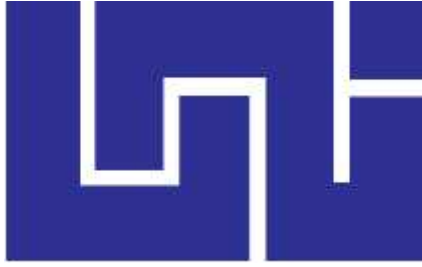


# UNIVERSIDAD NACIONAL DE INGENIERÍA



**Facultad de Electrotecnia y Computación  
(UNI-RUSB)**

## **TRABAJO MONOGRÁFICO**

Sistema web para la gestión de información médica de la “Clínica Pediátrica y Farmacia Cristina” (SIM-Cristina).

### **AUTORES:**

- Br. Cristhofer Enrique Espinoza Jiménez **Carné:** 2015-0109U
- Br. José Andrés Reyes Aguirre **Carné:** 2015-0522U

### **TUTOR:**

MSc. Ing. Adilson González López.

Managua, 13 de enero del 2023



## **Agradecimientos**

Agradecemos a Dios por la vida, la salud, sabiduría y por habernos acompañado y guiado a lo largo de nuestra carrera, por ser nuestra fortaleza en momentos de debilidad y por brindarnos una vida llena de aprendizajes y experiencias.

A nuestras familias que estuvieron ahí en nuestras noches de desvelo y dedicación para la culminación de este enriquecedor camino, que no termina aquí, sino que emerge una bifurcación nueva.

A nuestro tutor Adilson González por habernos brindado su apoyo incondicional al inicio de esta jornada, proveyendo excepcional retroalimentación para el desarrollo de este proyecto.

Al Doctor Carlos Gutiérrez por su colaboración con la realización de este proyecto y sus explicaciones en áreas que desconocíamos en el campo de las ciencias médicas.

A todos los maestros de la Universidad Nacional de Ingeniería que estuvieron involucrados en brindarnos la enseñanza requerida y adecuada a lo largo de estos 5 años de formación profesional y también por habernos brindado su amistad.

# Dedicatoria

## **A Dios:**

Porque nos da la oportunidad de existir, quien nos regala el don del entendimiento, inteligencia, sabiduría, fortaleza, voluntad, deseo de superación y en los momentos más difíciles nos ha enseñado a valorarnos cada día más para seguir formándonos como personas de éxitos.

## **A Nuestros Padres:**

Por ser los pilares más importantes de nuestras vidas, que día a día nos demuestran su amor con cariño y apoyo para seguir forjándonos en este camino y alcanzar nuestras metas anheladas y por el apoyo demostrado en todo momento

## **A Nuestros Familiares:**

Porque siempre contamos con ellos para brindarnos su confianza, cariño, respeto y apoyo moral.

## **A Nosotros Mismos:**

Porque realizamos este trabajo monográfico con el interés personal de superar las tareas difíciles que luego nos servirán de ejemplo y de guía, para superar otras más difíciles.

## Resumen

El presente trabajo monográfico titulado *Sistema web para la gestión de información médica de la “Clínica Pediátrica y Farmacia Cristina” (SIM-Cristina)* tiene como finalidad describir el desarrollo de un sistema web para el control y manejo de los historiales clínicos, citas médicas y control de pagos, dicho sistema se elaboró aplicando la metodología de desarrollo UWE.

El objetivo general es demostrar el desarrollo del sistema web para la gestión de información de historia clínica, control de citas y control de pagos de servicios de la Clínica Pediátrica y Farmacia Cristina.

El desarrollo del sistema inició con la recopilación de los requerimientos en base a las entrevistas realizadas con el personal de la clínica, obteniendo información de los procesos y tareas que se realizan en el día a día de la clínica. Se diseñó el sistema web, se realizaron pruebas y se validó el desempeño y facilidad de uso en comparación de los procesos que actualmente se realizan; y el cumplimiento de los objetivos establecidos.

## Índice

1	Introducción .....	1
2	Antecedentes .....	2
3	Justificación. ....	4
4	Objetivos .....	5
4.1	Objetivo general. ....	5
4.2	Objetivos específicos.....	5
5	Marco teórico. ....	6
5.1	Historial clínico. ....	6
5.2	Control de pagos. ....	6
5.3	Citas médicas .....	7
5.4	Farmacia.....	7
5.5	Sistema informático. ....	7
5.6	Base de dato. ....	7
5.7	UML.....	8
5.8	Ciclo de vida .....	8
5.9	Aplicación web.....	9
5.10	Arquitectura MVC .....	10
5.11	Modelo relacional .....	10
5.12	Herramientas de desarrollo. ....	11
5.12.1	Framework de Desarrollo Ruby on Rails.....	11
5.12.2	Lenguaje de programación Ruby .....	11
5.12.3	Librería de Ruby Devise (GEMA).....	12
5.12.4	Librería de Ruby CanCan (GEMA).....	12
5.12.5	Gestor de base de datos MySQL .....	13
5.12.6	Lenguaje de Marcas de Hipertexto HTML versión 5 .....	13
5.12.7	Hojas de estilo en cascada CSS .....	14
5.12.8	Biblioteca de JavaScript JQuery .....	14
5.12.9	Framework de desarrollo Bootstrap 4 .....	14
5.13	Metodología.....	14
6	Factibilidad del desarrollo. ....	17
6.1	Factibilidad técnica. ....	17
6.2	Factibilidad económica. ....	19

6.2.1	Costos de hardware. ....	19
6.2.2	Costos de software. ....	20
6.2.3	Costos de desarrollo. ....	21
6.2.4	Costos totales. ....	21
6.3	Factibilidad operativa. ....	22
7	Resultado del desarrollo. ....	22
7.1	Modelo de prototipo. ....	23
7.2	Fases de Desarrollo UWE. ....	24
7.2.1	Fase 1: Captura, análisis y especificación de requerimientos. ....	24
7.2.2	Fase 2: Diseño del sistema. ....	29
7.2.3	Fase 3: Codificación del software ....	64
7.2.4	Fase 4: Pruebas. ....	76
7.2.5	Fase 5: Implementación y despliegue. ....	83
7.3	Desempeño y facilidad de uso del sistema. ....	89
7.3.1	Desempeño. ....	89
7.3.2	Facilidad de uso. ....	90
8	Conclusiones. ....	93
9	Recomendaciones. ....	94
10	Bibliografía ....	95
11	Anexos. ....	97
11.1	Manual técnico ....	97
11.2	Manual de usuario ....	97
11.3	Cartas de aceptación. ....	97

**Indicie de Figura.**

Figura 1: Fases o etapas del ciclo de vida del software.....15

Figura 2: Ciclo del modelo de prototipos.....23

Figura 3: Diagrama de caso de uso login.....31

Figura 4: Diagrama de caso de uso historial clínico.....32

Figura 5: Diagrama de caso de uso citas médicas. ....33

Figura 6: Diagrama de caso de uso seguimiento.....34

Figura 7: Diagrama de caso de uso control de pago. ....36

Figura 8: Diagrama de caso de uso control de pago exámenes y servicios. ....37

Figura 9: Diagrama de caso de uso catalogo - producto. ....38

Figura 10: Diagrama de caso de uso catalogo - servicios.....39

Figura 11: Diagrama de caso de uso catalogo - exámenes.....40

Figura 12: Diagrama de caso de uso reporteria.....41

Figura 13: Diagrama de navegación del proceso para crear un historial clínico. ....42

Figura 14: Diagrama de navegación del proceso para crear una cita médica. ....43

Figura 15: Diagrama de navegación agregar productos o servicios a catálogos. ....44

Figura 16: Diagrama de navegación control de pagos de productos o servicios. ....45

Figura 17: Diagrama de navegación para la visualización de logs. ....46

Figura 18: Diagrama de navegación para configurar datos de la empresa.....47

Figura 19: Autenticación de usuario.....48

Figura 20: Crear nuevo paciente.....49

Figura 21: Seguimiento paciente. ....50

Figura 22: Agendar cita.....51

Figura 23: Agregar registro en catálogo.....52

Figura 24: Crear factura.....53

Figura 25: Diagrama de clase .....55

Figura 26: Vista de inicio de sesión. ....56

Figura 27: Vista del menú principal.....57

Figura 28: Historial clínico.....57

Figura 29: Formato historial clínico. ....58

Figura 30: Agenda de citas .....59

Figura 31: Control de pago .....59

Figura 32: Productos.....60



Figura 33: Roles.....	60
Figura 34: Creación de roles.....	61
Figura 35: Lista de usuarios.....	61
Figura 36: Lista de citas registradas .....	62
Figura 37: Lista de logs.....	62
Figura 38: Modelo arquitectónico .....	63
Figura 39: Descargar Visual Studio Code. ....	64
Figura 40: Creación del directorio con el nombre del proyecto. ....	64
Figura 41: Creación del repositorio. ....	65
Figura 42: Dirección del repositorio creado.....	65
Figura 43: Clonación del proyecto. ....	66
Figura 44: Repositorio de claves.....	66
Figura 45: Descarga del paquete Ruby.....	66
Figura 46: Instalación de RUBY ON RAILS 2.7.2. ....	66
Figura 47: Instalación del paquete build essential. ....	66
Figura 48: Instalación del bundle. ....	66
Figura 49: Abriendo la ruta del proyecto en Visual Studio Code.....	67
Figura 50: Estructura del proyecto. ....	67
Figura 51: Modelo entidad Relación. ....	69
Figura 52: Flujo de acceso a la web con Devise gema. ....	71
Figura 53: Seguridad de en la conexión de la DB usando Devise gema. ....	72
Figura 54: Código de se Session Expiry aplicado al código.....	72
Figura 55: Uso de CAN CAN. ....	73
Figura 56: Instancia de la Base de datos. ....	74
Figura 57: Vista de la administración de las copias de seguridad de la DB. ....	74
Figura 58: Copias de seguridad automatizadas activas.....	75
Figura 59: Esquema básico del ciclo de prototipado.....	76
Figura 60: Control de versiones - Cambios realizados en los prototipos. ....	79
Figura 61: Suite de pruebas Unitarias con el comando rspec -f d.....	80
Figura 62: Test del login.....	80
Figura 63: Test paciente. ....	81
Figura 64: Test agenda. ....	81
Figura 65: Test Catalogo producto.....	82

Figura 66: Test Catalogo servicio. ....	82
Figura 67: Test catalogo examen.....	82
Figura 68: Test control de pago. ....	83
Figura 69: Inicio de sesión AWS. ....	84
Figura 70: Selección de Easy create. ....	84
Figura 71: Configuración de la instancia de la DB. ....	85
Figura 72: Instancia de la DB en la lista.....	85
Figura 73: Comando para crear a heroku. ....	86
Figura 74: Comando para implementar el código en heroku. ....	86
Figura 75: Comando para iniciar heroku. ....	86
Figura 76: Despliegue de heroku. ....	87
Figura 77: Generación de la URL para la aplicación.....	87
Figura 78: Sistema desplegado. ....	88
Figura 79: Home page de la aplicación.....	88
Figura 80: Carta de aceptación del desarrollo. ....	97
Figura 81: Carta de aceptación del software.....	98
Figura 82: Formato de historial clínico ....	103
Figura 83: Formato de registro y control de citas. ....	104
Figura 84: Formato de facturación de medicamentos. ....	105
Figura 85: Formato de comprobante de examen de laboratorio. ....	106
Figura 86: Línea de tiempo del proyecto.....	107

## Índice de tablas

Tabla 1: Hardware de la clínica.....	17
Tabla 2: requisitos mínimos del software .....	18
Tabla 3: Requisitos óptimos del software.....	19
Tabla 4: Costos de hardware. ....	20
Tabla 5: Costos de software. ....	20
Tabla 6: Costos de desarrollo. ....	21
Tabla 7: Costos totales del proyecto.....	21
Tabla 8: actores del sistema. ....	24
Tabla 9: Requerimientos funcionales RF-01.....	25
Tabla 10: Requerimientos funcionales RF-02.....	25
Tabla 11: Requerimientos funcionales RF -03.....	25
Tabla 12: Requerimientos funcionales RF -04.....	26
Tabla 13: Requerimientos funcionales RF -05.....	26
Tabla 14: Requerimientos funcionales RF -06.....	26
Tabla 15: Requerimientos funcionales RF -07.....	26
Tabla 16: Requerimientos funcionales RF -08.....	26
Tabla 17: Requerimiento no funcional RNF-01.....	27
Tabla 18: Requerimiento no funcional RNF-02.....	27
Tabla 19: Requerimiento no funcional RNF-03.....	27
Tabla 20: Requerimiento no funcional RNF-04.....	27
Tabla 21: Requerimiento no funcional RNF-05.....	28
Tabla 22: Requerimiento no funcional RNF-06.....	28
Tabla 23: Diagrama de caso de uso login.....	31
Tabla 24: Caso de uso historial clínico.....	32
Tabla 25: Caso de uso citas médicas. ....	33
Tabla 26: Diagrama de caso de uso seguimiento. ....	35
Tabla 27: Caso de uso control de pago. ....	36
Tabla 28: Caso de uso control de pago exámenes y servicios. ....	37
Tabla 29: Caso de uso catalogo - producto. ....	38
Tabla 30: Caso de uso catalogo - servicio. ....	39
Tabla 31: Caso de uso control de pago catalogo - exámenes. ....	40
Tabla 32: Caso de uso reportearía. ....	41

Tabla 33: Resultados de evaluación de la razón de rendimiento .....91

Tabla 34: Resultados de evaluación de la funcionalidad de uso.....92

Tabla 35: Ficha de registro de prueba para proceso de historial clínico. ....99

Tabla 36: Ficha de registro de prueba para proceso de registro de citas. .... 100

Tabla 37: Ficha de registro de prueba para proceso de control de pago de productos. ..101

Tabla 38: Glosario de términos. .... 108

## 1 Introducción

El presente documento describe el desarrollo del Sistema web para la *Clínica Pediátrica y Farmacia Cristina*, que en lo sucesivo se referirá como SIM-Cristina, comprende la gestión de información para la clínica, organizada en cuatro módulos:

- ) Historial clínico.
- ) Control de citas.
- ) Control de pagos.
- ) Administración.

En el primer módulo que es el historial clínico se registran los datos relacionados con el seguimiento, la salud del paciente, atención y servicios prestados (consultas médicas, exámenes de laboratorio y tratamientos).

El módulo de control de citas, es el que se encarga de registrar la programación de las distintas atenciones que solicita el paciente, con el fin de mantener un orden en la atención del paciente.

El tercer módulo del sistema es referente al control de pagos, el cual se encarga del control de pago de los servicios y atenciones (Consultas médicas, exámenes de laboratorios y ventas de medicamentos). De igual manera se cuenta con el módulo de administración que permite la creación de roles de usuario, administración de permisos de usuarios, de esta manera se tiene un control de los permisos en cuanto a quienes tienen acceso y manipulan la información clasificada.

EL desarrollo del sistema web SIM-Cristina se describe en este documento de la siguiente manera:

- ) Antecedentes: En esta sección se describe la forma en que la *Clínica Pediátrica y Farmacia Cristina* trabaja actualmente.
- ) Justificación: Se especifican las razones por las que se propone la tesis y principales aportes del desarrollo para los procesos involucrados.
- ) Objetivos: Se describen los objetivos a seguir para el desarrollo completo del sistema web SIM-Cristina.
- ) Marco teórico: En este se describe el fundamento teórico utilizado para el planteamiento y desarrollo de la tesis en sus distintas etapas.
- ) Metodología: En esta sección se indica la metodología de desarrollo de software que se utilizó.

## 2 Antecedentes

La *Clínica Pediátrica y Farmacia Cristina* del Dr. Carlos Gutiérrez se encuentra ubicada de la cruz roja dos cuadras y media al oeste, barrio países bajos en la ciudad de Masaya.

En la clínica se lleva el control y manejo del historial clínico de cada paciente mediante un documento de texto en Microsoft Word (ver anexo 2), que contiene la siguiente información: datos generales, control de consultas, seguimiento de enfermedades (familiares, patológicas, hereditarias, actuales), antecedentes del parto, antecedentes posnatales, alimentación, vacunas, desarrollo psicomotor, examen físico, diagnósticos o problemas, los exámenes de laboratorio realizados y tratamiento o manejo médico.

Otra parte importante que se maneja es el registro y control de las citas, el manejo de estos datos se realiza en tablas de Word que detallan fecha y hora de próxima cita, nombre del paciente, diagnóstico y celular (ver anexo 3).

El control de pagos es uno de los proceso fundamentales en la *Clínica Pediátrica y Farmacia Cristina*, ya que este maneja el pago por consultas médicas, medicamentos y exámenes de laboratorio, de estos 3 el cobro de medicamentos en la farmacia es el único del cual se lleva registro en un documento de Excel (ver anexo 4), los demás son pagos directos al doctor sin generar ninguna evidencia y registro de los mismos, para el retiro de resultado de exámenes el paciente presenta un documento firmado y sellado por el doctor (ver anexo 5).

Como referencia al desarrollo de esta tesis encontramos la tesis llamada Desarrollo de un sistema web para la gestión de expedientes clínicos generados por las consultas ambulatorias de la Universidad Nacional Autónoma de Nicaragua (UNAN FAREM-Chontales) desarrollado en el año 2016, este comparte la característica del control y manejo del historial clínico. (Soto Flores, Pérez Duarte, & Rivera Martínez, 2016)

Otro trabajo a tomar en cuenta como referencia es el trabajo de fin de grado de la Universidad Carlos III de Madrid, desarrollada en el 2017, llamado Sistema Web para la gestión de una Clínica Médica el cual comparte tres características que se plantean en esta tesis las cuales son: el control y manejo del historial clínico, así como también registro y control de las citas. (Centenera, 2017).

De ambas referencias obtuvimos una premisa de como empezamos el desarrollo de nuestro proyecto, de la primera referencia se logra extraer un primer vistazo de cómo se manejan los expedientes de cada paciente, el cual nosotros llamamos historiales clínicos y la forma en cómo se es mostrado a los usuarios, de la segunda referencia se toma el control y organización de citase e historiales clínicos.

### 3 Justificación.

Tomando en cuenta el mecanismo actual de trabajo de la *Clínica Pediátrica y Farmacia Cristina*, descrito en la sección anterior, se propone el desarrollo de un sistema web que mejore la gestión de información relacionado a la historia clínica, planificación de citas y control de pagos.

El registro y control de la historia clínica es una tarea que requiere de mucho tiempo y esfuerzo, el sistema propuesto al que se ha denominado SIM-Cristina contempla un módulo el cual permite agilizar el registro y control de la historia clínica, garantizando una mejor organización, reduciendo tiempo y facilitando además la generación de reportes y estadísticas.

Con el módulo de control de citas de SIM-Cristina se garantiza un mejor registro de la información y que almacenada en el gestor de base de datos permite gestionar la información haciendo menos complejo la calendarización de citas, organización de citas, consultas de citas, recordatorios y generación de reportes.

El tercer módulo que se propone en SIM-Cristina es el control de todos los tipos de pagos; por consultas, exámenes y pago de medicamentos, que permite saber el ingreso total real por los servicios y atenciones brindados, el cual al no ser registrados en su totalidad limita a la generación de estadísticas y reportes que el sistema puede realizar.

En general la automatización de los procesos de la *Clínica Pediátrica y Farmacia Cristina* y el uso de medios de almacenamientos en bases de datos permite controlar mejor los datos y tener acceso a reportes gerenciales que faciliten la toma de decisiones.

Actualmente la clínica no cuenta con ninguna clase de seguridad en el almacenamiento y manejo de su información, lo cual se expone a individuos que no deberían tener acceso, debido a esta situación se implementa el módulo de administración que gestiona los roles de usuario y así brindar una mayor protección de los datos.



## 4 Objetivos.

### 4.1 Objetivo general.

Desarrollar un sistema web para la gestión de información de historia clínica, control de citas y control de pagos de servicios de la *Clínica Pediátrica y Farmacia Cristina*.

### 4.2 Objetivos específicos.

- ) Analizar los procesos de la *Clínica Pediátrica y Farmacia Cristina* el cual permita definir el alcance y los requerimientos funcionales y no funcionales del sistema.
- ) Diseñar el modelo del sistema a partir de los requerimientos obtenidos del análisis de los procesos, haciendo uso de las fases de desarrollo de la metodología UWE<sup>1</sup>.
- ) Codificar el sistema basado en los diseños obtenidos usando Ruby on Rails, Ruby, Devise (Gema), CanCan (Gema), MYSQL, HTML5, CSS, JQuery, Bootstrap4.
- ) Realizar pruebas de validación del sistema utilizando RSpec para retroalimenten las funcionalidades del sistema.
- ) Implementar el sistema web SIM-CRISTINA.

---

<sup>1</sup> UWE: Es una **metodología** basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web.

## 5 Marco teórico.

A continuación, se describe el fundamento teórico considerado para el desarrollo de SIM-Cristina en sus distintas etapas.

### 5.1 Historial clínico.

Es un documento, en el que se registran los antecedentes biológicos del individuo, es imprescindible para el control de la salud y el seguimiento de las enfermedades. Es fundamental en estudios epidemiológicos, es útil para la planificación educativa de las comunidades y de gran importancia médico-legal además de ser el punto de partida del razonamiento médico, es el instrumento utilizado por todos los profesionales de la salud y sirve de nexo entre las diferentes disciplinas (Schaposnik, 1977).

Dependiendo de la disciplina involucrada o del área del equipo de salud que se trate, la historia clínica podrá tener algunas características o matices especiales, Por ejemplo, la historia de una paciente tratada por un obstetra o la historia clínica de un pediatra, etc. Aunque los fundamentos de la historia clínica y su estructura sea siempre la misma.

### 5.2 Control de pagos.

El control de pago es una acción que se refiere a todos los actos relacionados con la elaboración, registro, envío y cobro de las facturas.

Una factura es un documento que refleja los datos de una transacción mercantil (venta de bienes o servicios) y contiene información como:

- ) Los datos relativos al emisor y receptor de la factura (nombre, NIF, dirección).
- ) La información sobre la transacción realizada (es el concepto de la factura).
- ) El importe facturado y el desglose por cada concepto.
- ) Los impuestos aplicables.
- ) El número de factura, la fecha y el lugar.
- ) El modo y plazo de pago.

El control de pago de un negocio engloba todas sus transacciones comerciales donde se incluyen las facturas de ventas y compras realizadas durante un período de tiempo establecido. (sage, 2020)

### 5.3 Citas médicas

Cita que un paciente tiene con un médico para tratarse de una dolencia o enfermedad. A instancias de la medicina, la palabra consulta presenta un uso híper popular ya que a través de la misma se denomina a aquella cita que un paciente o enfermo, realiza con un médico para tratar determinada afección o dolor que padece. (Ucha, 2013)

### 5.4 Farmacia

La farmacia es la disciplina que se encarga de la generación de productos que sirven como fármacos para tratar diversas dolencias. Por extensión se hace referencia a farmacia como el lugar físico en donde se comercializan dichos productos, según Oxford Languages “Ciencia y técnica de conocer las sustancias de acción terapéutica, de obtenerlas y combinarlas para preparar medicamentos. Establecimiento en el que se preparan y venden medicamentos”. (Languages, s.f.)

### 5.5 Sistema informático.

Un sistema informático es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático.

Un sistema informático se define como “un grupo de partes entrelazadas entre el hardware, software y recurso humano los mismos que permite realizar los registros y guardar toda la información que se haya procesado. El sistema informático forma parte del sistema de información; en este último la información y el acceso a la misma no está informatizada. Un ejemplo práctico sería un sistema de archivo de libros de una biblioteca su función general es la de servir como un sistema de información. Si dentro del mismo hay computadores que permiten organizar la biblioteca”. (Gallego, 2010)

### 5.6 Base de dato.

Se puede decir que una base de datos es un conjunto de archivos dedicados a guardar información relacionada entre sí, con referencia entre ellos de manera que se complementen y con la posibilidad de relacionarlos en base a diferentes criterios.

Según (Gómez, 2007) “una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso, es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real”. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción han de ser únicas estando almacenadas junto a los mismos.

Por otro lado (Juárez, 2006) dice que “una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización”.

## 5.7 UML

El lenguaje unificado de modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

UML es ante todo un lenguaje. “es un lenguaje de modelado visual que se usa para especificar, construir y documentar artefactos de un sistema de software. Captura de decisiones y conocimiento sobre los sistemas que se deben construir, está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios” (G. Booch, 1999).

## 5.8 Ciclo de vida

Según la norma 1074 IEEE<sup>2</sup> se define al ciclo de vida del software como “una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software” y la norma ISO<sup>3</sup> 12207 define como modelo de ciclo de vida al “marco de referencia, que contiene los procesos, las

---

<sup>2</sup> IEEE: El Instituto de Ingenieros Eléctricos y Electrónicos.

<sup>3</sup> ISO: Internacional Organization for Standardization (Organización Internacional de Normalización).

actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de requisitos hasta la finalización de su uso”. Ambas consideran una actividad como un subconjunto de tareas y una tarea como una acción que transforma las entradas en salidas. (Piattini, 1996).

## 5.9 Aplicación web

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

“En la actualidad todos somos usuarios de aplicaciones web. La mayoría de las veces que entramos a internet accedemos a través de una aplicación web; podemos decir que una aplicación web es aquella a la que accedemos a través de un navegador utilizando internet o una intranet” (Ramos Martín & Ramos Martín, 2014).

## 5.10 Arquitectura MVC

El patrón MVC. son las siglas de Model View Controller, en español Modelo Vista Controlador. Esto también se ve reflejado en que cada una de estas palabras representa cada uno de los 3 componentes del patrón MVC. Cada parte juega un rol fundamental para la completa integración del sistema.

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Alicante, 2012).

## 5.11 Modelo relacional

El modelo relacional se basa en el concepto matemático de relación, que gráficamente se representa mediante una tabla. Es decir, una relación es una tabla, con columnas y filas. En este modelo, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos.

Son los objetos principales sobre los que debe recogerse información y generalmente denotan personas, lugares, cosas o eventos de interés. Una relación tiene un nombre, un conjunto de atributos que representan sus propiedades y está formada por un conjunto de tuplas que incluyen los valores que cada uno de los atributos toma para cada una de las tuplas de la relación. Una relación se representa mediante una tabla bidimensional (las columnas representan los atributos y las filas representan las tuplas o registros) (Alcalde, 2017).

## 5.12 Herramientas de desarrollo.

### 5.12.1 Framework de Desarrollo Ruby on Rails.

Ruby on Rails, también conocido como RoR<sup>4</sup> o Rails es un framework<sup>5</sup> de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura MVC<sup>6</sup>. Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros framework y con un mínimo de configuración. El lenguaje de programación Ruby permite la meta programación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby. (RubyTeam, 2009)

### 5.12.2 Lenguaje de programación Ruby

Ruby es un lenguaje multiplataforma, interpretado y orientado a objetos. Ruby fue diseñado por Yukihiro Matsumoto ('Matz') en 1993. (Suarez, 2019).

Combina una sintaxis inspirada en Python, Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre. (RubyTeam, 2009)

---

<sup>4</sup> ROR: Ruby on Rails.

<sup>5</sup> Framework: un Framework es una estructura previa que se puede aprovechar para desarrollar un proyecto.

<sup>6</sup> MVC: Modelo Vista Controlador.

#### 5.12.3 Librería de Ruby Devise (GEMA)

Devise es una gema extraordinaria que nos ahorrará mucho trabajo. La utilizaremos para permitir que los usuarios creen cuentas en nuestra página Web, modifiquen sus perfiles, inicien o cierren su sesión, o también que les enviemos recordatorios de sus contraseñas y más. Todo esto es seguro y fácil de configurar. Podemos poner a mano todas las funcionalidades que Devise proporciona, pero no lo haremos por dos razones: Primero, toma mucho tiempo y en segundo, el nivel de arreglo que necesitamos para esta tesis no es tan complicado. Por lo tanto, una disposición estándar Devise es más que suficiente. (PeopleICanCode, s.f.)

#### 5.12.4 Librería de Ruby CanCan (GEMA)

CanCan es una biblioteca de autorización popular para Ruby on Rails que restringe el acceso de los usuarios a recursos específicos. La última joya (CanCanCan) es una continuación del proyecto muerto CanCan. (Tutorial de RIP, s.f.)

Todos los permisos se pueden definir en uno o varios archivos de capacidad y no se pueden duplicar entre controladores, vistas y consultas de bases de datos, manteniendo la lógica de permisos en un solo lugar para facilitar el mantenimiento y las pruebas. (GitHub, s.f.)



#### 5.12.5 Gestor de base de datos MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS<sup>7</sup>). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2<sup>8</sup>.

#### 5.12.6 Lenguaje de Marcas de Hipertexto HTML versión 5

HTML5 “es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que (como veremos más adelante) es extremadamente significativo. (Gauchat, 2012)

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio. (Gauchat, 2012)

Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero cómo esa estructura es construida. Crear fundamentos fuertes nos ayudará más adelante aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías.” (Gauchat, 2012)

---

<sup>7</sup> RDBMS: Sistema de gestión de bases de datos relacionales.

<sup>8</sup> DB2: es una familia de productos de sistema de gestión de bases de datos relacionales (RDBMS)

#### 5.12.7 Hojas de estilo en cascada CSS

**CSS** “fue siempre sobre estilo. En un intento por reducir el uso de código Javascript y para estandarizar funciones populares, CSS3 no solo cubre diseño y estilos web sino también forma y movimiento. La especificación de CSS3 es presentada en módulos que permiten a la tecnología proveer una especificación estándar por cada aspecto involucrado en la presentación visual del documento. Desde esquinas redondeadas y sombras hasta transformaciones y reposicionamiento de los elementos ya presentados en pantalla, cada posible efecto aplicado previamente utilizando JavaScript fue cubierto. Este nivel de cambio convierte CSS3 en una tecnología prácticamente inédita comparada con versiones anteriores. “ (Gauchat, 2012)

#### 5.12.8 Biblioteca de JavaScript JQuery

JQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, JQuery ha cambiado la forma en que millones de personas escriben JavaScript. (lancker, 2015)

#### 5.12.9 Framework de desarrollo Bootstrap 4

Bootstrap es una biblioteca multiplataforma también se define como conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. (Spurlock, 2014).

### 5.13 Metodología.

Se implementó la metodología UWE para la elaboración de esta tesis, que cubre todo el ciclo de vida de este tipo de sistemas web centrando además su atención en sistemas personalizados o adaptativos, UWE describe una metodología de diseño sistemático, basado en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML según (Galiano, 2012).

Se trabajó esta tesis mediante el desarrollo de las fases que plantea la metodología UWE:



**Figura 1: Fases o etapas del ciclo de vida del software.**

- 1) Captura, análisis y especificación de requisitos: para esta fase se hace uso de entrevistas, encuestas y cuestionarios con el fin de analizar y esclarecer las necesidades fundamentales de la clínica y farmacia donde se realizará la tesis.
- 2) Diseño del sistema: Se hace uso de UML para el diseño y modelado de todos los diagramas que conlleva la tesis en sus distintas fases.
- 3) Codificación del software: se procede a codificar el front-end y el back-end haciendo uso de los lenguajes y herramientas previstas a utilizar.
- 4) Pruebas: El software es testeado para detectar y mejorar posibles bugs en el funcionamiento del sistema con el fin de entregar una mejor calidad en el producto final.
- 5) La Instalación o Fase de Implementación: una vez concluida las fases anteriores se lleva a cabo la instalación de los insumos necesario para la implementación del sistema.

### **Evaluación del sistema.**

En esta fase se procede a la realización de pruebas para calidad del software en el sistema para las cuales existen diferentes maneras de realizar estas pruebas, en las que nos enfocaremos serán:

**Pruebas de funcionamiento:** Estas pruebas están dirigidas a testing de función y testing de seguridad.

**Pruebas de usabilidad:** La finalidad de estas pruebas es comprobar la relación del usuario con el sistema, prestando atención a la interfaz, la experiencia del usuario e interactividad.

**Pruebas funcionales:** Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio. Con estas pruebas funcionales se garantiza que las características y funcionalidades del software se comportan según lo esperado sin ningún problema.

**Pruebas unitarias:** Estas pruebas están dirigidas a cualquier función, procedimiento, método o módulo que puede ser una unidad que se someta a pruebas unitarias para determinar su corrección y comportamiento esperado.

## 6 Factibilidad del desarrollo.

En consideración de los recursos de cómputos (hardware y software) con los que cuenta la Clínica Pediátrica y Farmacia Cristina se muestran los siguientes aspectos, teniendo en cuenta que es un sistema web estos recursos son suficiente para su funcionamiento debido a que se está ejecutando desde un navegador mediante una url en la cual se encuentra alojada la aplicación web permitiendo que sea multiplataforma.

### 6.1 Factibilidad técnica.

#### Hardware y software de la clínica

Cantidad	Recurso	Descripción
1	Computadora de escritorio	Procesador Intel Core 2 duro, 500 GB HDD, 6 GB RAM.
1	Laptop	Procesador Intel inside 480 GB SSD, 4 GB RAM.
1	Laptop	Procesador Intel i3, 250 GB SSD, 4 GB RAM.
1	Conexión a internet	10 megas

#### Software

Cantidad	Recurso	Descripción
1	Sistema Operativo	Windows 8.1 Pro.
2	Sistema Operativo	Windows 10 Home.

Tabla 1: Hardware de la clínica

Cabe mencionar que estos son los recursos con los que cuenta la clínica, siendo óptimos para la puesta en producción del proyecto sobrepasando los recursos mínimos para el software.

## Hardware y software mínimos para el proyecto

Cantidad	Recurso	Descripción
1	Computadora de escritorio	Procesador Intel inside, 125 GB HDD, 2 GB RAM.
1	Laptop	Procesador Intel inside, 125 GB HDD, 2 GB RAM.
1	Laptop	Procesador Intel inside, 125 GB HDD, 2 GB RAM.
1	Conexión a internet	10 megas

### Software

Cantidad	Recurso	Descripción
1	Sistema Operativo	Windows 8 Home.
2	Sistema Operativo	Windows 8 Home.

**Tabla 2: requisitos mínimos del software**

En esta tabla se muestran los requisitos mínimos para el uso del software, actualmente la clínica cuenta con recursos superiores y óptimos para el uso del proyecto.

## Hardware y software óptimos

Cantidad	Recurso	Descripción
1	Computadora de escritorio	Procesador Intel Core i3 o AMD A10, almacenamiento 500 GB SSD, RAM 6 GB RAM.
1	Laptop	Procesador Intel Core i3 o AMD A10, almacenamiento 500 GB SSD, RAM 6 GB RAM.

1	Laptop	Procesador Intel Core i3 o AMD A10, almacenamiento 500 GB SSD, RAM 6 GB RAM.
1	Conexión a internet	100 megas

#### Software

Cantidad	Recurso	Descripción
1	Sistema Operativo	Windows 10 Pro.
2	Sistema Operativo	Windows 10 pro.

**Tabla 3: Requisitos óptimos del software**

En la tabla se muestran los requisitos óptimos para el uso del sistema.

## 6.2 Factibilidad económica.

En este punto, se realizó un desglose los gastos incurridos en el proyecto como tal, estos gastos serán clasificados como:

- Costos de hardware.
- Costos de software.
- Costos de desarrollo.
- Costos totales.

También se ha realizado las investigaciones correspondientes para determinar cuáles serán los costos más factibles y viables para hacer posible el desarrollo del Sistema web para la gestión de información médica de la “Clínica Pediátrica y Farmacia Cristina.

Se provee una lista detallada de estos gastos. ([Ver Tabla 2 - Costos de hardware.](#), [Tabla 3 - Costos de software.](#), [Tabla 4 - Costos de desarrollo.](#) y [Tabla 5 - Costos totales.](#))

### 6.2.1 Costos de hardware.

Estimación de costos.

#### Costos de hardware

---

Cantidad	Descripción	Precio unitario	Total
1	Computadora de escritorio, Procesador Intel Core 2 dúo, 500 GB HDD, 6 GB RAM.	350\$	350\$
1	Laptop, Procesador Intel inside 480 GB SSD, 4 GB RAM.	320\$	320\$
1	Laptop, Procesador Intel i3, 250 GB SSD, 4 GB RAM.	380\$	380\$
<b>Total</b>			<b>\$1050</b>

**Las computadoras para el desarrollo y el servidor de alojamiento del sistema no representaron ningún costo ya que la clínica cuenta con los recursos.**

**Tabla 4: Costos de hardware.**

#### 6.2.2 Costos de software.

##### **Costos de software**

Cantidad	Descripción	Precio unitario	Total
2	Ubuntu	\$0	\$0
2	Visual Studio Code	\$0	\$0
2	Acceso a Git	\$0	\$0
2	My SQL Server	\$0	\$0
2	Docker	\$0	\$0
2	Adobe Reader.	\$0	\$0
2	Windows 10 pro	\$21.21	\$21.21
1	Microsoft Windows 8.1 PRO	\$15.14	\$15.14
1	Hosting	\$ 7.99	\$ 7.99
<b>Total</b>			<b>\$44.34</b>

**Tabla 5: Costos de software.**



### 6.2.3 Costos de desarrollo.

#### Costos de desarrollo

Cantidad	Descripción	Salario mensual	Total	Cantidad
2	Desarrollador web senior.	\$900	6	\$5400
<b>Total</b>				<b>\$5400</b>

Tabla 6: Costos de desarrollo.

### 6.2.4 Costos totales.

#### Costos del proyecto

Descripción	Total (ya se cuenta con los recursos)	Total (adquiriendo recursos)
<b>Costos de hardware</b>	\$0	\$1050
<b>Costos de software</b>	\$0	\$44.34
<b>Costos de desarrollo</b>	\$5400	\$5400
<b>Total</b>	<b>\$5400</b>	<b>\$6,494.34</b>

Tabla 7: Costos totales del proyecto.

La inversión total en el proyecto es de \$ 6,486.35 este costo el cual involucra el total de los costos de hardware, software y desarrollo se lo está ahorrando la clínica y farmacia, ya que cuenta con el equipo que se debe comprar, así como también los softwares de desarrollo son de uso gratuito, no así el alojamiento del mismo que es el único costo que debe asumir la clínica, ya que actualmente el servicio de heroku es de pago lo que a su vez ofrece mejoras en el servicio de alojamiento. Cabe mencionar también que los equipos y recursos con los que cuenta la clínica y farmacia son óptimos para la puesta en marcha del proyecto y el correcto funcionamiento del mismo.

### 6.3 Factibilidad operativa.

Para el desarrollo de este apartado hay que tomar en cuenta que el uso de sistemas que automatizan procesos suelen ser un reto intimidante para personas que no estén familiarizadas con las computadoras y el software, en consecuencia, se tomara en cuenta que la interfaz de usuario sea lo más sencilla y amigable posible, de fácil uso, que permita una ágil navegación y no sea complicado en ningún aspecto, siempre enfocados en la comodidad del usuario y la facilidad que este perciba al migrar de un sistema de hojas de cálculo a un sistema web con mejoras tanto en el registro de información, como en su manipulación, la capacidad de generar reportes según la necesidad del usuario y la seguridad que este percibe al conceder permisos según los distintos roles que sean asignados, todo esto se lograra usando diagramas UML como base en el desarrollo.

La implementación de este sistema le dará la facilidad al usuario de SIM-Cristina de guardad de manera más organizada y sencilla grandes cantidades de información, así como el acceso a esta información será hará de forma más segura y rápida, se tendrá un control aun mayor y efectivo de la información de la clínica y el área de ventas de los medicamentos, con una estructura interna de los usuarios mejor organizada y controlando el acceso a la información para cada uno de ellos.

Según los descrito anteriormente, denotamos que el sistema es viable a nivel operativo, debido a que, con la implementación de este, se solucionan varios problemas, como el manejo de los procesos que abarca el sistema, generando mayor agilidad en los distintos procesos anteriormente descritos a los cuales el usuario tendrá un control mejor optimizado y organizado, evitando tiempos de espera innecesarios.

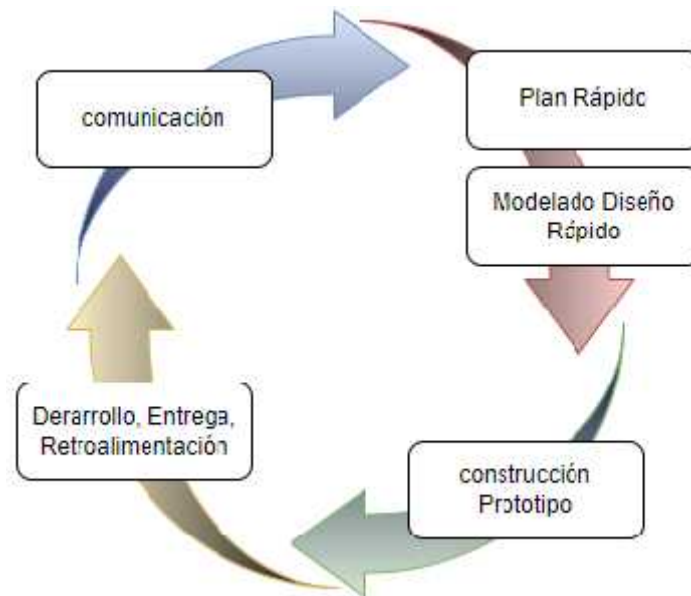
## 7 Resultado del desarrollo.

En este apartado se exponen cada recurso con el que cuenta la clínica, desde los aspectos más fundamentales del desarrollo de este proyecto pasando por la importancia que tiene la implementación del mismo, así como entender que es lo que el cliente realmente necesita por medio de la captura de requerimientos, y en consecuencia la presentación de los diseños a nivel de diagramas UML para que el usuario final tenga una idea clara y precisa de aquello que en función de sus necesidad se quiere lograr.

## 7.1 Modelo de prototipo.

Para la elaboración de SIM-Cristina utilizamos el método modelo de prototipo debido a los beneficios muy importante que nos aporta los cuales son:

- Los usuarios participan activamente en el desarrollo. Por lo tanto, los errores se pueden detectar en el primer paso del proceso de desarrollo de software.
- Se puede identificar la funcionalidad faltante, lo que ayuda a reducir el riesgo de fallas porque la creación de prototipos se considera una actividad de reducción de riesgos.
- Los comentarios de los usuarios ayudan a lograr mejores soluciones de desarrollo de software.
- El prototipo sirve como base para derivar una especificación del sistema.
- Los prototipos pueden ofrecer una formación temprana a los futuros usuarios del sistema de software.



**Figura 2: Ciclo del modelo de prototipos.**

Este modelo se utiliza para dar al usuario una vista preliminar de parte del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. (Ecured, s.f.)

Utilizando la plantilla básica ofrecida por Nice Admin -Free bootstrap admin HTML template y con el modelo de Prototipos Modular se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresa de esta manera se reduce el tiempo de retroalimentación entre el usuario y el equipo de desarrollo.

7.2 Fases de Desarrollo UWE.

7.2.1 Fase 1: Captura, análisis y especificación de requerimientos.

7.2.1.1 Actores

Los actores que son parte dentro del proyecto serán los usuarios que inician sesión en la plataforma y hacen uso del sistema de forma completa según sus permisos, dentro de los cuales están dependientes de farmacia, secretaria y un doctor, con un nivel de conocimiento suficiente para la comprensión de la forma en que funciona el sistema y como operarlo.

A continuación, se describirán los diferentes actores que intervienen en los casos de uso de sistema.

Actores	Funciones
Administrador del sistema	Súper usuario designado por la clínica y farmacia cristina, responsable de la administración de los usuarios del sistema, visualizar logs del sistema.
Doctor	Tiene la funcionalidad de crear usuarios, registrar pacientes, crear consultas, historiales clínicos, registrar seguimientos. Agendar exámenes de laboratorio, generar reportes de historiales clínicos, seguimientos, reporte por consultas y exámenes de laboratorio.
Secretaria	Tiene los permisos para registrar pacientes, agendar citas médicas, agendar exámenes de laboratorio.
Regente	Ingresar productos, registrar ventas de productos, modificar precios, generar reporte de ventas, generar facturas de ventas.

Tabla 8: actores del sistema.

7.2.1.2 Requerimientos funcionales.

Para esta fase se realizaron entrevistas al personal de la clínica y farmacia, esto permitió conocer el

funcionamiento actual de los procesos y como resultado obtener las especificaciones de los requerimientos funcionales y no funcionales que deben de representar el desarrollo de este proyecto de acuerdo con las necesidades expresadas en las entrevistas, usando modelos de caso de uso, esto permitió una adaptación en el prototipo del sistema de forma óptima y sistematizada.

ID: RF-01		Prioridad Esencial
Responsable:	Cristhofer Espinoza – José Reyes	
Descripción:	Recibimiento de credenciales de usuarios y su verificación para el de Inicio de Sesión.	

Tabla 9: Requerimientos funcionales RF-01.

ID: RF-02		Prioridad Esencial
Responsable:	Cristhofer Espinoza – José Reyes	
Descripción:	Contar con los diferentes módulos según las áreas descritas en el documento. (Historial clínico, agenda de citas, control de pagos del área de ventas en farmacia, control de pago de exámenes de laboratorio, control de pago de consultas, configuración de usuarios y roles, historial de acciones)	

Tabla 10: Requerimientos funcionales RF-02.

ID: RF-03		Prioridad Esencial
Responsable:	Cristhofer Espinoza – José Reyes	
Descripción:	Almacenamiento de datos en la Base de datos referente a los módulos definidos previamente.	

Tabla 11: Requerimientos funcionales RF -03.

**ID: RF-04****Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Visualización de datos almacenados en la base de datos.

**Tabla 12: Requerimientos funcionales RF -04.****ID: RF-05****Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Permitir la creación de nuevos historiales clínicos, así como la modificación y actualización de estos.

**Tabla 13: Requerimientos funcionales RF -05.****ID: RF-06****Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Mostrar las gráficas de las enfermedades con mayor frecuencia en las consultas por cada mes.

**Tabla 14: Requerimientos funcionales RF -06.****ID: RF-07****Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Generación de reportes con datos globales del sistema por módulos del sistema.

**Tabla 15: Requerimientos funcionales RF -07.****ID: RF-08****Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Sumatoria total de las ventas de cada día en la farmacia y la clínica con respecto a las consultas.

**Tabla 16: Requerimientos funcionales RF -08.**

**ID: RNF-01**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Se puede tener acceso a la aplicación web desde cualquier navegador.

**Tabla 17: Requerimiento no funcional RNF-01.**

**ID: RNF-02**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Para el despliegue del sistema web no se necesitará la instalación de un software adicional más que el navegador web.

**Tabla 18: Requerimiento no funcional RNF-02.**

**ID: RNF-03**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	Los mensajes presentados al usuario de error advertencia y éxito deben de ser claro y entendibles.

**Tabla 19: Requerimiento no funcional RNF-03.**

**ID: RNF-04**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	El sistema debe de tener un diseño responsive y adaptable a multiplataforma.

**Tabla 20: Requerimiento no funcional RNF-04.**

**ID: RNF-05**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	El sistema debe de tener interfaz gráfica intuitiva y amigable para el usuario.

**Tabla 21: Requerimiento no funcional RNF-05.**

**ID: RNF-06**

**Prioridad Esencial**

<b>Responsable:</b>	Cristhofer Espinoza – José Reyes
<b>Descripción:</b>	El sistema tendrá implementado un sistema de autenticación de usuarios, de esta manera sólo usuarios registrados podrán acceder a los distintos módulos.

**Tabla 22: Requerimiento no funcional RNF-06.**



#### *7.2.1.4 Restricciones de Diseño.*

El Desarrollo del sistema está estructurado bajo la arquitectura modelo, vista y controlador (MVC) el lenguaje con el que se incorporara será Ruby siendo este el que el equipo de desarrollo implementara para la codificación del sistema, con este MVC se permite que el proyecto tenga brecha de crecimiento en el futuro sin tener complicaciones de implementación.

#### *7.2.2 Fase 2: Diseño del sistema.*

##### *7.2.2.1 Diagramas de caso de uso.*

Las acciones que el usuario podrá realizar en el sistema estarán determinadas según el tipo de perfil y permisos que se le sea asignados y con los que podrá acceder:

#### **Administrador de sistema:**

- ) Administración del sistema.
- ) Generación de reportes.
- ) Creación de roles.

#### **Doctor:**

Es el encargado de atender en la clínica y también en farmacia de ser necesario, al ser un usuario con muchos accesos puede acceder a diferentes áreas, sus funciones son:

- ) Crear historiales clínicos.
- ) Crear seguimientos.
- ) Agendar citas.
- ) Designar precios de productos y servicios.
- ) Crear usuarios.
- ) Asignar roles.

#### **Secretaria:**

Se encarga de atender a los pacientes antes de pasar al consultorio, tomar datos personales y asignarles un orden de atención, en el sistema se encarga:

- ) Crear citas.
- ) Visualizar historiales clínicos.
- ) Visualizar seguimientos
- ) Registrar pagos por servicios y crear reportes.
- ) Registrar pagos por exámenes y crear reportes.

**Regente de farmacia:**

Encargado de atender la farmacia, sus funciones dentro del sistema son:

- ) Agregar productos al sistema.
- ) Registrar pagos por productos y crear reportes.

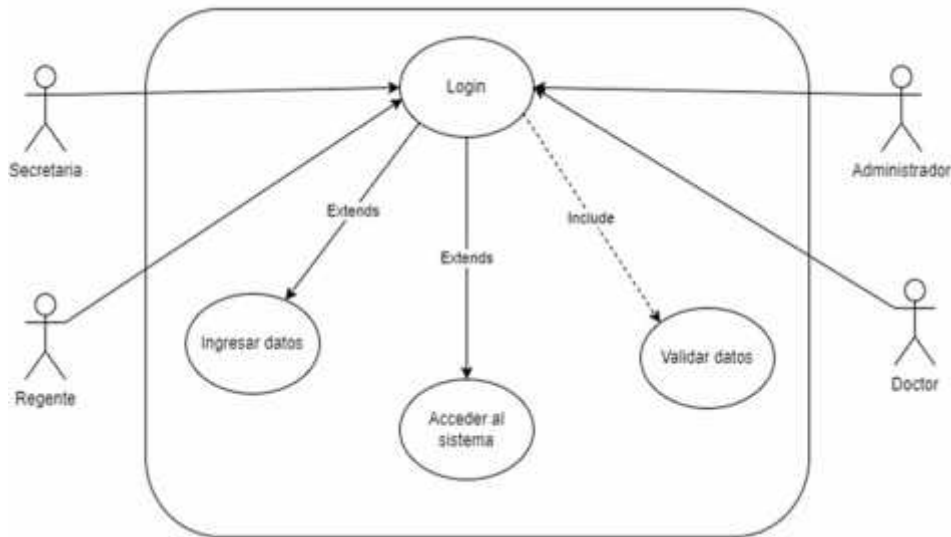


Figura 3: Diagrama de caso de uso login.

Nombre del caso de uso	Login
Actores	Administrador, doctor, secretaria y regente.
Descripción	El diagrama no indica la forma en como el usuario valida sus datos para poder acceder al sistema.
Disparador	Menú principal del sistema
Flujo básico de eventos	1.Ingresa usuario 2. El sistema valida los datos mediante la BD 3. Ingresa la contraseña 4. El sistema autoriza el ingreso
Flujo alterno	En caso de que el usuario ingrese datos incorrectamente, la página no permitirá el ingreso.  Se le notificará al usuario mediante un mensaje de “Email o contraseña inválidos”, luego se le permitirá ingresar los datos nuevamente hasta que sean correctos
Condición de entrada	El usuario debe de estar registrado en la base de datos para poder hacer uso del sistema.
Condición de salida	Cerrar sesión
Escenarios	Ninguno.

Tabla 23: Diagrama de caso de uso login.

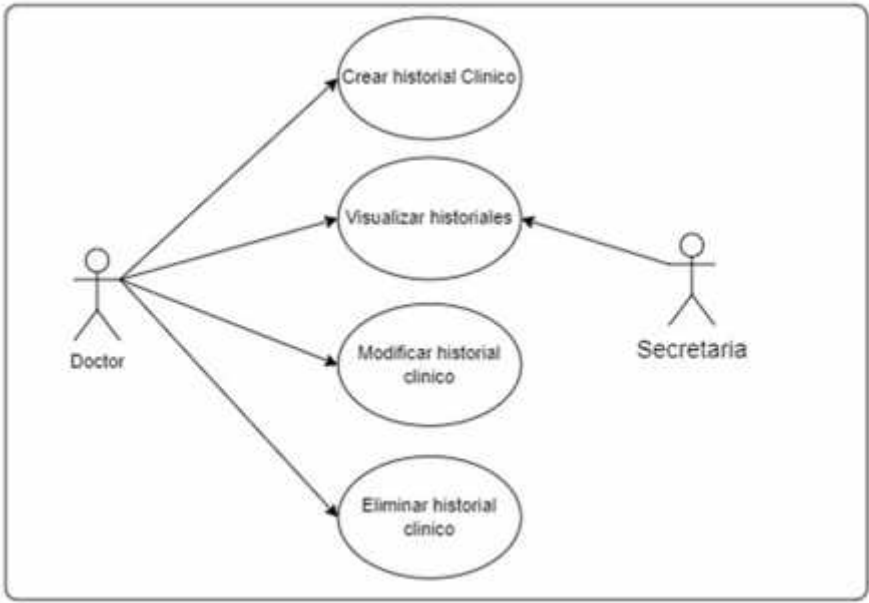


Figura 4: Diagrama de caso de uso historial clínico.

Nombre del caso de uso		Historial clínico
Actores	Doctor y secretaria.	
Descripción	Consultar paciente del día. Registro de ficha médica. (antecedentes personales y familiares-valoración de signos vitales).	
Disparador	Consultar, registrar.	
Flujo básico de eventos	1. Consultar los pacientes del día. 2. Registrar la ficha médica (de acuerdo a la información del peso, la estatura, la presión, etc.) 3. Cuando la ficha medica está llena, automáticamente se muestra la ficha medica luego de guardar.	
Flujo alternativo	No se ingresan los datos del paciente, no se guardan los datos.	
Condición de entrada	El usuario debe de haber ingresado al sistema.	
Condición de salida	El usuario puede salir de la opción regresando al módulo de historial clínico.	
Escenarios	Visualizar los pacientes del día en el sistema. Registrar ficha media con toda la información necesaria.	

Tabla 24: Caso de uso historial clínico.

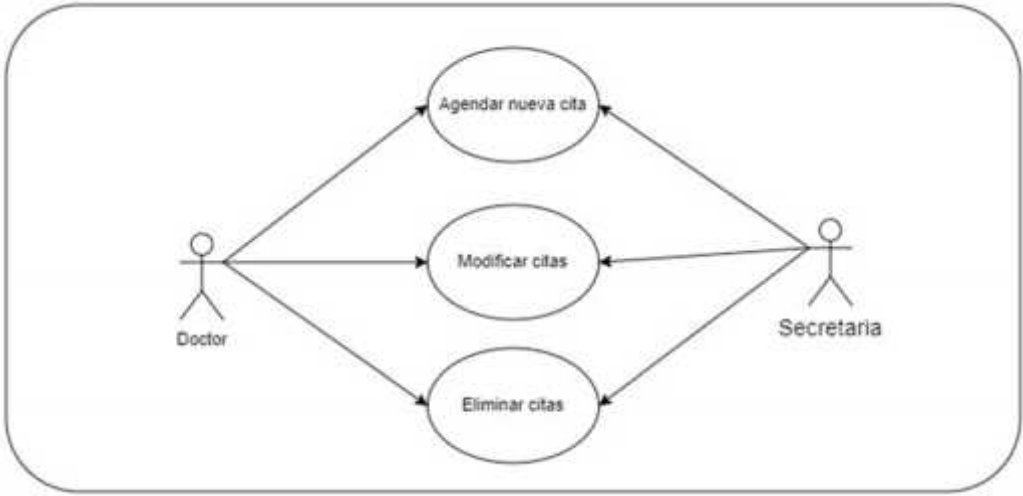


Figura 5: Diagrama de caso de uso citas médicas.

Nombre del caso de uso	Citas médicas
Actores	Doctor y secretaria.
Descripción	Consultar paciente del día. Registro de ficha de citas. Modificación de datos de cita.
Disparador	Consultar, registrar, modificar.
Flujo básico de eventos	1. Consultar los pacientes del día. 2. Registrar la ficha de citas. 3. Cuando la ficha de citas está llena, automáticamente se muestra la ficha luego de guardar y esta se puede actualizar o eliminar.
Flujo alterno	No se ingresan los datos del paciente, no se guardan los datos.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo de agenda.
Escenarios	Visualizar los pacientes del día en el sistema. Registrar ficha de citas con toda la información necesaria.

Tabla 25: Caso de uso citas médicas.

#### 7.2.2.1.4 Seguimientos.

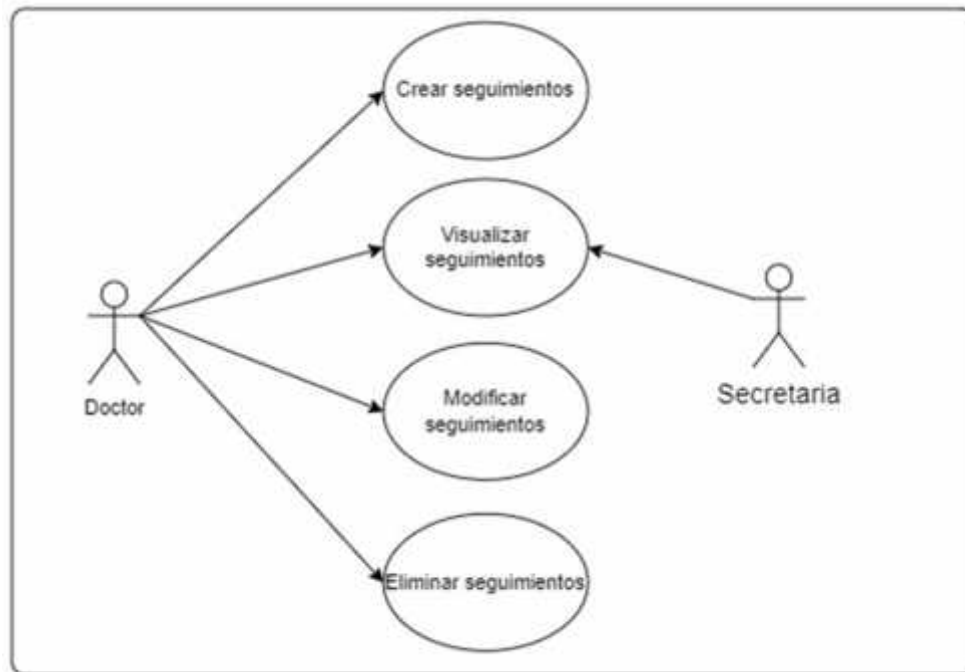


Figura 6: Diagrama de caso de uso seguimiento.

<b>Nombre del caso de uso</b>	<b>Seguimiento.</b>
<b>Actores</b>	Doctor y secretaria.
<b>Descripción</b>	Crear seguimientos. Visualizar, actualizar y eliminar seguimientos.
<b>Disparador</b>	Consultar, registrar, modificar, eliminar.
<b>Flujo básico de eventos</b>	1. Consultar los pacientes con seguimientos. 2. Registrar los datos para seguimiento. 3. Cuando los datos de seguimientos están completos, automáticamente se muestra la ficha luego de guardar y esta se puede actualizar o eliminar.
<b>Flujo alternativo</b>	No se ingresan los datos del paciente, no se guardan los datos.
<b>Condición de entrada</b>	El usuario debe de haber ingresado al sistema.
<b>Condición de salida</b>	El usuario puede salir de la opción regresando al módulo de reporteria.
<b>Escenarios</b>	Visualizar los pacientes que poseen seguimientos, así como crear nuevos seguimientos.
<b>Nombre del caso de uso</b>	<b>Seguimiento.</b>
<b>Actores</b>	Doctor y secretaria.
<b>Descripción</b>	Crear seguimientos. Visualizar, actualizar y eliminar seguimientos.
<b>Disparador</b>	Consultar, registrar, modificar, eliminar.
<b>Flujo básico de eventos</b>	1. Consultar los pacientes con seguimientos. 2. Registrar los datos para seguimiento. 3. Cuando los datos de seguimientos están completos, automáticamente se muestra la ficha luego de guardar y esta se puede actualizar o eliminar.
<b>Flujo alternativo</b>	No se ingresan los datos del paciente, no se guardan los datos.
<b>Condición de entrada</b>	El usuario debe de haber ingresado al sistema.
<b>Condición de salida</b>	El usuario puede salir de la opción regresando al módulo de reporteria.
<b>Escenarios</b>	Visualizar los pacientes que poseen seguimientos, así como crear nuevos seguimientos.

**Tabla 26: Diagrama de caso de uso seguimiento.**

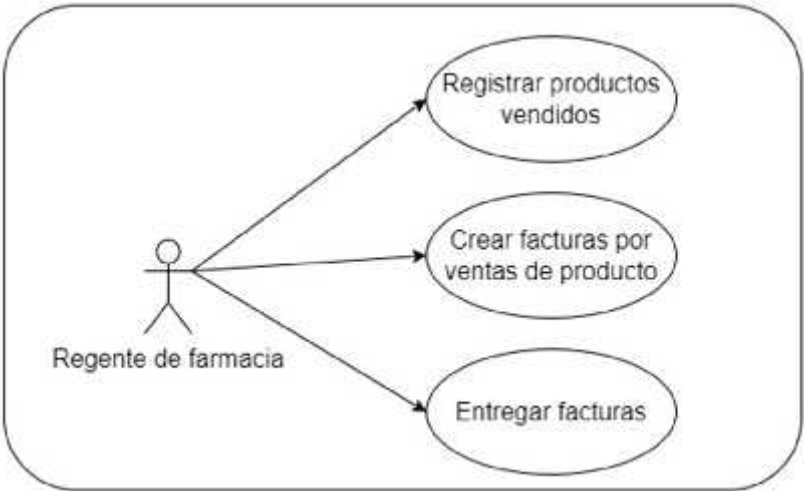


Figura 7: Diagrama de caso de uso control de pago.

Nombre del caso de uso		Control de pago.
Actores	Regente de farmacia.	
Descripción	Crear factura por productos vendidos. Visualizar, actualizar y eliminar factura por productos.	
Disparador	Consultar, registrar, modificar, eliminar.	
Flujo básico de eventos	1. Consultar las facturas existentes por ventas de productos. 2. Registrar los datos para una nueva factura. 3. Cuando los datos de la factura están completos, automáticamente se muestra la factura registrada luego de guardar y esta se puede actualizar o eliminar.	
Flujo alterno	No se ingresan los datos del paciente, no se guardan los datos.	
Condición de entrada	El usuario debe de haber ingresado al sistema.	
Condición de salida	El usuario puede salir de la opción regresando al módulo de control de pago.	
Escenarios	Visualizar las ventas por los distintos productos que vende la farmacia.	

Tabla 27: Caso de uso control de pago.



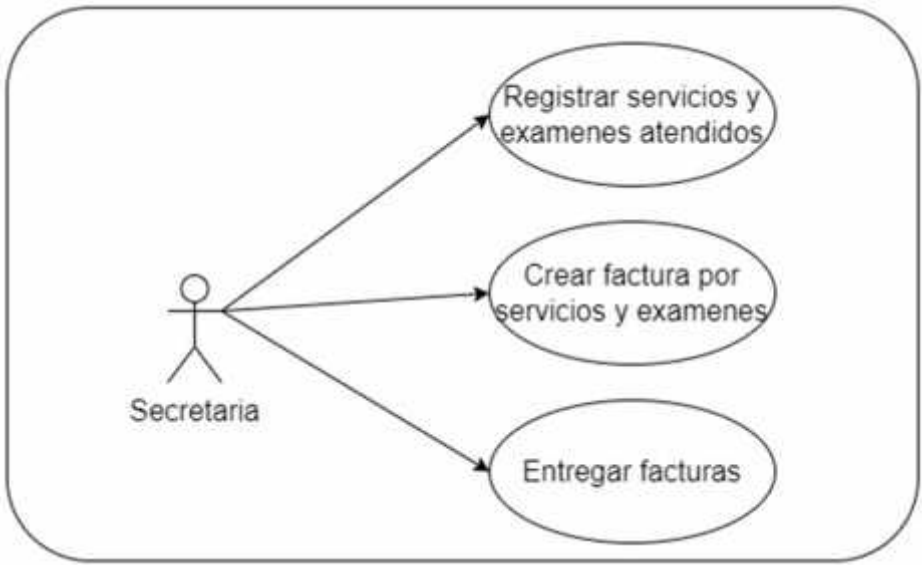


Figura 8: Diagrama de caso de uso control de pago exámenes y servicios.

Nombre del caso de uso	Control de pago exámenes y servicios.
Actores	Secretaria.
Descripción	Crear facturación por exámenes de laboratorio y servicios. Visualizar, actualizar y eliminar facturas por exámenes y servicios.
Disparador	Consultar, registrar, modificar, eliminar.
Flujo básico de eventos	1. Consultar los pacientes con exámenes pendientes. 2. Registrar los datos para exámenes nuevos y servicios ofrecidos. 3. Cuando los datos de la factura están completos, automáticamente se muestra la factura terminada, luego de guardar y esta se puede actualizar o eliminar.
Flujo alternativo	No se ingresan los datos del paciente, no se guardan los datos.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo facturación.
Escenarios	Visualizar las facturas por exámenes de laboratorio realizados

Tabla 28: Caso de uso control de pago exámenes y servicios.

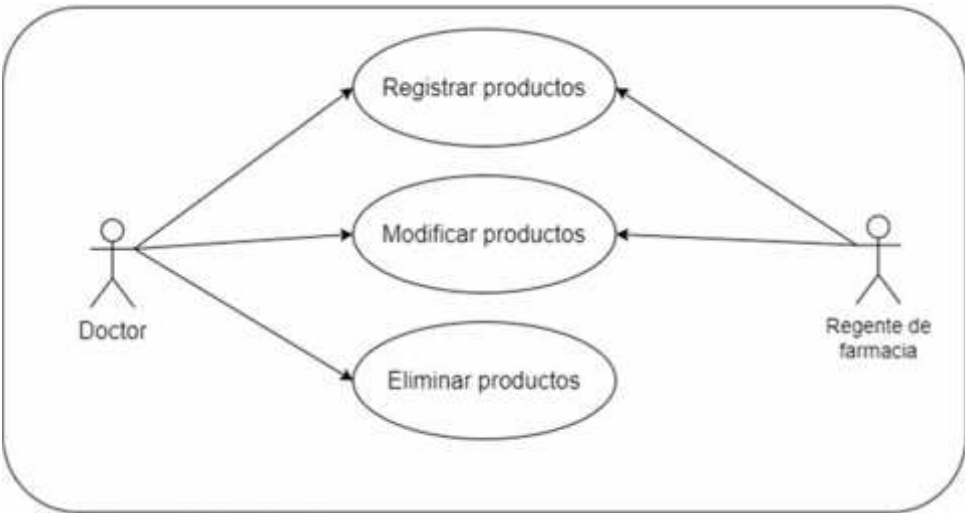


Figura 9: Diagrama de caso de uso catalogo - producto.

Nombre del caso de uso	Catalogo-producto.
Actores	Doctor, regente de farmacia.
Descripción	Crear y agregar productos y sus datos a la lista de productos que luego serán llamados en facturación.
Disparador	Consultar, registrar, modificar, eliminar.
Flujo básico de eventos	1. Consultar los productos existentes. 2. Registrar los datos para nuevos productos. 3. Cuando los datos de los productos estén agregados, se mostrarán en la lista de productos, para posteriormente ser llamados en facturación.
Flujo alterno	No se ingresan los datos del producto, no se guardan los datos.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo de facturación.
Escenarios	Visualizar las facturas por productos registrados.

Tabla 29: Caso de uso catalogo - producto.

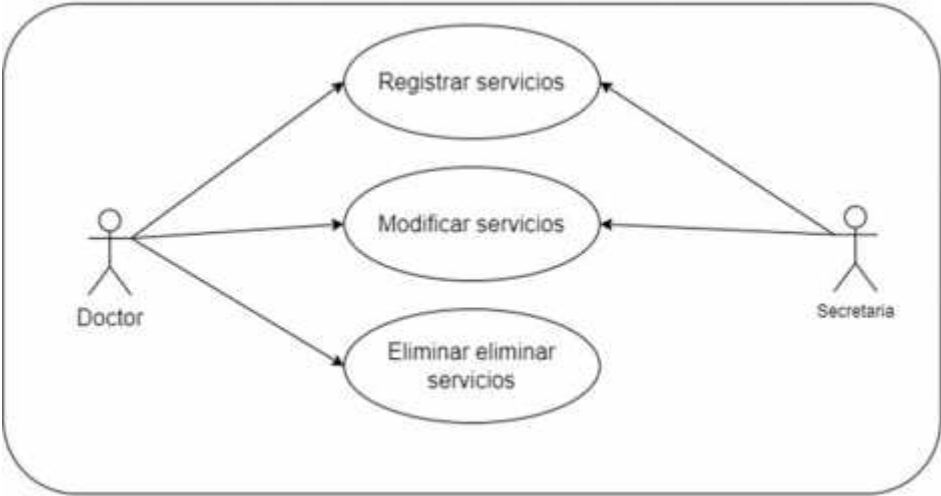


Figura 10: Diagrama de caso de uso catalogo - servicios.

Nombre del caso de uso	Catalogo-servicio.
Actores	Doctor, secretaria.
Descripción	Crear y agregar los servicios y sus datos a la lista de servicios ofertados que luego serán llamados en control de pago.
Disparador	Consultar, registrar, modificar, eliminar.
Flujo básico de eventos	1. Consultar los servicios ofertados. 2. Registrar los datos para los servicios. 3. Cuando los datos de los productos estén agregados, se mostrarán en la lista de servicios, para posteriormente ser llamados en control de pago.
Flujo alterno	No se ingresan los datos del producto, no se guardan los datos.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo de control de pago.
Escenarios	Visualizar las facturas por servicios registrados.

Tabla 30: Caso de uso catalogo - servicio.

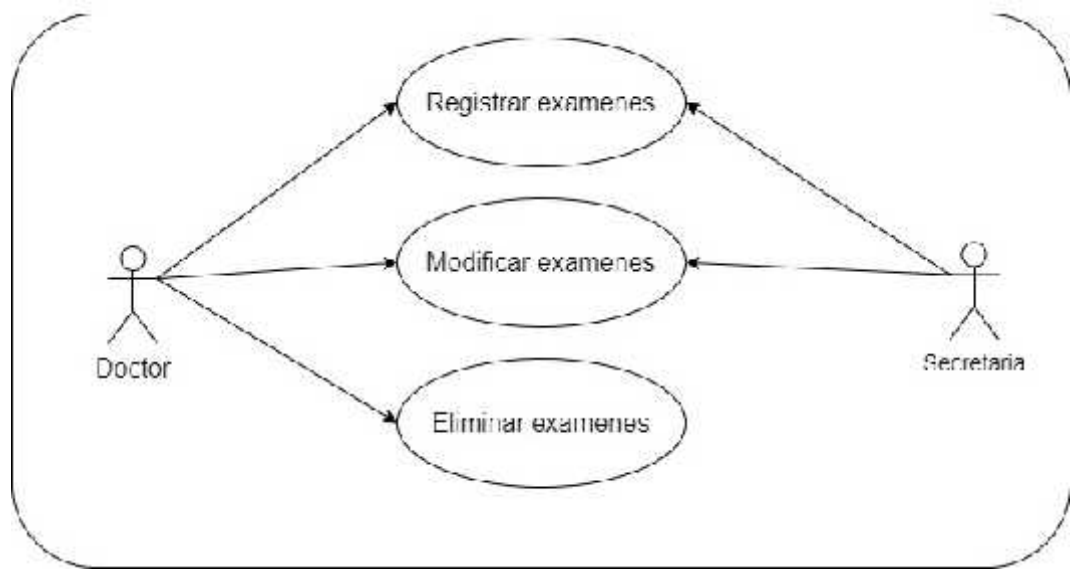


Figura 11: Diagrama de caso de uso catalogo - exámenes.

Nombre del caso de uso	Catalogo-exámenes.
Actores	Doctor, secretaria.
Descripción	Crear y agregar los exámenes y sus datos a la lista de exámenes ofertados que luego serán llamados en control de pago.
Disparador	Consultar, registrar, modificar, eliminar.
Flujo básico de eventos	1. Consultar los exámenes. 2. Registrar los datos para los exámenes. 3. Cuando los datos de los productos estén agregados, se mostrarán en la lista de exámenes, para posteriormente ser llamados en control de pago.
Flujo alterno	No se ingresan los datos del producto, no se guardan los datos.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo de control de pagos.
Escenarios	Visualizar las facturas por exámenes registrados.

Tabla 31: Caso de uso control de pago catalogo - exámenes.

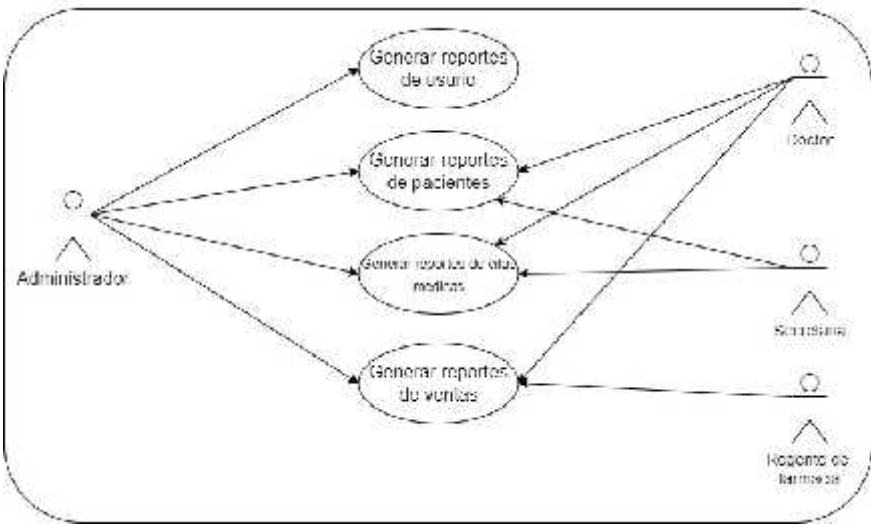


Figura 12: Diagrama de caso de uso reporteria.

Nombre del caso de uso	Reporteria.
Actores	Administrador, doctor, secretaria, regente de farmacia.
Descripción	Crear factura por exámenes de laboratorio. Visualizar, actualizar y eliminar facturas por exámenes.
Disparador	Consultar, registrar, modificar, eliminar.
Flujo básico de eventos	1. Consultar los reportes según el permiso de los roles asignados. 2. modificar datos de los reportes. 3. Cuando los datos de los reportes están completos, automáticamente se muestra el reporte terminado en una lista, luego se guarda y este se puede actualizar o eliminar.
Flujo alterno	No se genera reportes.
Condición de entrada	El usuario debe de haber ingresado al sistema.
Condición de salida	El usuario puede salir de la opción regresando al módulo de reportes.
Escenarios	Visualizar reportes según los módulos.

Tabla 32: Caso de uso reportearía.

### 7.2.2.2 Diagramas de navegación.

#### 7.2.2.2.1 Diagrama de navegación historial clínico.

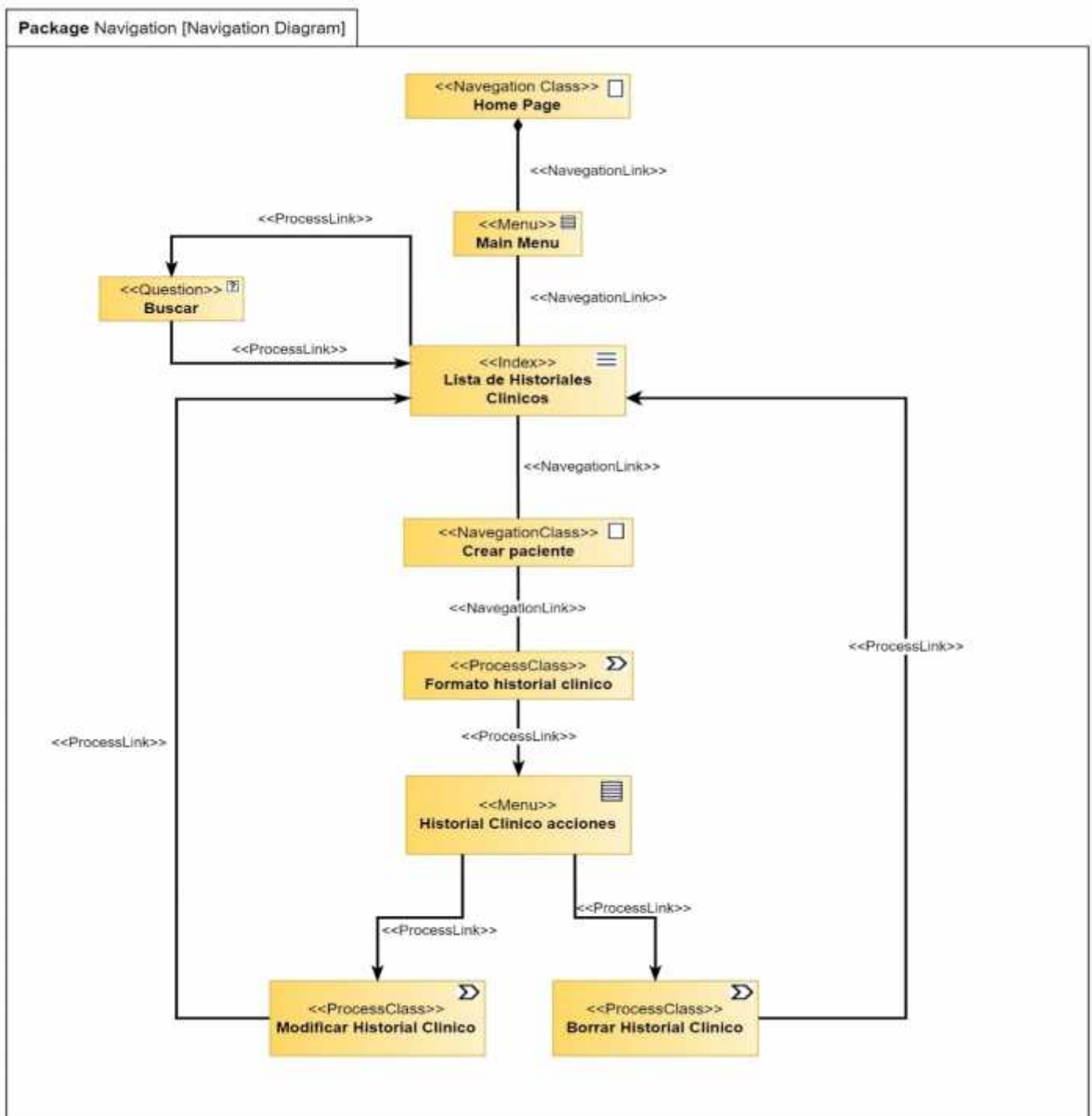


Figura 13: Diagrama de navegación del proceso para crear un historial clínico.

#### 7.2.2.2.2 Diagrama de navegación agendar cita.

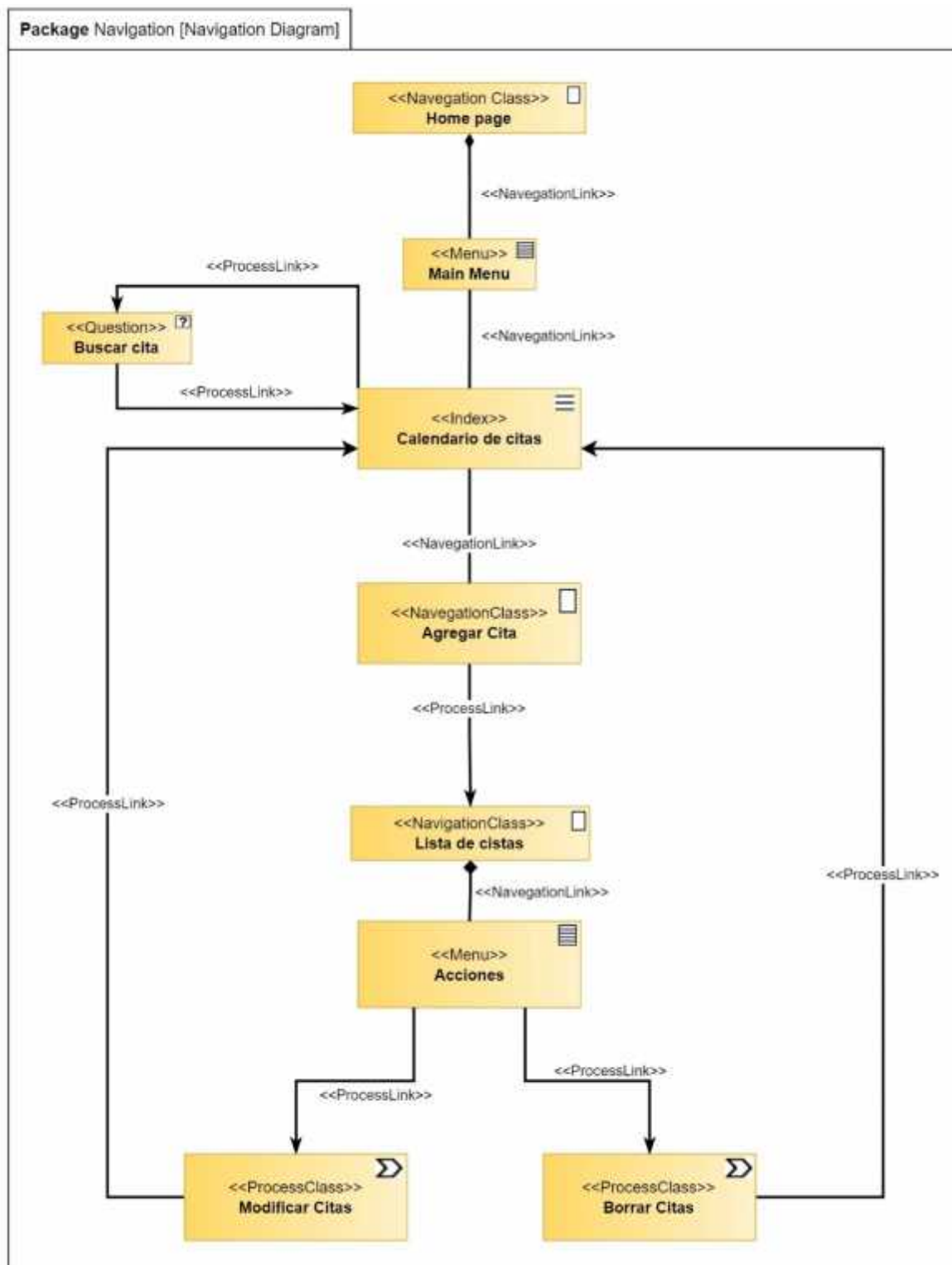


Figura 14: Diagrama de navegación del proceso para crear una cita médica.

### 7.2.2.2.3 Diagrama de navegación catálogo.

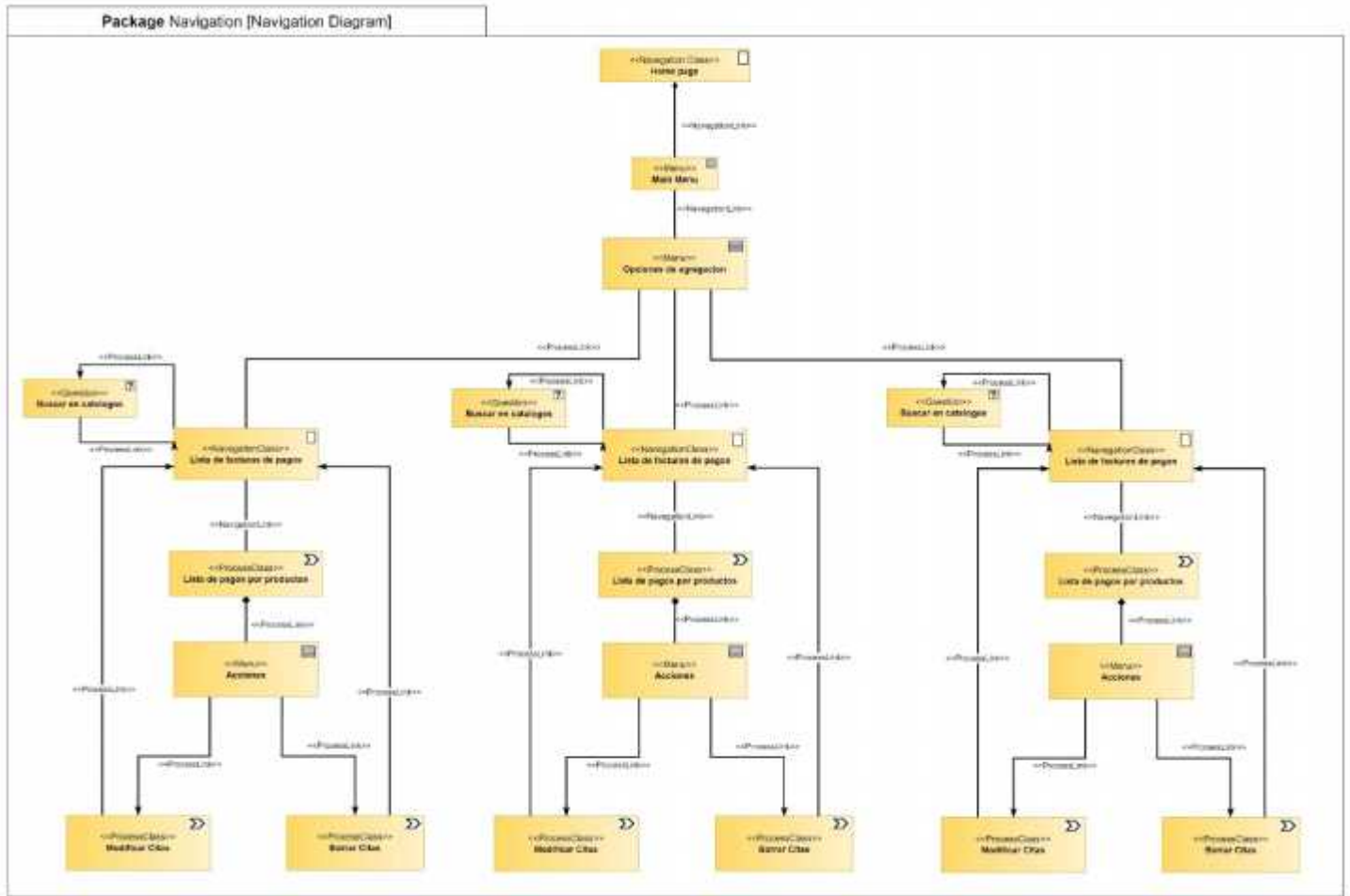


Figura 15: Diagrama de navegación agregar productos o servicios a catálogos.



#### 7.2.2.2.4 Diagrama de navegación control de pago.

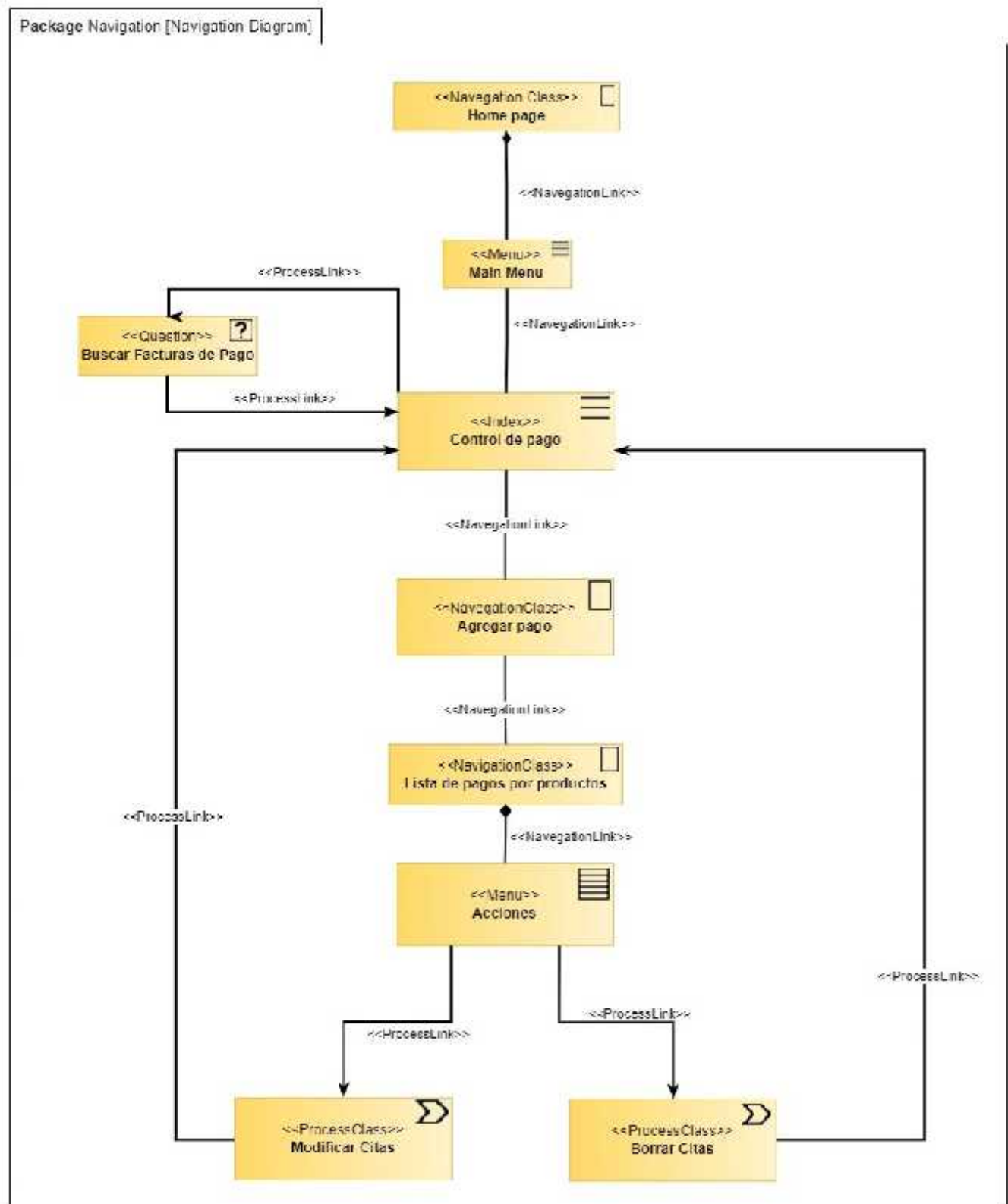


Figura 16: Diagrama de navegación control de pagos de productos o servicios.

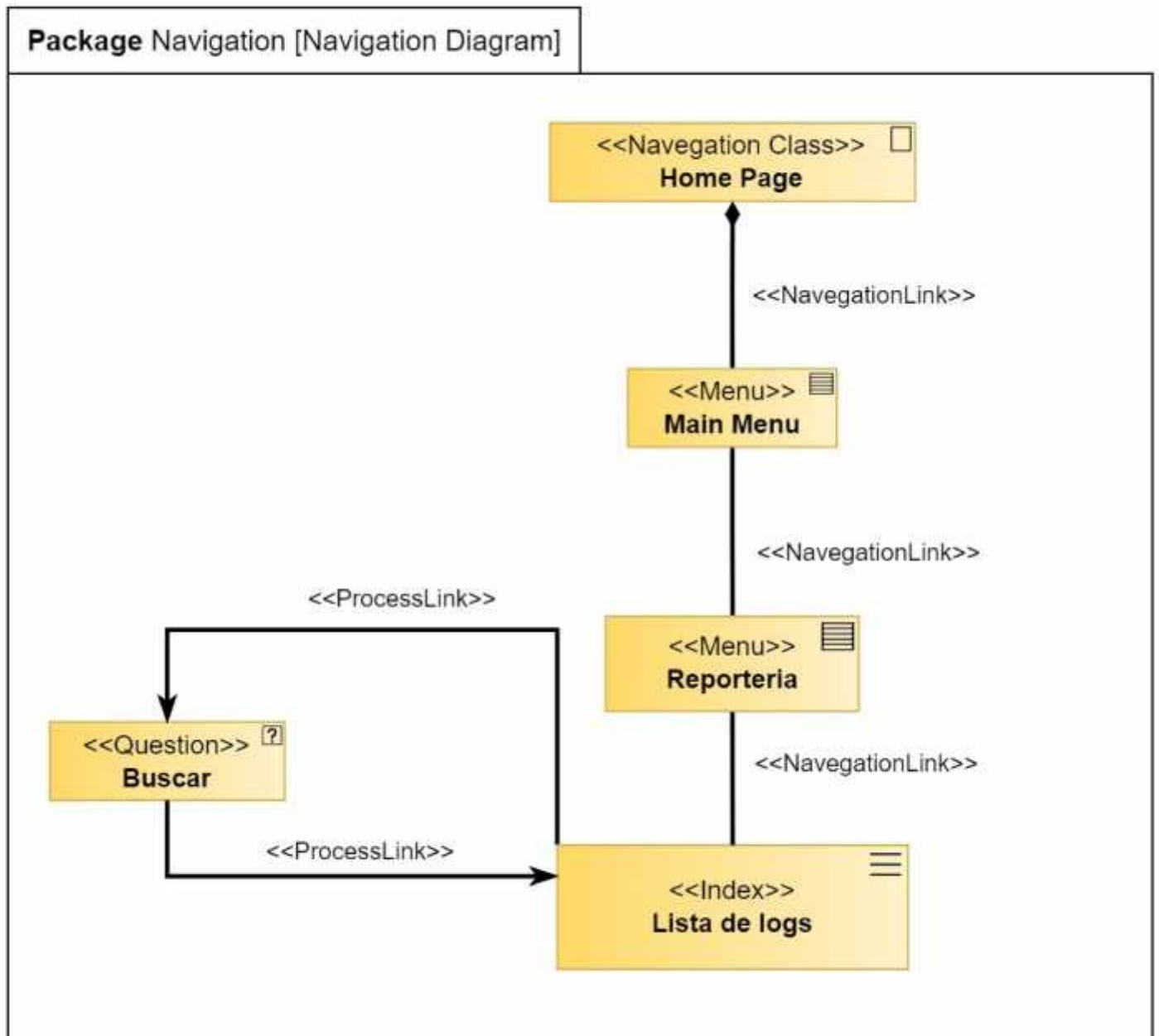


Figura 17: Diagrama de navegación para la visualización de logs.

#### 7.2.2.2.6 Diagrama de navegación configuración.

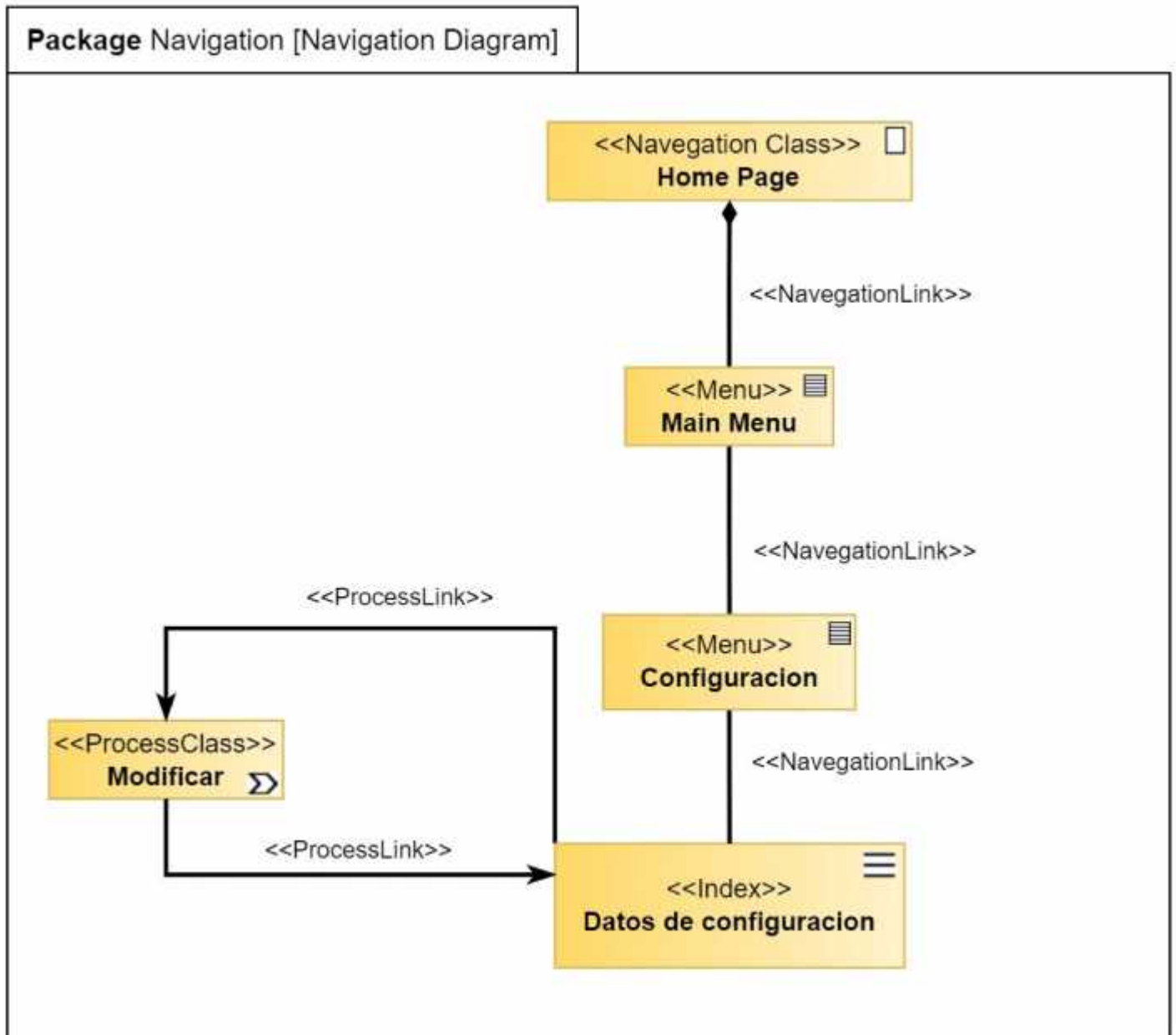


Figura 18: Diagrama de navegación para configurar datos de la empresa.

### 7.2.2.3 Diagrama de Secuencia.

En un sistema funcional, los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos. Estas interacciones entre objetos muestran una estructura de navegación que representa al sistema (SIM Cristina).

#### 7.2.2.3.1 Diagrama de secuencia – Autenticación de Usuario.

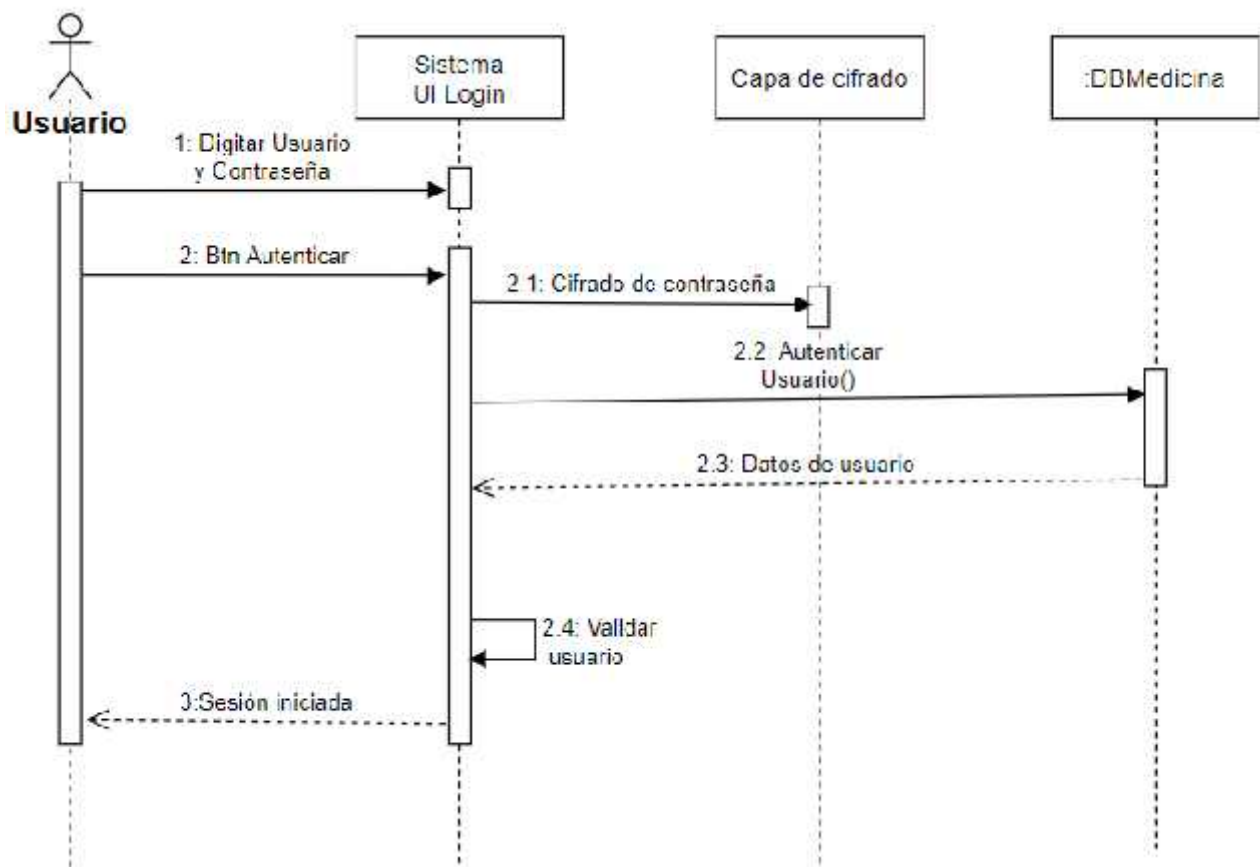


Figura 19: Autenticación de usuario.

#### 7.2.2.3.2 Diagrama de secuencia – Crear Nuevo Paciente.

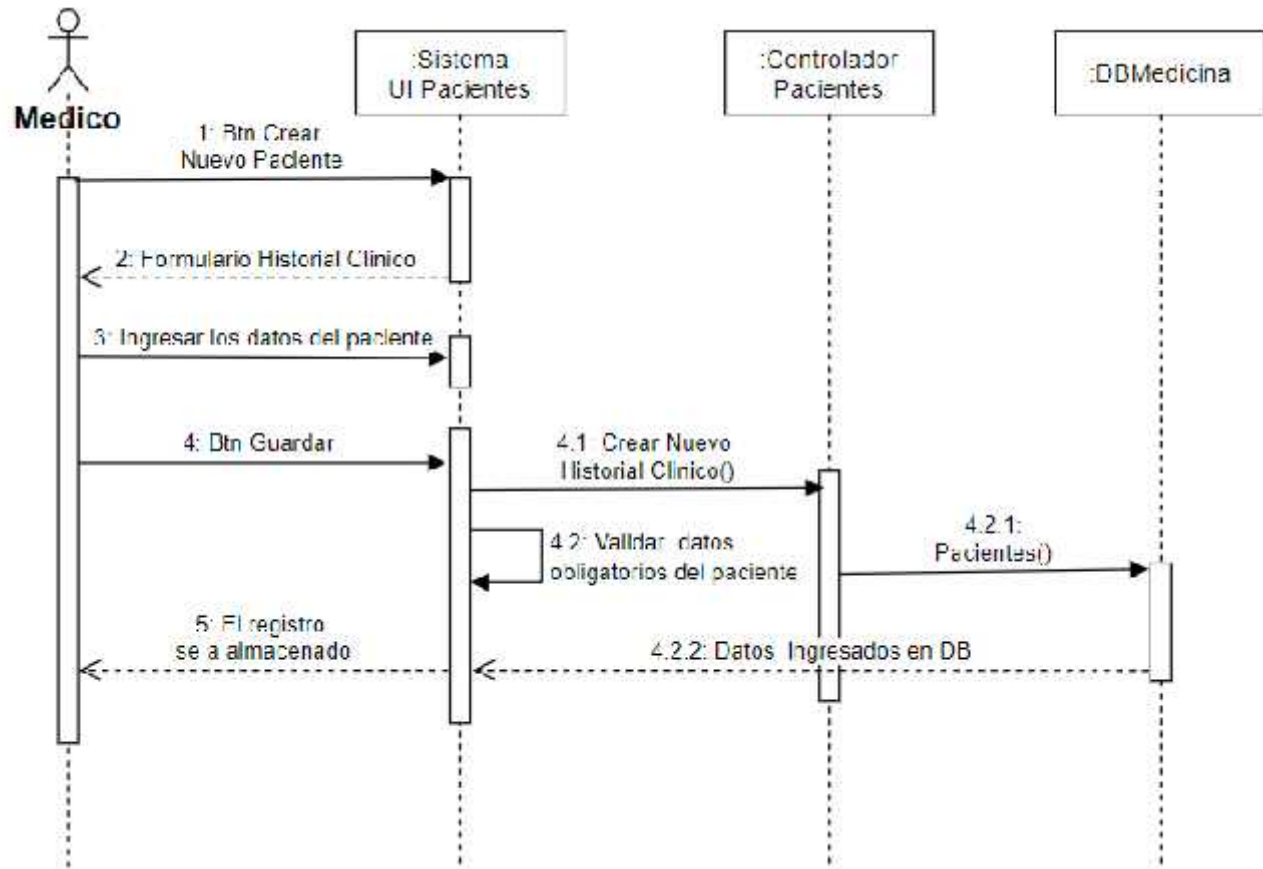


Figura 20: Crear nuevo paciente.

### 7.2.2.3.3 Diagrama de secuencia – Seguimiento Paciente.

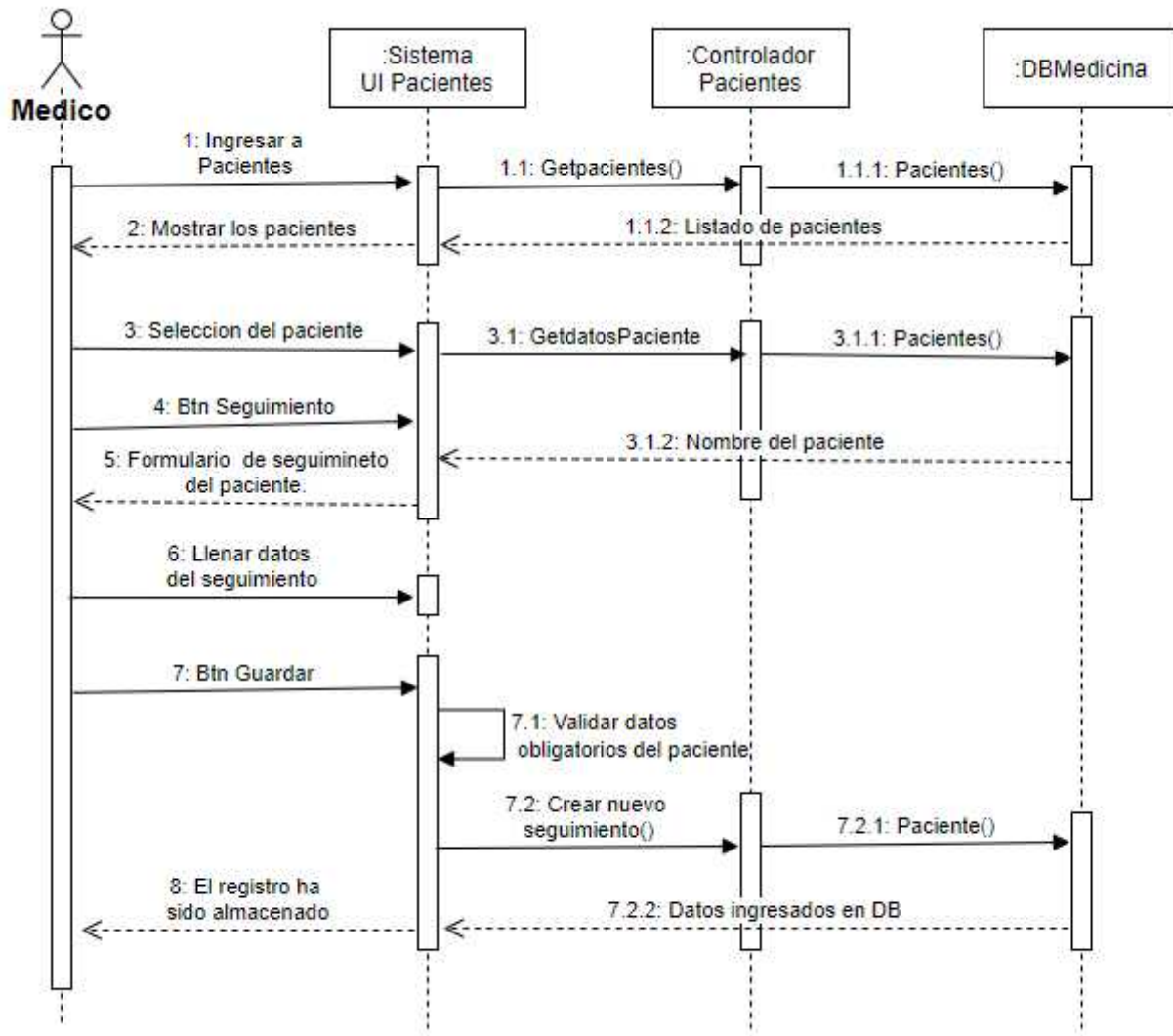


Figura 21: Seguimiento paciente.

#### 7.2.2.3.4 Diagrama de secuencia – Agendar Cita.

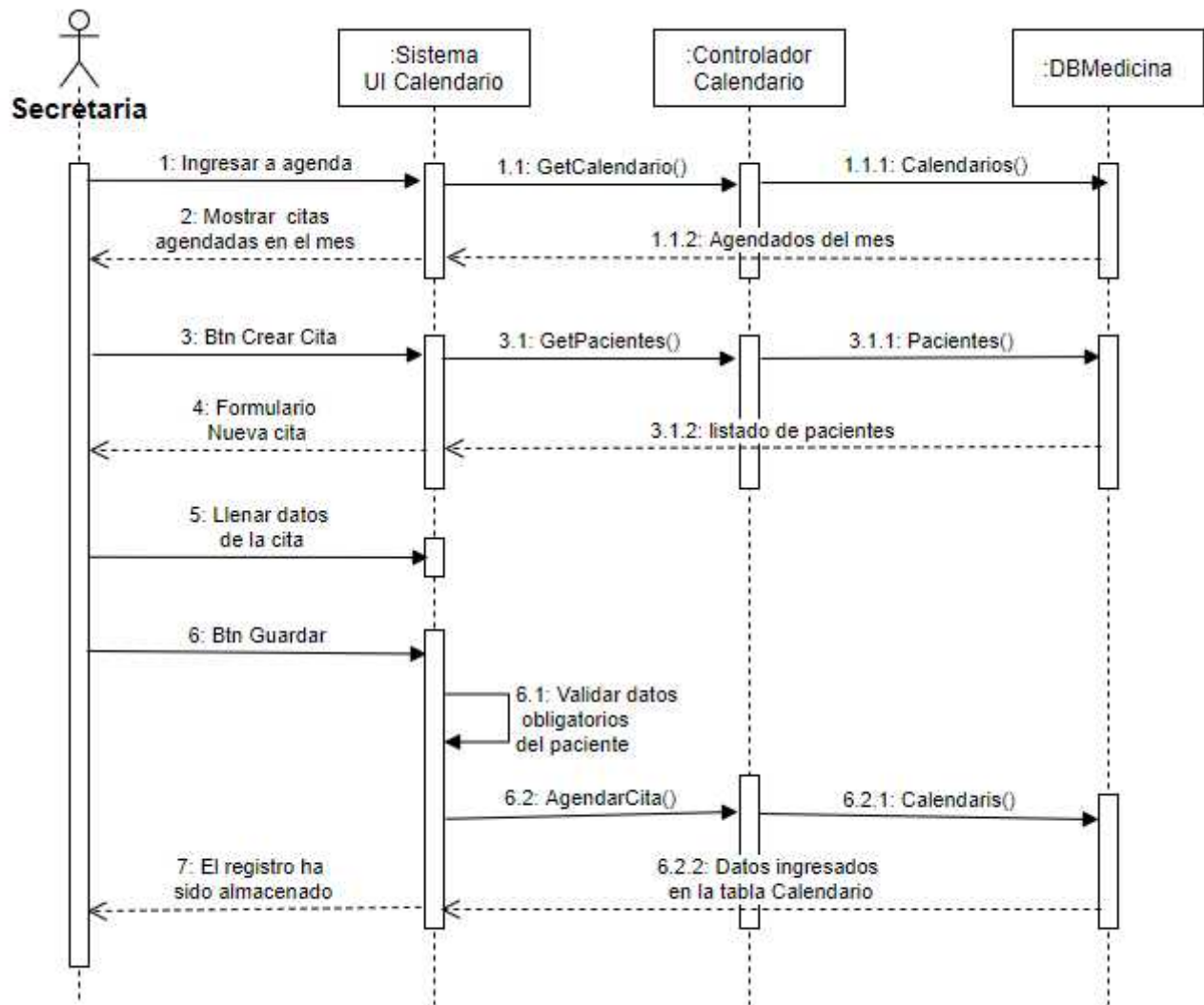


Figura 22: Agendar cita.

#### 7.2.2.3.5 Diagrama de secuencia – Agregar Registro en Catálogo.

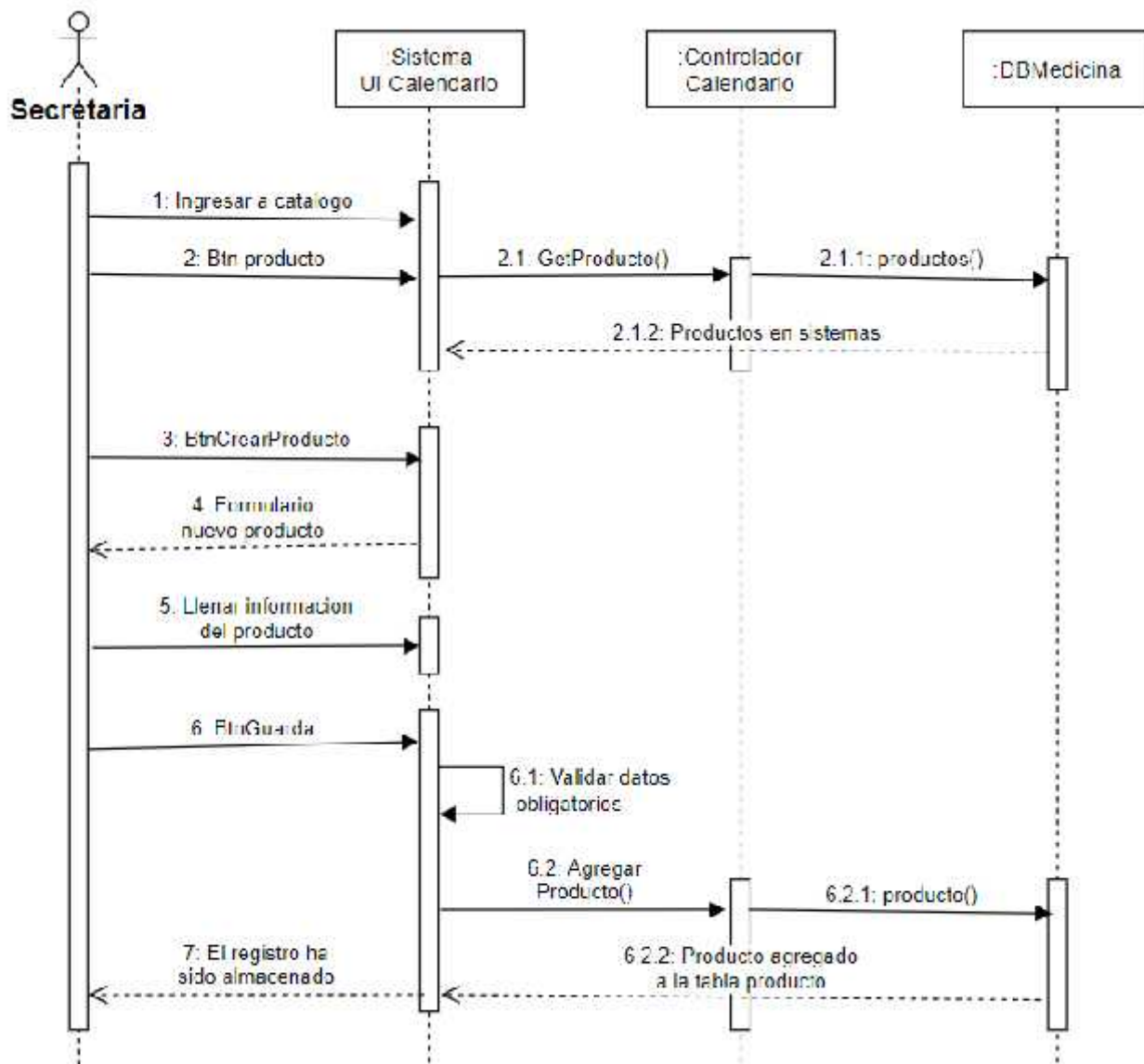


Figura 23: Agregar registro en catálogo.



#### 7.2.2.3.6 Diagrama de secuencia – Crear Factura.

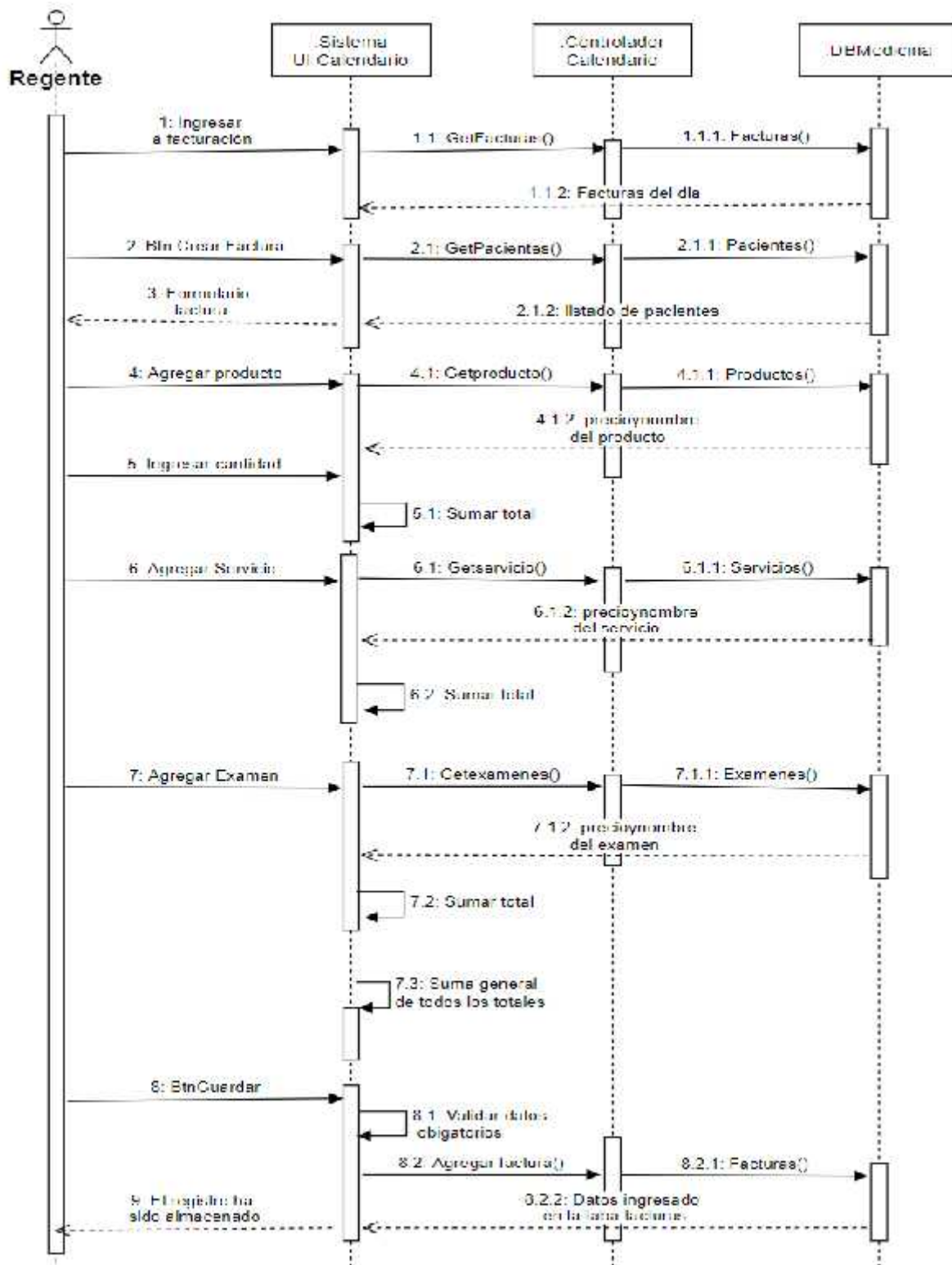


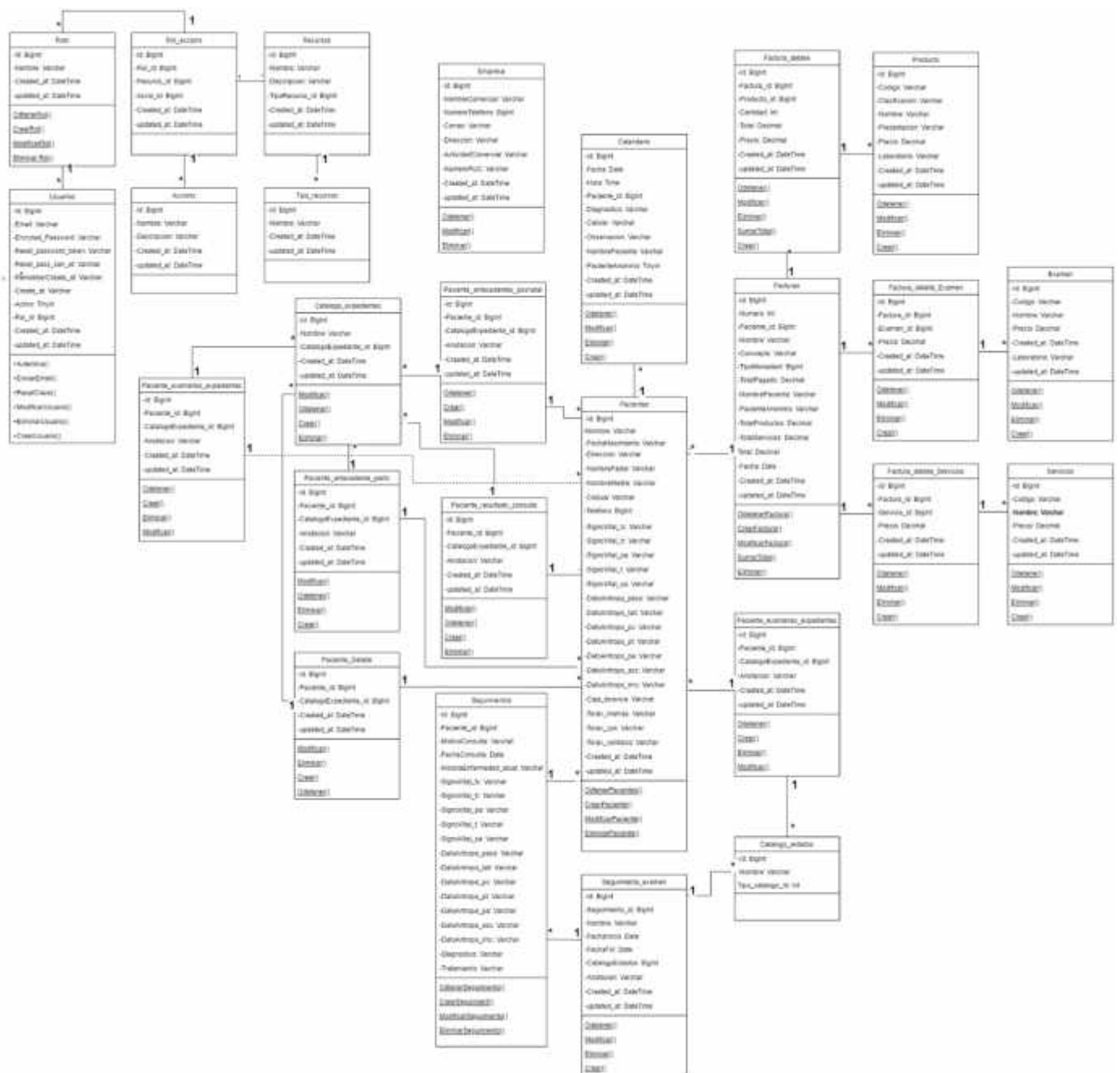
Figura 24: Crear factura.

#### 7.2.2.4 *Modelo conceptual.*

Los principales elementos de modelado UML utilizados en el modelo conceptual son: clase, asociación y paquete. Estos se representan gráficamente mediante la notación UML (1999). Podemos expresar estos diagramas como una representación gráfica que sirve para representar la estructura de un sistema que será implementado utilizando un lenguaje orientado a objetos.

##### 7.2.2.4.1 Diagrama de clases del sistema (SIM Cristina).

Descripción: En el diagrama de clases del sistema (SIM Cristina) se muestra la estructura mostrando las clases del sistema junto a sus atributos, operaciones, métodos y la relación entre los objetos.



#### 7.2.2.5 Modelo de presentación.

##### 7.2.2.5.1 Diagramas de presentación.

A continuación, se presentarán los bosquejos de diseño de cada vista que compondrá al prototipo de sistema web SIM-Cristina

The mockup shows a login interface for 'SIM-Cristina'. It features a title 'Autenticacion de usuario.' followed by the instruction 'Ingresa tu usuario y contraseña'. There are two input fields: one for 'Correo' (Email) with an '@' icon and a placeholder 'Ingresa tu correo contraseña', and another for 'Contraseña' (Password) with a placeholder 'Ingresa tu correo contraseña'. A blue 'Autenticar' button is positioned below the fields. At the bottom, there is a link that says '¿Has olvidado tu contraseña?'.

Figura 26: Vista de inicio de sesión.

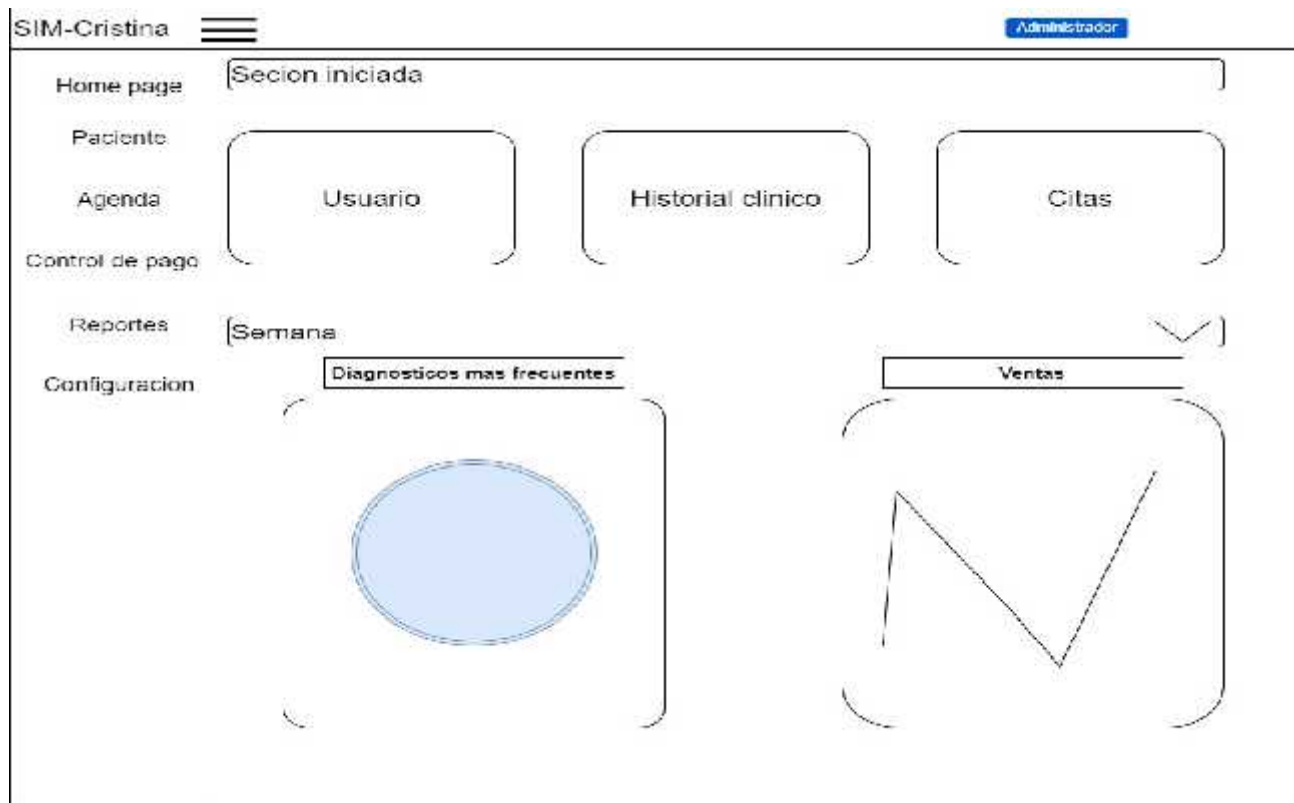


Figura 27: Vista del menú principal

SIM-Cristina

Administrador

Home page

Paciente

Agenda

Control de pago

Reportes

Configuración

Filtros

Historiales clinicos registrados

ID	Nombre	Fecha de	Nombre padre	Nombre madre	Fecha de	Fecha de	Acciones	Seguimiento
							<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>	<div>Seguimiento</div> <div>Cita</div>
							<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>	<div>Seguimiento</div> <div>Cita</div>
							<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>	<div>Seguimiento</div> <div>Cita</div>

Figura 28: Historial clínico

<div>Expediente PDF</div> <div>Crear seguimiento</div>		
Historial clínico	Seguimientos	Citas
<div>Datos del paciente</div> <div>Datos generales del paciente clínico</div> <div>Antecedentes del parto</div> <div>Antecedentes posnatales</div> <div>Examen físico</div> <div>Signos vitales</div>		

Figura 29: Formato historial clínico.

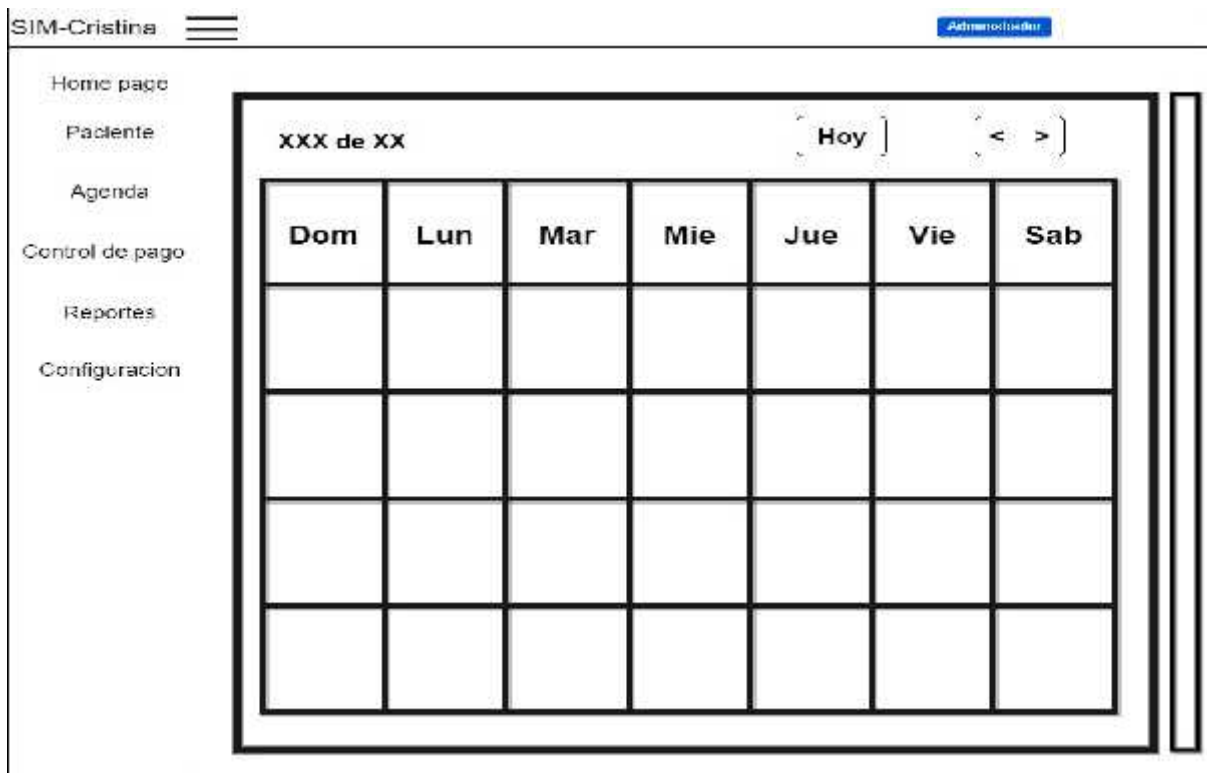


Figura 30: Agenda de citas

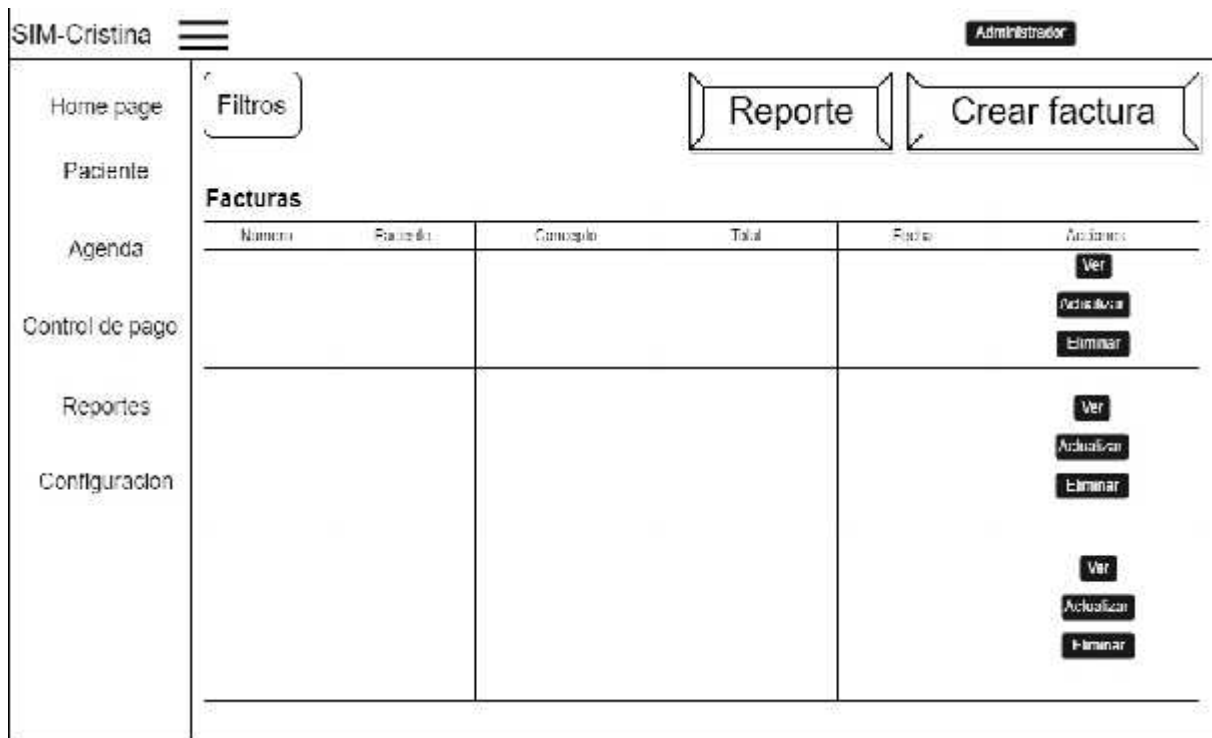


Figura 31: Control de pago





SIM-Cristina

Administrador

Home page

Paciente

Agenda

Control de pago

Reportes

Configuración

Información del rol

Nombre

Modulos del sistema

Pacientes	Citas	Facturacion	Reporteria	Configuracion
Recursos	Lectura	Escritura	Actualizar	Eliminar
Historial clinico				

Atas

Guardar

Figura 34: Creación de roles

SIM-Cristina

Administrador

Home page

Paciente

Agenda

Control de pago

Reportes

Configuración

Filtros

Reporte

Crear usuario

Usuarios registrados

ID	Correo	Nombre	Activo	Creado el	Actualizado el	Acciones
						<div><div>Ver</div><div>Actualizar</div><div>Eliminar</div></div>
						<div><div>Ver</div><div>Actualizar</div><div>Eliminar</div></div>
						<div><div>Ver</div><div>Actualizar</div><div>Eliminar</div></div>

Figura 35: Lista de usuarios

SIM-Cristina

Administrador

Home page

Paciente

Agenda

Control de pago

Reportes

Configuracion

Filtros

Crear reporte

Citas registradas

Fecha	paciente	hora	Diagnostico	Observaciones	Acciones
					<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>
					<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>
					<div>Ver</div> <div>Actualizar</div> <div>Eliminar</div>

Figura 36: Lista de citas registradas

SIM-Cristina

Administrador

Home page

Paciente

Agenda

Control de pago

Reportes

Configuracion

Filtros

Logs registrados

ID recurso	Recurso	Usuario	Acción	Fecha	Acciones
					<div>Ver</div>
					<div>Ver</div>
					<div>Ver</div>

Figura 37: Lista de logs

#### 7.2.2.5.2 Modelo arquitectónico.

Descripción: El diagrama de modelo arquitectónico de sistema (SIM Cristina) consta de una arquitectura sencilla y fácil de implementar.

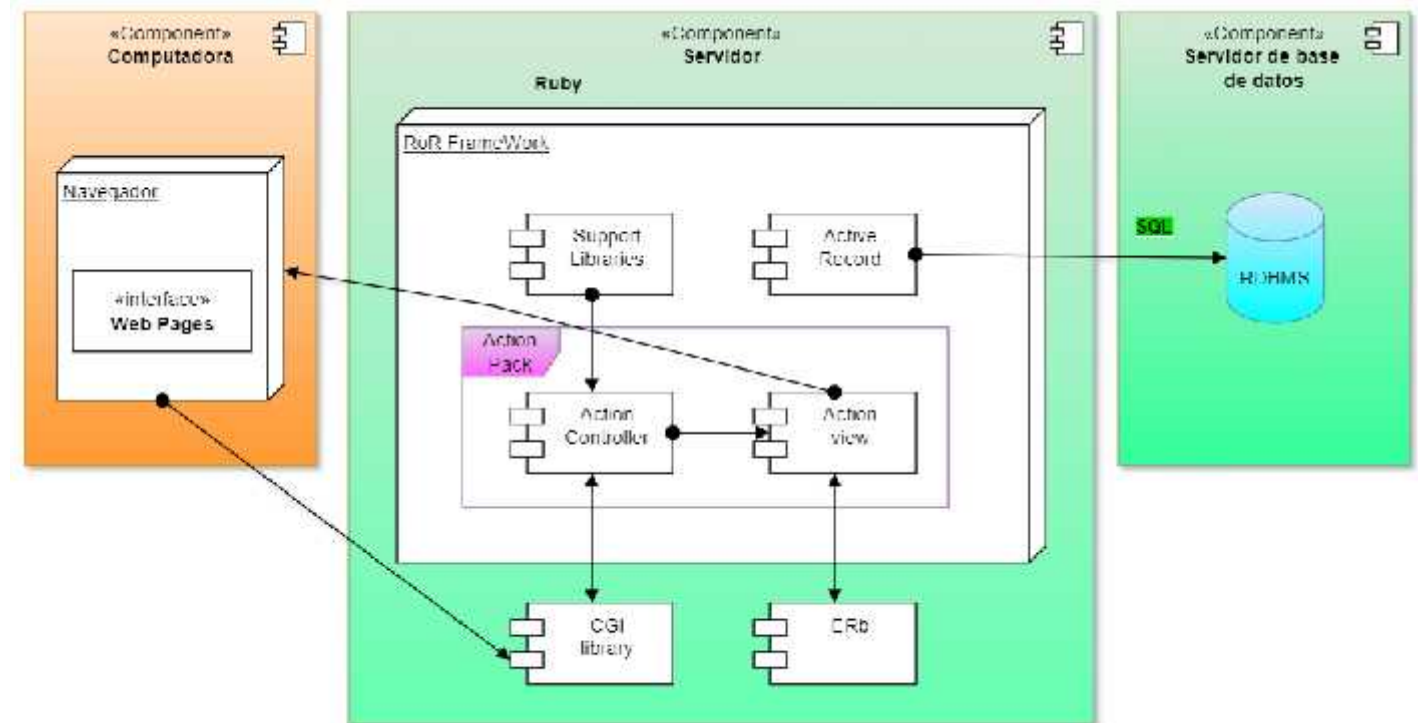


Figura 38: Modelo arquitectónico

### 7.2.3 Fase 3: Codificación del software

#### 7.2.3.1 Lenguaje de programación y Framework.

El lenguaje de programación para la creación del Web Server es Ruby utilizando Visual Studio code.

##### 7.2.3.1.1 Instalación de visual estudio code.

Para realizar la instalación de Visual Studio code se requiere ingresar a su página oficial <https://code.visualstudio.com>, luego se procede a realizar la descarga del instalador para Windows y lo ejecutamos.

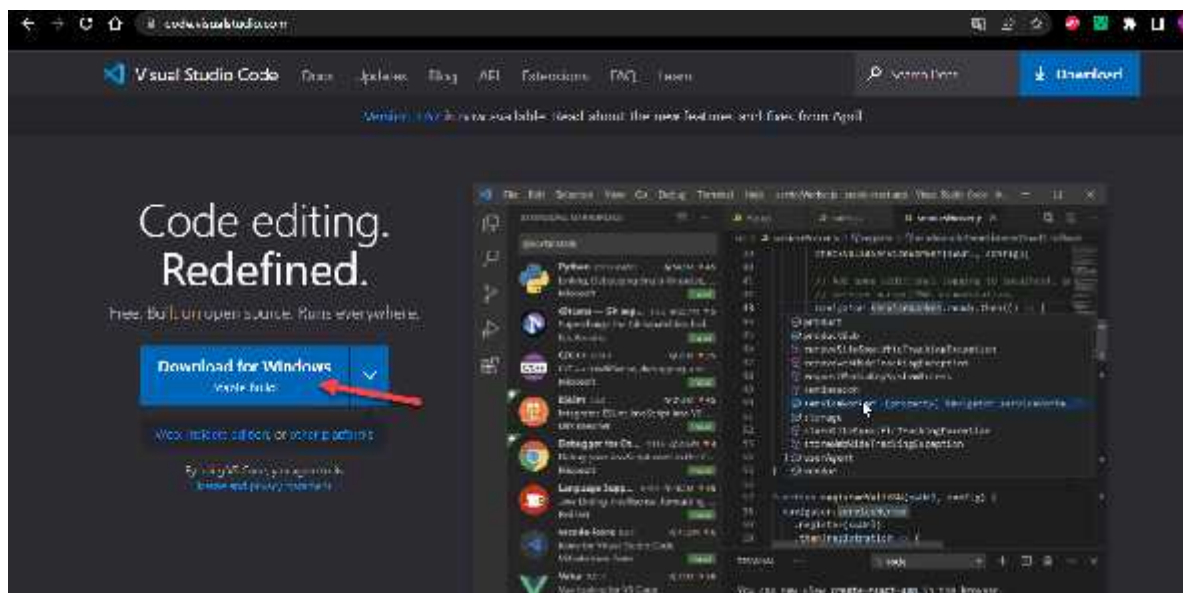
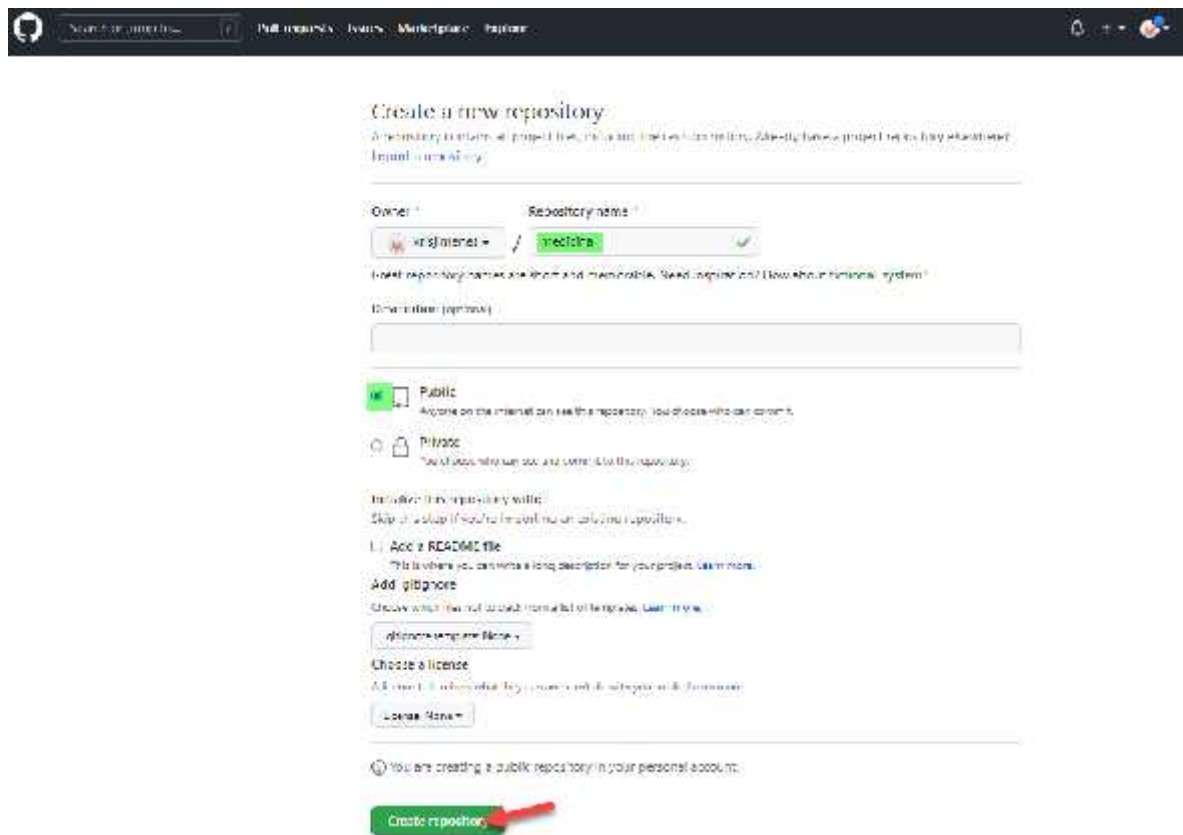


Figura 39: Descargar Visual Studio Code.

```
cerberus@DESKTOP-SJHVPGA:~/workspaces$ pwd
/home/cerberus/workspaces
cerberus@DESKTOP-SJHVPGA:~/workspaces$ mkdir medicina
```

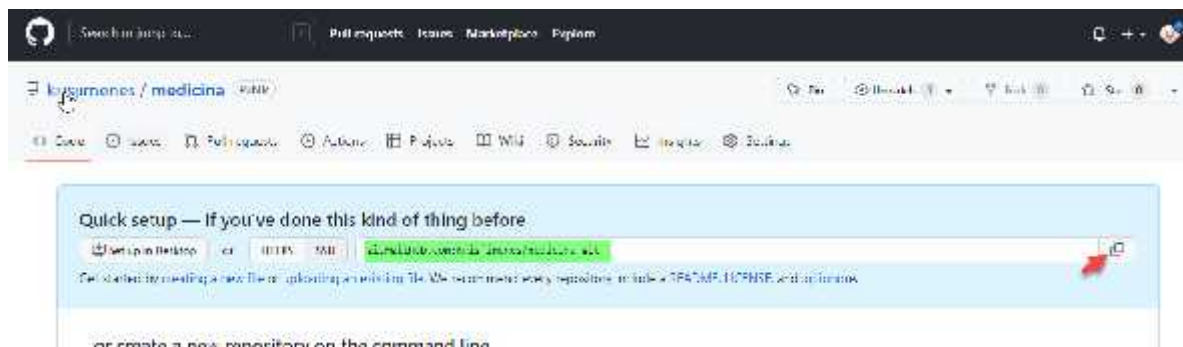
Figura 40: Creación del directorio con el nombre del proyecto.

previamente a la instalación de Visual Studio Code creamos he inicializamos el repositorio donde se contendrá el proyecto.



**Figura 41: Creación del repositorio.**

Ingresamos con nuestra cuenta de GitHub y creamos el repositorio medicina a como lo indica la imagen.



**Figura 42: Dirección del repositorio creado.**

Luego de haberlo creado copiamos la dirección para poder clonar el repositorio en nuestra carpeta que creamos previamente para de esta manera ya tenerlo inicializado en la rama por defecto.

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina$ pwd
/home/cerberus/Workspaces/medicina
cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina$ git clone git@github.com:krisjimenes/medicina.git
Cloning into 'medicina'...
Enter passphrase for key '/home/cerberus/.ssh/id_ed25519':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

```

Figura 43: Clonación del proyecto.

Luego de haber clonado el repositorio procedemos a la instalación de Ruby versión 2.7.2 que es en la que se desarrollará el proyecto para esto ejecutamos los siguientes comandos en el terminal.

Ejecutamos el repositorio de clave.

```

gpg --keyserver keyserver.ubuntu.com --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3
7D2BAF1CF37B13E2069D6956105BD0E739499BDB

```

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina$ gpg --keyserver keyserver.ubuntu.com --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
gpg: key 409B6B1796C275462A1703113804BB82D39DC0E3: public key is not available: No such file or directory
gpg: key 7D2BAF1CF37B13E2069D6956105BD0E739499BDB: public key is not available: No such file or directory
gpg: no valid OpenPGP data found.

```

Figura 44: Repositorio de claves.

Descargamos el paquete de Ruby a instalar.

```
\curl -sSL https://get.rvm.io | bash
```

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina/medicina$ \curl -sSL https://get.rvm.io | bash
Downloading https://github.com/rvm/rvm/archive/master.tar.gz

```

Figura 45: Descarga del paquete Ruby.

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina/medicina$ rvm install 2.7.2

```

Figura 46: Instalación de RUBY ON RAILS 2.7.2.

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina/medicina$ apt-get update && apt-get install -y build-essential

```

Figura 47: Instalación del paquete build essential.

```

cerberus@DESKTOP-SJHVPGA:~/Workspaces/medicina/medicina$ bundle install

```

Figura 48: Instalación del bundle.

Una vez finalizada toda la instalación y la preparación del ambiente procedemos a abrir el proyecto

creado en Visual Studio Code, donde encontraremos la siguiente estructura.

```
carberus@DESKTOP-SJHVPGA:~$ cd workspaces
carberus@DESKTOP-SJHVPGA:~/workspaces$ ls
medicina
carberus@DESKTOP-SJHVPGA:~/workspaces$ cd medicina
carberus@DESKTOP-SJHVPGA:~/workspaces/medicina$ ls
Dockerfile  Procfile  app  config.ru  lib  postcss.config.js  spec  tmp
Gemfile     README.md bin  db         log  public            storage vendor
Gemfile.lock Rakefile  config  docker-compose.yml  makefile  puma_entrypoint.sh  test
carberus@DESKTOP-SJHVPGA:~/workspaces/medicina$ code ..
```

Figura 49: Abriendo la ruta del proyecto en Visual Studio Code.

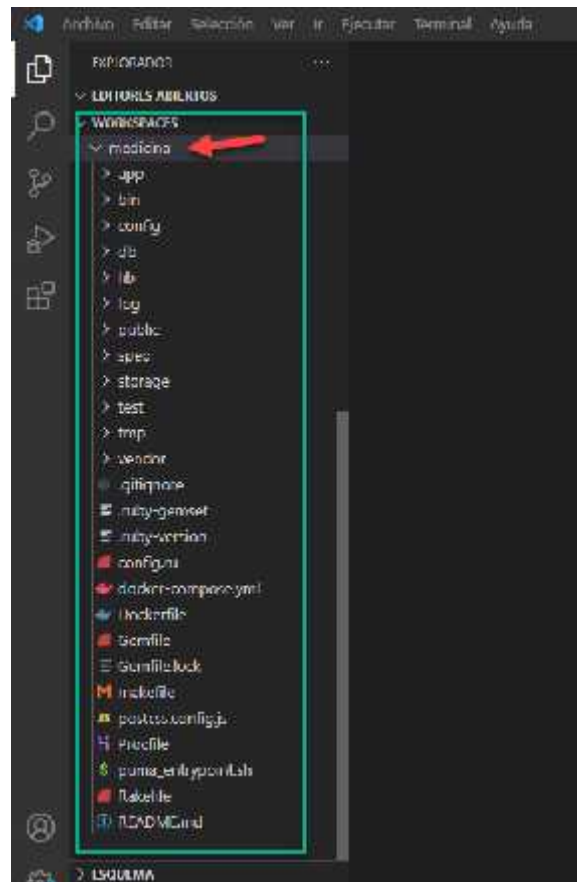


Figura 50: Estructura del proyecto.

#### 7.2.3.1.2 Bases de datos.

La base de datos empleada para el almacenamiento de la información fue MySQL debido a su gran escalabilidad y su confiabilidad, además de ser un proyecto de software libre con mucha información y documentación, además la empresa (Oracle Corporation) ya está familiarizada con el uso de este gestor de base de datos.

Se muestra el modelo EER (Entidad Relación) de la creación de la base de datos implementada en el

sistema donde se aprecia todas las tablas y relaciones necesarias para verlo en MySQL se agrega la instancia creada en el despliegue ( [Figura 70: Configuración de la instancia de la DB.](#)).

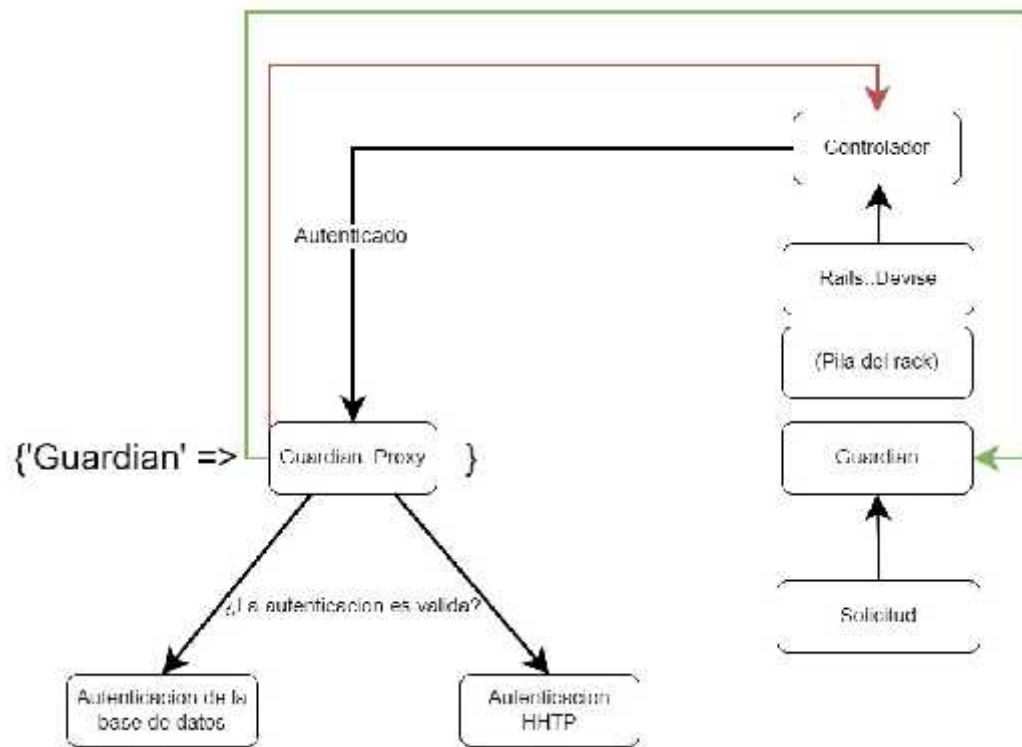




#### **Devise**

Para protección y seguridad de los datos del sistema se optó por usar una autenticación implementada mediante la gema Devise que mediante diferentes módulos proporciona diversos métodos de seguridad como son:

- ) database\_authenticatable: Los usuarios se van a poder autenticar con un nombre de usuario y contraseña almacenado en la base de datos.
- ) Registrable: Los usuarios van a poder registrarse, actualizar y eliminar su perfil.
- ) Recoverable: Los usuarios van a poder recuperar su contraseña
- ) Rememberable: Habilita la opción "Recordarme" en el login
- ) Trackable: Habilita el seguimiento al usuario (de dónde se autenticó, cuantas veces, de qué IP, etc.)
- ) Validatable: Valida el email y el password.



**Figura 52: Flujo de acceso a la web con Devise gema.**

1. La solicitud HTTP entra en la pila de rack.
2. El guardián recibe la solicitud se envía al controlador de Rails, que puede llamar a `¡autenticación_usuario!` de un filtro. Este es un alias para `request.env['warden'].authenticate! (:scope => :usuario)`.
3. El proxy del guardián elige una estrategia de autenticación true si es verdadero.
4. Cuando la autenticación se realiza correctamente, se devuelve un objeto de usuario al controlador. Cuando falla, el símbolo: Guardian se tira por la pila y es atrapado por la aplicación de rack de guardián. Este último devolverá una respuesta, que es una redirección a la página de inicio de sesión de forma predeterminada. Esto se puede anular llamando a `warden.custom_response!`.

```

app > models > usuario.rb > Usuario < ApplicationRecord
1  class Usuario < ApplicationRecord
2    ACTIVO = 1
3    INACTIVO = 0
4    ADMIN = 1
5    # Include default devise modules. Others available are:
6    # :confirmable, :lockable, :timeoutable, :trackable and :omniauthable
7    devise :database_authenticatable, :registerable,
8           :recoverable, :rememberable, :validatable
9

```

Figura 53: Seguridad de en la conexión de la DB usando Devise gema.

## Session Expiry

Las sesiones que nunca caducan amplían el plazo para ataques como la falsificación de solicitudes entre sitios, el secuestro de sesiones y la fijación de sesiones.

Una posibilidad es establecer la fecha de caducidad de la cookie con el ID de sesión, para implementar esta opción usamos Devise gema. Sin embargo, el cliente puede editar las cookies que se almacenan en el navegador web para que la expiración de sesiones en el servidor sea más segura y de esta forma establecer cierto tiempo de inactividad antes de que la sesión se cierre y evitar que los datos a los que el usuario tiene acceso sean manipulados por otro usuario.

```

> environments 41
initializers 42
  application_controller_re... 43
  assets.rb 44
  audited.rb 45
  backtrace_silencers.rb 46
  content_security_policy.rb 47
  cookies_serializer.rb 48
  core_ext.rb 49
  devise.rb 50
  filter_parameter_logging... 51
  inflections.rb 52
  kaminari_config.rb 53
  mime_types.rb 54
  require 'devise/orm/active_record' 55
  Devise.setup do |config| 56
    config.timeout_in = 1.hours 57
  end

```

Figura 54: Código de se Session Expiry aplicado al código.

## CAN CAN

CanCan es una biblioteca con enfoque en la autorización para Ruby y Ruby on Rails que concede accesos y bloquea recursos a los que un usuario determinado puede o no acceder.

Todos los permisos se pueden definir en uno o varios archivos de capacidad y no duplicarse entre controladores, vistas, y consultas de bases de datos, manteniendo su lógica de permisos en un solo lugar para facilitar el mantenimiento y las pruebas.

Consta de dos partes principales:

Biblioteca de autorizaciones que permite definir las reglas para acceder a diferentes objetos, y proporciona ayudantes para comprobar esos permisos.

Rails helpers para simplificar el código en Rails Controllers realizando la carga y comprobación de permisos de modelos automáticamente y reducir el código duplicado.

Métodos de seguridad de can can:

**Definir habilidades:** Los permisos de usuario se definen en una clase. CanCan 1.5 incluye un generador Rails 3 para crear esta clase.Ability.

**Comprobar capacidades y autorización:** Una vez inicializados, los permisos para el usuario actual se pueden comprobar utilizando ymethods en la vista y el controlador.can? cannot?. Una vez inicializados, los permisos para el usuario actual se pueden comprobar utilizando ymethods en la vista y el controlador.can?cannot?

**Manejar el acceso no autorizado:** Si se produce un error en la autorización del usuario, se generará una excepción. Puede detectar esto y modificar su comportamiento en el.CanCan::AccessDenied ApplicationController

**Bloqueado:** Si desea asegurarse de que la autorización se realiza en cada acción de la aplicación, agréguela a ApplicationController.check\_authorization.

```
class Ability
  include CanCan::Ability

  # Constantes del Menu
  RUTA = 1
  ICONO = 2

  def initialize(usuario)
    usuario ||= Usuario.new
    if usuario.rol_id == Rol::ADMINISTRADOR
      can :manage, :all
    else
      recurso = Recurso.all
      acciones_permitidas = RolAccion.where(rol_id: usuario.rol_id)
      validar_permiso_recurso(acciones_permitidas, recurso)
    end
  end
end
```

Figura 55: Uso de CAN CAN.

### 7.2.3.2 Respaldo de la información

Para poder resguardar los datos se habilita las copias de seguridad automatizadas que ofrece Amazon RDS.

Amazon RDS crea y guarda copias de seguridad automatizadas de la instancia de base de datos durante el periodo de copia de seguridad de la instancia de DB.

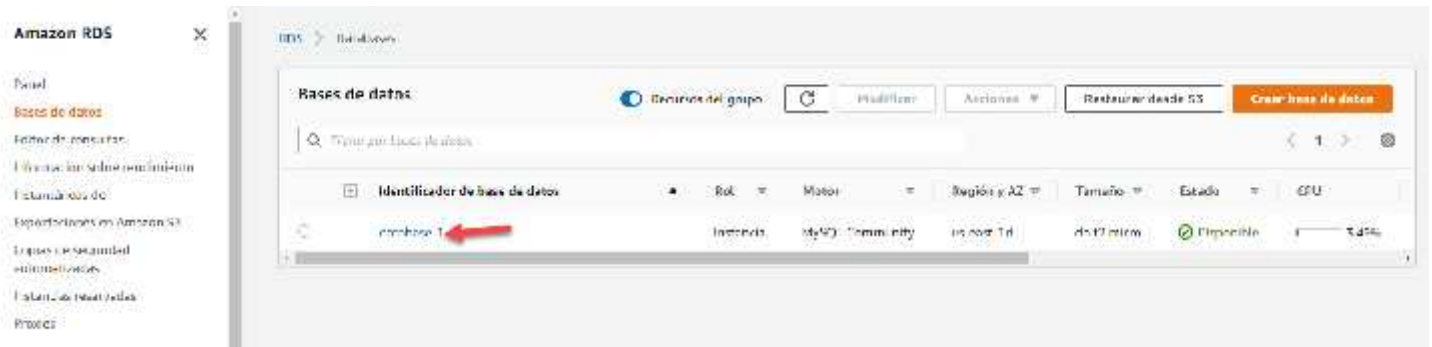


Figura 56: Instancia de la Base de datos.

Se crea una instantánea del volumen de almacenamiento de la instancia, creando una copia de seguridad de toda la instancia.

RDS guarda las copias de seguridad automatizadas de la instancia en función del periodo de retención de copia de seguridad especificado en este caso son 7 días. Con esta opción, es posible restaurar la base de datos a cualquier momento dado durante el periodo de retención de copia de seguridad.

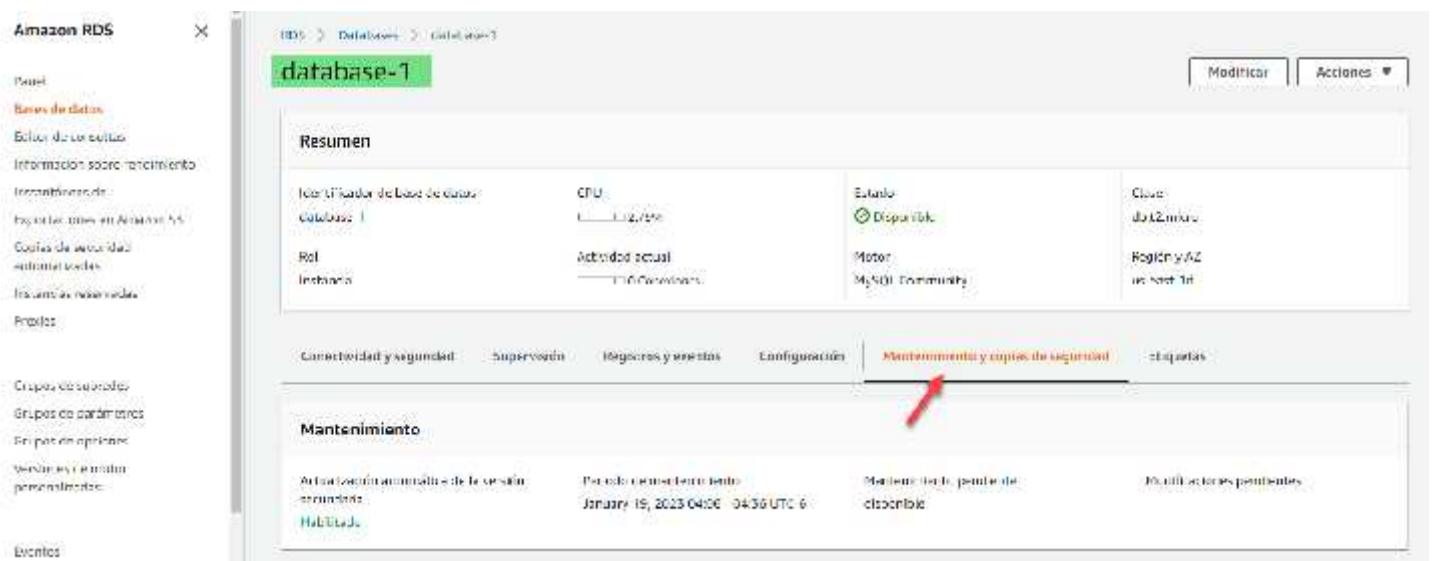


Figura 57: Vista de la administración de las copias de seguridad de la DB.





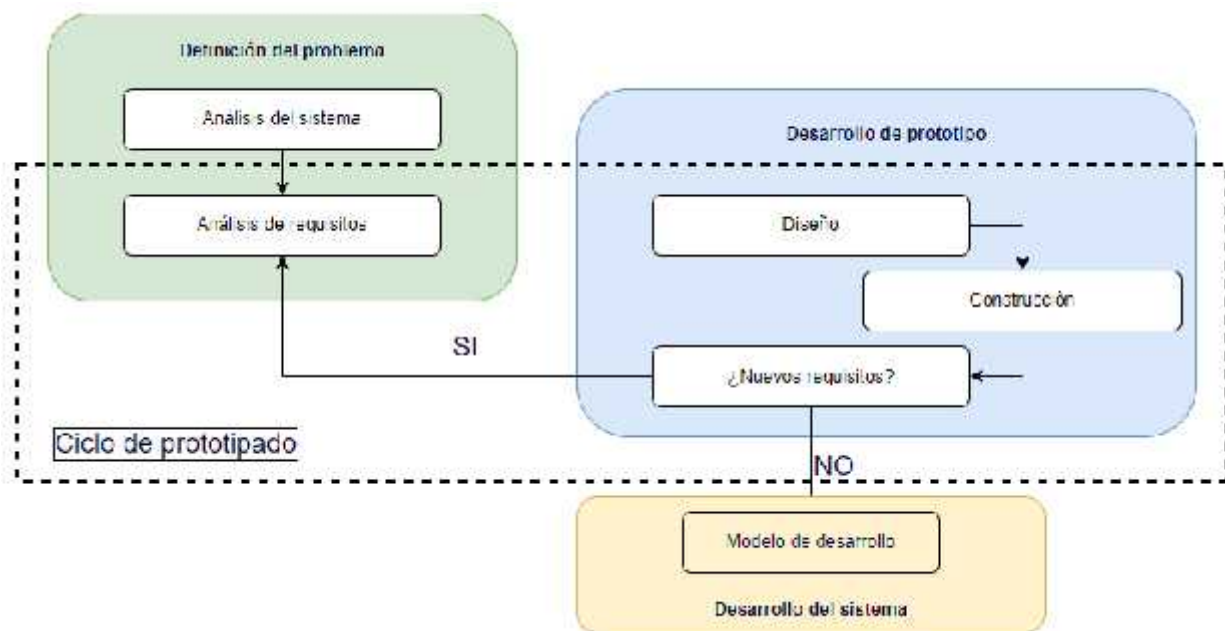
#### 7.2.4 Fase 4: Pruebas.

##### 7.2.4.1 Evolución del sistema.

En esta etapa se procederá con la ejecución de pruebas en cada módulo del sistema terminado, las cuales serán funcionales y unitarias para validar el correcto funcionamiento, del cual nos aseguraremos de no tener fallos.

##### 7.2.4.1.1 Modelo de prototipo evolutivo.

Se optó por el Modelo de prototipo evolutivo para el desarrollo del software debido a que fue estudiado en la clase de ingeniería de software y este permite que todo el sistema o algunas de sus partes, se construyan rápidamente de esta manera comprender o aclarar aspectos y estar más al pendiente de posibles carencias en los módulos en un tiempo temprano del desarrollo y así abastecer más en específico y al detalle las carencias de requerimientos del sistema porque se está en constante comunicación con el usuario final en cada una de sus etapas.



**Figura 59: Esquema básico del ciclo de prototipado.**



#### 7.2.4.1.2 Prototipo 1.

En el prototipado 1 se crearon las siguientes implementaciones:

- ) Página de inicio de sesión y recuperación de contraseña, se crearon las vistas, funcionalidades para iniciar sesión y recuperación de contraseña mediante correo electrónico.
- ) Página de inicio tras haber iniciado sesión, se implementó el home page donde se muestran card view informativos de la cantidad de usuarios, historiales clínicos y citas.
- ) Menú lateral izquierdo con la navegación de los menús (Home page, Pacientes, Agenda, control de pago, Configuración) y sub menús del sistema.
- ) Sección de pacientes crear historial clínico, creación del formulario de registro del paciente, se creó el CRUD del historial clínico cuando un paciente es nuevo.
- ) Creación y configuración de roles de usuario, en esta sección se crearon las funcionalidades y formularios del CRUD en los roles y restricción de los mismos en el sistema.
- ) Creación y administración de usuarios del sistema, se implementó las funcionalidades del CRUD de usuarios con su rol.
- ) Se creó la funcionalidad y vista de un log informativo de las acciones realizadas en el sistema.

#### 7.2.4.1.3 Prototipo 2.

En el prototipado 2 se crearon las siguientes mejoras en base a los comentarios y pruebas por el usuario y tutor realizadas al prototipo 1 al igual que las nuevas implementaciones:

- ) Se agregó la mejora de crear un filtro en la vista del paciente, de esta manera hay más exactitud al momento de las búsquedas.
- ) Se agregó la mejora en la sección de paciente, se creó la opción de añadir seguimiento y cita a los pacientes ya existentes al igual de agendamiento a una cita, también se implementó mejora a la vista el historial del paciente creando las secciones de seguimientos y citas del paciente.
- ) Se creó la vista de la agenda y sus funcionalidades CRUD se implementaron validaciones para fechas y hora de citas ya ocupadas.
- ) En el Menú lateral la opción de reportera con su submenú Citas, Seguimientos y se alojó la vista de los logs.
- ) Se creó el formulario de configuración de los datos de la empresa.

#### 7.2.4.1.4 Prototipo 3.

En el prototipado 3 se crearon las siguientes mejoras en base a los comentarios y pruebas por el usuario y tutor realizadas al prototipo 2 al igual que las nuevas implementaciones:

- ) Se creó la mejora en el historial agregando la opción de exportar el historial y los seguimientos de los pacientes.
- ) Se realizó la mejora de poder agendar citas para pacientes que no están en sistema, igualmente las validaciones de no poder agendar citas si ya la hora y fecha está agendada.
- ) El CRUD para los productos de la farmacia.
- ) La sección de control de pago, en esto se muestra el total de las ventas hasta del día. Se añadieron las acciones CRUD para la elaboración de una factura.
- ) La exportación de la factura creada actual como de cualquier factura en sistema.
- ) Los filtros en las secciones de paciente, agenda, pago, productos y reportes en general.

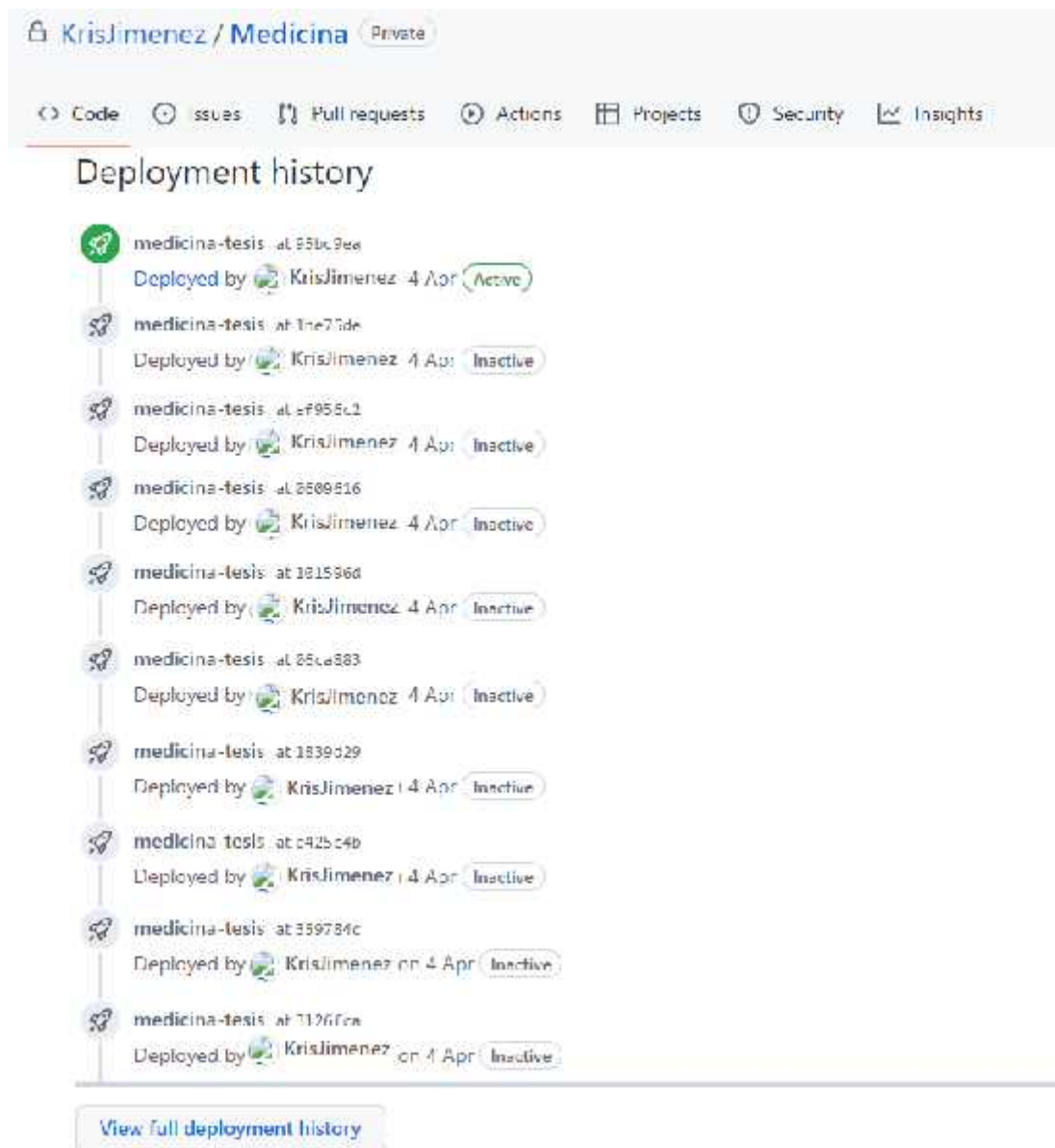
#### 7.2.4.1.5 Prototipo 4 y Final.

En el prototipado 4 y final se realizaron las correcciones de errores y mejoras en el sistema en base a los comentarios del usuario final las cuales fueron:

- ) Se crearon mejoras en el diseño de todas las vistas y menú del sistema.
- ) Se implementó paginación y segmentación a todas las tablas y listas informativas.
- ) En el home page se agregó una gráfica correspondiente a las enfermedades para visualizar en porcentaje estadístico de cual son las más recurrentes.
- ) En el home page se agregó grafica de tiempo de los ingresos totales del control de pago del día el cual se filtra por mes y semana.
- ) En el menú se creó la opción de catálogo y se dividió en tres sesiones los cuales son los productos, consultas y exámenes.
- ) Para las secciones del catálogo se creó también la tabla informativa de las consulta y exámenes de la misma manera se creó el CRUD de estas secciones del sistema.
- ) En la creación de facturas se dividieron los filtros para agregar productos, consultas y exámenes y de esta manera se suma el total final.
- ) En la vista de facturas creadas se agregó una visualización de total venta del día.

#### 7.2.4.2 Control de versiones.

Se utilizó GitHub para la gestión y el control de los cambios realizado en el código a lo largo de todo el proceso de creación de los prototipos y evolución del proyecto antes expuesto.



**Figura 60: Control de versiones - Cambios realizados en los prototipos.**

#### 7.2.4.3 Pruebas Unitarias.

Para la creación de pruebas unitarias se utilizó RSpec el cual es una gema que se usa para testear las aplicaciones de Ruby on Rails a continuación se presentan los casos de pruebas creados y los resultados. En dónde se logra constatar que los tiempos de respuestas y validación de métodos son eficientes y satisfactorios para un desempeño óptimo del sistema.

```
cerberus@DESKTOP-SJHVPGA:~/workspaces/unit_test_medicina$ rspec -f d
Finished in 1 minute 20.11 seconds (files took 1.25 seconds to load)
50 examples, 0 failures
```

**Figura 61: Suite de pruebas Unitarias con el comando rspec -f d.**

Se realizaron un total de 24 test donde en conjunto se ejecutaron 50 validaciones a los procesos claves para el funcionamiento del sistema, siendo:

- ) Validación del login.
- ) Recuperación de contraseña.

```
Test Login
#validación_logear
  Se espera validar que: Email o contraseña exista / inicie sesión.
  Se espera validar que: no se inicie sesión con Email o contraseña inválidos.
  Se espera validar que: no se inicie sesión con Email o contraseña Null.
#validación_Recuperar_contraseña
  Se espera validar que: se pueda recuperar la contraseña.
```

**Figura 62: Test del login.**

- ) creación de un paciente.
- ) modificación de paciente.
- ) Creación de seguimiento.
- ) Modificación de seguimiento.
- ) Eliminación de paciente.
- ) Exportación de reporte del historial del paciente.

```

Test Paciente
#validación_crear_paciente
  Se espera validar que: el paciente no exista.
  Se espera validar que: no haya campos requeridos en Null.
  Se espera validar que: se cree el paciente.
#validación_modificar_paciente
  Se espera validar que: se obtengan datos de el paciente a modificar.
  Se espera validar que: no haya campos requeridos en Null.
  Se espera validar que: se modifique el paciente.
#validación_crear_seguimiento_paciente
  Se espera validar que: no haya campos requeridos en Null.
  Se espera validar que: se agregue el seguimiento al historial del paciente.
#validación_modificar_seguimiento_paciente
  Se espera validar que: no haya campos requeridos en Null.
  Se espera validar que: se agregue el seguimiento al historial del paciente.
#validación_eliminar_paciente
  Se espera validar que: se elimine el historial del paciente.
#validación_exportar_reporte_paciente
  Se espera validar que: se pueda exportar el historial del paciente.

```

Figura 63: Test paciente.

- ) Creación de cita.
- ) Editar cita.
- ) Eliminar cita.

```

Test Agenda
#validación_crear_cita
  Se espera validar que: la fecha y hora no este ocupada.
  Se espera validar que: los campos obligatorios no sean aceptados en Null.
  Se espera validar que: se cree la cita.
#validación_editar_cita
  Se espera validar que: la fecha y hora a cambiar no este ocupada.
  Se espera validar que: los campos obligatorios no sean aceptados en Null.
  Se espera validar que: se modifique la cita.
#validación_eliminar_cita
  Se espera validar que: la cita se elimine.

```

Figura 64: Test agenda.

- ) creación de producto.
- ) Edición de producto.
- ) Eliminación producto.



```

Test Catalogo_producto
#validación_crear_producto
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se cree el producto.
    Se espera validar que: el identificador del producto no exista.
#validación_editar_producto
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se modifique la el producto.
#validación_eliminar_producto
    Se espera validar que: la el producto se elimine.

```

Figura 65: Test Catalogo producto.

- ) Creación de servicio.
- ) Edición de servicio.
- ) Eliminación de servicio.

```

Test catalogo_servicio
#validación_crear_servicio
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se agregue el servicio.
    Se espera validar que: el identificador del servicio no exista.
#validación_editar_servicio
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se modifique el servicio.
#validación_eliminar_servicio
    Se espera validar que: el servicio se elimine.

```

Figura 66: Test Catalogo servicio.

- ) Creación de examen.
- ) Edición de examen.
- ) Eliminación de examen.

```

Test catalogo_examen
#validación_crear_examen
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se agregue el examen.
    Se espera validar que: el identificador del examen no exista.
#validación_editar_examen
    Se espera validar que: los campos obligatorios no sean aceptados en Null.
    Se espera validar que: se modifique el axemen.
#validación_eliminar_examen
    Se espera validar que: el axemen se elimine.

```

Figura 67: Test catalogo examen.

- ) creación de registro de pago.
- ) Edición de registro de pago.

- ) Eliminación de registro de pago.
- ) Exportación de registro de pago.

```
test control de pago
#validación crear registro.de.pago
  Se espera validar que: se pueda traer nombre de pacientes en sistema.
  Se espera validar que: los campos obligatorios no sean aceptados en Null.
  Se espera validar que: se sume el total general de los producto,servicios y exámenes.
  Se espera validar que: se cree el registro del pago.
#validación_editar_registro.de.pago
  Se espera validar que: los campos obligatorios no sean aceptados en Null.
  Se espera validar que: se sume el total general de los producto,servicios y exámenes modificados.
  Se espera validar que: se modifique el registro de pago.
#validación_eliminar_registro.de.pago
  Se espera validar que: el registro de pago se elimine.
#validación exportar registro.de.pago
  Se espera validar que: el registro de pago se permita exportar.

finished in 1 minute 20.12 seconds (files took 1.64 seconds to load)
50 examples, 0 failures
```

Figura 68: Test control de pago.

#### 7.2.5 Fase 5: Implementación y despliegue.

Una vez concluida la etapa de pruebas y de evaluación de los resultados estamos listos para implementar el sistema en un servidor, procedemos a realizar la implementación comenzando con la creación de una instancia de base de datos MySQL.

Usamos **Easy create** para crear una instancia de base de datos que ejecute MySQL con la Consola de administración de AWS.

Primero se inició sesión en la consola de administración de AWS y se abrió la consola de Amazon RDS en <https://console.aws.amazon.com/rds/>.

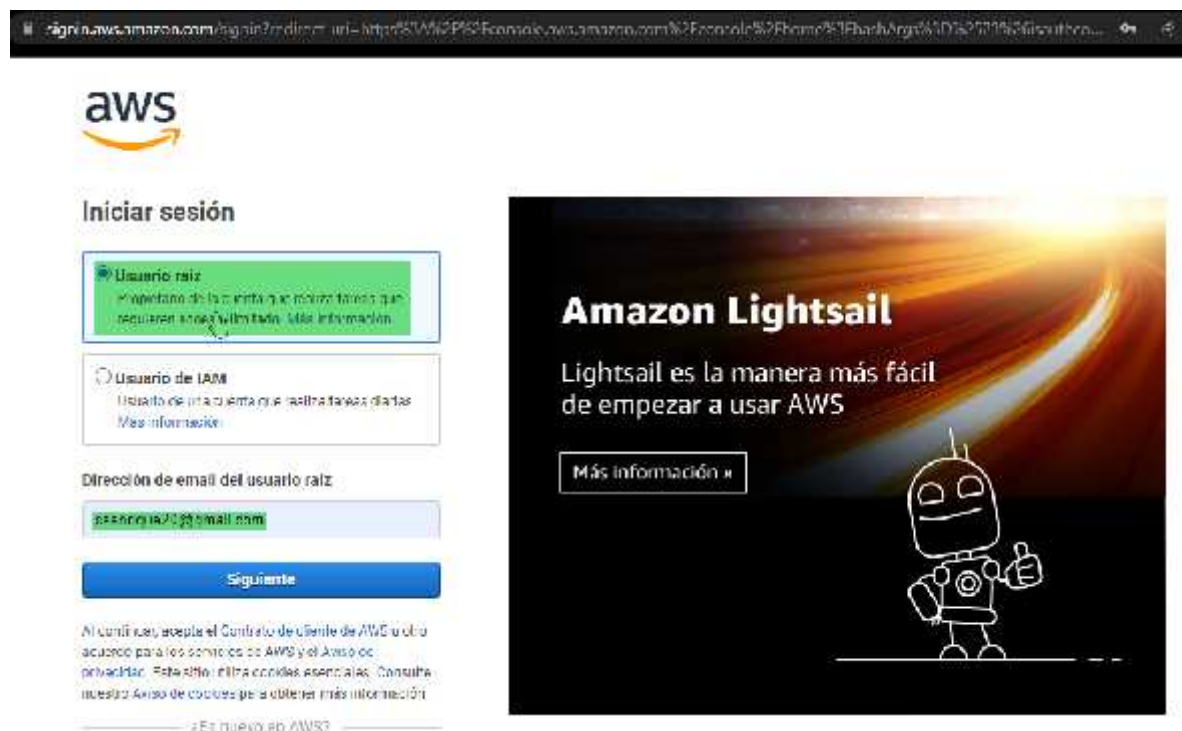


Figura 69: Inicio de sesión AWS.

En la esquina superior derecha de la consola de Amazon RDS se elige la región de AWS en la que desea crear la instancia de base de datos. En el panel de navegación, elegimos Bases de datos y crear base de datos seleccionamos la opción Creación fácil.

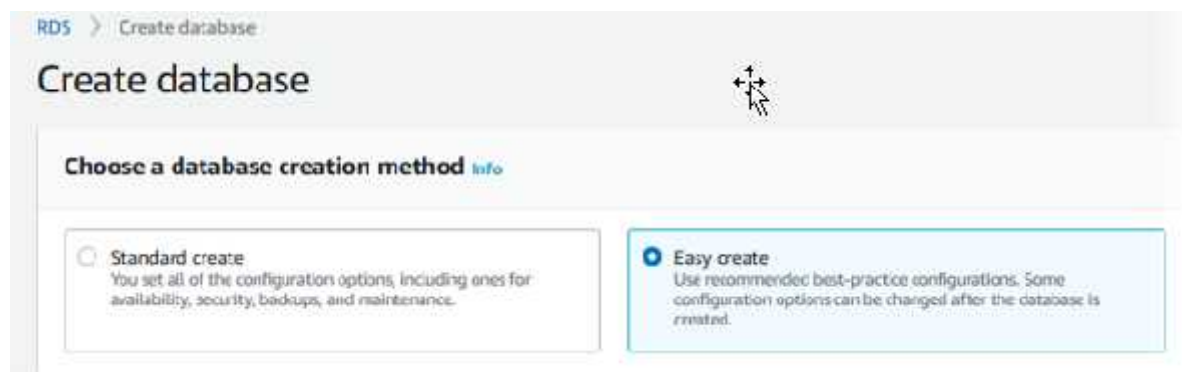



Figura 70: Selección de Easy create.


En Configuración, se eligió MySQL, para el tamaño de la instancia de base de datos, seleccionamos el nivel Gratis. Para el identificador de instancia de base de datos, ingresamos un nombre y un nombre de usuario maestro.





### Configuration


Engine type [Info](#)


☐ Amazon Aurora


☒ MySQL


☐ MariaDB


☐ PostgreSQL


☐ Oracle


☐ Microsoft SQL Server


DB instance size

☐ Production

db.r6g.xlarge  
4 vCPUs  
32 GiB RAM  
500 GiB

☐ Dev/Test

db.r6g.large  
2 vCPUs  
16 GiB RAM  
100 GiB

☒ Free tier

db.t2.micro  
1 vCPUs  
1 GiB RAM  
20 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)

Type a login ID for the master user of your DB instance.

admin

Figura 71: Configuración de la instancia de la DB.

En la lista Bases de datos se aprecia el nombre de la nueva instancia de base de datos MySQL

RDS > Databases > database-1

## database-1

[Modificar](#) [Acciones](#)

Resumen			
Identificación de base de datos database-1	CPU 6.27%	Estado Disponible	Clase db.t2.micro
Rol	Actividad actual 0 Conexiones	Motor MySQL Community	Región y AZ us-east-1d
Instancia			

Figura 72: Instancia de la DB en la lista.

Luego se procede a Implementar la aplicación en Heroku en el directorio raíz, el cual es una plataforma de servicios en la nube que permite manejar los servidores y sus configuraciones, escalamiento y la administración.

```
$ heroku create
Creating app... done, radiant-sands-01405
https://radiant-sands-01405.herokuapp.com/ | https://git.heroku.com/radiant-sands-01405.git
```

Figura 73: Comando para crear a heroku.

Procedemos a implementar el código a Heroku.

```
$ git push heroku main
remote: Compressing source files... done.
remote: Building source:
remote:
remote: ----> Building on the Heroku-20 stack
remote: ----> Determining which buildpack to use for this app
remote: ----> Ruby app detected
remote: ----> Installing bundler 2.2.21
remote: ----> Removing BUNDLED WITH version in the Gemfile.lock
remote: ----> Compiling Ruby/Rails
remote: ----> Using Ruby version: ruby-3.0.2
remote: ----> Installing dependencies using bundler 2.2.21
remote:      Running: BUNDLE_WITHOUT='development:test' BUNDLE_PATH=vendor/bundle BUNDLE_
remote:      Fetching gem metadata from https://rubygems.org/
remote:      Fetching gem metadata from https://rubygems.org/.....
remote:      Fetching rake 13.0.6
remote:      Installing rake 13.0.6
remote:      Fetching concurrent-ruby 1.1.9
remote:      Fetching minitest 5.14.4
remote:      Fetching erubi 1.10.0
remote:      Fetching builder 3.2.4
```

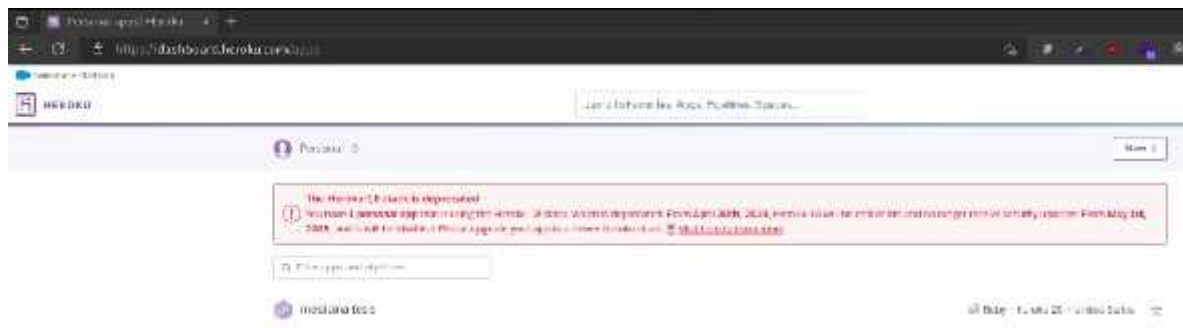
Figura 74: Comando para implementar el código en heroku.

Una vez terminada la implementación de código se procede a migrar la DB de nuestro proyecto con el comando: **heroku run rake db:migrate**.

Luego iniciamos la aplicación en el navegador con el comando: **heroku open**.

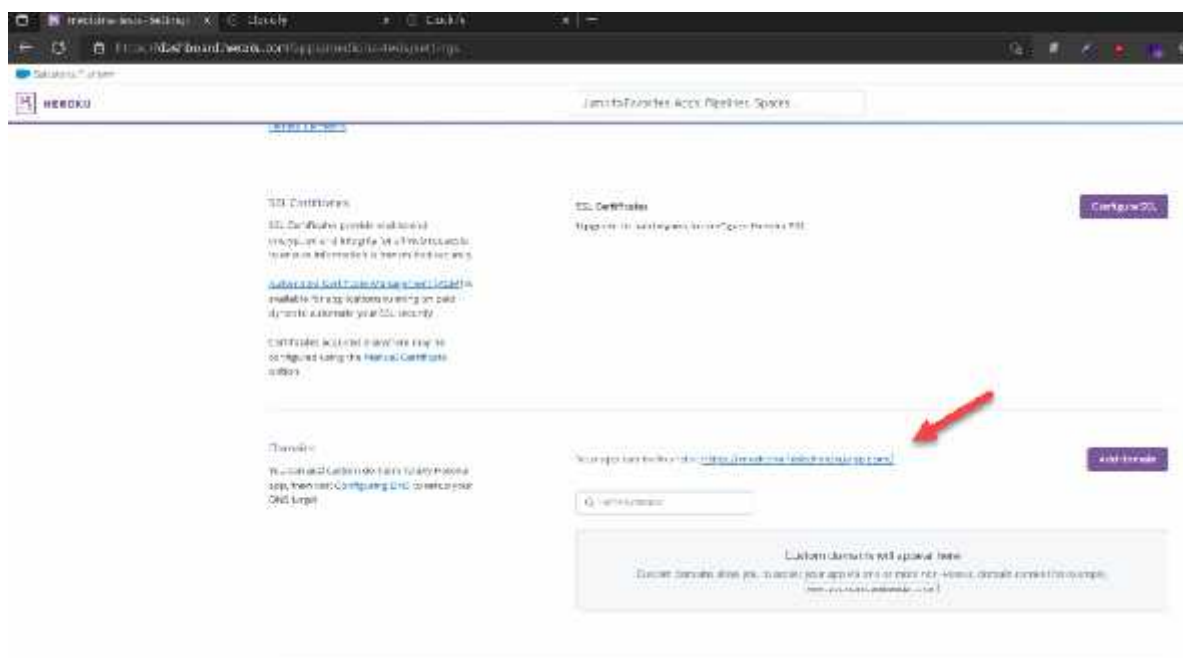
```
$ heroku open
```

Figura 75: Comando para iniciar heroku.



**Figura 76: Despliegue de heroku.**

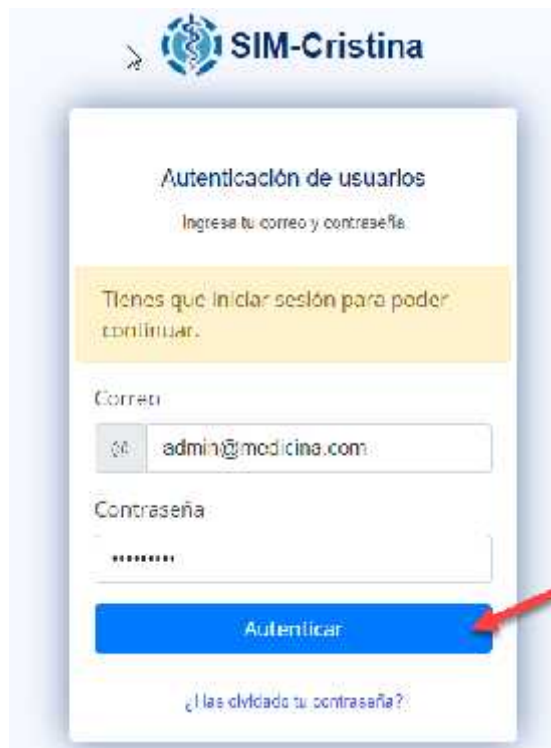
Heroku nos proporciona una URL web predeterminada para la aplicación.



**Figura 77: Generación de la URL para la aplicación.**

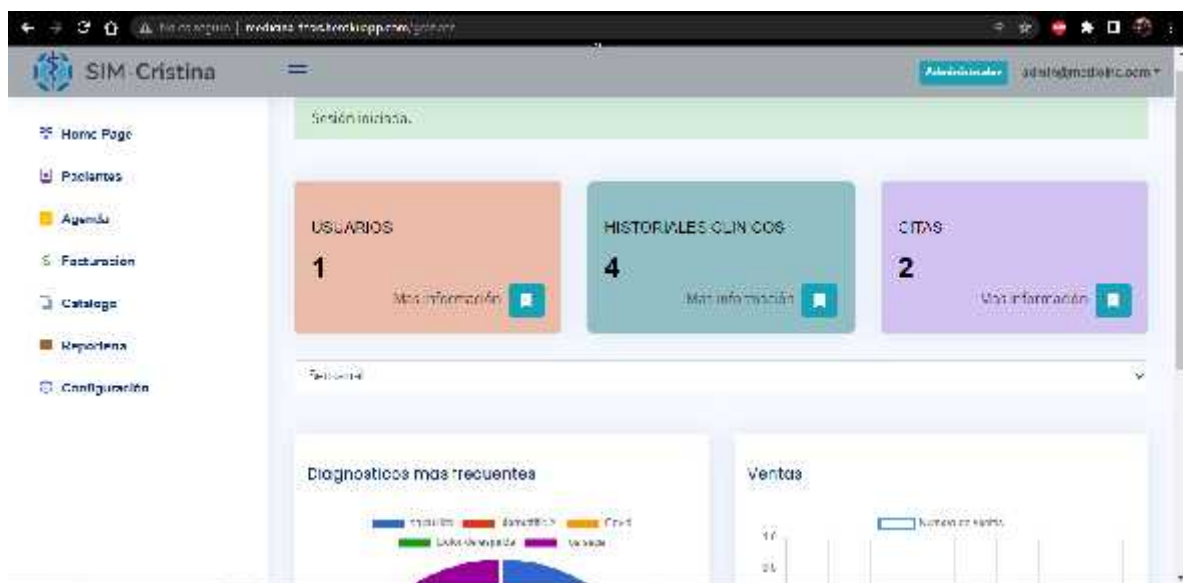
Una vez que obtenemos el link procedemos acceder a nuestra aplicación web.

<https://medicina-tesis.herokuapp.com>



**Figura 78: Sistema desplegado.**

Si iniciamos sesión entraremos al home page de la aplicación.



**Figura 79: Home page de la aplicación.**

## 7.3 Desempeño y facilidad de uso del sistema.

### 7.3.1 Desempeño.

A modo de prueba, se realizarán evaluaciones de la manera convencional que el proceso se realiza (estas pueden ser manualmente, o usando una hoja de cálculo de Excel y Word).

Actualmente, es usado por profesionales en hojas de cálculo en dónde ellos tienen que ingresar los datos manualmente en fórmulas que ya poseen diseñadas en Excel, lo que puede encontrarse un poco tedioso.

La medida del desempeño es el tiempo de respuesta que se requiere hacer para una determinada tarea o procedimiento.

- Tiempo de respuesta (T): tiempo requerido para generar un reporte, una función, y diferentes funcionalidades del sistema.  $T_{\text{nuevo}}, T_{\text{anterior}}$
- Razón de rendimiento (R): Expresa en qué medida el uso del nuevo sistema supera en rendimiento el método anterior.

**Fórmula 1:** razón de rendimiento

$$R = \frac{T_{\text{nuevo}}}{T_{\text{anterior}}} = \frac{T_{\text{anterior}}}{T_{\text{nuevo}}}$$

En donde, si el resultado es mayor a 1, entonces el sistema es mejor que el proceso antiguo. Si es igual a 1 entonces el sistema y el proceso son iguales. Y si es menor que 1 el sistema es inferior en rendimiento.

**Fórmula 2:** ventaja neta porcentual de rendimiento

$$n\% = (n - 1) * 100\%$$

Siendo n% el porcentaje de expresión en la medida que es mejor el proceso actual el sistema.

### 7.3.2 Facilidad de uso.

La facilidad de uso es la simplicidad de la herramienta en este caso el sistema, que solo se puede medir mediante pregunta directa al usuario en función de su percepción al momento de usarla.

Para medir la facilidad de uso lo haremos consultando con el usuario en donde se tomará en cuentas las medidas, esfuerzo y la simplicidad de uso.

A continuación, se detallan los aspectos a tomar en cuenta:

- **Esfuerzo:** cantidad de pasos de procedimientos necesarios para lograr una tarea determinada es expresada a través de una variable numérica entera.
- **Simplicidad:** grado de simplificación y adecuación al diseño que facilite la interacción con el sistema. Variable discreta con valores simple: simple, complejo.
- **La medida de la facilidad de uso:** es el esfuerzo, que en este contexto se puede cuantificar en función del número de pasos del procedimiento.

### Método de evaluación

El proceso de evaluación cuantitativa seleccionado fue el método de pruebas para desempeño y facilidad de uso, para esto se realizaron 3 casos de pruebas con los principales módulos del sistema, los ejecutores de las pruebas fueron los usuarios con los roles de doctor, secretaria y regente de farmacia, los cuales realizaron pruebas sobre el historial clínico, agenda de citas y facturación de productos respectivamente.

### Prueba 1: Historial clínico

Se realizará un historial clínico y un seguimiento, donde se guardarán los datos requeridos para un historial clínico completo.

### Prueba 2: Citas

Se realizará la recepción de una cita, se guardarán los datos para agendarla, asignarle una hora y una fecha determinada, y verla en el calendario.

### Prueba 3: Control de pago.

En esta prueba se registran compras en el módulo de control de pago, capturando los datos necesarios para realizar una compra, y posteriormente ser visualizado en la lista de facturas.

Estas pruebas se realizarán a partir de la estructura actual de trabajo de la clínica.

Se cronometró el tiempo y la cantidad de pasos que tomar realizar estas tareas, así como la simplicidad de uso que los usuarios perciben.

Los resultados obtenidos de las pruebas se tabularán para comparar y verificar si el sistema es viable.

### **Evaluación de desempeño**

La siguiente tabla resume los resultados de evaluación de la razón de rendimiento (n) y mejora neta porcentual de rendimiento (n%).

<b>Caso de prueba</b>	<b>Método convencional t(s)</b>	<b>Sistema SIM-Cristina t(s)</b>	<b>n</b>	<b>n%</b>
<b>Prueba 1</b>	400	230	1.7391	73.91%
<b>Prueba 2</b>	250	160	1.5625	56.25%
<b>Prueba 3</b>	140	60	2.3333	133.33%
<b>Global</b>	790	450	1.7555	75.55%

**Tabla 33: Resultados de evaluación de la razón de rendimiento**

Los resultados prueban que el uso del sistema SIM-Cristina tiene mejor desempeño que el método anterior de trabajo, debido a que el tiempo requerido para completar la misma operación varía a favor del sistema SIM-Cristina, además el indicador de mejora del rendimiento global muestra un 75.55 % mejora neta porcentual de rendimiento, reduciendo prácticamente a más de la mitad el tiempo el proceso de ejecución de las operaciones.

### **Evaluación de la facilidad de uso**

Para medir la facilidad de uso para los distintos usuarios fue necesario hacer un conteo de la cantidad de esfuerzo requerido (número de pasos) para realizar las pruebas asignadas y se tomó en cuenta la percepción que tuvo el usuario según el nivel de complejidad de cada caso.

Evolución de facilidad de uso						
Caso	Prueba NO	Esfuerzo (Pasos)		Mejora de facilidad	Simplicidad	
		Método anterior	Sistema		Método anterior	Sistema
1	Prueba 1	37	15	146.66%	Complicado	Simple
2	Prueba 2	5	3	66.66%	Normal	Simple
3	Prueba 3	5	3	66.66%	Normal	Simple

**Tabla 34: Resultados de evaluación de la funcionalidad de uso**

Los resultados obtenidos demuestran que el sistema SIM-Cristina requiere menos pasos de procedimientos para completar las operaciones a diferencia del método anterior. El índice de mejora de facilidad de uso fue de 146.66 % para el primer caso, para el segundo se obtuvo una mejora de 66.66 % y finalmente en el tercer y último caso se obtuvo una mejora sustancial del 66.66 %.

En cuanto a la percepción del usuario la evaluación demostró que para el caso 1 paso de ser de un proceso complejo a uno normal para el caso 2 se pasó de proceso normal a uno simple y finalmente el caso 3 paso de ser de media dificultad a uno completamente simple.



## 8 Conclusiones.

Como solución a la problemática expuesta en el documento de manera metodológica y exacta, se planteó un objetivo general y los objetivos específicos, los cuales en el transcurso del documento han sido cumplido y de los cuales llegamos a las siguientes conclusiones:

) Por medio de las técnicas de recolección de información y datos se nos fue posible el analizar en cada fase de los módulos el tener una visión clara de las necesidades del cliente, y en base a esas necesidades enfocarnos en un desarrollo mejor orientado.

) La implementación de la metodología en cada fase del desarrollo nos permitió mejorar más el desarrollo de este proyecto, descubriendo aún más detalles que los ya obtenidos previamente con el cliente y de igual manera la recopilación de los requerimientos funcionales y no funcionales.

) Para el desarrollo se utilizó Ruby on Rails, distintas gemas de este lenguaje como CanCan y Devise, así como MySQL, Bootstrap, JavaScript, JQuery tecnologías que permitieron un desarrollo sencillo y la creación de un ambiente amigable con el usuario final.

En conclusión, mediante las pruebas que se realizaron se observó que el tiempo en cada aspecto del desarrollo, implementación y uso del sistema, es un factor positivo sobre todo en este último aspecto. Según lo anterior mencionado se ha demostrado que cada objetivo se alcanzó y realizó con éxito a lo largo de este proyecto, se han adquirido y consolidado conocimientos, que luego se volvieron herramientas para lograr una finalización exitosa y un producto de valor para el cliente.

## 9 Recomendaciones.

Ante el desarrollo del sistema web consideramos las siguientes recomendaciones con el fin de que se tenga un funcionamiento y manejo óptimo del sistema para satisfacer las necesidades del cliente.

- ) Considerar el mantenimiento y evolución del software, con el fin de amplificar más módulos y funcionalidades el cual será ventajoso para la clínica y farmacia SIM\_Cristina.
- ) Mantener actualizado el manual técnico en base a futuros cambios que se le pueda aplicar al sistema.
- ) Realizar un upgrade a un plan de pago más sofisticado del Host que ofrece herokuapp para mejorar el rendimiento del sistema.
- ) Se propone la creación de una aplicación para Android y iOS con la misma funcionalidad que la aplicación web actualmente realiza, debido a que este sistema tiene un gran potencial de escalabilidad a futuro.

## 10 Bibliografía

- Alcalde, A. (1 de 10 de 2017). *El baúl del programador*. Obtenido de Diseño de Bases de Datos ( I ) - Conceptos del Modelo Relacional: <https://elbauldelprogramador.com/disenio-de-bases-de-datos-i-conceptos/>
- Alicante, U. d. (20 de 08 de 2012). *Servicio de InformáticaASP.NET MVC 3 Framework*. Obtenido de Modelo vista controlador (MVC): [https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20\(MVC\)%20es,control%20en%20tres%20comp onentes%20distintos.](https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20(MVC)%20es,control%20en%20tres%20comp onentes%20distintos.)
- Centenera, E. G. (febrero de 2017). *Sistema Web para la gestión de una Clínica Médica* . Obtenido de e-archivo.uc3m.es: [https://e-archivo.uc3m.es/bitstream/handle/10016/27020/TFG\\_Eduardo\\_Garcia\\_Centenera.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/27020/TFG_Eduardo_Garcia_Centenera.pdf)
- Ecured. (s.f.). *Ecured*. Obtenido de [https://www.ecured.cu/Modelo\\_de\\_prototipos](https://www.ecured.cu/Modelo_de_prototipos)
- G. Booch, J. R. (1999). *El Lenguaje Unificado de Modelado*. Addison Wesley.
- Galiano, L. (3 de 11 de 2012). *Metodología UWE aplicada a mi solución informática de mi proyecto*. Obtenido de <http://elproyectodeluigaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
- Gallego, J. C. (2010). *Mantenimiento de sistemas microinformáticos - técnicas*.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript* . Barcelona: MARCOMBO, S.A.
- GitHub*. (s.f.). Obtenido de <https://github.com/CanCanCommunity/cancancan>
- Gómez, A. (2007). *Access 7.0 para Windows*. Madrid, España: Pearson.
- Juárez, M. (2006). *Bases de Datos Relacionales*. Mexico: Pearson-mexico.
- lancker, L. V. (2015). *JQuery: El framework Java Script de la web 2.0*.
- Languages, O. (s.f.). *Definicion de Farmacia*. Obtenido de [https://www.google.com.ni/search?q=definicion+de+farmacia&sxsrf=ALeKk01PrpQddBlgf-RT1nmkaKLbeiERTQ%3A1621281763433&ei=48uiYJLnGYHH\\_Qa-krqQDA&oq=definicion+de+farma&gs\\_lcp=Cgdnd3Mtd2l6EAMYADIHCAAQRhD5ATICCAAyAggAMg](https://www.google.com.ni/search?q=definicion+de+farmacia&sxsrf=ALeKk01PrpQddBlgf-RT1nmkaKLbeiERTQ%3A1621281763433&ei=48uiYJLnGYHH_Qa-krqQDA&oq=definicion+de+farma&gs_lcp=Cgdnd3Mtd2l6EAMYADIHCAAQRhD5ATICCAAyAggAMg)

IIADICCAAyAggAMgIIADICCAAyAggAMgIIADoHCAAQ.

*People/CanCode*. (s.f.). Obtenido de <http://peoplecancode.com/es/tutorials/how-to-manage-users-with-devise-ruby-on-rails>

Piattini, M. G. (1996). *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Madrid. : Rama. .

Ramos Martín, A., & Ramos Martín, M. J. (2014). *Aplicaciones Web 2.<sup>a</sup> edición*. España: Paraninfo.

RubyTeam. (Noviembre de 2009). *EcuRed*. Obtenido de Ruby on Rails:  
[https://www.ecured.cu/Ruby\\_on\\_Rails](https://www.ecured.cu/Ruby_on_Rails)

sage. (2020). Obtenido de <https://www.sage.com/es-es/blog/diccionario-empresarial/facturacion/>

Schaposnik, F. (1977). *Semiología*. Texas: El Ateneo.

Soto Flores, S. A., Pérez Duarte, C. J., & Rivera Martínez, A. J. (2016). *Desarrollo de un sistema web para la gestión de expedientes clínicos*. Obtenido de Repositorio Institucional UNAN-Managua:  
<https://repositorio.unan.edu.ni/3853/1/11082.pdf>

Spurlock, J. (2014). Bootstrap: Responsive Web Development. En J. Spurlock, *Bootstrap: Responsive Web Development* (pág. 19). New York: Scripting News.

Suarez, A. (2019). *Aprende a programar con ruby*.

*Tutorial de RIP*. (s.f.). Obtenido de <https://riptutorial.com/ruby-on-rails/example/10259/getting-started-with-cancan>

Ucha, F. (03 de 2013). *Definición ABC*. Obtenido de Definición de Consulta:  
<https://www.definicionabc.com/general/consulta.php>

Write Less, D. (2016). JQuery the future. En D. Write Less, *JQuery the future* (págs. 10-11). California: Angel City Press.

## 11 Anexos.

### 11.1 Manual técnico

### 11.2 Manual de usuario

) Manual técnico y manual de usuario agregados en el CD.

### 11.3 Cartas de aceptación.



Sistema web para la gestión de información médica de la “Clínica  
Pediátrica y Farmacia Cristina” (SIM-Cristina).

Masaya 10 de enero de 2022

A quien corresponda,

Por este medio la clinica y farmacia Cristina ha revisado y ha aprobado de manera satisfactoria la propuesta de desarrollo de un **Sistema web para la gestión de información médica** de dicha clinica, que será elaborado por los estudiantes Cristhofer Enrique Espinoza Jiménez y José Andrés Reyes Aguirre, egresados de la carrera de ingenieria en computación de la universidad Nacional de ingenieria.

Sin más que agregar me despido, deseando éxitos en sus labores.



Firma y sello

Email: [carlosmanuelgutierrezgomez6@gmail.com](mailto:carlosmanuelgutierrezgomez6@gmail.com)  
Cruz roja 2C ½ al oeste, barrio paisés bajos, Masaya.

Figura 80: Carta de aceptación del desarrollo.



*Dr. Carlos Manuel Gutiérrez G.*



Medico y Cirujano  
Especialista En Pediatría Atencion adulto y niños.  
Emergencia las 24 Horas.

Codigo MINSA: 18665 Celular:5854-8571

Aceptación: Sistema web para la gestión de información médica de la  
"Clínica Pediátrica y Farmacia Cristina" (SIM-Cristina).

Masaya 05 de Octubre de 2022

A quien corresponda.

Por este medio, la clinica y farmacia Cristina da a conocer que se ha revisado detalladamente el funcionamiento y ha aprobado de manera satisfactoria el desarrollo del **Sistema web para la gestión de información médica** de dicha clínica, debido a que cumple con las necesidades y la solución planteada, que elaboro por los estudiantes Cristhofer Enrique Espinoza Jiménez y José Andrés Reyes Aguirre, egresados de la carrera de ingeniería en computación de la universidad Nacional de ingeniería.

Sin más que agregar me despido, deseando éxitos en sus labores.

Firma y sello

Email: [carlosmanuelgutierrezgomez6@gmail.com](mailto:carlosmanuelgutierrezgomez6@gmail.com)  
Cruz roja 2C ½ al oeste, barrio paisés bajos, Masaya.

Figura 81: Carta de aceptación del software

## Anexo 1 Formato de fichas de pruebas.

Ficha de prueba para el proceso de historial clínico		
<p><b>Indicaciones:</b> Rellene los valores que se le solicitan, en cada prueba anote los datos que solicitan:</p> <p><b>Tiempo:</b> Hora de inicio y fin (con minutos y segundos).</p> <p><b>Pasos:</b> Conteo de la cantidad de pasos que requiere para generar este reporte.</p> <p><b>Simplicidad:</b> Percepción que tenga de la facilidad de uso del servidor de reporte, es simple, normal o complejo. Considere la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva y los controles usados son conocidos y de fácil uso.</p>		
Prueba 1		
Medida	Método Anterior	Método nuevo
Tiempo:	6.6	3.8
Hora de inicio:	1:35:00 pm	2:30:00 PM
Hora Finalización:	1:41:06 pm	2:33:08 PM
Pasos de procedimiento:	37	15
Simplicidad de uso	[ ] simple, [ ] normal, [✓] complejo.	[✓] simple, [ ] normal, [ ] complejo

Tabla 35: Ficha de registro de prueba para proceso de historial clínico.

## Ficha de prueba para el proceso de registro de citas

**Indicaciones:** Rellene los valores que se le solicitan, en cada prueba anote los datos que solicitan:

**Tiempo:** Hora de inicio y fin (con minutos y segundos).

**Pasos:** Conteo de la cantidad de pasos que requiere para generar este reporte.

**Simplicidad:** Percepción que tenga de la facilidad de uso del servidor de reporte, es simple, normal o complejo. Considere la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva y los controles usados son conocidos y de fácil uso.

### Prueba 2

Medida	Método Anterior	Método nuevo
<b>Tiempo:</b>	<b>4.1</b>	<b>2.6</b>
<b>Hora de inicio</b>	<b>01:20:00 PM</b>	<b>01:35:00 PM</b>
<b>Hora Finalización:</b>	<b>01:24:04 PM</b>	<b>01:37:06 PM</b>
<b>Pasos de procedimiento:</b>	<b>5</b>	<b>3</b>
<b>Simplicidad de uso</b>	[ ] simple, [✓] normal, [ ] complejo.	[ ✓] simple, [ ] normal, [ ] complejo

Tabla 36: Ficha de registro de prueba para proceso de registro de citas.



## Ficha de prueba para el proceso de control de pago de productos

**Indicaciones:** Rellene los valores que se le solicitan, en cada prueba anote los datos que solicitan:

**Tiempo:** Hora de inicio y fin (con minutos y segundos).

**Pasos:** Conteo de la cantidad de pasos que requiere para generar este reporte.

**Simplicidad:** Percepción que tenga de la facilidad de uso del servidor de reporte, es simple, normal o complejo. Considere la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva y los controles usados son conocidos y de fácil uso.

### Prueba 3

Medida	Método Anterior	Método nuevo
<b>Tiempo:</b>	<b>2.3</b>	<b>1</b>
<b>Hora de inicio</b>	<b>05:00:00 PM</b>	<b>05:30:00 PM</b>
<b>Hora Finalización:</b>	<b>05:02:03 PM</b>	<b>05:31:00 PM</b>
<b>Pasos de procedimiento:</b>	<b>5</b>	<b>3</b>
<b>Simplicidad de uso</b>	[ ] simple, [✓] normal, [ ] complejo.	[✓] simple, [ ] normal, [ ] complejo

Tabla 37: Ficha de registro de prueba para proceso de control de pago de productos.

## Anexo 2 Formato de historial clínico.

Este es el formato de historial clínico que se utiliza para cada paciente.

**Dr. Carlos Manuel Gutiérrez Gómez.**

**MEDICO Y CIRUJANO.**

**ESPECIALISTA EN PEDIATRIA.**

### Historia clínica pediátrica y seguimiento

#### I. Datos generales

Nombre.

Fecha.

Fecha y lugar de nacimiento.

Dirección.

Nombres de los padres.

II. Motivo de consulta

III. Historia de la enfermedad actual.

IV. Antecedentes familiares patológicos.

V. Enfermedades hereditarias

VI. Enfermedades personales patológicas

VII. Antecedentes del parto(vía, edad gestacional, atención del parto, eventos durante el parto)

VIII. Antecedentes posnatales (apgar, peso, asfixia, hospitalización)

IX. Alimentación.

X. Vacunas.

XI. Desarrollo psicomotor.

XII. Examen físico

-Signos vitales. FC    FR    P/A    T°    S/A

-Datos antropométricos

Peso    Talla    PC    PT    PA    ASC    IMC

---

- Aspecto general.
- Piel y mucosa.
- Gráneo y cuero cabelludo
- Ojo, orejas y oído.
- Nariz, boca y cuello.
- Tórax
- Caja torácica, mamas, CPS
- Cardíaco.
- Abdomen y pelvis.
- Ano y recto.
- Extremidades superiores e inferiores.
- Genitourinario
- Neurológico.

XIII. Diagnóstico o Problemas

XIV. Tratamiento o manejo médico, incluye alimentación.

XV. Citas

## 1. Seguimientos

-fecha

-peso

-signos vitales FC FR P/A T° S/A Talla IMC  
ASC

2. Motivo de consulta.

3. Historia de la enfermedad actual.

4. Examen físico.

5. Diagnóstico o problema

6. Tratamiento.

Figura 82: Formato de historial clínico

### Anexo 3 Formato de registro y control de cita.

Formato utilizado en el registro y control de citas.



CITA MEDIDAS CLÍNICAS DR. CARLOS MANUEL GUTIÉRREZ GÓMEZ

FECHA	HORA	NOMBRE Y APELLIDO	DIAGNOSTICO	CELULAR	OBSERVACIÓN
25/11/2019	2:30PM	ZABDY ÁLVAREZ	RENT-AGE	81575384	
02/12/2019	10:00AM	ETHAN MERCADO	ESTREÑIMIENTO	76214667	
03/12/2019	10:00 AM	CRISTIAN FLÓREZ	IVRS	81291789	EXAMENES
06/12/2019	10:00 AM	EMMA FLÓREZ	SOB	86484841	
06/12/2019	6:00 PM	KEYSI FUENTES	EDA	87818865	
06/12/2019	10:00AM	NOHEMÍ HURTADO	FAA		SEGUIMIENTO
06/12/2019	3:30PM	JOSHUA GONZÁLEZ		85993681	
10/12/2019	9:00AM	JEFERSON CARCACHÉ	UÑA ENCARNADA	81888870	
11/12/2019	10:00AM	DOMINIC MENDOZA	QUINTIS ALERGICA	83328192	
12/12/2019	4:00 PM	MATHIAS ACEVEDO	GINGIVITIS	88980988	EXAMENES
12/12/2019	5:30PM	EMMA ZELEDÓN	FARINGITIS	86015801	
13/12/2019	2:00PM	JAHSLIA GONZALEZ	GRANULOMA	85993681	
14/12/2019	4:00PM	CRISTIAN FLÓREZ	PARASITOSIS	81291789	EXÁMENES
16/12/2019	10:00 AM	ALONDRA FLÓREZ	SEPSIS A REPETICIÓN	88425050	SEGUIMIENTOS
16/12/2019	2:00 PM	EMMA FLÓREZ	SOB	86484841	
21/12/2019	10:00 AM	ZABDY ÁLVARES	IVRS	81575384	
21/12/2019		JHONATAN LÓPEZ	EDA PARASITOSIS	75143597	EXAMENES
21/12/2019	11:30	NALANI ESCOBAR	INTOLERANCIA	76081749	400
26/12/2019	10:00AM	DOMINIC MENDOZA	RINITIS	83328192	
27/12/2019	4:00 PM	RAFAELA LOPEZ	ERICEPELA	88829492	EXÁMENES
27/12/2019	10:00 AM	JOSHUA GONZALES	GRANULOMA	85993681	
27/12/2019	3:30 PM	DYLAN SANCHEZ	PÓLIPO NASAL	81636319	
29/12/2019	9:00 AM	THIAGO MARTINEZ	SOB	50681139728	SEGUIMIENTO
02/01/2020	10:00 AM	THIAGO MARTINEZ	SOB	506811397328	SEGUIMIENTO

Figura 83: Formato de registro y control de citas.

Anexo 4 Formato de control de pago de medicamentos.

Formato de facturación de medicamentos en la farmacia.

Nº FACTURA					
FECHA					
DATOS DEL CLIENTE					
CÉDULA/NIT	4011409810001Y	DIRECCIÓN	CRUZ ROJA 30 AL ESTE		
NOMBRE/RAZÓN SOCIAL	MIGUEL VEGA	TELÉFONO	25223350		
DETALLES DE LA COMPRA					
FORMA DE PAGO					
CÓDIGO	PRODUCTO	CANTIDAD	PRECIO UNITARIO	TOTAL	
1		2	C\$ -	C\$ -	
2			C\$ -	C\$ -	
4			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
			C\$ -	C\$ -	
TOTAL ARTÍCULOS		2	SUB TOTAL	C\$ -	
Recibí Conforme			IVA	C\$ -	
			TOTAL	C\$ -	

Figura 84: Formato de facturación de medicamentos.

Anexo 5 Formato de comprobante de examen de laboratorio.

Formato de orden médica para examen de laboratorio.

Dr. Carlos Manuel Gutiérrez G.  
Medico y Cirujano  
Especialista En Pediatría Atención Adultos y Niños  
Emergencia las 24 Horas.  
Código Minsa: 18665 Celular: 9854-8571

Fecha: 15/07/2021  
Nombre: CRISTOFER ANDRÉS TRUJILLO  
Peso: 56 kg Edad: 10 años

Buc + Plog  
enealítico  
Perfil L. p. de co  
Glicemia  
EGG en Ayuno

21 1150 610L

Dr. Carlos M. Gutiérrez G.  
MEDICO - CIRUJANO  
ESPECIALISTA EN  
PEDIATRIA  
C.O. MINS. Firma y Sello

Email: carlosmanuelgutierrezgomez@gmail.com  
Cruz roja 20 1/2 al Oeste, Barrio países bajos, Masaya

Figura 85: Formato de comprobante de examen de laboratorio.



Anexo 6 Línea de tiempo del proyecto.

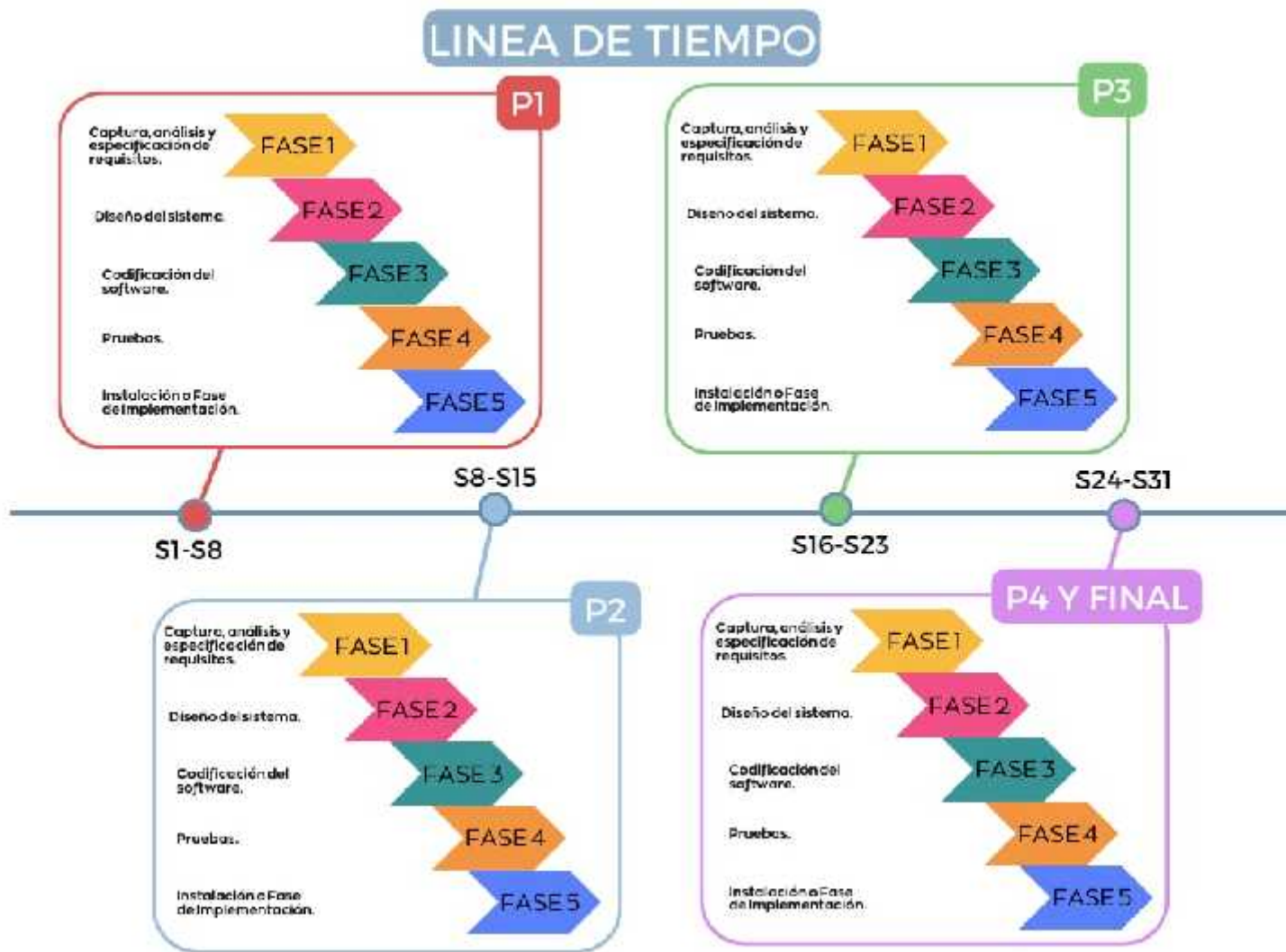


Figura 86: Línea de tiempo del proyecto.

## Glosario de términos.

Termino	Definición
<b>SIM-Cristina</b>	Sistema de Información Medica Cristina.
<b>Historial clínico</b>	Documento legal de la rama médica que surge del contacto entre un paciente y un profesional de la salud.
<b>UWE</b>	Es una metodología basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web.
<b>IEEE</b>	El Instituto de Ingenieros Eléctricos y Electrónicos.
<b>ISO</b>	Internacional Organization for Standardization (Organización Internacional de Normalización).
<b>MVC</b>	Model View Controller.
<b>ROR</b>	Ruby on Rails.
<b>Framework</b>	Es una estructura previa que se puede aprovechar para desarrollar un proyecto.
<b>RDBMS</b>	Sistema de gestión de bases de datos relacionales.
<b>DB2</b>	Es una familia de productos de sistema de gestión de bases de datos relacionales (RDBMS).
<b>HTML5</b>	HyperText Markup Language, versión 5
<b>CSS</b>	Siglas en inglés para: hojas de estilo en cascada.
<b>Javascript</b>	es un lenguaje de programación interpretado, dialecto.
<b>UML</b>	El Lenguaje Unificado de Modelado.
<b>front-end</b>	Es la parte del desarrollo web que se dedica a la parte frontal de un sitio web.
<b>back-end</b>	Es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione.
<b>CRUD</b>	(Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada.
<b>RSpec</b>	Es una gema que se usa para testear tu aplicación.
<b>AWS</b>	Amazon Web Services.
<b>RDS</b>	Radio Data System.
<b>DB</b>	Data base.
<b>Heroku</b>	es una plataforma de servicios en la nube.

**Tabla 38: Glosario de términos.**