Un dibujo de una cara feliz

Descripción generada automáticamente con confianza media

**UNIVERSIDAD DON BOSCO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE COMPUTACIÓN**

**Diseño y Programación de Software Multiplataforma DPS941**

**​Actividad:  
Proyecto - Fase I**

**Integrantes:  
Alberto Elena, Bryan Josué - AE210567**

**Colocho Díaz, Jairo Rafael - CD210488**

**González Crespín, Fátima Argentina - GC200404**

**Montano González, Fernando Josué - MG210111**

**Menjívar Ramírez, David Gerardo - MR210455**

**Salas Bojórquez, Kallahan Andrea - SB210537**

**Encargado:  
Alexander Alberto Siguenza Campos**

**24/03/2024**

**Introducción**

El desarrollo de aplicaciones móviles ha alcanzado una importancia fundamental en la era actual, donde los dispositivos móviles se han convertido en herramientas que siempre están presentes en la vida diaria de las personas. En este proyecto, vamos a desarrollar una aplicación móvil destinada a mejorar la gestión y operatividad del Asilo Nueva Esperanza, abordando diversas necesidades y facilitando la prestación de servicios para sus residentes.

Para llevar a cabo este proyecto, hemos decidido adoptar una metodología ágil, en particular, orientada a Scrum. Esta elección se basa en la necesidad de realizar un proceso de desarrollo colaborativo y al mismo tiempo eficiente, y nos proporcionará un marco de trabajo flexible que nos permitirá adaptarnos rápidamente a los cambios y responder de manera efectiva a los requerimientos del usuario. Utilizaremos herramientas como Trello para la gestión de tareas y seguimiento del progreso del proyecto, lo que nos permitirá mantenernos organizados y enfocados en los objetivos establecidos.

**Objetivos**

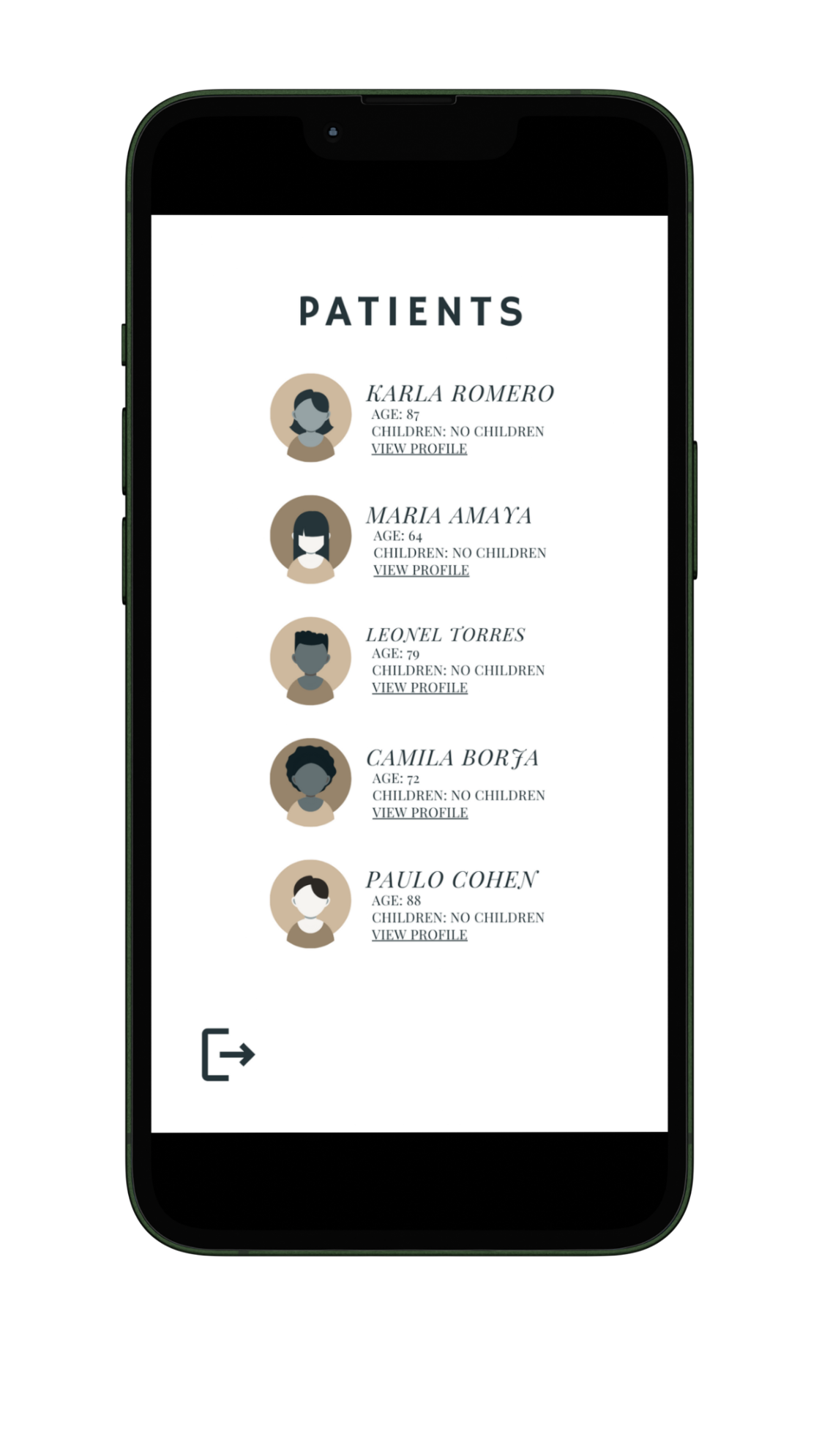
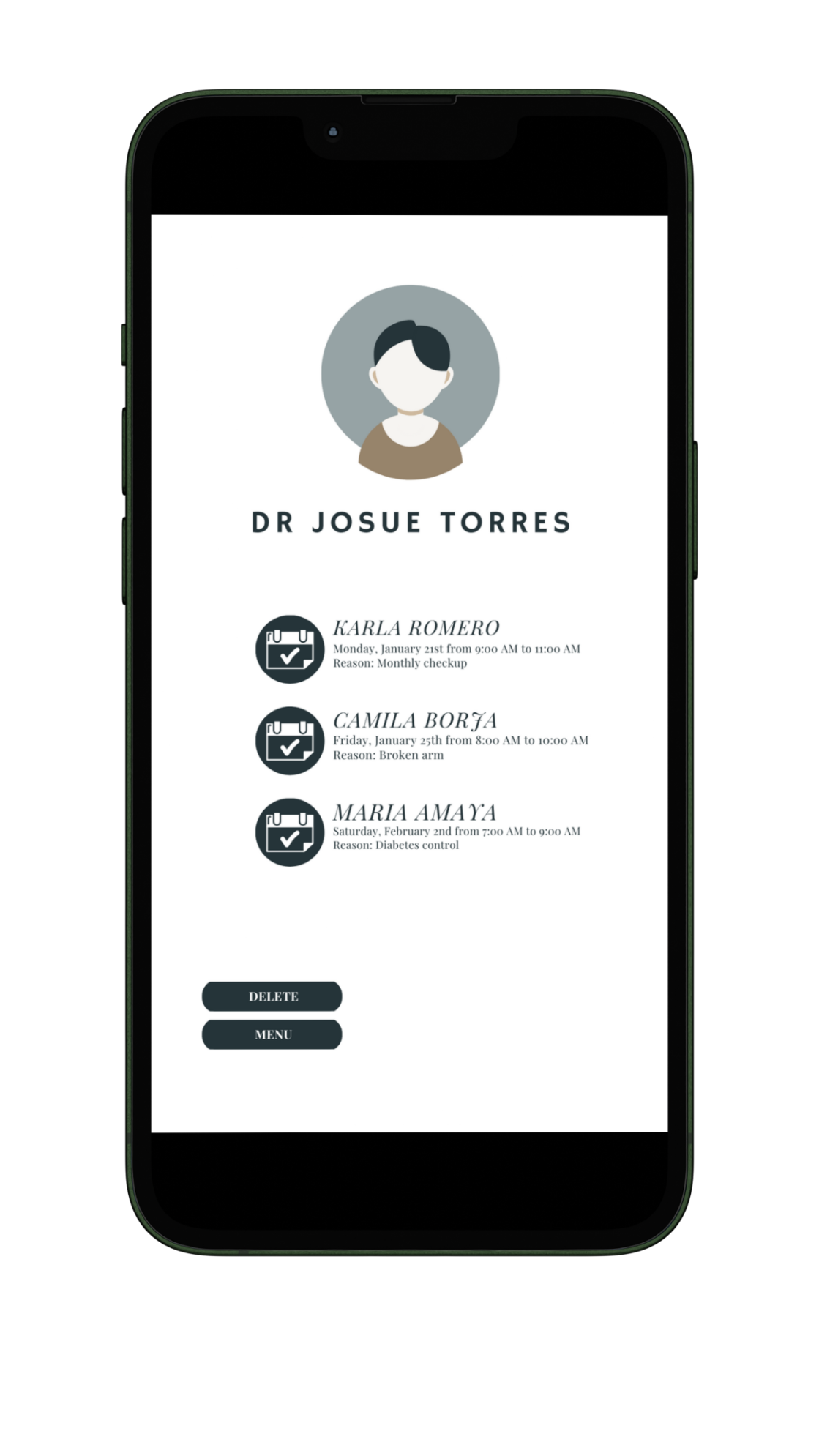
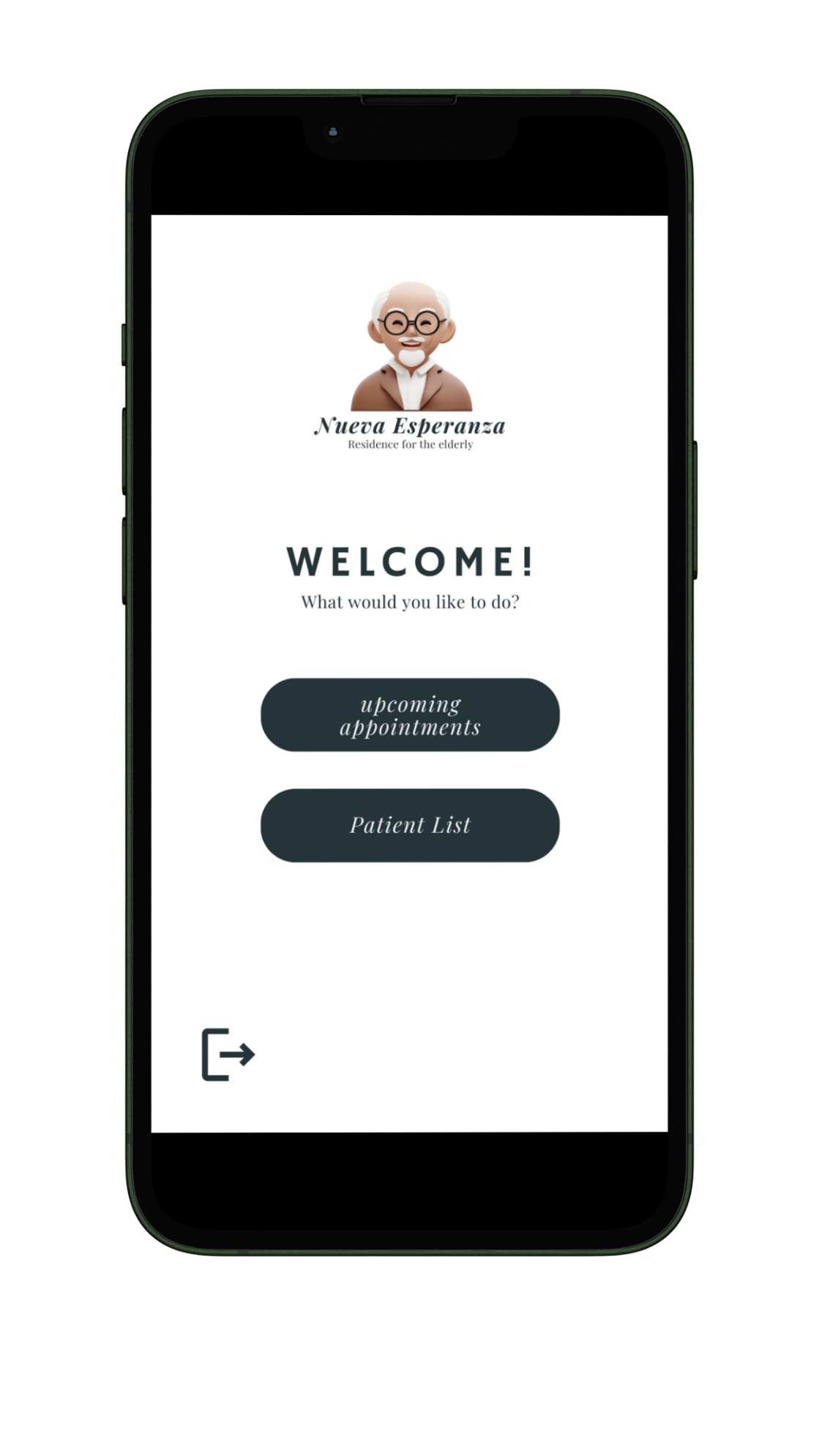
Objetivo General: Crear una aplicación móvil segura y completa para así poder gestionar eficientemente los registros médicos en el Asilo de Ancianos Esperanza de Santa Ana. Esta herramienta tiene como meta principal mejorar la administración de pacientes, médicos y citas, evitando pérdidas de información y asegurando un historial de citas completo y preciso.

Objetivos Específicos:

1. Crear un mecanismo de autenticación seguro que posibilite el acceso a la aplicación mediante credenciales de usuario y contraseña. Además, se implementará la opción de inicio de sesión mediante cuentas de Gmail para brindar mayor comodidad y seguridad a los usuarios.
2. Diseñar una interfaz intuitiva y práctica en la aplicación móvil, optimizando la navegación y permitiendo a los usuarios ejecutar acciones como agregar, actualizar y visualizar información relacionada con pacientes, médicos y citas de forma intuitiva y sin complicaciones.
3. Implementar un sistema de permisos y roles personalizados en la aplicación, asegurando que los médicos accedan exclusivamente a las funciones asociadas con citas y pacientes, mientras que los administradores tengan el control total sobre la gestión de la información y los usuarios del sistema.

**Captura de pantalla de un celular

Descripción generada automáticamenteDiseño y mockups**

**Lógica para seguir**

Actores:

* Paciente: Usuario del asilo Nueva Esperanza, que busca atención médica y necesita gestionar su historial y citas.
* Médico: Profesional de la salud encargado de atender a los pacientes y registrar información clínica importante.
* Administrador: Encargado de administrar la plataforma, gestionar usuarios, citas y garantizar su correcto funcionamiento.

Casos de uso:

1. Autenticación:

* Actores: Paciente, Médico, Administrador
* Descripción: Los usuarios tienen la posibilidad de acceder a la aplicación mediante autenticación, ya sea ingresando sus credenciales personales (usuario y contraseña) o utilizando sus cuentas de Gmail para una experiencia más ágil.

1. Visualizar información del paciente:

* Actores: Médico, Administrador
* Descripción: Los médicos y administradores pueden explorar en detalle la información relacionada con un paciente, incluyendo un registro exhaustivo de citas anteriores, padecimiento, medicación y citas próximas. Esto les permite brindar una atención más personalizada y efectiva.

1. Gestión de citas:

* Actores: Paciente, Médico
* Descripción: Los pacientes tienen la capacidad de solicitar nuevas citas o cancelar citas existentes de manera fácil y conveniente. Por otro lado, los médicos pueden acceder a su agenda de citas, gestionar su disponibilidad y marcar las citas realizadas para mantener un seguimiento de su agenda de trabajo.

1. Actualizar datos personales:

* Actores: Administrador
* Descripción: Se brinda a los usuarios la posibilidad de mantener actualizada su información personal dentro de la plataforma, lo que incluye detalles como dirección, números de contacto y cualquier información médica relevante que pueda impactar en su atención.

1. Registro de historial de citas:

* Actores: Médico
* Descripción: Este caso de uso permite a los médicos registrar los acontecimientos y observaciones relevantes de cada consulta. Desde diagnósticos hasta tratamientos, pasando por notas específicas, toda esta información se consolida en el historial médico del paciente.

1. Generación de informes:

* Actores: Administrador
* Descripción: El administrador tiene la capacidad de generar informes y estadísticas exhaustivas sobre la actividad de citas, pacientes y médicos en un periodo específico.

1. Administración de usuarios y roles:

* Actores: Administrador
* Descripción: Se otorga al administrador la responsabilidad de gestionar los usuarios y sus roles dentro de nuestro sistema. Esto incluye la creación, modificación y eliminación de cuentas de usuario, así como la asignación de permisos de acuerdo con las necesidades y responsabilidades de cada usuario.

1. Envío de notificaciones:

* Actores: Sistema y/o Administrador
* Descripción: Se implementa un sistema de notificaciones que permite informar a los usuarios sobre citas programadas, cambios en las citas planificadas o recordatorios importantes. Esta función contribuye a mejorar la comunicación entre médicos y pacientes, así como a garantizar la puntualidad y la eficiencia en la gestión de citas.

1. Cerrar sesión:

* Actores: Paciente, Médico, Administrador
* Descripción: Los usuarios tienen la opción de cerrar sesión en la aplicación para proteger la privacidad y seguridad de sus datos personales. Esta medida de seguridad garantiza que la información confidencial no esté accesible para terceros cuando no se esté utilizando activamente la aplicación.

Diagrama, Texto

Descripción generada automáticamente**DIAGRAMAS UML**

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

**Tecnologías para utilizar**

* **GitHub**: GitHub es una plataforma de desarrollo colaborativo de software que utiliza el sistema de control de versiones Git. Permite a los equipos de desarrollo trabajar de manera colaborativa en proyectos de software, gestionar el código fuente, realizar seguimiento de cambios, y facilitar la revisión y colaboración entre desarrolladores.

Es importante trabajar con Git en nuestro proyecto por su capacidad para separar las áreas de desarrollo en diferentes ramas (branches), lo que permite un desarrollo seguro y la realización de pruebas de nuevas características sin afectar el código funcional principal (branch master).

* **Trello**: Trello es una herramienta de gestión de proyectos que permite a los equipos crear tableros virtuales para planificar y organizar tareas de manera colaborativa. Utilizando metodologías ágiles como Scrum o Kanban (que en nuestro caso, será Scrum), Trello facilita la asignación de tareas, el seguimiento del progreso y la gestión de proyectos. Es importante utilizar Trello en nuestro proyecto porque nos ayuda a tener una visión clara y estructurada de las tareas pendientes, en curso y completadas.
* **Python:** Python ha sido elegido como el lenguaje de programación para el trabajar el backend de nuestro proyecto por varias razones. Primeramente, Python es conocido por su simplicidad y legibilidad de código, lo que facilita el desarrollo y mantenimiento del software. Además, cuenta con una amplia variedad de bibliotecas y frameworks que simplifican el desarrollo de aplicaciones web y su integración con bases de datos y otros sistemas. Python también es altamente escalable y flexible, lo que lo hace adecuado para proyectos de diferentes tamaños y complejidades. Adicional a eso, hemos tomado en cuenta la comunidad activa y la documentación que se encuentra, completa y amplia, porque simplifica la resolución de problemas y la actualización.

Hemos tomado en cuenta al mismo tiempo, algunas alternativas similares a Python para el desarrollo del backend de nuestro proyecto siendo estas opciones: Node.js y Ruby on Rails.

* **SQL Server o PostgreSQL:** La elección entre SQL Server y PostgreSQL para la gestión de bases de datos en nuestro proyecto se basa en ciertos criterios, tomando en cuenta que ambos DBMS son bastante utilizados. SQL Server, desarrollado por Microsoft, brinda un sistema de integración con otras herramientas de la suite de Microsoft, como Azure, lo que puede puede traer beneficios a futuro se utiliza el ecosistema de Microsoft. Por otro lado, PostgreSQL es una opción de código abierto que destaca por su fiabilidad, escalabilidad y compatibilidad con los estándares SQL.
* **React Native:** React Native es un framework de desarrollo de aplicaciones móviles que utiliza React, una biblioteca de JavaScript para crear interfaces de usuario. Utilizando un enfoque de desarrollo basado en componentes reutilizables, React Native permite a los desarrolladores crear aplicaciones móviles multiplataforma con una sola base de código. Es importante utilizar React Native en nuestro proyecto para el frontend de la aplicación debido a su capacidad para compilar código nativo para iOS y Android, lo que nos ayuda a asegurar un rendimiento óptimo del sistema y una experiencia de usuario fluida en ambas plataformas. Además, React Native ofrece una curva de aprendizaje más rápida para los desarrolladores que ya están familiarizados con JavaScript y React, lo que acelera el proceso de desarrollo y reduce los costos.

**Presupuesto del sistema**

**Bibliografía**

* Kline, K., & Jones, M. (2012). SQL Server MVP Deep Dives, Volume 2. Manning Publications.
* Riggs, R., & Rogelio, A. (2011). PostgreSQL 9 Administration Cookbook. Packt Publishing Ltd.
* Lutz, M. (2013). Learning Python. "O'Reilly Media, Inc.".
* Hettinger, R. (2017). Transforming Code into Beautiful, Idiomatic Python. PyCon US 2013.