



PRIMERA FASE PROYECTO

Diseño y Programación de Software Multiplataforma

ELABORADO POR:

Samuel Fernando Calderon Reyes CR202814

Walter Daniel Mejia Palacios MP202829

Docente:

Alexander Sigüenza

Facultad de Ingeniería / Virtual

Soyapango, septiembre 2023

Índice

Introducción	3
Diseño UX/UI -> Mock Ups	4
Lógica a utilizar	8
Diagramas UML	10
Herramientas a utilizar	12
Presupuesto del costo de la aplicación	15
Fuentes de Consulta	17

Introducción

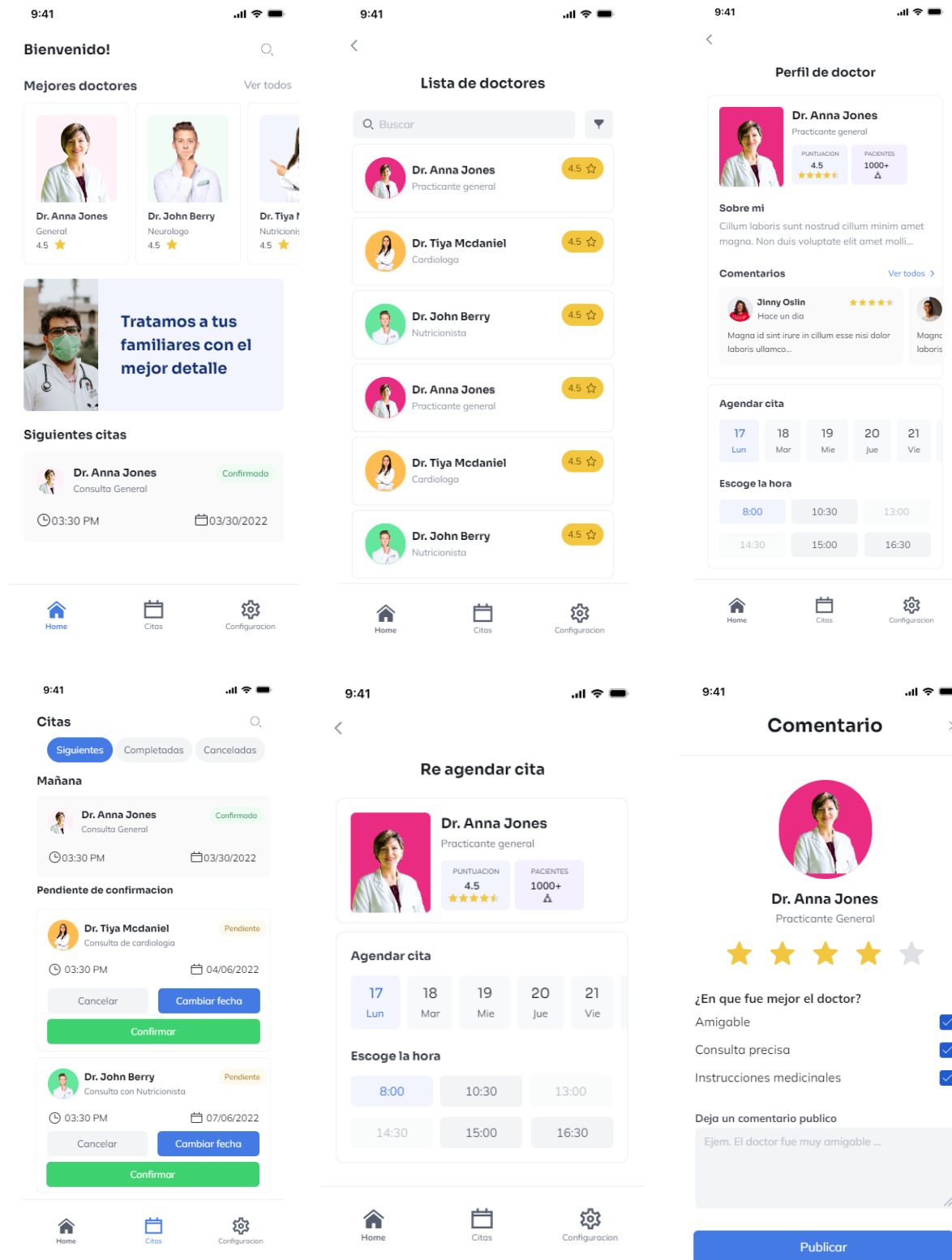
El proyecto "Asilo la Esperanza" busca una solución avanzada diseñada para modernizar la gestión de citas médicas en el ámbito profesional. Este proyecto tiene como objetivo proporcionar una plataforma digital que permita a los doctores optimizar y coordinar sus agendas de citas, potenciando la eficiencia en la atención sanitaria y garantizando una experiencia mejorada para sus pacientes.

"Asilo la Esperanza" es una aplicación móvil intuitiva y segura que facilita la creación, actualización y consulta de horarios de citas de manera ágil y eficaz. Permite a los profesionales médicos mantener un control total sobre su programación, priorizando la atención al paciente y optimizando su tiempo en el consultorio.

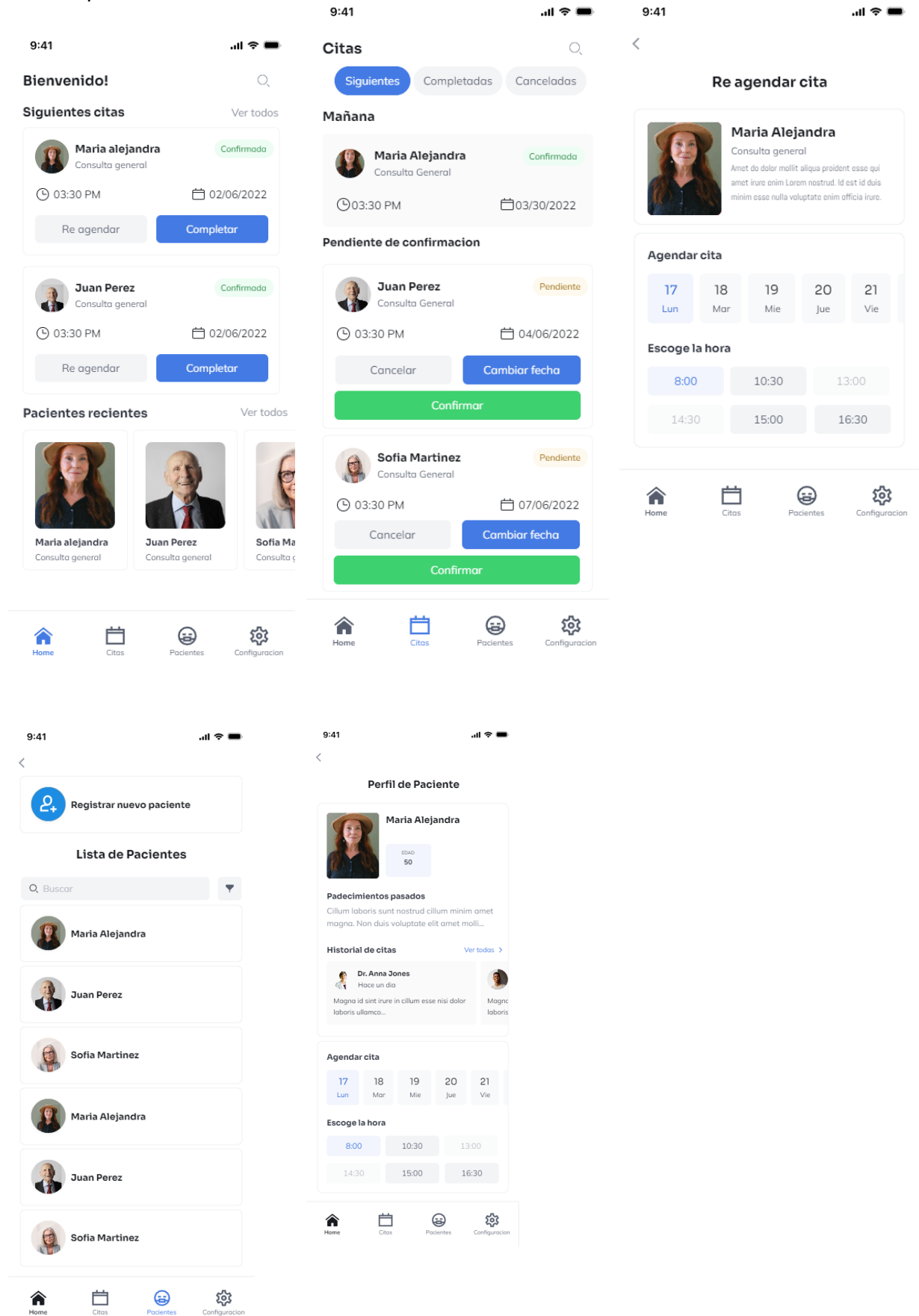
La innovación de este proyecto radica en su capacidad para adaptarse a las necesidades cambiantes del entorno médico, ofreciendo una solución tecnológica que redefine la gestión tradicional de citas. Estamos comprometidos a brindar un producto de alta calidad que mejore la eficiencia de los servicios de salud, estableciendo así un estándar elevado en la administración de citas médicas.

Diseño UX/UI -> Mock Ups

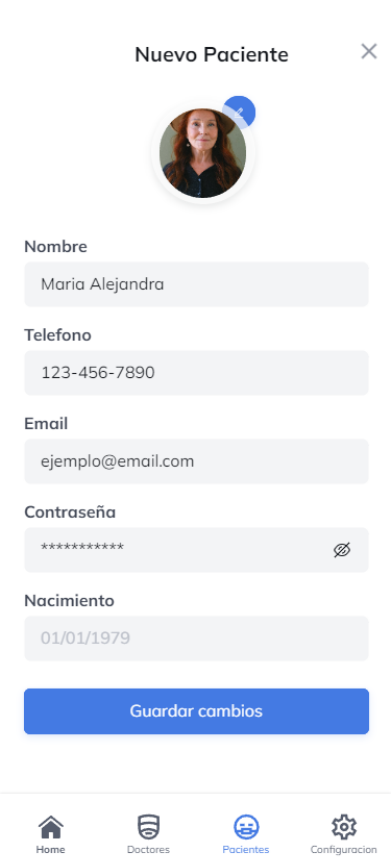
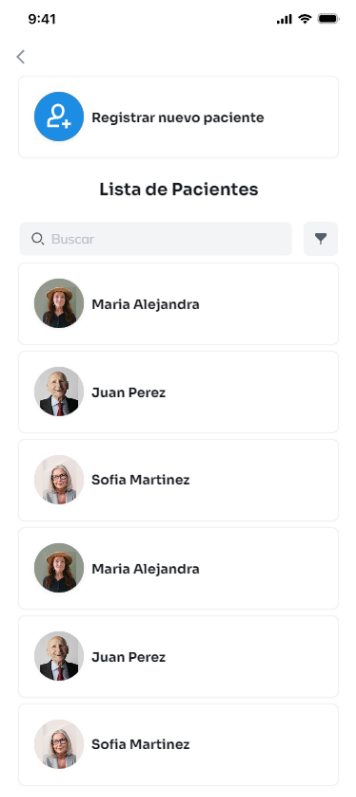
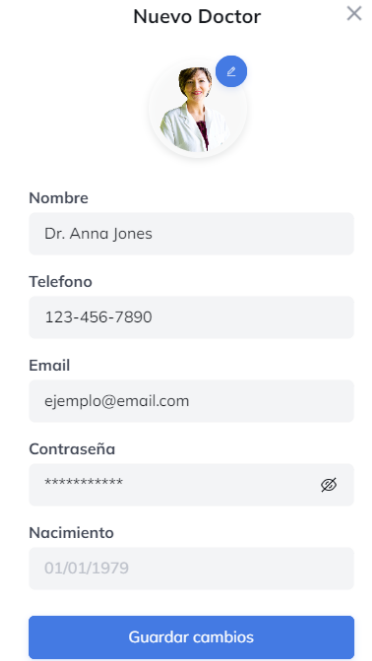
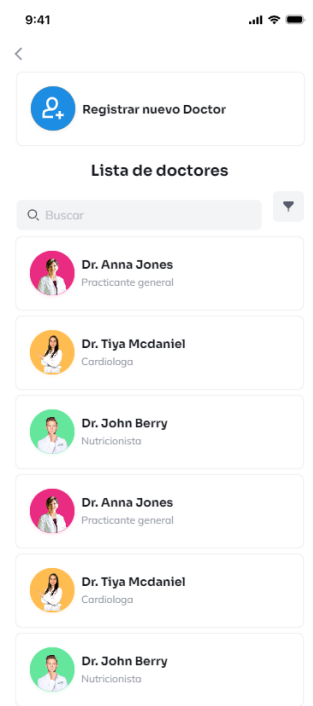
Interfaz para pacientes/familiares de pacientes



Interfaz para Doctores




Interfaz para Admin




Interfaz de configuración general


9:41





User name


example@gmail.com


 Editar perfil


 Preferencias


 Modo noche


 Centro de ayuda

 Cerrar sesion

 Home

 Citas


 Pacientes

 Configuracion

9:41

Configuracion de perfil

X



Nombre

Dr. Anna Jones


Telefono

123-456-7890

Email

ejemplo@email.com


Contraseña


*****


Nacimiento


01/01/1979

Guardar cambios

 Home

 Citas

 Pacientes

 Configuracion

Lógica a utilizar

Nuestra lógica se basará alrededor de 3 columnas principales, las cuales serán nuestros tipos de usuario:

1. Administrador: Este será el usuario que estará encargado de poder crear el perfil de nuevos doctores y nuevos pacientes, así como obviamente editar y listar a todos ellos en sus respectivas pantallas.
2. Doctor: Este usuario será el que tenga la capacidad de poder registrar nuevos pacientes, editar su información pero también tendrá la capacidad de agendar citas con los pacientes y acordar una fecha de dicha cita, podrá cancelar, editar y confirmar la cita, al igual que ver los registros de las citas que ha tenido con todos sus pacientes o con un paciente en específico.
3. Paciente: En este caso reconocemos que dado el contexto de que los pacientes serán adultos mayores dejaremos la opción a 3 situaciones:
 - a. El adulto mayor tiene capacitación tecnológica: El paciente tendrá sus credenciales de ingreso a la aplicación donde el mismo podrá solicitar citas, confirmar, editar y cancelar citas existentes, y ver el listado de Doctores para poder agendar sus citas.
 - b. El adulto mayor no tiene capacitación tecnológica pero si un familiar de apoyo: En este caso tendremos una situación en la que pasará lo mismo que con anterioridad pero todas las funciones las tendrá un encargado.
 - c. El adulto mayor no tiene capacitación tecnológica y tampoco un familiar que le apoye: En este caso tendremos una situación en la cual el adulto mayor no tendrá que realizar ninguna gestión y tampoco tendrá credenciales de inicio de sesión, únicamente será controlado y monitoreado por los doctores, quienes serán ellos los responsables de agendar las citas para llevar el control del adulto mayor en su totalidad.

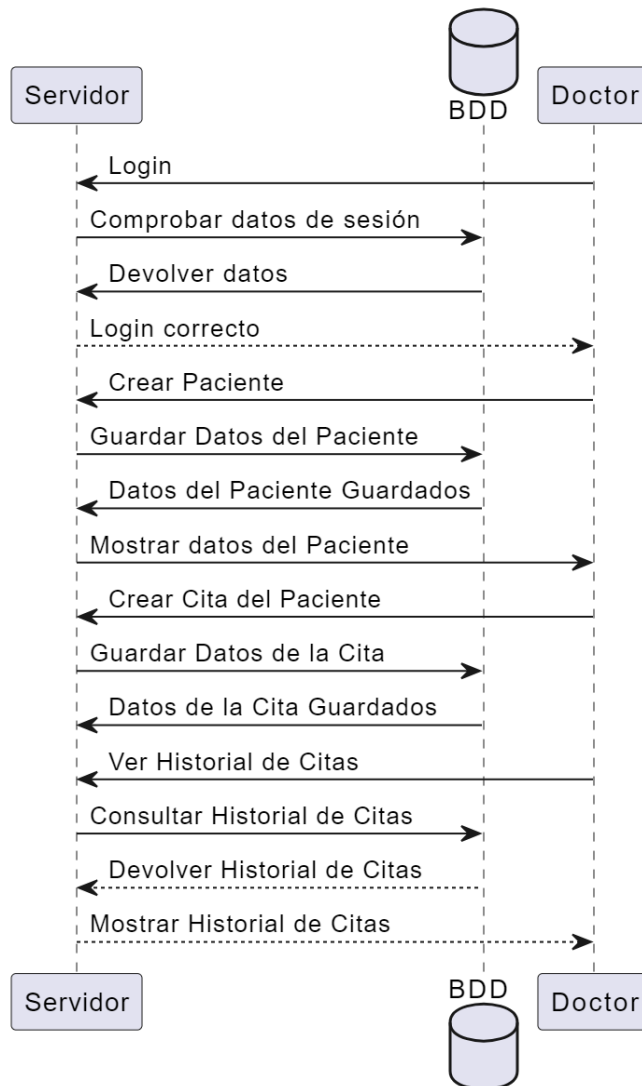
Para todos los usuarios antes mencionados se creará una plataforma de ingreso con credenciales asignadas por la misma plataforma y todos tendrán igualmente un acceso rápido con Google o sus cuentas de Gmail.

Para la lógica de las citas y la edición de las mismas planteamos un control por confirmación para poder facilitar la comprensión de todos los usuarios. Este control por confirmación comprende de la siguiente forma:

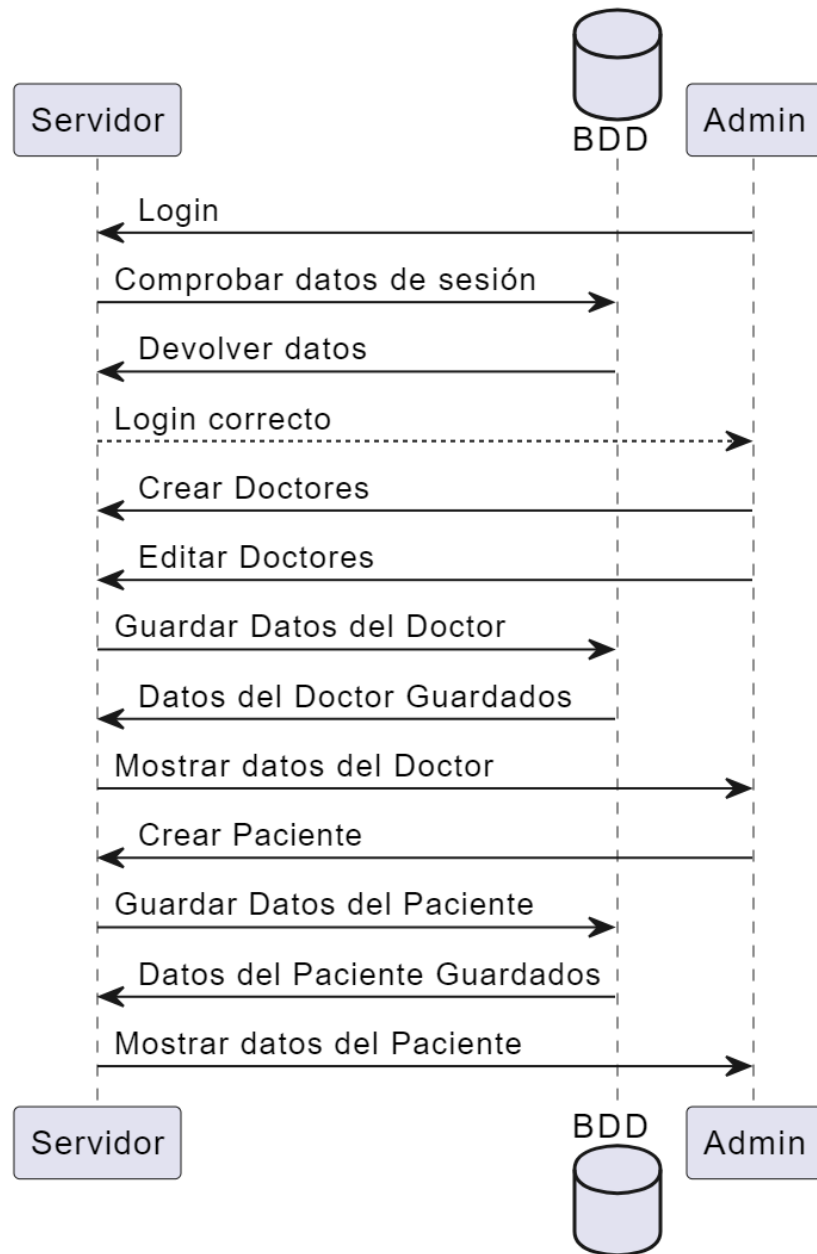
1. Inicio/Solicitud/Registro de cita: En este momento un doctor puede registrar/solicitar una cita a un paciente, o viceversa, donde un paciente puede solicitar una cita con un doctor, en este caso la cita quedará en modo de “pendiente”.
 - a. En la única situación donde no se quedará en estado “pendiente” es cuando la cita la registra un doctor y está citando a un paciente de tipo de usuario/situación “C)” (adulto mayor incapacitado tecnológicamente y sin familiar de apoyo), en esta situación la cita será automáticamente estado de “confirmada”.
2. Cita en estado “Pendiente”: En este momento el usuario recibirá una notificación que tiene una nueva solicitud de cita junto con la hora y fecha de la misma, este usuario tendrá la capacidad de poder cancelarla, agendarla y confirmar.
3. Cita en estado de “Confirmada”: esto significa que la fecha y hora fue corroborada por ambos usuarios y se guardará como una fecha importante en las listas de calendarios de ambos usuarios, al igual que se les notificara anterior a la misma de la llegada próxima a atender la cita.
4. Cita “cancelada”: En este caso se entiende que uno de los usuarios no estaba de acuerdo con dicha cita y no quería atender a ella y por ello entra al estado de cancelada el cual, quedará en registro que se inició la solicitud de la cita pero por una razón en concreto esta cita no se efectuó.
5. Cita “reprogramada”: En esta situación lo que vemos es que dado un imprevisto en la fecha que se había propuesto, y por ello se genera un cambio en la fecha, dicho cambio hace que el otro usuario sea notificado al respecto y pueda repetir el mismo proceso desde el estado de pendiente, este proceso puede repetirse cuantas veces sea necesario hasta llegar a una fecha en que ambos usuarios estén de acuerdo y ambos confirman la cita.

Diagramas UML

Utilizando el actor Doctor, quién será el encargado de Registrar Pacientes, Registrar Citas y Consultar el Historial de Citas.



Desde la perspectiva del Administrador, quién tendrá acceso a la mayor parte de los registros, y podrá tanto crear y actualizar valores de Doctores y Pacientes.



Herramientas a utilizar

Para el desarrollo exitoso del proyecto "Asilo la Esperanza", se utilizarán diversas herramientas y tecnologías especializadas tanto en el frontend (aplicación móvil) como en el backend (servidor y base de datos). Aquí se presentan las herramientas clave a utilizar en el frontend, backend y base de datos en cada etapa del proyecto:

Frontend (Aplicación Móvil - React Native):

- React Native:

Plataforma de desarrollo para crear aplicaciones móviles multiplataforma, permitiendo la construcción de interfaces de usuario rápidas y eficientes.

- JavaScript y ES6+:

Lenguaje de programación fundamental para el desarrollo en React Native, con especial énfasis en las características modernas de ES6+ para escribir código más limpio y eficiente.

- Redux:

Biblioteca para el manejo del estado de la aplicación, permitiendo una gestión centralizada de los datos y un flujo de información predecible.

- React Navigation:

Librería para gestionar la navegación en la aplicación, facilitando la creación de rutas y pantallas.

- Axios:

Ciente HTTP basado en promesas para realizar peticiones a la API del backend.

Backend (Servidor - Express.js):

- Node.js:

Entorno de ejecución de JavaScript del lado del servidor que permite la construcción de aplicaciones escalables y de alto rendimiento.

- Express.js:

Framework de aplicación web de Node.js utilizado para construir y administrar la API del servidor.

- MySQL:

Sistema de gestión de bases de datos relacional utilizado para almacenar y gestionar la información relacionada con pacientes, doctores y citas.

- JSON Web Tokens (JWT):

Estándar abierto para la creación de tokens de acceso, utilizado para autenticar a los usuarios y garantizar la seguridad de las comunicaciones.

IDEs:

Para el proyecto se utilizarán varios entornos de desarrollo adecuados para el desarrollo tanto del frontend (aplicación móvil) como del backend (servidor). Dentro de estos usaremos:

- Visual Studio Code (VSCode):

Un entorno de desarrollo de código abierto ampliamente utilizado, con soporte para React Native y una amplia gama de extensiones que facilitan el desarrollo y la depuración.

- WebStorm:

Un poderoso entorno de desarrollo de JetBrains, con excelentes capacidades para el desarrollo de aplicaciones JavaScript y React Native, incluyendo características avanzadas de depuración y análisis de código.

- Android Studio y Android Emulator (para desarrollo en Android):

Herramienta oficial para el desarrollo de aplicaciones Android y un emulador de Android para probar la aplicación en un entorno controlado.

- Postman:

Herramienta para probar y documentar APIs, fundamental para verificar la funcionalidad y el rendimiento de la API backend.

- HeidiSQL:

Herramienta de gestión de bases de datos MySQL de código abierto, que ofrece una interfaz gráfica amigable para administrar y consultar la base de datos MySQL.

Presupuesto del costo de la aplicación

Para calcular el presupuesto del costo de la aplicación, se deben considerar varios factores, como el tiempo de desarrollo, el costo de los programadores, el costo de las herramientas y tecnologías utilizadas, entre otros. Podemos iniciar dividiendo el trabajo en etapas de desarrollo para hacernos una idea del costo que puede tener la aplicación.

Etapas del desarrollo:

1. Análisis y Diseño:

- a. Definición de requisitos y especificaciones: 10 horas
- b. Diseño de la arquitectura: 20 horas
- c. Diseño de la interfaz de usuario: 30 horas
- d. Total: 60 horas**

2. Desarrollo:

- a. Configuración del entorno de desarrollo: 20 horas
- b. Desarrollo del frontend (React Native): 250 horas por programador (500 horas en total)
- c. Desarrollo del backend (Express, integración con MySQL): 100 horas por programador (200 horas en total)
- d. Integración de acceso por usuario y contraseña, y acceso por Gmail: 40 horas
- e. Total: 740 horas**

3. Pruebas y Optimización:

- a. Pruebas de funcionalidad y rendimiento: 60 horas
- b. Optimización de rendimiento: 30 horas
- c. Correcciones y ajustes según pruebas: 30 horas
- d. Total: 120 horas**

4. Despliegue y Entrega:

- a. Configuración de servidores y despliegue: 20 horas
- b. Documentación y entrega: 20 horas
- c. **Total: 40 horas**

Total de horas estimadas: 960 horas

Ahora, tomando en cuenta que son dos programadores full stack que cobran \$2000 al mes cada uno y el proyecto se desarrollará en 3 meses, podemos calcular el costo total del proyecto:

- Costo por hora por programador:

\$2000 al mes / 160 horas al mes \approx \$12.5 por hora

Costo por hora por programador: \$12.5 por hora

- Costo total del desarrollo:

Total de horas estimadas: 960 horas

Costo por hora por programador: \$12.5

Horas que realizará cada programador: **480 horas**

Costo total del desarrollo: (480 horas * \$12.5) * 2 programadores = **\$12,000**

El costo final del desarrollo sería de **\$12,000**.

Fuentes de Consulta

- Setting up the development environment · React native. (2023, 29 agosto). <https://reactnative.dev/docs/environment-setup>
- *Express 4.X - referencia de API*. (s. f.). <https://expressjs.com/es/4x/api.html>
- Documentación. (s. f.). Guía para publicar herramientas digitales. <https://el-bid.github.io/guia-de-publicacion/documents/documentacion/>
- Dieker, N. (2023, 27 septiembre). Cuánto le cuesta a un desarrollador de aplicaciones crear una aplicación. Cisin. <https://www.cisin.com/coffee-break/es/technology/cost-of-app-developer-to-make-an-app.html>