

GESTION DE CITAS MEDICAS –SALUDPLUS

JOSEPH CAMILO URQUIJO BAUTISTA

JUAN MANUEL GUERRERO RUEDA

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERIAS

FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA

JUAN CARLOS MARINÑO MORANTES (Docente Universitario)

BUCARAMANGA, COLOMBIA

2024

1.Introducción	4
2.Situacion Problema	5
2.1 Pregunta problema	5
3. Justificación	6
4.Objetivos	7
4.1 Objetivo General	7
4.2 Objetivo Especifico	7
5. Metodología	8
5.1 Fases	8
5.1.1 Análisis	8
5.1.2 Diseño	8
5.1.3 Implementación	8
5.1.4 Verificación	9
5.1.5 Mantenimiento	9
6. Marco Tecnológico	10
6.1 Lenguaje de programación	10
6.2 Diseño de interfaces	10
6.3 Entorno de Desarrollo integrado	10
6.4 Control De Versiones	10
6.5 Plataforma de desarrollo	10
6.6 Diseño de diagramas	10
6.7 Gestor de base de datos	10
7. Marco Conceptual	11
7.1 Sistema	11
7.2 Programación orientada a objetos	11
7.3 Paradigma	11
7.4 Programación orientada a eventos	11
7.5 Estructura de datos	12
7.6 Listas	12
7.7 Interfaz	12
7.8 Algoritmo	12
7.9 Base de datos	12
7.10 Seguridad Informática	13
7.11 Diagrama de clases	13
7.12 Diagrama de casos de uso	13
7.13 Diagrama de secuencia	13
7.14 Paciente	13
7.15 Cita médica	13
7.16 Orden médica	13
7.17 Examen médico	14

7.17 Autorización.....	14
8. Especificación de requerimientos	14
8.1 Requisitos Funcionales.....	14
8.1.1 Registro de clientes.....	14
8.1.2 Gestión de citas.....	14
8.1.3 Taquilla de pagos	14
8.1.4 Llamado de pacientes.....	14
8.1.5 Registro de ordenes medicas.....	14
8.1.6 Autorización de exámenes	15
8.2 Requisitos No Funcionales.....	15
8.2.1 Seguridad	15
8.2.2 Eficiencia	15
8.2.3 Usabilidad	15
8.2.4 Fiabilidad	15
8.2.5 Compatibilidad.....	15
9. Diseños	16
9.1 Diagrama de clases	16
9.2 Diagrama de casos de uso	17
9.3 Diagrama de secuencia.....	18
9. Cronograma de actividades.....	19
10. Cronograma de actividades.....	20

1. Introducción

El presente documento aborda el desarrollo de un sistema de gestión de citas médicas para el centro de salud privado SALUDPLUS en Bucaramanga, Colombia. Este proyecto surge como respuesta a la necesidad de mejorar la eficiencia y calidad en la atención médica ofrecida a los usuarios, especialmente en lo concerniente al agendamiento de citas y la autorización de órdenes médicas para exámenes.

El sistema busca atender a los desafíos de largas esperas y gestión tardía de citas y exámenes médicos, con el propósito de mejorar los procesos internos del centro de salud y la experiencia del paciente. Además, se pretende garantizar la seguridad y confidencialidad de la información del paciente, cumpliendo con los estándares de privacidad y protección de datos en el sector de la salud.

En consecuencia, este documento detalla la situación problemática, la pregunta problema, la justificación, los objetivos y la metodología a seguir en el desarrollo del sistema. Además, se presenta un marco tecnológico, conceptual y de requerimientos, junto con un cronograma de actividades para guiar el proceso de implementación del sistema de gestión de citas médicas en SALUDPLUS.

2. Situación Problema

En el contexto del Centro de Salud Privado SALUDPLUS en Bucaramanga, Colombia, se enfrentan a una serie de desafíos operativos que están afectando negativamente la calidad de la atención médica brindada a sus pacientes.

Entre estos desafíos se encuentran largas esperas para agendar citas médicas, demoras significativas en la autorización de órdenes médicas para exámenes diagnósticos y una gestión ineficiente en la administración de pacientes en todas las áreas del centro, desde la recepción hasta la sala de espera y las consultas médicas.

Estas dificultades han generado una experiencia insatisfactoria y frustrante para los usuarios del centro, quienes se ven obligados a enfrentar largos períodos de espera y una falta de claridad en el proceso de atención médica.

Además, la falta de un sistema integrado de gestión de citas y órdenes médicas ha llevado a errores administrativos, duplicación de esfuerzos y una falta de seguimiento adecuado de la atención médica prestada, lo que resulta en una experiencia fragmentada y poco satisfactoria para los pacientes, quienes a menudo se encuentran en la incertidumbre sobre el estado y el seguimiento de sus citas y tratamientos médicos.

Esta situación ha provocado un aumento en la frustración y la ansiedad entre los pacientes, quienes esperan una solución que les brinde mayor transparencia y control sobre su atención médica.

En este contexto crítico, SALUDPLUS reconoce la urgente necesidad de implementar una solución integral que le permita mejorar la eficiencia de sus operaciones, optimizar la gestión de pacientes y ofrecer una experiencia de atención médica más fluida, transparente y satisfactoria para sus pacientes, reafirmando su compromiso con la excelencia en el cuidado de la salud y el bienestar de la comunidad a la que sirve.

2.1 Pregunta problema

¿Cómo desarrollar un sistema de gestión de pacientes en Java que permita agendar citas, administrar colas de espera y autorizar órdenes médicas para exámenes en un centro de salud privado en Bucaramanga, garantizando una atención oportuna y de calidad para los usuarios?

3. Justificación

El desarrollo de este sistema de gestión de pacientes es crucial para abordar los desafíos operativos que enfrenta el Centro de Salud Privado SALUDPLUS en Bucaramanga, Colombia. Estos desafíos, como largas esperas para agendar citas médicas, demoras en la autorización de órdenes médicas para exámenes y una gestión ineficiente de pacientes, afectan negativamente la calidad de la atención médica brindada a los pacientes.

El desarrollo de los procesos internos del centro de salud es esencial para garantizar una atención médica más eficiente y efectiva para los pacientes. Al reducir los tiempos de espera para agendar citas y recibir autorización para exámenes médicos, se optimiza el flujo de pacientes y se agiliza el proceso de atención médica.

Además, al desarrollar este sistema, SALUDPLUS podrá asignar mejor los recursos disponibles, tanto humanos como materiales. La planificación y programación más rápida de citas y exámenes permitirá una distribución más equitativa de la carga de trabajo entre el personal médico y una asignación más eficiente de las instalaciones y equipos médicos.

Este sistema también contribuirá significativamente a mejorar la experiencia del paciente. La reducción de los tiempos de espera y la mejora en la gestión de citas médicas y exámenes crearán una experiencia más satisfactoria para los usuarios del centro de salud. Esto fortalecerá la relación entre el centro de salud y sus pacientes, aumentando la fidelidad y la confianza en los servicios médicos ofrecidos.

Además, el desarrollo de un sistema integrado de gestión de pacientes garantizará la seguridad y confidencialidad de la información del paciente. Al proporcionar un almacenamiento seguro de datos médicos y un acceso controlado a la información, se cumplirán los estándares de privacidad y protección de datos en el sector de la salud.

4. Objetivo general

Desarrollar un sistema de gestión de pacientes en el centro de salud de Bucaramanga. Este sistema busca agilizar el agendamiento de citas, reducir los tiempos de espera y mejorar la atención médica ofrecida a los pacientes.

4.1 Objetivos Específicos:

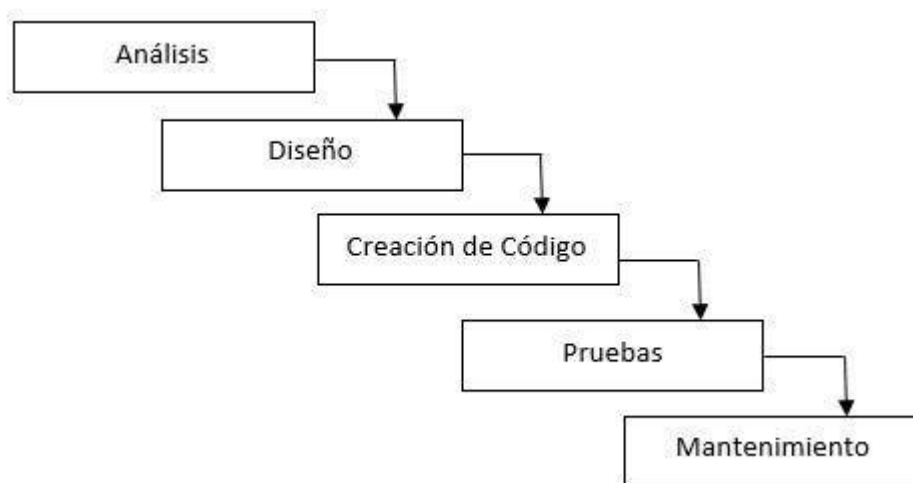
- Analizar exhaustivamente los requisitos del sistema de gestión de citas médicas, identificando las necesidades específicas del centro de salud de Bucaramanga, para comprender completamente el alcance del proyecto y garantizar su relevancia y pertinencia.
- Diseñar de manera detallada la arquitectura del sistema, la estructura de la base de datos y la interfaz de usuario, asegurando la coherencia y la integridad del diseño, con el fin de proporcionar una representación clara y comprensible del funcionamiento del sistema.
- Implementar el sistema de gestión de citas médicas utilizando Java como lenguaje de programación, siguiendo una metodología en cascada, para garantizar la claridad en el proceso de desarrollo y la consistencia en la implementación de las funcionalidades requeridas.
- Realizar pruebas unitarias sistemáticas para validar la funcionalidad del sistema, detectando posibles errores tempranos en el proceso de desarrollo y asegurando la calidad y confiabilidad del producto final.
- Ejecutar pruebas periódicas durante todo el ciclo de desarrollo para identificar y corregir errores de manera oportuna, manteniendo la integridad y el rendimiento del sistema en todo momento y garantizando su conformidad con los requisitos establecidos y su capacidad para satisfacer las necesidades específicas del centro de salud.

5. Metodología

La metodología para el desarrollo del sistema de gestión de pacientes será cascada, la cual se caracteriza por su enfoque secuencial y su planificación detallada. Esta metodología se divide en varias fases que se ejecutan de manera lineal, lo que permite una estructura clara y definida para el desarrollo del proyecto.

Figura 1

Representación gráfica de la metodología cascada



Nota. Imagen obtenida de https://miro.medium.com/v2/resize:fit:478/1*2UcWBIItGxECz1i-AxkLh2w.jpeg esta imagen ilustra las fases de la metodología en cascada.

5.1 Fases:

5.1.1 Análisis:

En esta fase se llevará a cabo un análisis detallado de los requisitos del sistema, centrándose en las necesidades específicas del centro de salud de Bucaramanga. Se recopilará información sobre las funcionalidades requeridas, los usuarios finales y los objetivos del sistema.

5.1.2 Diseño: En esta etapa se elaborará el diseño del sistema, definiendo la arquitectura general, la estructura de la base de datos y la interfaz de usuario. Se crearán diagramas de clases, diagramas de casos de uso y otros artefactos que ayuden a visualizar la estructura y el funcionamiento del sistema.

5.1.3 Implementación:

Una vez completado el diseño, se procederá a la implementación del sistema, donde se traducirán los diseños y especificaciones en código fuente utilizando el lenguaje de programación Java. Se desarrollarán las diferentes funcionalidades del sistema, asegurando la coherencia y la integridad del código.

5.1.4 Verificación: En esta fase se realizarán pruebas exhaustivas para verificar el correcto funcionamiento del sistema y detectar posibles errores o fallos. Se llevarán a cabo pruebas unitarias para asegurar la calidad y la fiabilidad del producto final.

5.1.5 Mantenimiento:

Una vez que el sistema esté en funcionamiento, se iniciará la fase de mantenimiento, donde se realizarán actualizaciones, correcciones de errores y mejoras según sea necesario. Se atenderán las solicitudes de los usuarios y se garantizará el correcto funcionamiento continuo del sistema a lo largo del tiempo.

6. Marco Tecnológico

6.1 Lenguaje de programación

- **Java:** Se implementará Java como el lenguaje principal para el desarrollo del sistema de Gestión de Citas Médicas – SALUDPLUS. Java ofrece una amplia adopción en el ámbito empresarial y proporciona una plataforma robusta y escalable para el desarrollo de aplicaciones.

6.2 Diseño de interfaces

- **Figma:** Se utilizará Figma para el diseño de las interfaces gráficas de usuario (GUI). Esta herramienta permite crear prototipos interactivos y diseños de alta calidad, facilitando la comunicación visual entre el equipo de diseño y desarrollo.

6.3. Entorno de Desarrollo Integrado (IDE)

- **NetBeans:** El proyecto se desarrollará y desplegará en el IDE NetBeans. NetBeans es un IDE popular y confiable para el desarrollo de aplicaciones Java, que ofrece diversas herramientas para la edición de código, depuración y pruebas.

6.4 Control de versiones

- **Git:** Se empleará Git como el sistema de control de versiones para rastrear y administrar los cambios en el código. Git permite crear ramificaciones y mantener un historial de desarrollo confiable, facilitando la colaboración entre los miembros del equipo.

6.5 Plataforma de Desarrollo

- **IDE:** Se utilizará el IDE de despliegue de la aplicación en NetBeans, sin embargo, cada miembro del equipo de desarrollo tendrá la flexibilidad de utilizar su IDE preferido. Es importante tener en cuenta la incorporación y adaptación del código de un IDE a otro, garantizando así la coherencia en el desarrollo del programa.

6.6 Diseño de diagramas

- **StarUML:** Para la creación de diagramas de diseño, se empleará StarUML. Esta herramienta facilita la elaboración de diagramas UML, permitiendo visualizar la estructura y el comportamiento del sistema de manera clara y concisa.

6.7 Gestor de base de datos

- **MySQL:** Se utilizará MySQL como el gestor de base de datos para el sistema de Gestión de Citas Médicas – SALUDPLUS. MySQL es una opción popular y ampliamente adoptada en el desarrollo de aplicaciones web y empresariales debido a su robustez, escalabilidad y facilidad de uso. Proporciona una plataforma confiable para almacenar y gestionar los datos del sistema, asegurando un rendimiento óptimo y una gestión eficiente de la información médica de los pacientes.

7. Marco Conceptual

7.1 Sistema: En el contexto del desarrollo de software, un sistema se refiere a un conjunto de componentes interrelacionados que trabajan juntos para lograr un objetivo común. Puede incluir software, hardware, datos, procesos y personas. Un sistema está diseñado para cumplir con ciertas funciones o proporcionar ciertos servicios para satisfacer las necesidades de los usuarios. Por ejemplo, un sistema de gestión de citas médicas es un conjunto de software que se utiliza para programar citas, administrar registros de pacientes y facilitar la atención médica en un centro de salud. Los sistemas pueden ser simples o complejos, dependiendo de la naturaleza y el alcance de las funciones que realizan.

7.2 Programación Orientada a Objetos: La Programación Orientada a Objetos (POO) es un enfoque utilizado en el desarrollo de software que se basa en la idea de modelar el mundo real a través de la creación de objetos. En este paradigma, un objeto representa una entidad con características (atributos) y comportamientos (métodos) asociados. La programación orientada a objetos permite organizar el código de manera más estructurada y modular, lo que facilita la reutilización y la mantenibilidad del software. Al modelar entidades del mundo real como objetos, podemos abstraer la complejidad y representarlas de manera más intuitiva en nuestro código. Esto resulta en un desarrollo de software más eficiente y adaptable a cambios futuros.

7.3 Paradigma: Un paradigma de programación es un conjunto de reglas y principios que guían la forma en que se escribe el código de un programa de computadora. Es como un conjunto de instrucciones que define cómo se deben organizar y estructurar las instrucciones de un programa para resolver un problema de manera eficiente. En otras palabras, es una metodología que proporciona un marco de trabajo para el desarrollo de software, dictando cómo se deben abordar los problemas y cómo se deben escribir las instrucciones de manera que la computadora pueda entenderlas y ejecutarlas correctamente.

7.4 Programación Orientada a Eventos: La Programación Orientada a Eventos (POE) es un enfoque en el desarrollo de software donde el comportamiento del programa está dirigido por eventos que suceden en el sistema. Estos eventos pueden ser cualquier tipo de acción o suceso, como hacer clic en un botón, mover el mouse, o recibir datos de una red. En la POE, el programa está diseñado para responder automáticamente a estos eventos, ejecutando código específico en función de lo que suceda en el sistema. En resumen, la POE se trata de escribir programas que actúen de manera automática en respuesta a las acciones del usuario o a eventos que ocurran en el sistema, lo que permite una interacción más dinámica y fluida con la aplicación.

7.5 Estructura de Datos: La Estructura de Datos se refiere a la forma en que los datos se organizan, almacenan y manipulan en un sistema informático para facilitar su acceso y utilización de manera eficiente. En términos más simples, se trata de la organización de la información de manera que sea fácil de manejar y procesar por parte de un programa o sistema. Las estructuras de datos pueden incluir arreglos, listas, pilas, colas, árboles, entre otros, cada uno diseñado para un propósito específico y con operaciones definidas para manipular los datos de manera adecuada. En pocas palabras, la Estructura de Datos es el marco que define cómo se organiza y se accede a la información dentro de un sistema informático, lo que permite realizar operaciones como búsqueda, inserción, eliminación y modificación de datos de manera eficiente.

7.6 Listas: Las listas son estructuras de datos que consisten en una colección ordenada de elementos donde cada elemento tiene una posición relativa dentro de la lista. Cada elemento puede ser de cualquier tipo y puede ser accedido y manipulado mediante su posición en la lista. Las listas proporcionan flexibilidad para almacenar y manipular conjuntos de datos de manera dinámica, permitiendo la inserción, eliminación y modificación de elementos según sea necesario. Son utilizadas en programación para organizar y gestionar datos de forma eficiente.

7.7 Interfaz: Una interfaz en el contexto de la informática se refiere a un punto de interacción entre un usuario y un sistema informático, o entre diferentes componentes de un sistema. En términos más formales, una interfaz puede definirse como un conjunto de métodos, operaciones o funciones que permiten la comunicación y la interacción entre dos entidades, como un usuario y una aplicación, o entre diferentes partes de un software. Las interfaces pueden presentarse en forma de interfaces gráficas de usuario (GUI), interfaces de línea de comandos (CLI), interfaces de programación de aplicaciones (API), entre otros. Su propósito es facilitar la comunicación y el intercambio de información de manera eficiente y efectiva.

7.8 Algoritmo: Un algoritmo es un conjunto finito de pasos o instrucciones bien definidas y ordenadas que se utilizan para resolver un problema o realizar una tarea específica. Estos pasos deben ser precisos, no ambiguos y capaces de llevarse a cabo en un tiempo finito. Los algoritmos son fundamentales en la informática y en la programación, ya que son la base para desarrollar software y resolver problemas de manera eficiente. Un algoritmo puede ser representado mediante un diagrama de flujo, pseudocódigo o en un lenguaje de programación específico. Su diseño y eficiencia son aspectos críticos para el desarrollo de sistemas informáticos robustos y eficientes.

7.9 Base de Datos: Una base de datos es un sistema organizado para almacenar y gestionar conjuntos de datos relacionados. Es como un archivo digital donde se almacena información de manera estructurada y accesible para su posterior consulta y manipulación.

7.10 Seguridad Informática: La seguridad informática se refiere a las medidas y prácticas utilizadas para proteger los sistemas de información contra accesos no autorizados, daños o cualquier otra amenaza. Incluye técnicas como el cifrado de datos, la autenticación de usuarios y la implementación de firewalls.

7.11 Diagrama de Clases: Un diagrama de clases es una representación visual de la estructura estática y las relaciones entre las clases de un sistema orientado a objetos. Muestra las clases del sistema, sus atributos, métodos y las relaciones entre ellas, como la herencia, la asociación, la composición, etc. Es como un plano que describe la arquitectura y las relaciones entre los diferentes componentes del sistema.

7.12 Diagrama de Casos de Uso: Un diagrama de casos de uso es una representación visual de las interacciones entre los actores (usuarios o sistemas externos) y el sistema en un escenario particular. Describe los diferentes casos de uso del sistema, que son acciones o funciones que el sistema proporciona a los usuarios para lograr un objetivo específico. Los actores se representan como entidades externas que interactúan con el sistema a través de estos casos de uso. Es como un mapa que muestra las diferentes formas en que los usuarios pueden interactuar con el sistema y los resultados de esas interacciones.

7.13 Diagrama de Secuencia: Un diagrama de secuencia es una representación visual de cómo interactúan los objetos en un sistema a lo largo del tiempo para realizar una determinada funcionalidad. Muestra la secuencia de mensajes intercambiados entre los objetos durante la ejecución de un escenario específico, lo que permite visualizar el flujo de control y la comunicación entre los componentes del sistema. Es como una película que muestra la interacción entre los actores (objetos) en una escena particular.

7.14 Paciente: El paciente es el usuario principal del sistema, quien solicita citas médicas y exámenes dentro del centro de salud. Debe registrar sus datos personales, como nombres, apellidos, edad e identificación, para acceder a los servicios del centro. Se le asigna un identificador único para su historia clínica y para el seguimiento de sus citas y exámenes.

7.15 Cita Médica: El sistema permite la programación de citas médicas para distintas especialidades, como medicina general y especialidades médicas. Cada cita médica se genera con base en la especialidad requerida, el profesional asignado y el motivo de la consulta (valoración, examen o control). Se asigna un ticket a cada cita médica, el cual contiene información sobre la especialidad, motivo de la cita y turno del paciente.

7.16 Orden Médica: Una orden médica es una instrucción escrita por un profesional de la salud, como un médico o un especialista, que prescribe un tratamiento, procedimiento o medicamento para un paciente. En el contexto del sistema de gestión de citas médicas, una orden médica se refiere a la solicitud formal de un médico para que un paciente realice un examen específico, como análisis de sangre, radiografías u otros procedimientos diagnósticos.

7.17 Examen: Un examen es un procedimiento médico que se realiza para evaluar la salud de un paciente, diagnosticar una enfermedad o monitorear el progreso de un tratamiento. En el contexto del sistema de gestión de citas médicas, un examen se refiere a un servicio médico específico solicitado por un médico y programado para un paciente.

7.18 Autorización: La autorización es el proceso mediante el cual se aprueba formalmente una solicitud o acción específica. En el contexto del sistema de gestión de citas médicas, la autorización se refiere a la aprobación de una orden médica para realizar un examen específico. Este proceso puede implicar la revisión de la orden por parte de personal autorizado para determinar si el examen es necesario, si está cubierto por el seguro médico del paciente y si cumple con los criterios médicos establecidos. Una vez autorizado, el paciente puede proceder con el examen según lo programado.

8. Especificación de Requerimientos

8.1 Requisitos Funcionales

8.1.1 Registro de Pacientes: El sistema debe permitir el registro de pacientes ingresando sus datos personales, incluyendo nombres, apellidos, edad, identificación y número de teléfono. En caso de ser la primera visita, se asignará un ID de historia clínica al paciente.

8.1.2 Gestión de Citas: El sistema debe contar con una funcionalidad para agendar citas médicas. Para cada cita, se deben registrar la especialidad requerida, el profesional asignado y el motivo de la cita (valoración, examen o control). Se generará un ticket para cada cita que contenga un identificador representativo de especialidad, motivo de la cita y turno del paciente. Se deberá asignar un valor para la consulta, indicando si el pago ha sido realizado o no.

8.1.3 Taquilla de Pagos: Debe existir una taquilla de pagos donde se registren las transacciones de pago de las citas médicas. Al registrar la transacción, se actualizará el estado de pago en el sistema. Todos los tickets, independientemente del costo o tipo de visita, deben pasar por la taquilla para ingresar a la cola de espera.

8.1.4 Llamado de Pacientes: Se debe implementar un sistema de llamado de pacientes que notifique cuándo deben ser atendidos. Al ingresar un comando específico por especialidad, se tomará un ID de la cola de espera respetando el orden de llegada. Una vez atendido un paciente, su ID debe ser removido de la cola de espera.

8.1.5 Registro de Órdenes Médicas: Se debe desarrollar un módulo para registrar órdenes médicas de exámenes. Cada orden debe incluir un identificador único, la especialidad correspondiente, el tipo de examen, el ID del paciente, el costo del examen, una breve descripción y un flag que indique su estado.

8.1.6 Autorización de Exámenes: Se implementará una funcionalidad para autorizar los exámenes pendientes por paciente. Solo se puede autorizar un examen a la vez. El acceso a las órdenes registradas se realizará utilizando un esquema LIFO (Last In, First Out).

8.2 Requisitos no funcionales

8.2.1 Seguridad: El sistema debe garantizar la seguridad y confidencialidad de la información del paciente, cumpliendo con los estándares de privacidad y protección de datos en el sector de la salud.

8.2.2 Eficiencia: El sistema debe ser eficiente en el agendamiento de citas y en la gestión de la cola de espera, minimizando los tiempos de espera para los pacientes.

8.2.3 Usabilidad: La interfaz del sistema debe ser intuitiva y fácil de usar para los usuarios, permitiendo una interacción fluida y comprensible.

8.2.4 Escalabilidad: El sistema debe ser escalable para adaptarse a un aumento en la demanda de servicios médicos sin comprometer su rendimiento.

8.2.5 Fiabilidad: El sistema debe ser confiable y robusto, minimizando la posibilidad de fallos o errores que puedan afectar la atención médica prestada.

8.2.6 Compatibilidad: El sistema debe ser compatible con diferentes dispositivos y navegadores web para garantizar su accesibilidad desde diferentes plataformas.

9. Diseños

Figura 2

9.1 Diagrama de clases

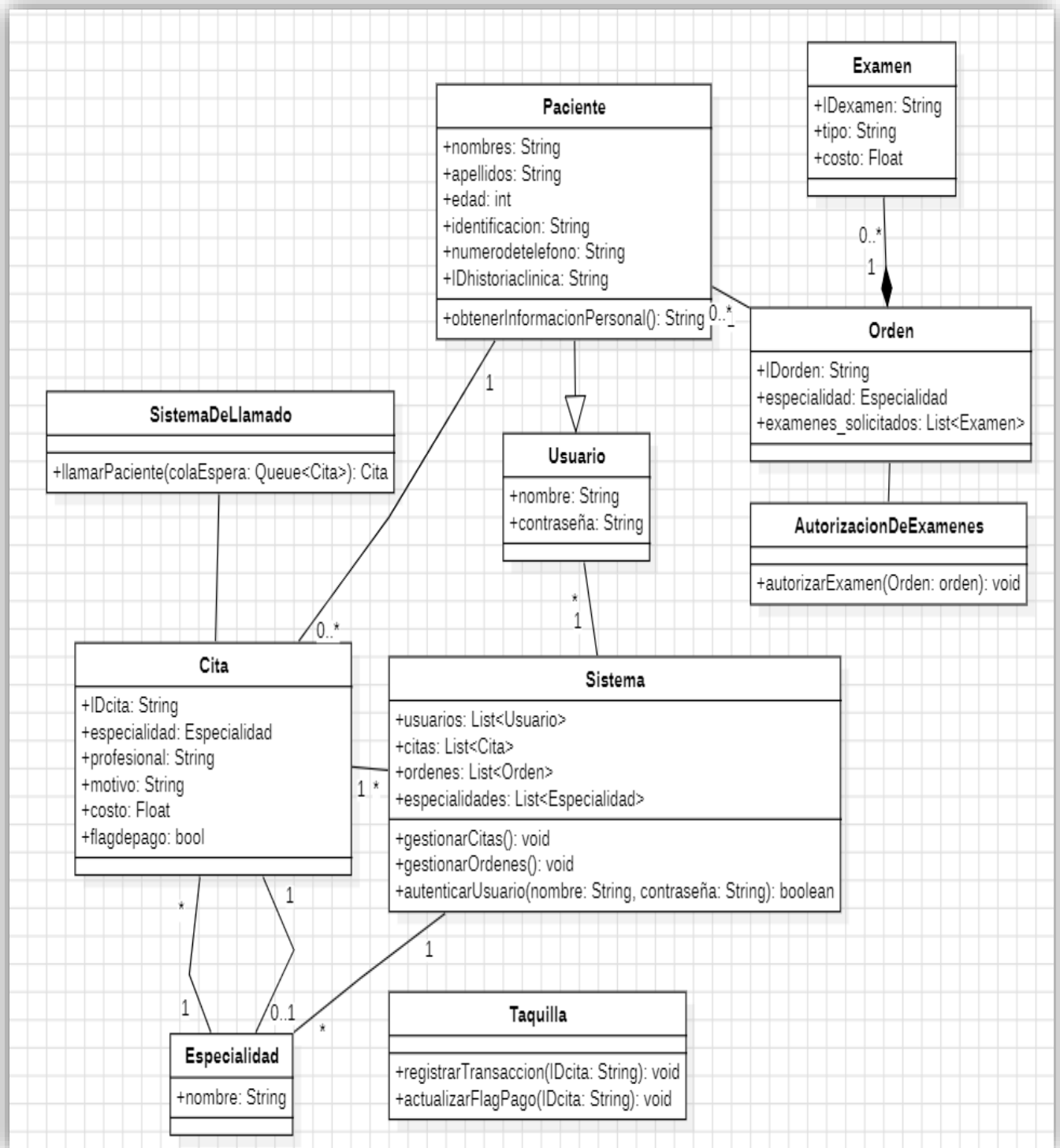


Figura 3

9.2 Diagrama de casos de uso

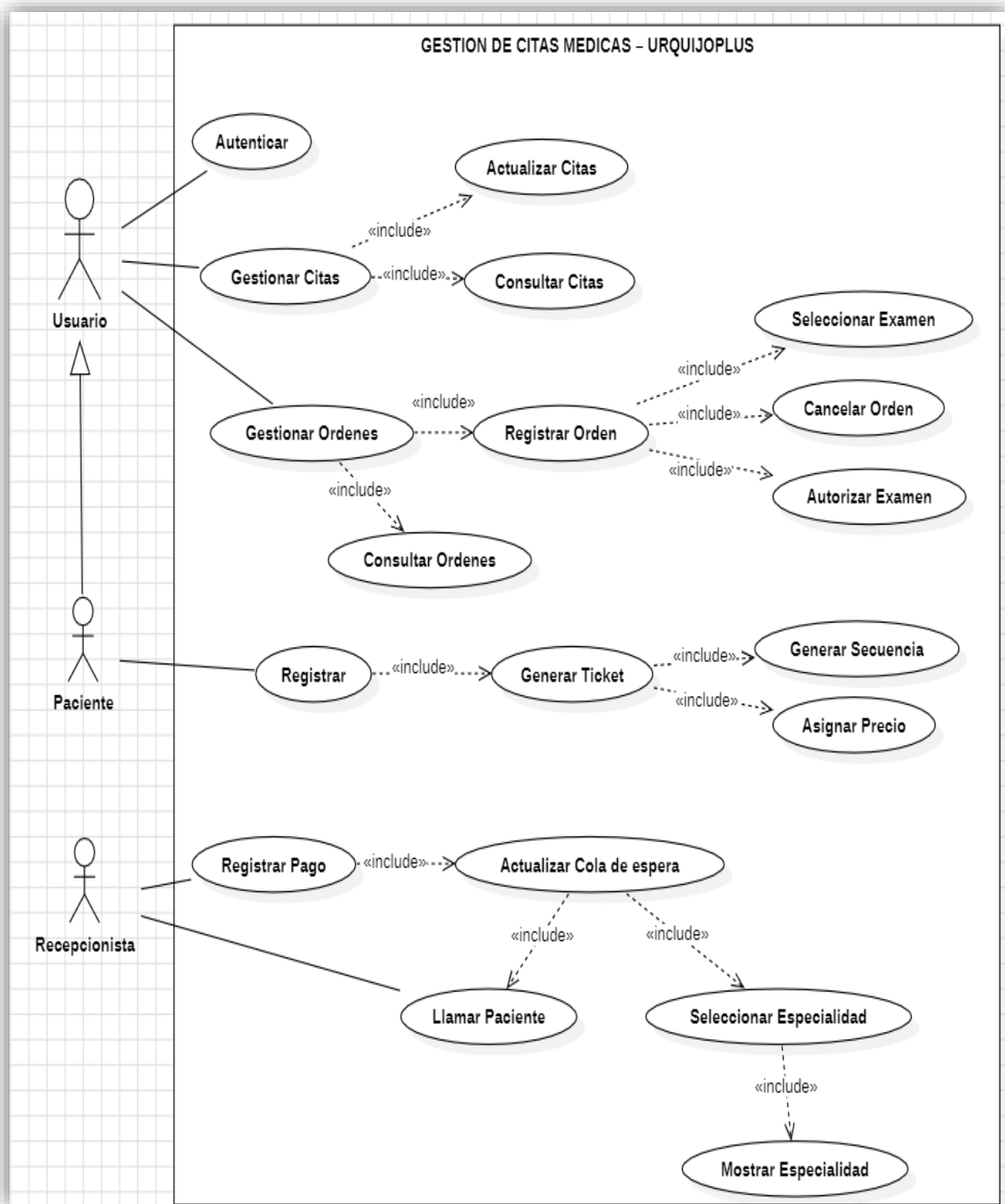
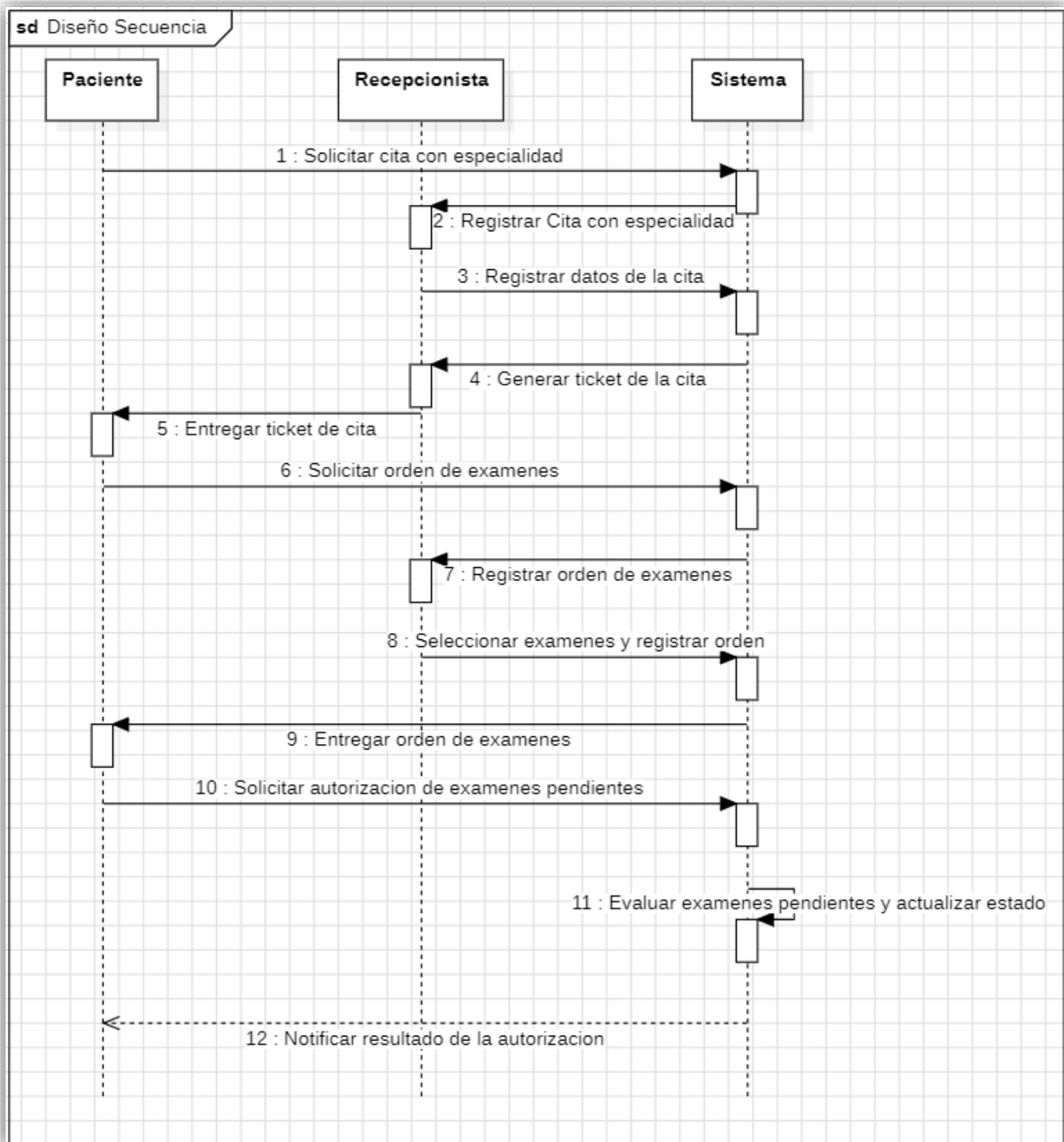


Figura 4

9.3 Diagrama de secuencia



10. Cronograma de Actividades

ACTIVIDADES A REALIZAR	FECHAS
Bloque 1: Preparación para la Primera Entrega	5 de febrero - 8 de marzo
Definición del Equipo y Roles	5-7 de febrero
Investigación Preliminar	5-10 de febrero
Revisión de Formatos de Entrega (IEE o APA)	8-10 de febrero
Elaboración de Portadas y Formato del Documento	8-10 de febrero
Introducción, Planteamiento del Problema y Justificación	8-15 de febrero
Definición de Objetivos del Proyecto	12-15 de febrero
Marco Referencial (Conceptual, Tecnológico y/o Legal)	12-22 de febrero
Metodología de Trabajo	18-22 de febrero
Resultados Preliminares (Especificación de Requerimientos y Diseños)	20-28 de febrero
Elaboración del Cronograma de Actividades	25-28 de febrero
Revisión y Ajustes	5-7 de marzo
Entrega 1: Propuesta de Proyecto	8 de marzo
Entregables: Especificación de Requerimientos, Diseño (solo para esta entrega)	
Bloque 2: Desarrollo y Preparación para la Segunda Entrega	9 de marzo - 19 de abril
Planificación detallada del Desarrollo (Fase de Análisis)	9-15 de marzo
Diseño Detallado del Software	16-22 de marzo
Desarrollo del Código Fuente	23 de marzo - 5 de abril
Pruebas Unitarias	6-12 de abril
Preparación de Documentación Técnica	13-16 de abril
Validación del Software	17-18 de abril
Evidencia de Puesta en Marcha o Implementación	19 de abril
Entrega 2: Propuesta de Proyecto Final	19 de abril
entregables: Código Fuente, Validación del Software, Evidencia de Puesta en Marcha o Implementación	

Referencias

Joyanes Aguilar, Luis, and Ignacio Zahonero Martínez. *Programación En C/C++ JAVA y UML*. Segunda edición. México, D. F.: McGraw-Hill Interamericana Editores, 2014. Print.

Martin C., Robert, and Gómez Celador, José Luis traductor. *Código Limpio: Manual de Estilo Para El Desarrollo ágil de Software*. Madrid: Anaya-Multimedia, 2012. Print.

Joyanes Aguilar, Luis., Ignacio. Zahonero Martínez, and e-libro, Corp. *Programación En C Metodología, Algoritmos y Estructura de Datos*. 2a ed. Madrid [etc]: McGraw-Hill/Interamericana de España, 2005. Print.