

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3058 - Ingeniería de Software 1

Sección 10

Ing. Lynette Garcia Perez



Gestión del control de las operaciones en clínicas médicas privadas

Javier Alejandro Prado Ramirez - 21486

Bryan Carlos Roberto España - 21550

Javier Alejandro Ramírez Heredia - 21600

Mario Andres Cristales Cardona - 21631

GUATEMALA, 12 de abril de 2023

Resumen

Este proyecto tiene como objetivo desarrollar una solución en línea para clínicas privadas que les permita mejorar la eficiencia en el proceso de agendar citas para sus pacientes. La solución propuesta busca abordar los desafíos que enfrentan las clínicas privadas en la gestión de una gran cantidad de citas diarias, lo que puede resultar en largas esperas para los pacientes y posibles errores en el agendamiento. El programa en línea permitirá a las clínicas registradas agregar a sus médicos y especialidades, y los pacientes podrán asignar citas en línea y tener acceso a su historial médico, incluyendo citas pasadas y futuras, diagnósticos y recetas médicas. Los médicos, por su parte, podrán ver sus citas con pacientes y realizar diagnósticos y recetar medicamentos. La necesidad de esta solución es actual, ya que las clínicas privadas necesitan mejorar su eficiencia en el proceso de agendar citas para sus pacientes, y los pacientes buscan una atención médica de calidad y fácil de gestionar. Los objetivos concretos son desarrollar una plataforma en línea fácil de usar, permitir un control preciso del calendario de citas y ofrecer una atención de calidad a los pacientes.

Introducción

La solución propuesta está dirigida a clínicas privadas que buscan mejorar la eficiencia en el proceso de agendar citas para sus pacientes. Estas clínicas pueden variar en tamaño y especialidades médicas, pero comparten la necesidad de ofrecer a sus pacientes un sistema de citas fácil de usar y eficiente. Además, esta solución también está diseñada para beneficiar directamente a los pacientes que asisten a estas clínicas, quienes podrán gestionar sus citas en línea, ver su historial médico, recibir diagnósticos y recetas médicas.

Las clínicas privadas a menudo enfrentan el desafío de gestionar una gran cantidad de citas diarias, lo que puede resultar en largas esperas para los pacientes y posibles errores en el agendamiento. Nuestra solución busca abordar estas dificultades al proporcionar una plataforma en línea fácil de emplear, que les permita a las clínicas y a sus médicos mantener un control preciso sobre su calendario de citas y asegurar una atención de calidad a sus pacientes. En este proyecto, se desarrollará un programa en línea que permitirá a las clínicas registradas con nosotros agregar a sus médicos y especialidades. Los pacientes podrán asignar citas en línea y tendrán acceso a su historial médico, incluyendo citas pasadas y futuras, diagnósticos y recetas médicas. Por su parte, los médicos podrán ver sus citas con pacientes y realizar diagnósticos y recetar medicinas a los pacientes.

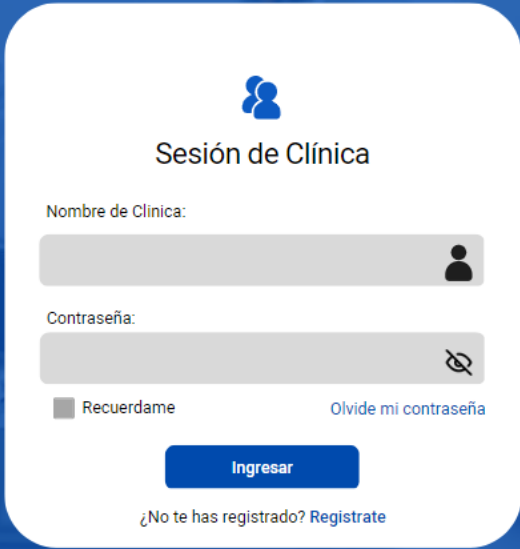
Los objetivos generales del informe son describir la solución propuesta y su utilidad para clínicas privadas, mientras que los objetivos específicos son destacar las funcionalidades específicas del programa, la importancia de su implementación y las potenciales ventajas que ofrecerá a los pacientes y a los médicos.

Design Thinking


Refinición del Prototipo:


Fase 3 - Prototipo.


Clínicas:




A login form titled "Sesión de Clínica" is centered on a blue background with a faint image of a doctor's hands. The form is white with rounded corners and contains the following elements: a blue icon of two people, the title "Sesión de Clínica", a label "Nombre de Clínica:" followed by a text input field and a person icon, a label "Contraseña:" followed by a password input field and an eye icon, a "Recuerdame" checkbox, a link "Olvide mi contraseña", a blue "Ingresar" button, and a link "¿No te has registrado? Registrate" at the bottom.

 Untilted Admin

 Search

 Inicio

 Doctores

 Perfil

@UsuarioAdministrador


Inicio



Login:



Inicia Sesión como: **Doctor** | Paciente



Inicio de Sesión


Usuario:


Contraseña:

☐ Recuérdame [Olvide mi contraseña](#)

Ingresar


[¿No te has registrado? Regístrate](#)


 Untilted Admin

 Search

 Dashboard

 Citas

 Avatar

 Configuración

@UsuarioAdministrador

Agregar Cita

Especialidad:

Seleccionar

No lo tengo claro

Cardiólogo

Odontología

Area:

Seleccionar

No lo tengo claro

No aplica

Odontopediatría

Clinica:

Seleccionar

Clinica Médica Gamarro, Edificio Clinicas

Blue Medical Portales

Le Mans, S.A. (Calle Martí)

Profesional:

Seleccionar

No lo tengo claro

Alvaro Ramirez

Oscar Sandoval

« Agosto 2020 »						
Lu	Ma	Mi	Ju	Vi	Sa	Do
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

16:00

16:30

17:00

18:00

18:30

19:00

Siguiente

Proximas citas:

Fecha: Viernes 24 de febrero 2023 15:00 h

CONSULTA DERMATOLOGÍA MÉDICO-QUIRÚRGICA

Clinica San Pedro, Zona 10

Dr. Alvaro Ramirez

Anular Cita

Ver detalle completo

Listado de citas:

Fecha: Viernes 24 de febrero 2023 15:00 h

CONSULTA DERMATOLOGÍA MÉDICO-QUIRÚRGICA

Clinica San Pedro, Zona 10

Dr. Alvaro Ramirez

Ver detalle completo

Fecha: Viernes 24 de febrero 2023 15:00 h

CONSULTA DERMATOLOGÍA MÉDICO-QUIRÚRGICA

Clinica San Pedro, Zona 10

Dr. Alvaro Ramirez

Ver detalle completo

Doctores:

Inicio



Prototipo Fase 4:

https://www.canva.com/design/DAFYKsbC-64/k6b39iO7q-chiTZNn7q8aQ/edit?utm_content=DAFYKsbC-64&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Prototipo Fase Final:

https://www.canva.com/design/DAFf3vdhqYs/H2y_ZkvAAw-TsuoqoPdi7Q/edit?utm_content=DAFf3vdhqYs&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Análisis

Requisitos Funcionales

Pacientes:

- a) Log in
- b) Registrar
- c) Olvidé mi contraseña
- d) Cerrar sesión
- e) Editar perfil
- f) Agendar cita
- g) Seleccionar tipo de médico
- h) Seleccionar área específica del médico
- i) Ver diagnóstico
- j) Descargar diagnóstico

Admin. Clínica:

- a) Registrar clínica (contratar servicio)
- b) Elegir entre médico único o grupo de médicos
- c) Verificar credenciales (una vez, información de la clínica)
- d) Registrar especialidad médica
- e) Registrar área por especialidad
- f) Registrar médico/s
- g) Cambiar contraseñas
- h) Editar médico/s
- i) Editar clínica
- j) Cerrar sesión

Médicos y Asistentes:

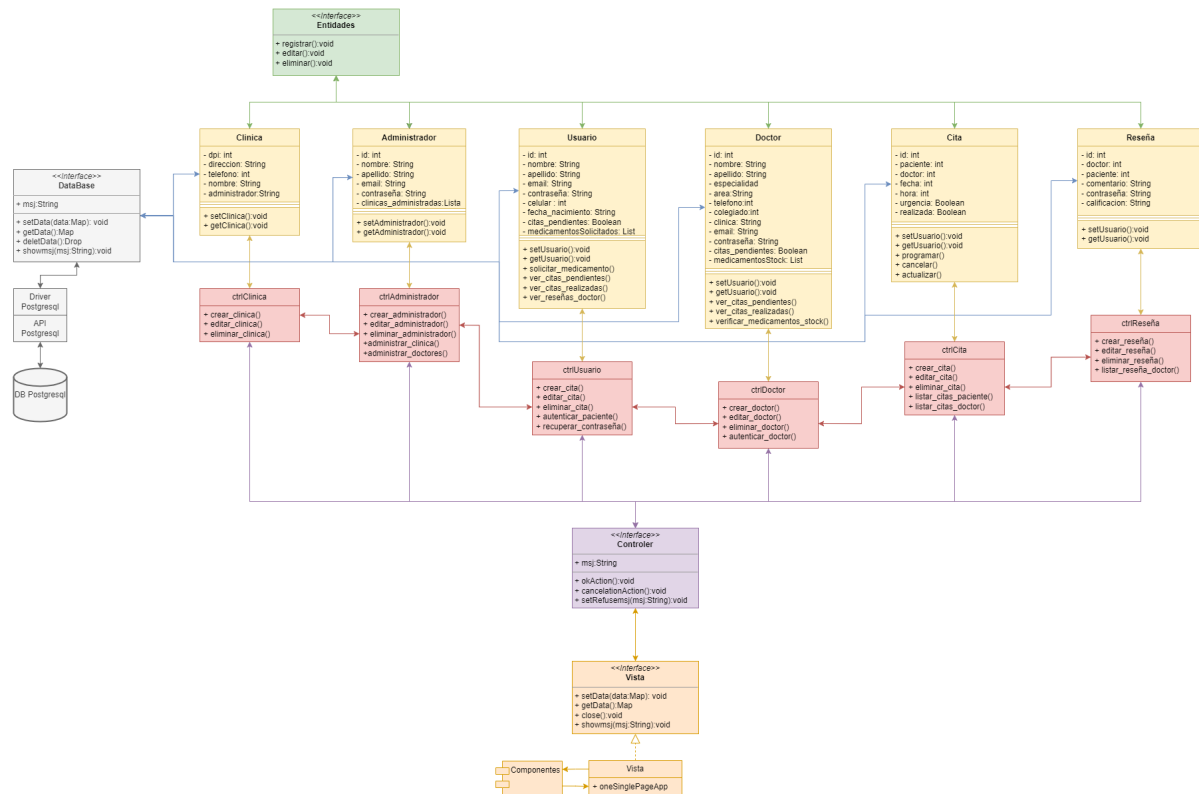
- k) Editar perfil
- l) Calendario de citas
- m) Ver paciente
- n) Asignar diagnósticos
- o) Recetar medicinas
- p) Log in
- q) Cerrar sesión

Historias de Usuario:

- Como paciente, quiero registrar mi cuenta para poder empezar a llevar el control de mis registros médicos.
- Como paciente, quiero iniciar sesión en mi cuenta para llevar el control de mis registros médicos.
- Como paciente, quiero agendar citas para lograr acordar una visita al médico en determinada fecha.
- Como paciente, quiero editar mi perfil para cambiar datos personales en momentos que yo quiera.
- Como paciente, quiero cerrar sesión para poder cambiar de usuario o cuando deje de usar el programa.
- Como paciente, quiero ver mis citas para poder llevar el registro del médico que me atenderá, ver los diagnósticos de la cita, citas pasadas y ver mis medicinas recetadas.
- Como doctor, quiero registrar mi cuenta personal para llevar el control de mis pacientes con el sistema.
- Como doctor, quiero iniciar sesión para poder acceder a los registros de mis pacientes y llevar un control de ellos.
- Como doctor, quiero poder ver mis citas para poder aceptar y organizar el tiempo que tengo que estar trabajando y teniendo un control de entrada de mis pacientes.
- Como doctor, quiero aceptar o rechazar las citas dependiendo de la disponibilidad de agenda de esos días.
- Como doctor, quiero ver pacientes para asignar medicinas que sirvan al paciente la cantidad y hora de las medicinas que tiene que tomar.
- Como doctor, quiero ver a mis pacientes para ver sus diagnósticos pasados y sus antecedentes médicos.
- Como doctor, quiero ver a mis pacientes para subir su diagnóstico y actualizar su perfil médico.
- Como doctor, quiero ver a mis pacientes para sugerirles una nueva cita y proceder con sus consultas y progreso médico.
- Como administrador de clínicas, quiero registrarme para poder escoger un plan de cantidad de médicos y pagarlo.
- Como administrador de clínicas, quiero iniciar sesión para llevar control de los doctores de la clínica.
- Como administrador de clínicas quiero crear mi clínica para registrar los doctores y poder agregarlos.
- como administrador de clínicas, quiero editar los médicos para cambiar su información y dar detalles.
- Como administrador de clínicas, quiero eliminar doctores para controlar la cantidad, por si algún doctor decide retirarse o no adquirir el servicio.

Backend

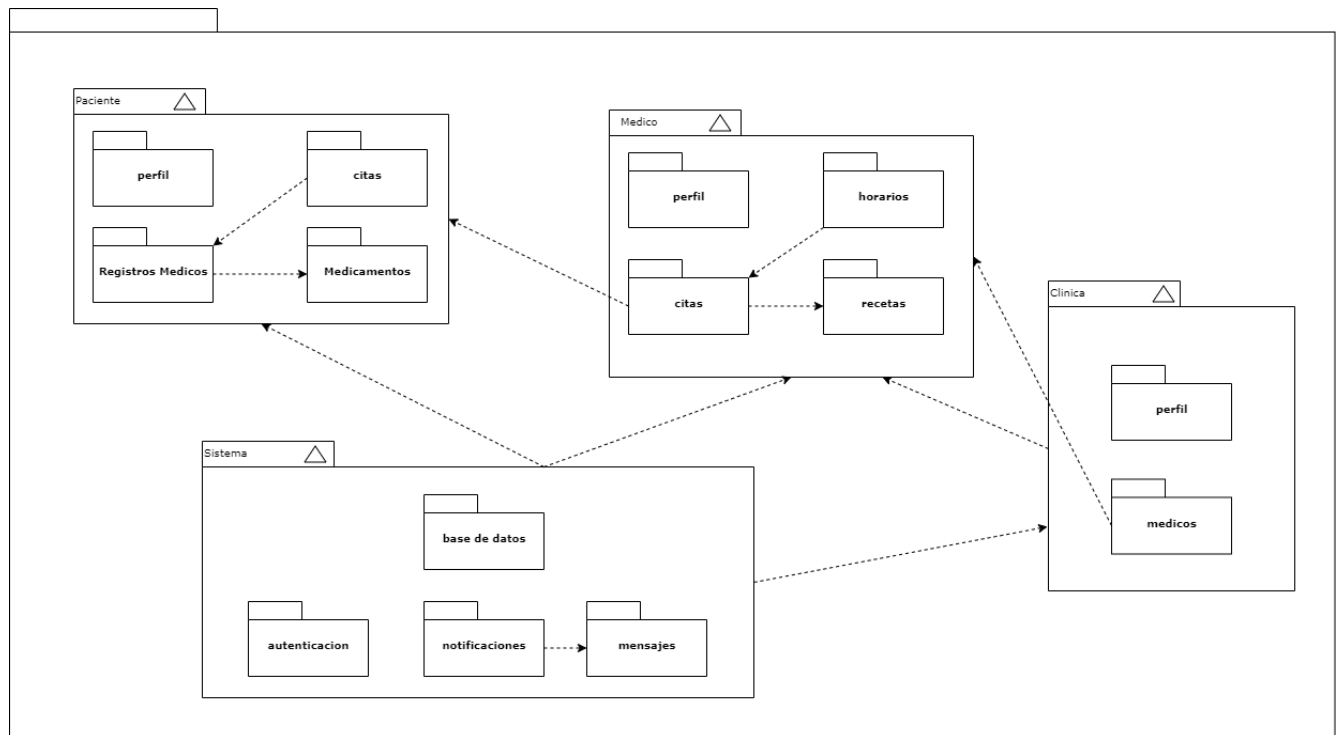
Diagrama de Clases



- La clase Paciente representa a los pacientes que utilizan el sistema. Tiene atributos como ID, nombre, apellido, correo electrónico, contraseña, citas pendientes, citas realizadas y medicamentos solicitados. Los métodos incluyen solicitar medicamento, ver citas pendientes, ver citas realizadas y ver reseñas de médicos.
- La clase Doctor representa a los médicos que utilizan el sistema. Tiene atributos como ID, nombre, apellido, correo electrónico, contraseña, citas pendientes, citas realizadas y medicamentos en stock. Los métodos incluyen ver citas pendientes, ver citas realizadas y verificar medicamentos en stock.
- La clase Administrador representa a los administradores que utilizan el sistema. Tiene atributos como ID, nombre, apellido, correo electrónico, contraseña y clínicas administradas. Los métodos incluyen administrar clínica y administrar médicos.
- La clase Clinica representa a las clínicas que utilizan el sistema. Tiene atributos como ID, nombre, ubicación y doctores. Los métodos incluyen agregar doctor y eliminar doctor.

- La clase Cita representa una cita programada entre un paciente y un médico. Tiene atributos como ID, paciente, médico, fecha, hora, tipo (regular o emergencia) y estado. Los métodos incluyen programar cita, cancelar cita y actualizar cita.
- La clase Reseña representa una reseña de un paciente sobre un médico. Tiene atributos como ID, médico, paciente, comentario y calificación. Los métodos incluyen crear reseña, editar reseña y eliminar reseña.
- Los controladores son clases que se encargan de gestionar y controlar la interacción entre el usuario y el sistema.
- El PacienteController gestiona la creación, edición y eliminación de pacientes, la autenticación de pacientes y la recuperación de contraseñas.
- El DoctorController gestiona la creación, edición y eliminación de médicos, y la autenticación de médicos.
- El AdministradorController gestiona la creación, edición y eliminación de administradores, y la autenticación de administradores.
- El ClinicaController gestiona la creación, edición y eliminación de clínicas.
- El CitaController gestiona la creación, edición y eliminación de citas, y la lista de citas para pacientes y médicos.
- El ReseñaController gestiona la creación, edición y eliminación de reseñas y la lista de reseñas de médicos.

Diagrama de Paquetes



Paquetes:

- Pacientes:

El paquete de pacientes es el que incluye toda la información de los usuarios que atienden a las clínicas. Aquí se guardará toda la información del paciente y también todas las citas pendientes/realizadas del paciente, al igual que los diagnósticos hechos por los médicos en cada cita. Sus componentes son:

- Perfil: este paquete guarda la información personal del paciente.
- Citas: este paquete guarda la información de las citas programadas del paciente.
- Registros Médicos: este paquete guarda todos los diagnósticos realizados por los médicos a través de la aplicación.
- Medicamentos: este paquete guarda los medicamentos recetados por los médicos en cada cita.

- Médicos:

El paquete de médicos incluye la información de los médicos, como su teléfono, su horario y la clínica en la que trabaja. Sus componentes son:

- Perfil: este paquete guarda la información personal del médico.
- Horarios: este paquete incluye los horarios en los que ejerce el médico
- Citas: este paquete guarda las citas programadas del médico.
- Recetas: este paquete contiene las recetas que ha dejado el médico.

- Clínica:

El paquete de clínica incluye toda la información de la clínica, como su ubicación, su nombre y sus médicos. Sus componentes son:

- Perfil: este paquete guarda la información de la clínica.
- Médicos: este paquete contiene a todos los médicos de la clínica.

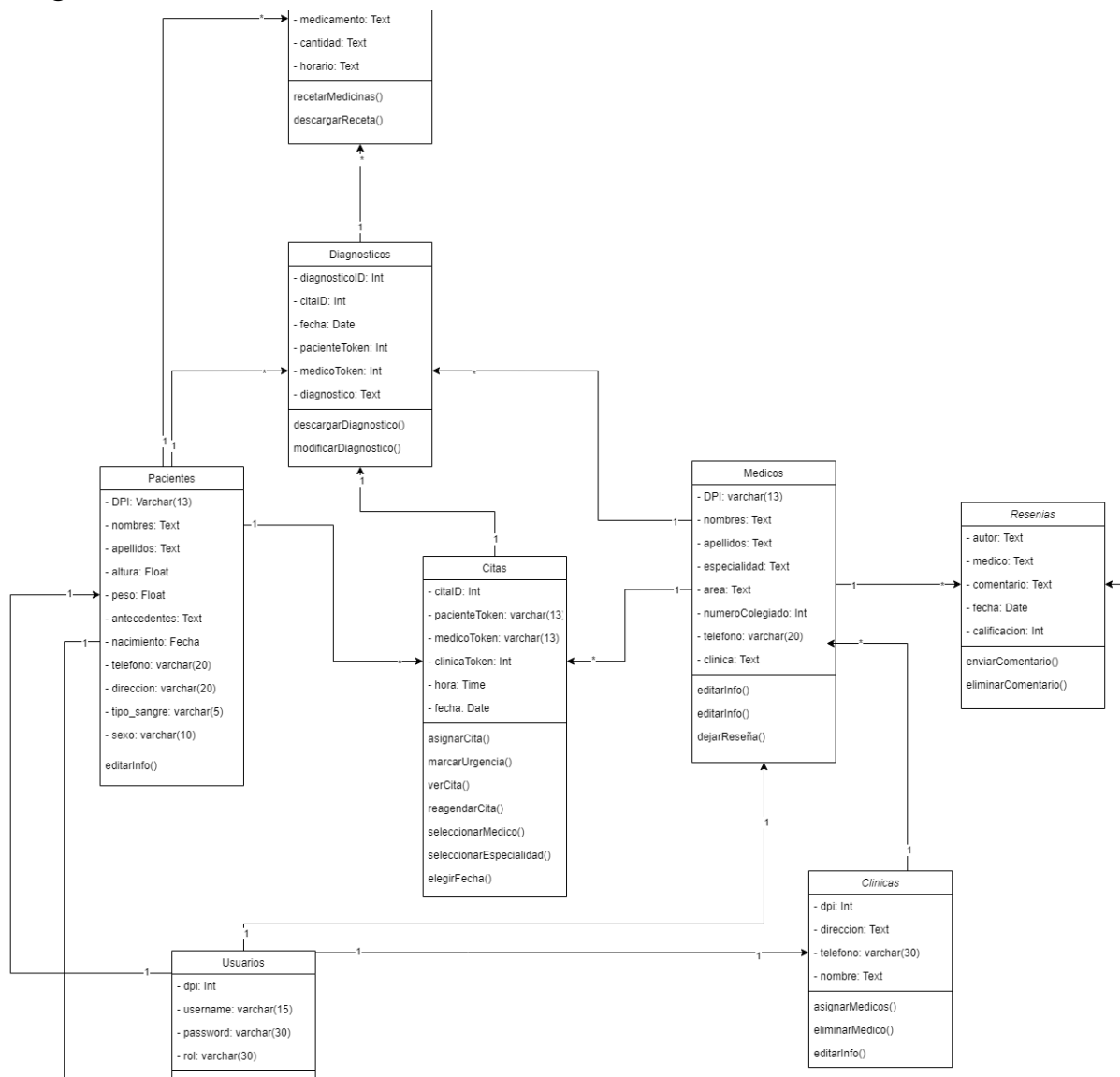
- Sistema:

El paquete de sistema incluye todo lo que se trabajará en el backend, es decir las funciones de las cuales el usuario no debe preocuparse y son las que harán el trabajo de la aplicación. Sus componentes son:

- Base de datos: este paquete contiene toda la base de datos que guarda la información de cada paquete
- Autenticación: este paquete se encarga de la autenticación por usuario.
- Notificaciones: este paquete se encarga de las notificaciones a los usuarios.
- Mensajes: este paquete incluye los mensajes que son enviados a cada usuario a través de las notificaciones.

Persistencia de Datos

Diagrama de Clases Persistentes



Tipo de Almacenamiento de Datos

El tipo de almacenamiento de datos seleccionado es de tipo relacional.

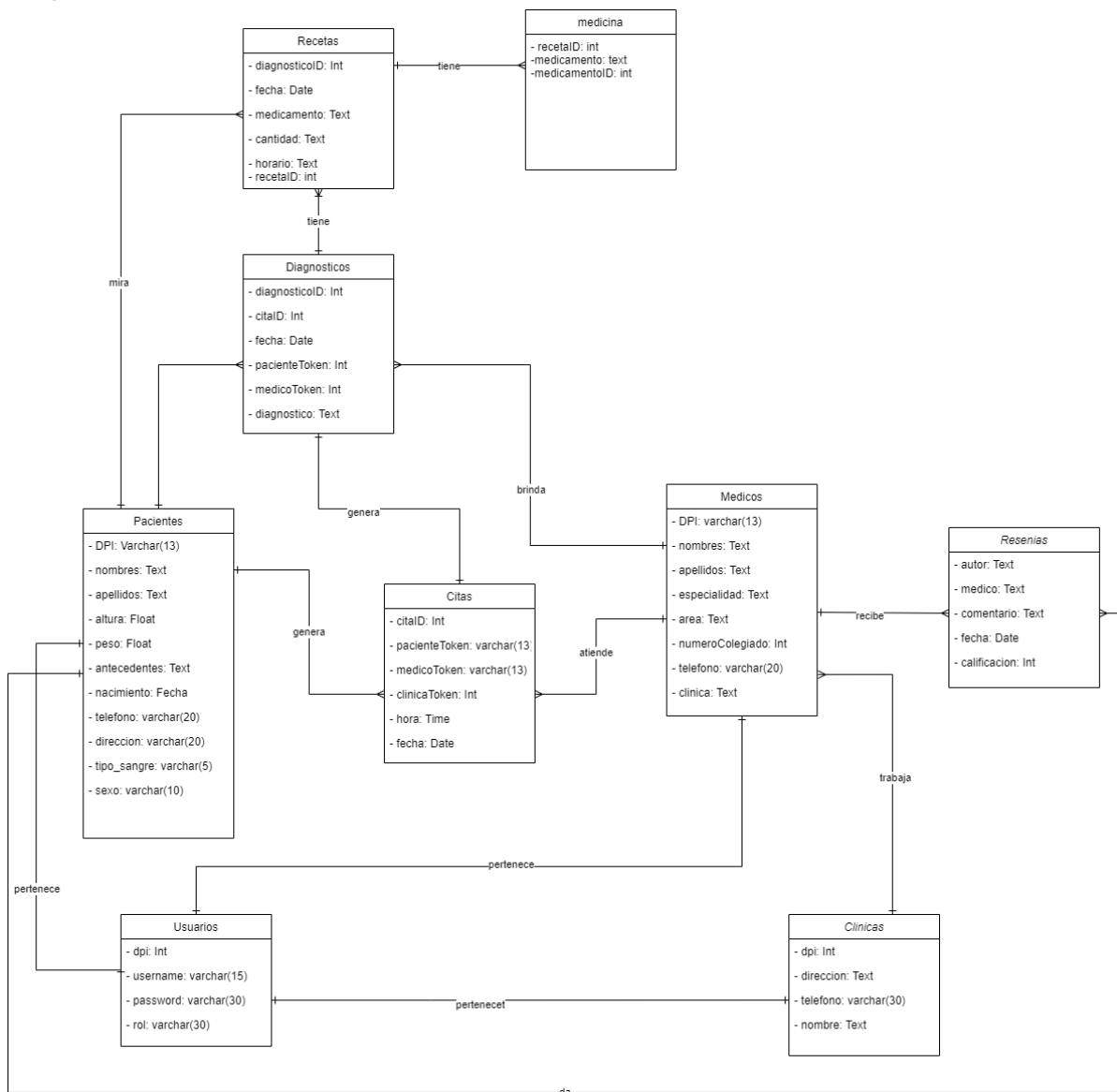
Razones:

- Estructura de datos: Un modelo de datos relacional permite estructurar los datos en tablas relacionales, lo que permite organizar y almacenar los datos de pacientes, médicos y citas de manera eficiente. Esto permite que la información esté bien organizada y sea fácil de acceder y actualizar.
- Integridad de datos: Un sistema de almacenamiento de datos relacional permite definir restricciones de integridad referencial para garantizar la coherencia de los datos en toda la aplicación. Por ejemplo, se puede establecer que un paciente solo

puede agendar citas con un médico existente, lo que asegura que no se creen registros de citas no relacionadas.

- Escalabilidad: Los sistemas de almacenamiento de datos relacionales pueden ser escalables, lo que significa que se pueden agregar más recursos para manejar grandes volúmenes de datos y una gran cantidad de usuarios.
- Consultas y análisis: Un modelo de datos relacional permite realizar consultas y análisis complejos en los datos almacenados. Por ejemplo, se pueden realizar consultas para obtener información sobre pacientes específicos o médicos que han tratado a ciertos pacientes. Además, las herramientas de análisis y generación de informes pueden ayudar a extraer información valiosa para la toma de decisiones de la clínica.

Diagrama de Entidad Relación



Diseño

Requisitos No Funcionales

Requisito no funcional	Forma en que se medirá su cumplimiento.	Estimaciones
Cantidad de pacientes utilizando la aplicación al mismo tiempo.	Simulación de pacientes interactuando con la aplicación al mismo tiempo.	200 - 800 pacientes agendando citas.
Actualización de información en tiempo real	Agendando citas al mismo tiempo en distintas sesiones de pacientes.	Tiempo estimado de actualización en tiempo real: 0.25 segundos
La aplicación debe ser capaz de manejar un gran volumen de datos sin disminuir la velocidad o la calidad del servicio.	Evalutando el tiempo de respuesta, tasa de errores, uso de recursos y ancho de banda donde tienen que tener menos del 5% de error.	Tiempo de respuesta estimado por interacción: 0.5 segundos
La aplicación debe cumplir con los estándares de seguridad de la industria para proteger la información del paciente.	Se pueden obtener certificaciones de seguridad, como ISO 27001, para demostrar el cumplimiento de la aplicación con las mejores prácticas de seguridad.	Si es confiable
El programa debe ser capaz de manejar un gran volumen de pacientes, médicos y registros médicos a medida que la clínica crece.	Utilizando pruebas de carga y rendimiento. Utilizando herramientas que generan una carga simulada en el sistema.	Volumen de pacientes estimado: 10,000 Volumen de médicos: 500 Volumen de clínicas: 100 Volumen de registros: 70,000
La aplicación debe cargar y mostrar información de manera rápida y eficiente.	utilizando métricas como el tiempo de carga, el rendimiento de la base de datos, el uso de la memoria y la CPU y las métricas de red.	Tiempo estimado de respuesta: 0.25 segundos
El sistema debe de estar funcionando las 24 horas del día todos los días.	Se puede medir en porcentaje, donde un 100% significa que el sistema está funcionando todo el tiempo. Se puede utilizar una herramienta de monitoreo de tiempo de actividad para medir esto.	Estimación: la aplicación funciona 24/7

Tecnologías a utilizar

Angular

Angular es un framework de desarrollo de aplicaciones web de front-end desarrollado por Google. Angular utiliza el lenguaje de programación de TypeScript que es una extensión de JavaScript que tiene una sintaxis más concisa. La estructura de un proyecto en Angular se basa en módulos, componentes y servicios, lo que permite una mayor reutilización de código y una estructura más clara y organizada.

Ventajas:

- Modularidad: Angular utiliza un enfoque modular para el desarrollo de aplicaciones web, lo que permite una mayor reutilización de código y una estructura más clara y organizada.
- Amplio conjunto de características: Angular proporciona un gran conjunto de características para el desarrollo de aplicaciones web, incluyendo enrutamiento, validación de formularios, animaciones, inyección de dependencias y más.
- Soporte de TypeScript: TypeScript es un lenguaje de programación más seguro y fácil de usar que JavaScript, lo que permite una mayor seguridad y una sintaxis más concisa.

Desventajas:

- Curva de aprendizaje empinada: Angular es un framework complejo con una curva de aprendizaje empinada, lo que puede ser un desafío para los desarrolladores nuevos en el framework.
- Tamaño del paquete: Debido a su gran conjunto de características y módulos, las aplicaciones Angular tienden a tener un tamaño de paquete más grande que las aplicaciones construidas con otros frameworks.
- Compatibilidad con versiones antiguas: Las actualizaciones de Angular pueden ser complicadas y pueden requerir cambios significativos en el código de una aplicación, lo que puede ser un problema para las aplicaciones más grandes y complejas.

Razones de selección:

- Los integrantes del equipo de desarrollo están familiarizados con Angular, lo cual facilitará el desarrollo de la aplicación.
- La comunidad de Angular es bastante amplia, por lo tanto, conseguir información sobre posibles problemas que se puedan presentar será fácil de obtener.
- La cantidad de librerías que existen de Angular es muy amplia.
- TypeScript es un lenguaje muy fácil de utilizar a comparación de JavaScript, el equipo de desarrollo se encuentra ya acostumbrado a JS, por lo tanto la migración a TS no será un problema.

Flask

Flask es un framework para aplicaciones web para Python, utilizado para construir aplicaciones ligeras y medianas. Flask proporciona una variedad de características útiles, como rutas, sesiones, autenticación y plantillas HTML.

Ventajas:

- Flexibilidad: Flask es un framework altamente flexible que permite a los desarrolladores crear aplicaciones web personalizadas y adaptadas a sus necesidades específicas.
- Ligereza: Flask es un framework minimalista y ligero que no requiere una gran cantidad de recursos del sistema, lo que lo convierte en una buena opción para aplicaciones web más pequeñas y medianas.
- Curva de aprendizaje: Flask es fácil de aprender y utilizar, especialmente para los desarrolladores de Python que están familiarizados con la programación orientada a objetos.

Desventajas:

- Limitaciones para aplicaciones web más grandes: Debido a que Flask es un framework minimalista, puede no ser la mejor opción para aplicaciones web más grandes y complejas que requieren una mayor cantidad de características y funcionalidades.
- Falta de estructura: Flask no proporciona una estructura rígida para la organización del proyecto, lo que puede hacer que sea más difícil para los desarrolladores mantener el código limpio y organizado.
- Necesidad de integrar manualmente algunas características: Aunque Flask proporciona algunas características útiles, como rutas y sesiones, los desarrolladores pueden tener que integrar manualmente algunas características adicionales, lo que puede ser un desafío para aquellos que no están familiarizados con Python.

Razones de selección:

- Flask es un framework bastante flexible y simple de utilizar, además uno de los integrantes de desarrolladores ya ha trabajado con Flask. El equipo confía en que es posible aprender todos juntos con la ventaja de que un integrante ya tiene conocimiento sobre el framework.
- Es un framework bastante ligero, no es tan complejo como muchos otros en el mercado.

PostgreSQL

Es un sistema de código abierto de administración de bases de datos del tipo relacional que permite trabajar con conjuntos de datos complejos sin ralentizaciones, facilitando el almacenamiento, lectura y escritura de información.

Ventajas:

- Gran escalabilidad: Nos permite configurar PostgreSQL en cada equipo según el hardware. Por lo que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima.
- Extensibilidad: Gran variedad de extensiones distribuidas por el grupo de desarrolladores de PostgreSQL. También por terceros o incluso nosotros mismos podemos crear nuestras propias extensiones.
- Potencia y robustez: Cumple con la característica de ACID Compliant que es el acrónimo de Atomicity, Consistency, Isolation y Durability

Desventajas:

- Sintaxis: Alguno de los comandos o sentencias puede llegar a no ser intuitiva si no tienes un nivel medio de conocimientos en lenguaje SQL.
- Curva de aprendizaje: Para aquellos que no están familiarizados con PostgreSQL, puede haber una curva de aprendizaje pronunciada para entender su estructura y sintaxis.
- Consumo de recursos: PostgreSQL puede consumir más recursos del sistema en comparación con otros sistemas de gestión de bases de datos, lo que puede afectar el rendimiento de otras aplicaciones en el mismo servidor.

Razones de selección:

- Fiabilidad y seguridad: PostgreSQL dado que es conocido por su alta fiabilidad y seguridad, nos ayudará bastante para el control de datos críticos con la información crucial de pacientes, hospital, clínica o centro de salud.
- Escalabilidad: Dado que Postgre puede manejar grandes volúmenes de datos, es una excelente opción para nuestro proyecto debido a toda la información esperada que almacenaremos de cada paciente, médico, hospital, centro de salud o clínica.
- Compatibilidad con Python: PostgreSQL es compatible con Python y cuenta con bibliotecas y módulos que facilitan su integración con Flask y Angular, frameworks utilizados en el proyecto de administración de pacientes.

Python

Python es un lenguaje de programación a alto nivel orientada a objetos para desarrollo de aplicaciones, este lenguaje a comparación de otros cuenta con una semántica dinámica integrada. Con este lenguaje de programación podemos ser capaces de hacer muchas cosas relacionadas con desarrollo web, análisis de datos, seguridad informática entre otros.

Ventajas:

- Lenguaje de alto nivel: Para programar en Python se pueden usar elementos del lenguaje natural, ya que tiene una sintaxis similar al inglés.

- Bibliotecas y Frameworks: Python tiene una amplia variedad de bibliotecas y frameworks dado que contiene muchos módulos integrados, además pueden encontrar bibliotecas adicionales disponibles en PyPI.
- Baja curva de aprendizaje: Dado que la sintaxis es muy sencilla en Python, se pueden escribir programas con pocas líneas de código, por lo que su línea de aprendizaje, cualquier persona puede aprender programas sencillos.

Desventajas:

- Consumo de memoria: El consumo de memoria de Python es muy alto, y esto se debe a la flexibilidad de los tipos de datos.
- Desarrollo móvil: Python no es el mejor lenguaje para desarrollar aplicaciones móviles, dado que existen mejores lenguajes para desarrollar.
- Procesamiento lento: Python puede llegar a procesar de forma lenta, debido a la versatilidad.

Razones de selección:

- Facilidad de aprendizaje: Python es un lenguaje de programación "fácil" de aprender, por eso como grupo decidimos usar este lenguaje de programación ya que cada integrante del grupo cuenta con conocimientos aceptables y un buen dominio.
- Amplia variedad de bibliotecas y módulos: Python cuenta con una amplia variedad de bibliotecas y módulos que pueden ser de gran utilidad para la implementación del proyecto, lo que reduce significativamente el tiempo y esfuerzo requerido para su desarrollo.
- Integración con manejo de bases de datos: Python cuenta con varias bibliotecas y módulos que se acoplan muy bien con el manejo de bases de datos, lo que facilita su integración con Flask y Angular, frameworks utilizados en el proyecto.

Gestor de Bases de Datos

DBeaver

DBeaver es una herramienta de administración de bases de datos y cliente SQL gratuita y de código abierto que es compatible con una amplia variedad de bases de datos, incluyendo MySQL, PostgreSQL, Oracle, SQL Server, MongoDB y muchos más.

Ventajas:

- Soporte de múltiples bases de datos: DBeaver es compatible con una amplia variedad de bases de datos, lo que lo convierte en una herramienta versátil para los desarrolladores y administradores de bases de datos.
- Funcionalidades avanzadas: DBeaver cuenta con una amplia gama de funcionalidades avanzadas, como autocompletado de código SQL, visualización de datos en gráficos y diagramas, y soporte para transacciones y control de versiones.
- Interfaz de usuario intuitiva: DBeaver cuenta con una interfaz de usuario intuitiva y fácil de usar que permite a los usuarios navegar fácilmente por las diferentes funciones y características de la herramienta.

Desventajas:

- Curva de aprendizaje: DBeaver es una herramienta avanzada y puede tener una curva de aprendizaje empinada para los usuarios nuevos o menos experimentados.
- Recursos del sistema: DBeaver puede requerir una gran cantidad de recursos del sistema, lo que puede ser un problema para aquellos que tienen recursos limitados en sus computadoras o servidores.
- Falta de soporte para algunas características avanzadas: Aunque DBeaver cuenta con una amplia gama de funcionalidades avanzadas, algunas características pueden no estar disponibles o tener un soporte limitado para algunas bases de datos.

Razones de selección:

- Soporte multiplataforma: DBeaver es compatible con diferentes sistemas operativos, lo que lo hace ideal para equipos de desarrollo que trabajan en diferentes plataformas.
- Conexión con diferentes tipos de bases de datos: DBeaver permite la conexión con diferentes tipos de bases de datos, lo que le brinda la flexibilidad para trabajar con diferentes sistemas de gestión de bases de datos.
- Funcionalidades avanzadas: DBeaver cuenta con una gran cantidad de funcionalidades avanzadas que facilitan la gestión de la base de datos, como por ejemplo: edición de tablas, creación de vistas, importación/exportación de datos, entre otras.

NaviCat

Navicat es una herramienta de administración de bases de datos y cliente SQL que es compatible con una amplia variedad de bases de datos, incluyendo MySQL, PostgreSQL, Oracle, SQL Server y muchos más.

Ventajas:

- Soporte de múltiples bases de datos: Navicat es compatible con una amplia variedad de bases de datos, lo que lo convierte en una herramienta versátil para los desarrolladores y administradores de bases de datos.
- Funcionalidades avanzadas: Navicat cuenta con una amplia gama de funcionalidades avanzadas, como la importación y exportación de datos, la sincronización de bases de datos y la generación de informes y gráficos.
- Interfaz de usuario intuitiva: Navicat cuenta con una interfaz de usuario intuitiva y fácil de usar que permite a los usuarios navegar fácilmente por las diferentes funciones y características de la herramienta.
- Funcionalidades colaborativas: Navicat cuenta con funcionalidades para trabajar de forma colaborativa con otras personas, lo que facilita la colaboración en proyectos de bases de datos.

Desventajas:

- Precio: Navicat es una herramienta comercial y puede ser costosa para algunos usuarios o empresas.
- Recursos del sistema: Navicat puede requerir una gran cantidad de recursos del sistema, lo que puede ser un problema para aquellos que tienen recursos limitados en sus computadoras o servidores.
- Curva de aprendizaje: Navicat es una herramienta avanzada y puede tener una curva de aprendizaje empinada para los usuarios nuevos o menos experimentados.

Razones de selección:

- Colaboración en equipo: NaviCat permite trabajar en equipo en la administración de bases de datos. Con NaviCat, múltiples usuarios pueden trabajar simultáneamente en la misma base de datos, lo que facilita la colaboración y el intercambio de información en tiempo real.
- NaviCat es compatible con una amplia gama de bases de datos, lo que significa que puedes usarla con diferentes tipos de bases de datos, incluyendo MySQL, PostgreSQL, Oracle, SQL Server, SQLite y más.
- NaviCat tiene una comunidad activa y sólida de usuarios y desarrolladores que comparten sus conocimientos y experiencias en foros y grupos de discusión en línea. Esto significa que siempre hay alguien dispuesto a ayudarte si tienes alguna pregunta o problema.

pgAdmin 4

PgAdmin 4 es una herramienta de administración de bases de datos gratuita y de código abierto para PostgreSQL, una base de datos relacional muy popular.

Ventajas:

- Soporte para múltiples plataformas: PgAdmin 4 está disponible en varias plataformas, incluyendo Windows, macOS y Linux, lo que la hace accesible para una gran variedad de usuarios.
- Interfaz de usuario intuitiva: La interfaz de usuario de PgAdmin 4 es fácil de usar y permite a los usuarios navegar rápidamente a través de las diferentes funciones y características de la herramienta.
- Amplia gama de características: PgAdmin 4 ofrece una amplia gama de características y funcionalidades para ayudar a los usuarios a administrar sus bases de datos de forma efectiva, como la creación y edición de tablas, la ejecución de consultas SQL y la administración de usuarios y permisos.

Desventajas:

- Inestabilidad: Aunque PgAdmin 4 es una herramienta confiable, algunos usuarios han reportado problemas de inestabilidad y errores inesperados al utilizar la herramienta.
- Curva de aprendizaje: PgAdmin 4 es una herramienta avanzada y puede tener una curva de aprendizaje empinada para los usuarios nuevos o menos experimentados.
- Limitaciones en la versión gratuita: Si bien PgAdmin 4 es gratuito, algunas de sus características y funcionalidades avanzadas pueden estar disponibles solo en la versión de pago.

Razones de selección:

- Elegimos PgAdmin como herramienta de administración de bases de datos porque es una herramienta de código abierto y gratuita, compatible con múltiples plataformas, con una interfaz de usuario intuitiva y una amplia gama de características y funcionalidades para ayudar a los usuarios a administrar sus bases de datos de manera efectiva. Además, nuestro equipo de desarrollo tiene experiencia previa con esta herramienta, lo que puede facilitar la adopción y el aprendizaje. En general, consideramos que PgAdmin 4 es una herramienta confiable y eficiente que puede ayudar a nuestro equipo a trabajar de manera más eficiente y efectiva en proyectos que requieren la administración de bases de datos PostgreSQL.

Informe de gestión

- Prototipos
- Diagrama de clases
- Persistencia
- Selección de la tecnología a utilizar

Formulario LOGT

Javier Prado	21486					
Fecha	Inicio	Fin	Tiempo de interrupcion	Delta tiempo	Fase	Comentarios
10/04/2023	18:00	18:30	3 minutos	27 minutos	corte 3	fui por agua
11/04/2023	11:00	02:50	30 minutos	03:30	corte 3	fui hacer mi comida
12/04/2023	04:00	07:00		3	corte 3	

Mario Cristales						
Fecha Inicial	Inicio	Fin	Tiempo de interr	Tiempo Total	Fase	Comentarios
29-03-2023	17:00	17:35	0	0:35	3	
11-04-2023	18:19	23:33	1:40	3:34	3	Cené
12-04-2023	1:00	1:41	0	0:41	3	
12-04-2023	13:58	18:31	1:00	3:33	3	Fui a clase

Javier Heredia						
Fecha Inicial	Inicio	Fin	Tiempo de interrupcion	Tiempo Total	Fase	Comentarios
10-04-2023	16:25	17:10	15	0:45	3	comi un snak
11-04-2023	17:20	18:10	15	0:50	3	fui al baño
12-04-2023	14:25	19:10	15	4:45	3	almorce

Bryan España						
Fecha inicial	Inicio	Fin	Tiempo interrupción	Tiempo total	Fase	Comentario
10-04-2023	16:00:00	17:00:00	20m	40m	3	Comida
11-04-2023	17:00:00	18:00:00	15	45m	3	Comida
12-04-2023	14:00:00	16:00:00	10	1:50m	3	Comida