). Introducción a los Servicios Web. Invocación de servicios web SOAP. *Universidad de Alicante*, 1–46. http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html

Duarte Vega, G. E. (2016). *Arquitectura Propuesta para un Servicio Web Completo: Metodología de Desarrollo e Implementación*. 1–154.

Encalada, C. (2018). *ANALISIS, DESARROLLO, IMPLEMENTACION E IMPLANTACION DE UN SISTEMA PARA LA AUTOMATIZACION DE PROCESOS EN UN CENTRO MEDICO. CASO DE ESTUDIO: E.M. CENTRO DE ESPECIALIDADES MEDICAS, UTILIZANDO LA METODOLOGIA EXTREME PROGRAMMING (XP).*

Express.js. (2021). *Introducción a Express.js*. https://expressjs.com/

Facebook Inc. (2017). *Jest*. https://doc.ebichu.cc/jest/es-ES/

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf

HYPERS. (2016). *Rsuite*. https://github.com/rsuite/rsuite

IBM. (2019). *¿Qué es SOAP?* https://www.ibm.com/support/knowledgecenter/es/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ac55770_.htm

Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. *Universidad de Vigo. España*, 1–22. http://iie.fing.edu.uy/~josej/docs/XP - Jose Joskowicz.pdf

Jovaldiv, V. C. (2016). *Modelo De Procesos Para El Desarrollo Del*. 187–208.

Laínez Fuentes, J. R. (2015). *Desarrollo de Software Ágil: Extremme Programming y Scrum*. (IT Campus Academy (ed.); Segunda ed).

Lamarca M. (2013). *Tesis Doctoral Hipertexto: el nuevo concepto de documento en la cultu- ra de la imagen. Universidad Complutense de Madrid.* http://www.hipertexto.info/documentos/serv_web.htm

Lezama López, J. A., & Machuca Abanto, E. (2017). Universidad Privada Antonio

Guillermo Urrelo. *Universidad Privada Antonio Guillermo Urrelo*. http://repositorio.upagu.edu.pe/handle/UPAGU/269

López Menéndez de Jiménez, R. (2015). *Metodologías ágiles de desarrollo de Software aplicadas a la gestión de proyectos empresariales*. 6–11.

Molina Rios, J. R., Zea Ordonez, M. P., Contento Segarra, M. J., & Garcia Zerda, F. G. (2017). State of Art: Development Methodologies in Web Applications. *3C Tecnologia*, *6*(3), 54–71. https://www.3ciencias.com/articulos/articulo/estado-del-arte-metodologias-desarrollo-aplicaciones-web/

Montero Molina, B., Cevallos Vite, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales Revista Multidisciplinaria de Investigación ISSN: 2550-6862*, *2*(17), 114–121. https://www.researchgate.net/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software

Moreno Saiz, S. (2015). *Estudio de Arquitecturas Software para Servicios de Internet de las Cosas*. http://oa.upm.es/37339/

Muñoz, A. (2013). *Introducción a NodeJS a través de Koans*. 242. http://nodejskoans.com/

Node.js. (2021). *Introducción a Node.js*. https://nodejs.org/es/

Pinto, S., Avila, M., & Mejia, M. (2016). *Diseño y desarrollo de una prueba de concepto de un lector electrónico (e-reader) de textos en inglés*. *1*, 1917–1921.

Pulido, E., Escobar, Ó., & Nuñez, J. Á. (2019). *Base de datos* (Primera ed). Patria Educación.

Red Hat Inc. (2021). *¿Qué es una arquitectura de aplicaciones?*

Rick, L. (2016). *Express. js Guide Book on Web framework for Node. js*.

Rodríguez, Á. I., Padilla, J. I., Parra, H. A., & Rodríguez, P. (2019). *Arquitectura basada en micro-servicios para aplicaciones web Microservices based architecture for web applications*.

Salud, D. E. (2006). *Guía de diseño y mejora continua de procesos asistenciales*.

Sataloff, R. T., Johns, M. M., & Kost, K. M. (2018). *COMPARACIÓN DE TENDENCIAS TECNOLÓGICAS EN APLICACIONES WEB*.

Sayago, J., Flores, E., & Recalde, A. (2019). *Análisis Comparativo entre los Estándares Orientados a Servicios Web SOAP , REST y GRAPHQL Comparative Analysis between Standards Oriented to Web Services SOAP , REST and GRAPHQL*. *9*, 10–22.

Semantic-Org. (2018). *Semantic UI*. https://github.com/Semantic-Org/Semantic-UI

Sequelize. (2021). *Introducción a Sequelize*. https://sequelize.org/

Severance, C. (2020). Uso de Servicios Web. In *Python para todos: explorando la información con Python 3*.

Veps, J. (2017). *SurviveJS - React De aprendiz a maestro Juho Vepsäläinen y Raúl Expósito*.

Vercel Inc. (2021). *Vercel*. https://vercel.com/docs

Yanez, D. (2019). Método descriptivo: características, etapas y ejemplos. *1*, 1. https://www.lifeder.com/metodo-descriptivo/

ANEXOS

Reunión No. 1

Fecha
22 de febrero 2021 – 16:00

Lugar
Av. Orellana y Av. 6 de diciembre, Centro médico ANEZ.

Temas tratados

- Análisis de requerimientos.
- Formulación inicial de historias de usuario.



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 2

Fecha
24 de marzo 2021 – 10:00

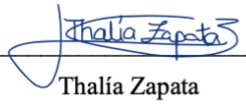
Lugar
Av. Orellana y Av. 6 de diciembre, Centro médico ANEZ.

Temas tratados

- Recopilación de información complementaria.
- Revisión de historias de usuario.
- Revisión de planificación.
- Presentación del diseño inicial de interfaces



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 3

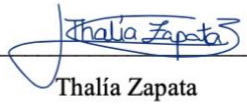
Fecha
13 de abril 2021 – 16:00

Lugar
Reunión virtual.

Temas tratados
• Presentación del módulo de autenticación y autorización. • Presentación del modulo de gestión de pacientes.



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 4

Fecha
29 de abril 2021 – 16:00

Lugar
Reunión virtual.

Temas tratados
• Presentación del módulo de gestión de citas. • Presentación del modulo de gestión de historias clínicas.



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 5

Fecha
14 de mayo 2021 – 16:00

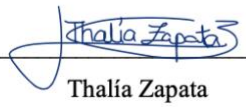
Lugar
Reunión virtual.

Temas tratados

- Presentación del módulo de evoluciones.
- Presentación del modulo de gestión de usuarios.



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 6

Fecha
28 de mayo 2021 – 16:00

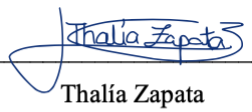
Lugar
Reunión virtual.

Temas tratados

- Presentación del módulo de médicos.
- Presentación del modulo de gestión de consultorios.



Dra. Estefanía Zapata



Thalía Zapata

Reunión No. 7

Fecha
7 de junio 2021 – 16:00

Lugar
Av. Orellana y Av. 6 de diciembre, Centro médico ANEZ.

Temas tratados

- Entrega del sistema web y manual de usuario.
- Planificación para capacitación al personal del centro médico.

Dra. Estefanía Zapata

Thalía Zapata

CENTRO MÉDICO ANEZ

Aceptación del Sistema Web

Quito, 7 de junio de 2021

Yo, Estefanía Zapata, con C.I. 1721444972, en calidad de Gerente General del Centro médico ANEZ, informo que el sistema web desarrollado para el trabajo de disertación titulado "DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DE PROCESOS ASISTENCIALES DE UN CENTRO MÉDICO, CASO DE ESTUDIO: CENTRO MÉDICO ANEZ", consta con los siguientes módulos, cuyas funcionalidades han sido comprobadas y aprobadas:

- Módulo de autenticación y autorización
- Módulo de gestión de pacientes.
- Módulo de gestión de citas.
- Módulo de gestión de historias clínicas.
- Módulo de gestión de evoluciones.
- Módulo de gestión de usuarios.
- Módulo de gestión de médicos.
- Módulo de gestión de consultorios.

En favor de lo expuesto, doy por concluido el proyecto y acepto el software desarrollado por la Srta. Thalía Zapata con C.I. 1721444980.

Dra. Estefanía Zapata
C.I. 1721444972
GERENTE GENERAL ANEZ.

Av. Orellana y Av. 6 de Diciembre – Edificio la Viña
Quito – Ecuador

CENTRO MÉDICO ANEZ

Capacitación

Quito, 8 de junio de 2021

Yo, Estefanía Zapata con C.I. 1721444972, en calidad de Gerente General del Centro médico ANEZ, informo que la Srta. Thalía Zapata con C.I. 1721444980 realizó la capacitación respectiva sobre el manejo del sistema web desarrollado, dirigido al personal del centro de salud, programa de capacitación llevado a cabo el día 8 de junio de 2021 de forma virtual.

Dra. Estefanía Zapata
C.I. 1721444972
GERENTE GENERAL ANEZ.

Av. Orellana y Av. 6 de Diciembre – Edificio la Viña

Quito – Ecuador

MANUAL DE USUARIO

Tabla de contenidos

Página de iniciar sesión ***3***

Página de inicio ***3***

CITAS ***4***

Página citas 4

Página agendar cita 4

Página detalle de cita 5

Página recordatorio de citas 6

PACIENTES ***6***

Página pacientes 6

Página crear paciente 6

Página detalle de paciente 7

HISTORIAS CLÍNICAS ***8***

Página historias clínicas 8

Página crear historia clínica 8

Página detalle de historia clínica 9

EVOLUCIONES ***10***

Página evoluciones 10

Página crear evolución 10

Página detalle de evolución 11

Página generar receta 12

Página generar certificado 12

CONSULTORIOS ***13***

Página consultorios 13

Página crear consultorio 14

Página detalle de consultorio 14

MÉDICOS ***15***

Página médicos 15

Página crear médico....15
Página detalle de médico....16
USUARIOS....***17***
Página usuarios....17
Página crear usuario....17
Página detalle de usuario....18
Página perfil....18
Página actualizar datos....19
Página cambiar contraseña....19

Página de iniciar sesión

Se presenta una página para iniciar sesión en la que el usuario debe ingresar su nombre de usuario, contraseña y presionar en el botón “Login”.

En caso de haber olvidado la contraseña se muestra la opción “¿Olvidó contraseña?”, la cual al presionarla dirige a una página (Figura 2) donde el usuario debe ingresar su nombre de usuario y presionar en “Restaurar contraseña” para restablecer su contraseña.

Figura 1. Página de iniciar sesión

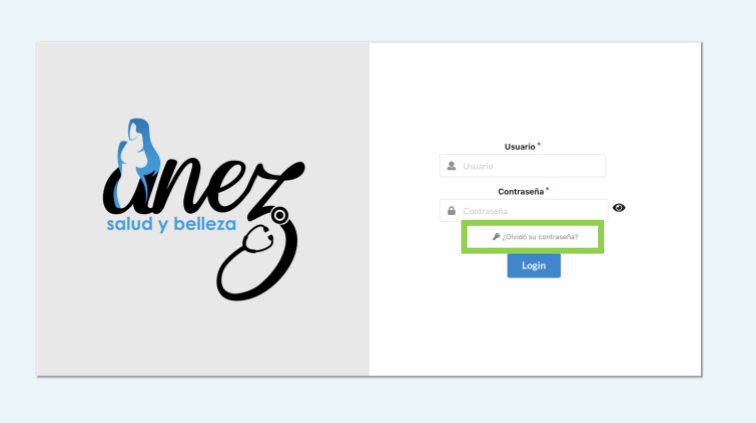


Figura 2. Página restablecer contraseña

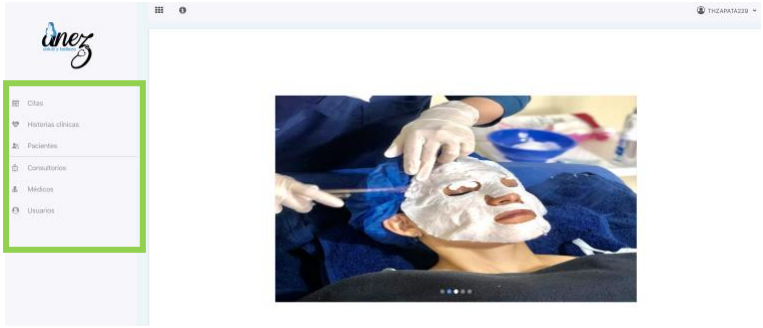


Página de inicio

Una vez iniciada sesión se muestra la pantalla de la Figura 2, dependiendo del rol del usuario se despliegan las siguientes opciones:

- *Administrador:* acceso a Citas, Historias clínicas, Pacientes, Consultorios, Médicos y Usuarios.
- *Médico:* acceso a Citas, Historias clínicas y Pacientes.
- *Recepcionista:* acceso a Citas y Pacientes.

Figura 3. Página de inicio.

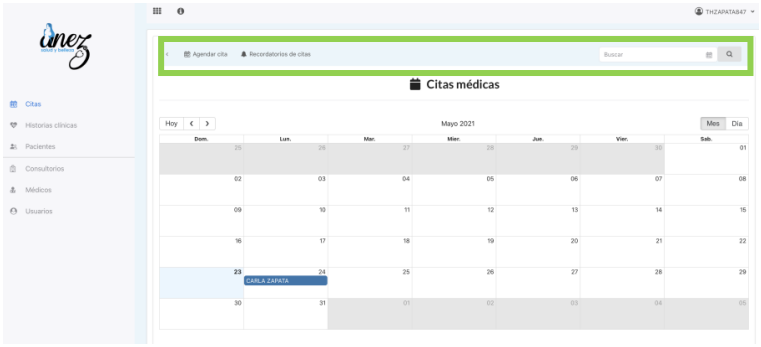


CITAS

Página citas

Al seleccionar "Citas" en el menú izquierdo, como se aprecia en la Figura 3, se presenta un calendario con las citas agendadas en el mes y en la barra superior las opciones "Agendar cita" y "Recordatorio de citas", adicionalmente se muestra la opción para buscar citas en un rango de fechas.

Figura 4. Página de citas.

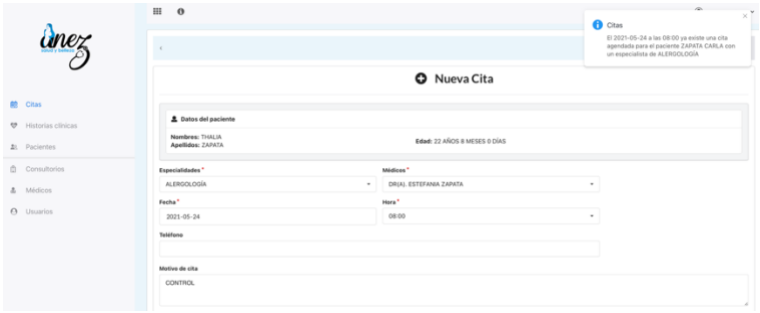


Página agendar cita

Al presionar "Agendar cita" se muestra un modal, el cual permite crear un paciente o seleccionar uno ya existente, una vez seleccionado el paciente desde la página "Pacientes", se presentará un formulario para seleccionar la fecha, hora y demás campos necesarios para el agendamiento de la cita, una vez completado correctamente el formulario se requiere presionar en "Guardar".

En caso de existir un cruce de horarios con otras citas se muestra un mensaje informativo.

Figura 5. Página agendar cita.



Página detalle de cita

Al presionar sobre una cita en el calendario (Figura 4), se despliega una pantalla con los datos del paciente, de la cita agendada y en la barra superior las opciones “Editar” la cual al presionarla presenta una página similar a la Figura 5 y “Eliminar” la misma que al presionarla muestra un modal (Figura 7) para confirmar la eliminación de la cita.

Figura 6. Página detalle de cita.

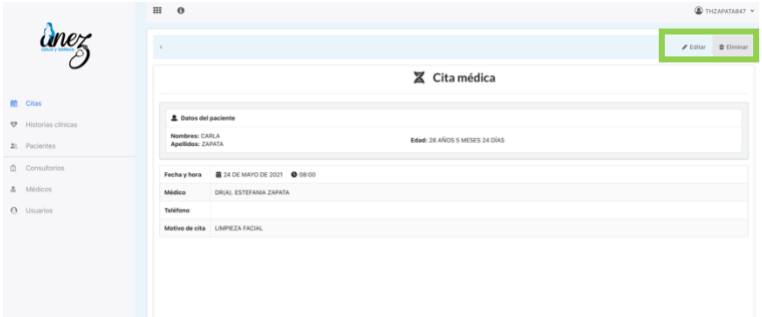
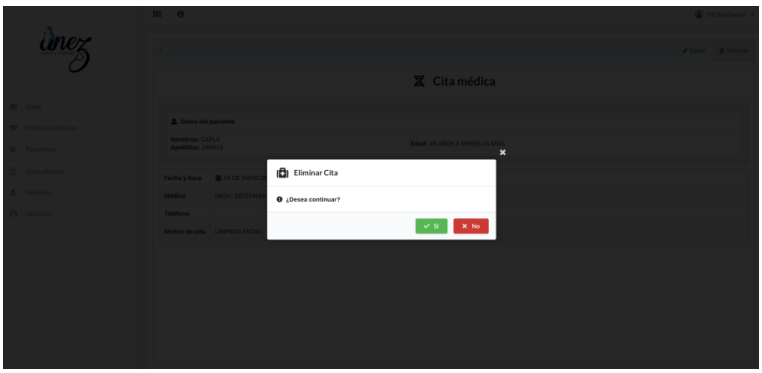


Figura 7. Modal eliminar cita.



Página recordatorio de citas

Al presionar en el botón “Recordatorio de citas” (Figura 4), se despliega una página con las próximas citas agendadas, en esta el usuario puede seleccionar todas o una cita en específico y pulsar el botón “Enviar notificaciones” para enviar un correo electrónico recordando al paciente la fecha y hora de su cita médica.

Figura 8. Página recordatorio de citas.

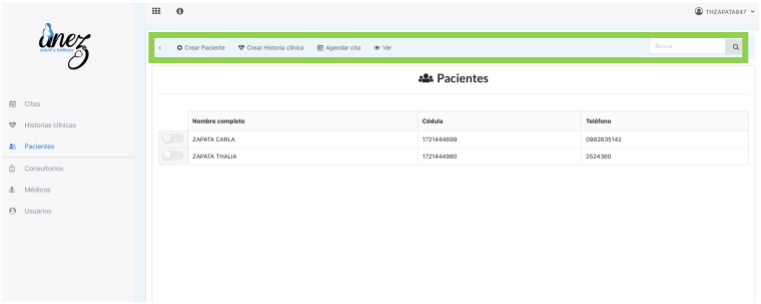


PACIENTES

Página pacientes

Al presionar en “Pacientes” en el menú izquierdo, se despliega una pantalla con todos los pacientes registrados y en la parte superior una barra con las opciones para “Crear Paciente”, “Crear Historia clínica”, “Agendar cita”, “Ver” y realizar búsquedas de pacientes por su nombre, apellido o número de cédula.

Figura 9. Página pacientes.

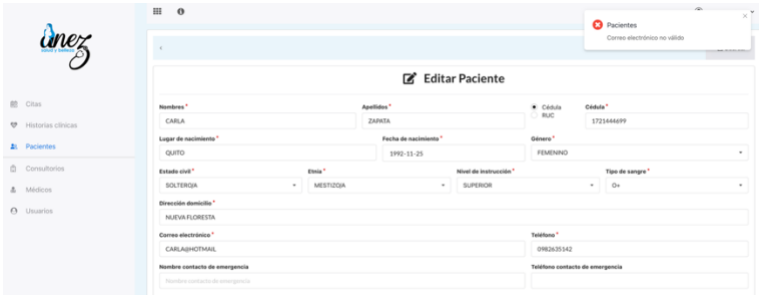


Página crear paciente

Al presionar el botón “Crear Paciente”, se presenta un formulario (Figura 10) para llenar los datos personales del paciente, una vez completados los campos requeridos se debe presionar en el botón “Guardar” ubicado en la barra superior.

En caso de no completar correctamente el formulario, se mostrarán mensajes de error.

Figura 10. Página crear paciente.



Página detalle de paciente

Al seleccionar un paciente y presionar en la opción “Ver” (Figura 9), se mostrará una pantalla como en la Figura 11 con los datos del paciente y las opciones “Editar” y “Eliminar”, esta última no disponible para usuarios con el rol de “Recepcionista”.

En caso de presionar “Editar” se mostrará un formulario similar a la Figura 10 y en caso de presionar “Eliminar” se mostrará un modal como en la Figura 12.

Figura 11. Página detalle de paciente.

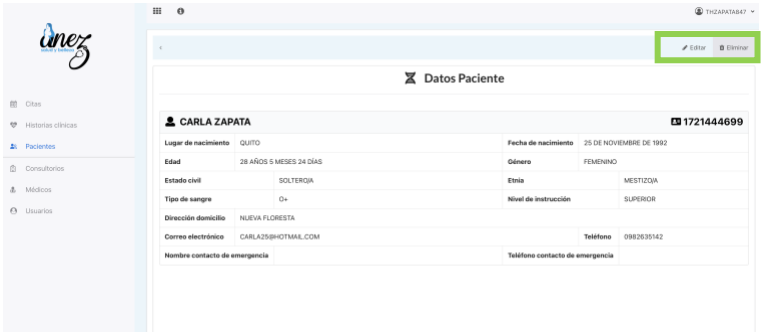
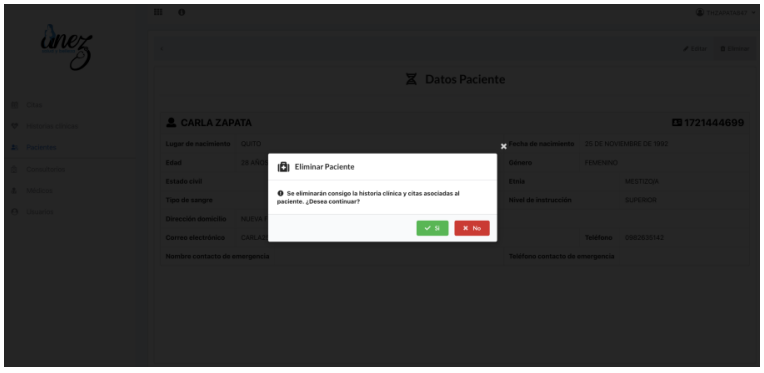


Figura 12. Modal eliminar paciente.



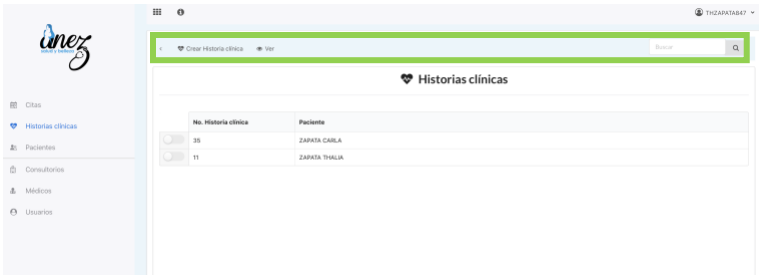
HISTORIAS CLÍNICAS

Página historias clínicas

Al presionar “Historias Clínicas” desde el menú lateral izquierdo, se despliegan todas las historias clínicas ingresadas y en la parte superior una barra con la opción “Crear Historia clínica” la cual al presionarla muestra un modal que permite crear un paciente o buscar uno ya existente, una vez seleccionado el paciente es posible proceder a la creación de su historia clínica y la opción “Ver” para mostrar a detalle la historia clínica seleccionada.

Asimismo, en la barra superior se encuentra un cuadro de texto para realizar la búsqueda de historias clínicas por el nombre, apellido o número de cédula del paciente.

Figura 13. Página historias clínicas

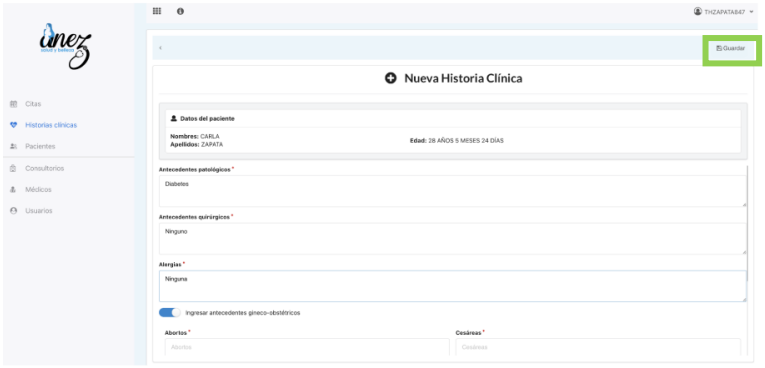


Página crear historia clínica

Una vez seleccionado el paciente y presionado en “Crear Historia clínica”, se presentan los datos del paciente y un formulario para ingresar sus antecedentes patológicos, quirúrgicos, gineco-obstétricos, entre otros.

Al completar el formulario, se requiere presionar el botón “Guardar”.

Figura 14. Página crear historia clínica.



Página detalle de historia clínica

Al presionar en la opción "Ver" (Figura 13), se despliega una página con los datos del paciente e historia clínica, en el lado izquierdo de la barra superior las opciones "Crear evolución" y "Ver evoluciones" y en el lado derecho las opciones "Editar" y "Eliminar"

Al presionar en "Editar" se mostrará un formulario similar al de la Figura 14, en donde el usuario podrá modificar los datos deseados y al presionar en "Eliminar" se presentará un modal como en la Figura 16.

Figura 15. Detalle de historia clínica.

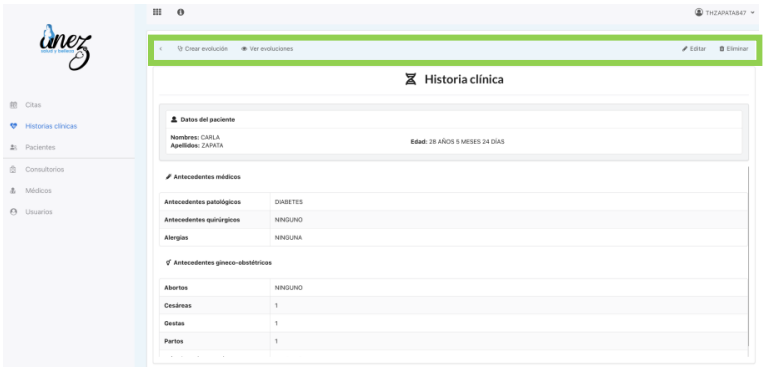
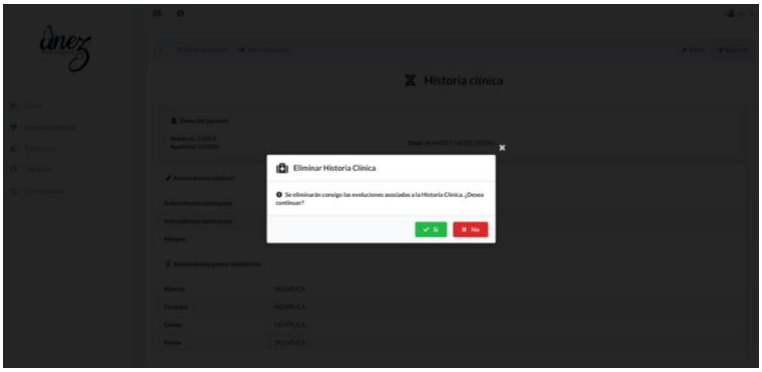


Figura 16. Modal eliminar historia clínica.



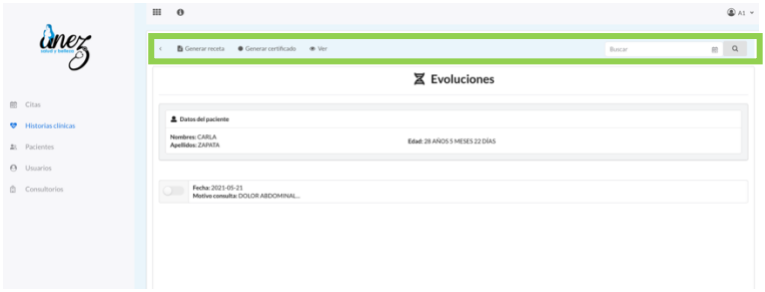
EVOLUCIONES

Página evoluciones

Al presionar la opción “Ver evoluciones” (Figura 15) se presenta una página con todas las evoluciones registradas en orden cronológico descendente.

En la parte superior se muestran las opciones “Generar receta”, “Generar certificado” y en el lado derecho la posibilidad de buscar evoluciones por fecha.

Figura 17. Página evoluciones.

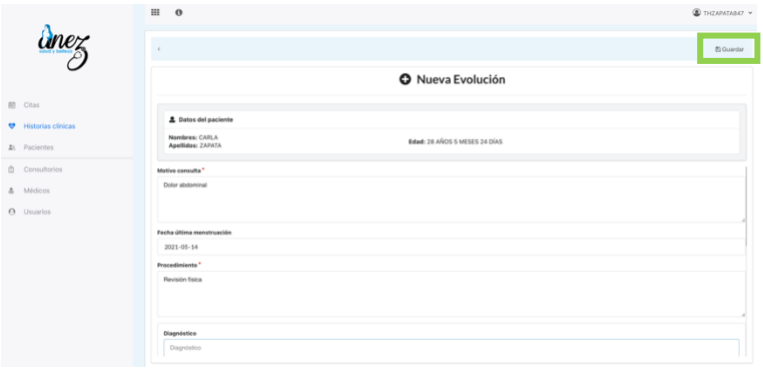


Página crear evolución

Se accede a esta página presionando en el botón “Crear evolución” (Figura 15), en esta se presentan los datos del paciente y un formulario para ingresar y seleccionar datos del procedimiento, diagnóstico, medicación, indicaciones y fotos en caso de ser necesario.

Al llenarse todos los campos requeridos se requiere presionar en el botón “Guardar”.

Figura 18. Página crear evolución.



Página detalle de evolución

Una vez seleccionada la evolución y presionado en el botón "Ver" (Figura 17), se muestran los datos del paciente, evolución y las opciones para "Editar" y Eliminar".

Al presionar en el botón "Editar" se presenta una pantalla similar a la Figura 18 para modificar los datos que se requieran y al presionar en "Eliminar" se despliega un modal para confirmar la eliminación de la evolución (Figura 20).

Figura 19. Página detalle de evolución.

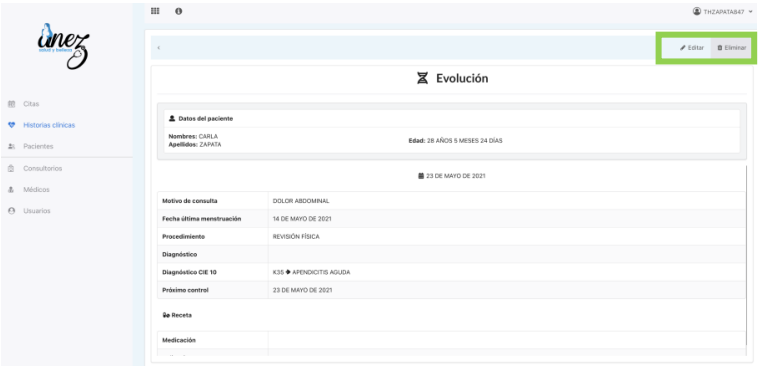
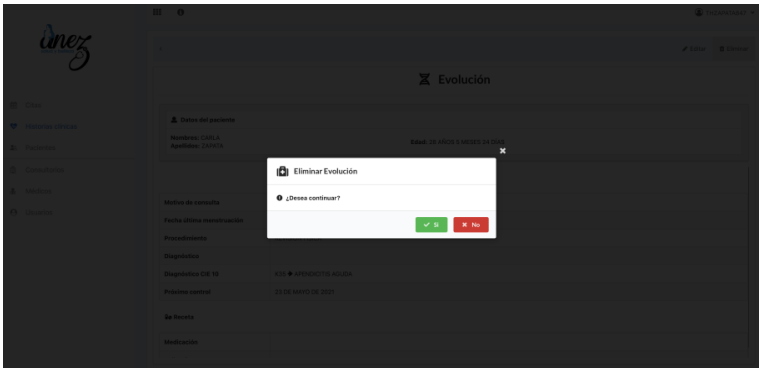


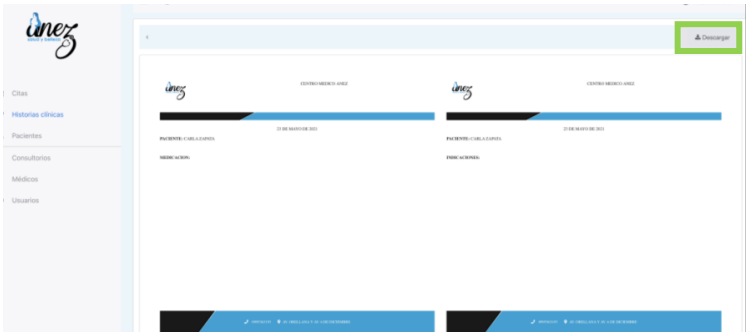
Figura 20. Modal eliminar evolución.



Página generar receta

Una vez seleccionada la evolución y al presionar en el botón "Generar receta" (Figura 17) se puede previsualizar la receta y descargarla desde el botón "Descargar" ubicado en la barra superior.

Figura 21. Página generar receta.



Página generar certificado

Una vez seleccionada la evolución y presionar el botón "Generar certificado" (Figura 17) se presenta un página con dos opciones, la primera para seleccionar los días de reposo, fechas de ingreso y egreso (Figura 22), y la segunda para previsualizar y descargar el certificado médico (Figura 23).

Figura 22. Página generar certificado (Primera parte).

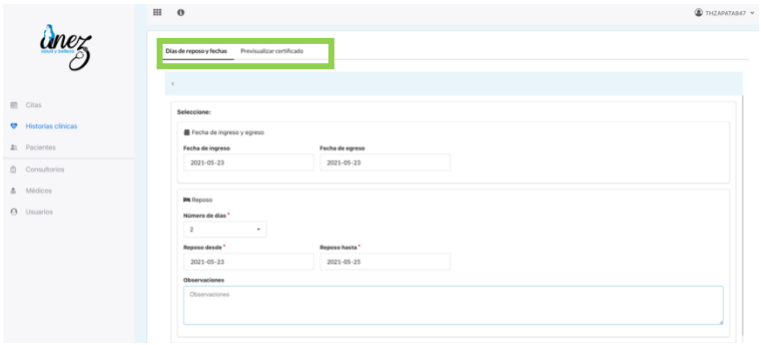
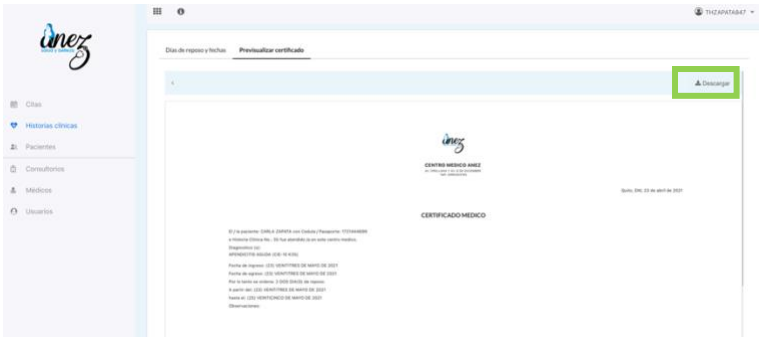


Figura 23. Página generar certificado (Segunda parte).



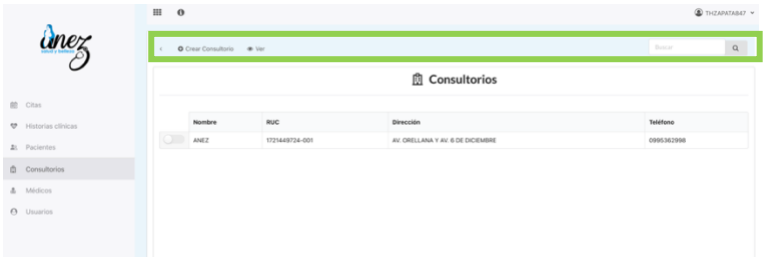
CONSULTORIOS

Página consultorios

Al seleccionar “Consultorios” en el menú izquierdo, se despliega una página con todos los consultorios registrados y en la parte superior las opciones para “Crear consultorio” y “Ver”.

Adicionalmente, en la barra superior se muestra un cuadro de texto que permite buscar consultorios por su nombre o RUC.

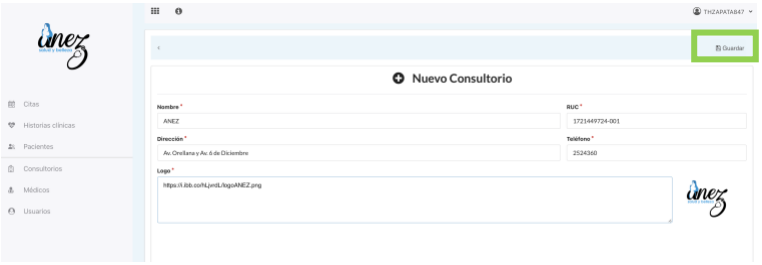
Figura 24. Página consultorios.



Página crear consultorio

Al presionar en “Crear consultorio” (Figura 26), se muestra un formulario para llenar los datos principales del consultorio, una vez completo el formulario se requiere presionar en el botón “Guardar” ubicado en la parte superior.

Figura 25. Página crear consultorio.



Página detalle de consultorio

Al seleccionar un consultorio y presionar en “Ver” (Figura 26), se despliega una página con los datos ingresados del centro médico y en la barra superior las opciones para “Editar” y “Eliminar”.

En la opción “Editar” se presenta un formulario similar a la Figura 25, en el cual es posible modificar los datos ingresado y en la opción “Eliminar” un modal para confirmar la eliminación (Figura 27).

Figura 26. Detalle de consultorio.

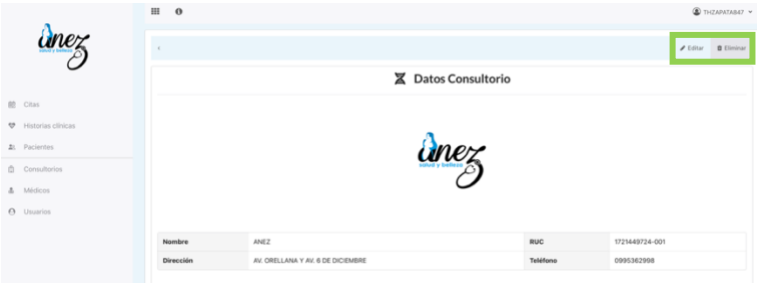
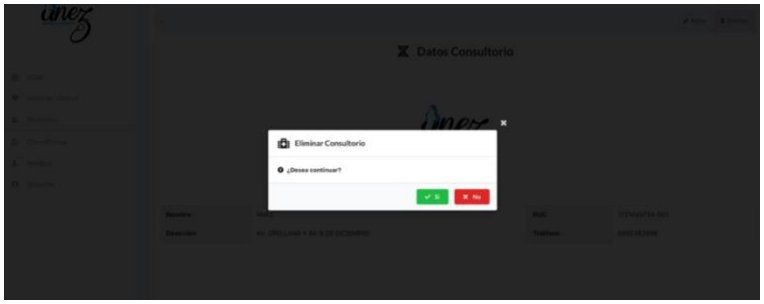


Figura 27. Modal eliminar consultorio.

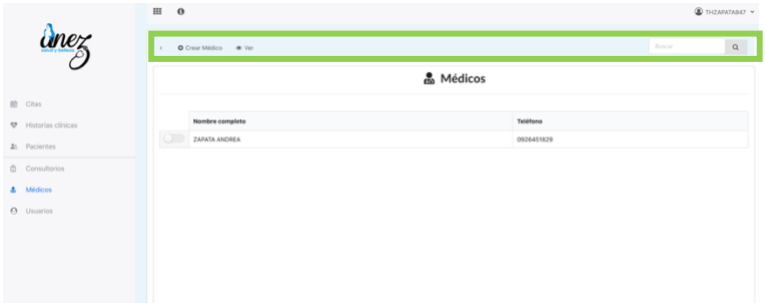


MÉDICOS

Página médicos

Al presionar en "Médicos" desde el menú lateral izquierdo, se presenta una página con el listado de médicos disponibles y en la barra superior, lado izquierdo las opciones "Crear médico", "Ver" y en el lado derecho un cuadro de texto para la búsqueda de médicos por su nombre, apellido o cédula del médico.

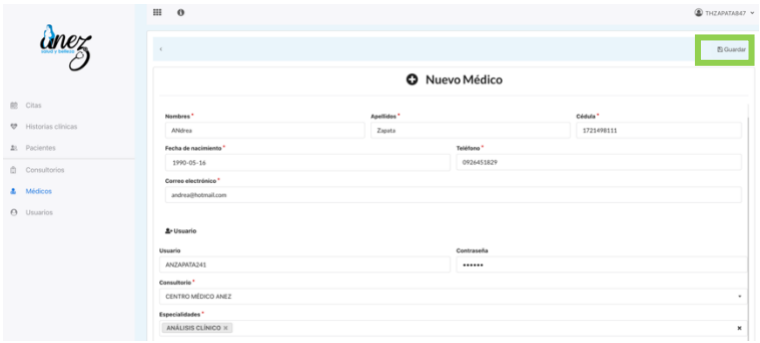
Figura 28. Página médicos.



Página crear médico

Al presionar en "Crear médico" (Figura 28), se despliega un formulario para el ingreso de los datos personales del médico, seleccionar sus especialidades y generar su nombre de usuario y contraseña, los mismo que son enviados al correo electrónico registrado.

Figura 29. Crear médico.



Página detalle de médico

Al seleccionar un médico y presionar en "Ver" (Figura 28), se muestran los datos del médico y en la barra superior la opción para "Editar", la cual al presionarla despliega un formulario similar al de la Figura 29 y la opción "Eliminar" muestra un modal (Figura 31).

Figura 30. Página detalle de médico.

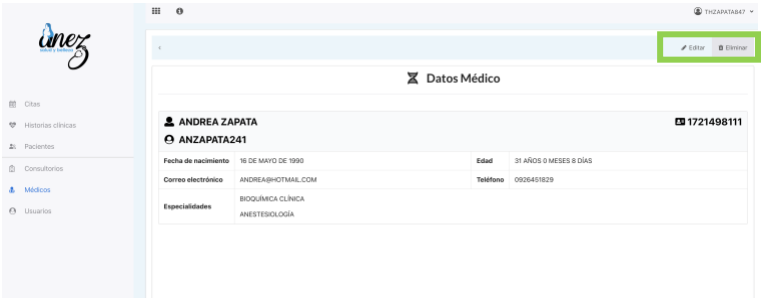
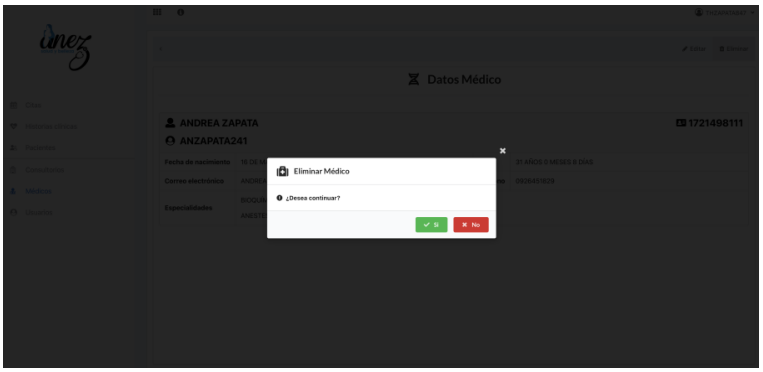


Figura 31. Modal eliminar médico.

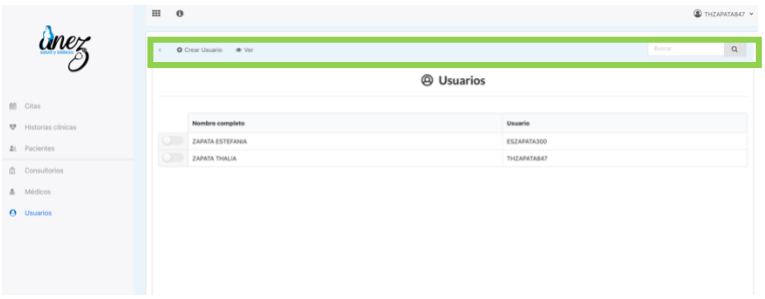


USUARIOS

Página usuarios

Al presionar en “Usuario” desde el menú izquierdo, se presenta una página con todos los usuarios ingresados y en la parte superior las opciones de “Crear usuario”, “Ver” y para un cuadro de texto para buscar usuarios por su nombre, apellido y nombre de usuario.

Figura 32. Página usuarios.

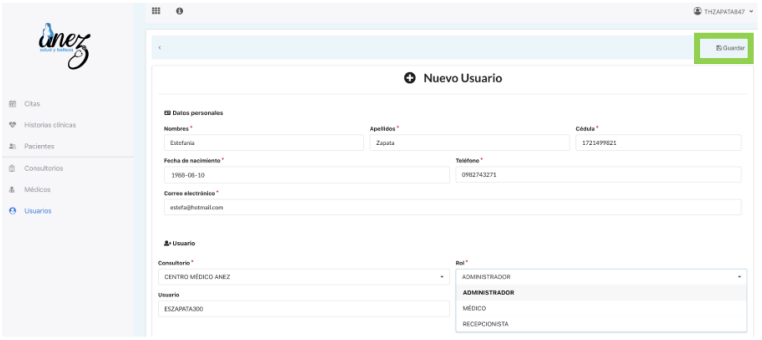


Página crear usuario

Al presionar en “Crear usuario” (Figura 32), se presenta un formulario para el ingreso de los datos personales del usuario, seleccionar su rol, consultorio al que pertenece y generar su nombre de usuario y contraseña, mismos que son enviados a su correo electrónico.

Una vez completo el formulario se requiere presionar en el botón “Guardar”.

Figura 33. Página crear usuario.



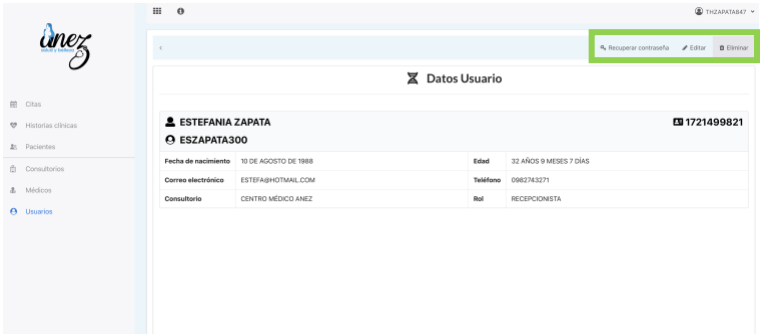
Página detalle de usuario

Al seleccionar un usuario y presionar en “Ver”, se despliega una página con los datos del usuario y en la barra superior las opciones “Recuperar contraseña”, “Editar” y “Eliminar”.

Al presionar en:

- “Recuperar contraseña” se genera y envía una nueva contraseña al correo electrónico del usuario.
- “Editar” se presenta un formulario similar al de la Figura 33, en el cual es posible modificar los datos del usuario
- “Eliminar” se muestra un modal para confirmar la eliminación del usuario.

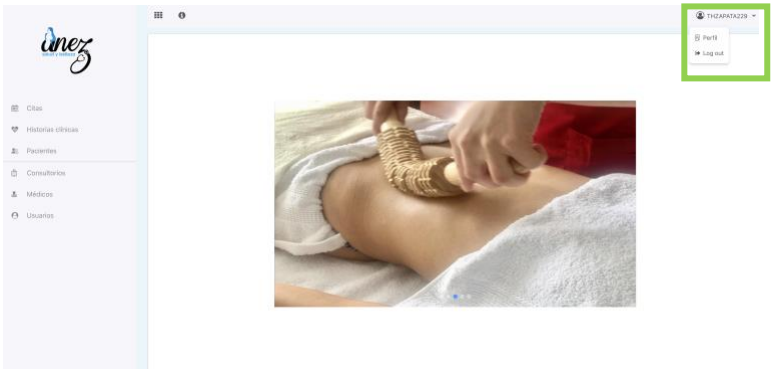
Figura 34. Página detalle de usuario.



Página perfil

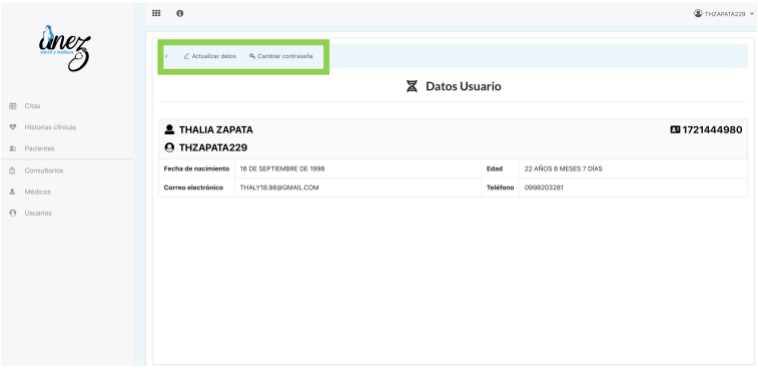
Al presionar sobre el nombre de usuario ubicado en la parte superior derecha, se despliegan las opciones “Perfil” y “Logout”, la primera opción muestra una página como en la Figura 36 y la segunda permite cerrar sesión y direcciona a la página de iniciar sesión (Figura 1).

Figura 35. Acceder a perfil.



Al presionar en “Perfil” se muestran los datos del usuario y en la barra superior las opciones “Actualizar datos” y “Cambiar contraseña”.

Figura 36. Página perfil.

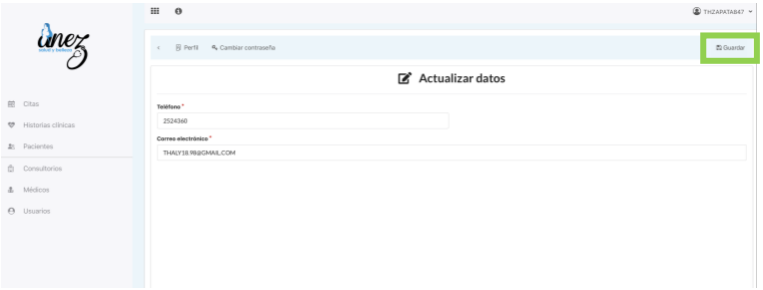


Página actualizar datos

Al presionar el botón “Actualizar datos” (Figura 36), se muestra una página donde el usuario puede modificar su correo electrónico o número de teléfono.

Una vez modificados los campos, se requiere presionar en “Guardar”.

Figura 37. Página actualizar datos.



Página cambiar contraseña.

Al presionar en “Cambiar contraseña” (Figura 36), se presenta un formulario, en el cual el usuario debe ingresar su contraseña actual y una nueva que cumpla con las reglas establecidas.

Una vez completados los 3 campos requeridos se presiona en “Guardar”.

Figura 38. Página cambiar contraseña.

