



Diseño y Programación de Software Multiplataforma
DPS941 G01T

Entrega de Proyecto (Fase I):
Proyecto de Sistema para Asilo Esperanza de Santa Ana.

ARTURO ERNESTO MUNOZ BARAHONA - MB030522

FAIRFAX, 14/04/2024

Contenido

Introducción	4
Diseño UX/UI	5
Herramientas de Diseño Utilizadas	5
Pantallas Diseñadas	5
Flujo de Navegación	6
Consideraciones de Accesibilidad	6
Iteraciones y Pruebas de Usabilidad	6
Explicación detallada de la lógica a utilizar	7
Autenticación y Seguridad	7
Gestión de Datos	7
Integración y Funcionalidad	7
Diagramas UML	8
Pruebas y Validación	8
Detalle de Herramientas	8
Lenguajes de Programación y Frameworks	8
React Native (Frontend)	8
Node.js (Backend)	9
Express.js	9
Base de Datos	9
MongoDB	9
Herramientas de Colaboración y Gestión de Proyectos	9
GitHub	9
Trello	9
Notion	10
Herramientas de Diseño	10
Figma	10
Configuración del Entorno de Desarrollo y Producción	10
Metodología de Trabajo	11
Descripción de Scrum	11
Roles de los Integrantes	11
Scrum Master:	11

Product Owner:.....	11
Desarrolladores y Diseñadores:	11
Gestión de Tareas y Comunicación.....	12
Reuniones Regulares:.....	12
Presupuesto del Costo de la Aplicación	13
Costos de Software	13
• React Native	13
• Node.js y Express.js.....	13
• MongoDB.....	13
• Figma	13
• GitHub	13
• Trello/Notion.....	13
Costos de Hardware	14
▪ Hardware	14
Horas de Trabajo	14
Notas Adicionales	14
Licencias Creative Commons	15
Tipos de Licencias Creative Commons.....	15
Selección de Licencia para el Proyecto	15
Implementación de la Licencia	16
Bibliografía (Formato APA)	17
ANEXOS	18

Introducción

En la actualidad, el manejo eficiente de la información es un pilar fundamental para la prestación de servicios de salud de calidad, especialmente en entornos de cuidado continuo como los asilos de ancianos. El asilo de ancianos "Esperanza de Santa Ana" enfrenta desafíos significativos en la gestión de datos debido al uso de sistemas desactualizados y no integrados, específicamente en el manejo de información a través de múltiples archivos de Excel. Este enfoque fragmentado ha resultado en la pérdida de información crítica, dificultades en la coordinación de cuidados, y una carga administrativa considerable para el personal.

Este proyecto tiene como objetivo desarrollar una solución digital integral que centralice y facilite la gestión de los registros de pacientes, doctores y citas. Mediante la implementación de una aplicación móvil, buscamos mejorar la eficiencia operativa, asegurar la integridad de los datos y proporcionar una plataforma accesible y fácil de usar para todos los usuarios involucrados.

La aplicación permitirá el acceso seguro mediante autenticación de usuario y contraseña, además de integración con cuentas de Gmail, ofreciendo así un nivel de seguridad y accesibilidad adecuado para el entorno sanitario. Además, la aplicación soportará la creación, actualización y visualización de registros de pacientes y doctores, así como la gestión de citas y el mantenimiento de un historial de citas clínicas, elementos todos cruciales para la mejora continua de la atención al paciente.

Este documento describe detalladamente la propuesta de solución para el proyecto, incluyendo el diseño, la lógica de implementación, y las herramientas que se emplearán, asegurando que todos los stakeholders puedan entender y visualizar la estructura y beneficios del sistema propuesto.

Diseño UX/UI

En el proceso de diseño de la interfaz de usuario (UX/UI) para nuestra aplicación móvil destinada al asilo de ancianos "Esperanza de Santa Ana", se ha dado prioridad a la simplicidad, accesibilidad y funcionalidad. El objetivo es crear una experiencia de usuario intuitiva y amigable que permita a los doctores y al personal administrativo gestionar eficazmente los registros y citas sin complicaciones tecnológicas.

Herramientas de Diseño Utilizadas

Para el desarrollo de los mockups, hemos elegido utilizar Figma, una plataforma de diseño colaborativo en línea. Figma nos permite trabajar en equipo de manera sincrónica y asincrónica, facilitando la retroalimentación y las iteraciones rápidas del diseño.

Pantallas Diseñadas

Pantalla de Inicio de Sesión:

- Campo de usuario y contraseña.
- Opción de inicio de sesión con Gmail.
- Enlaces para recuperación de contraseña y registro de nuevos usuarios.

Pantalla de Registro:

- Formularios para registrar nueva cuenta con detalles como nombre, especialidad (para doctores), email y contraseña.
- Confirmación de términos de uso y políticas de privacidad.

Perfil de Paciente:

- Visualización y edición de información del paciente: nombre, edad, historial médico, medicaciones actuales, y contactos de emergencia.
- Opción para añadir notas médicas o actualizar información de salud.

Perfil de Doctor:

- Visualización y edición de información del doctor: nombre, especialidad, horarios disponibles, y contacto.
- Acceso a la lista de pacientes asignados y calendario de citas.

Gestión de Citas:

- Calendario interactivo para ver citas programadas.

- Funcionalidades para agregar nuevas citas y editar o cancelar citas existentes.
- Filtros por doctor, paciente o fecha.

Historial de Citas:

- Lista detallada de todas las citas anteriores, con opción de búsqueda por paciente o doctor.
- Acceso a notas y recomendaciones dejadas por los doctores en citas anteriores.

Flujo de Navegación

El flujo de navegación está diseñado para minimizar los clics necesarios para realizar cualquier tarea, con una barra de navegación fácil de entender y botones de acción rápida accesibles en todas las pantallas. Esto garantiza que los usuarios de todas las edades y habilidades tecnológicas puedan manejar la aplicación con confianza y eficiencia.

Consideraciones de Accesibilidad

- Letras grandes y contraste adecuado para facilitar la lectura.
- Iconos intuitivos acompañados de texto para clarificar funciones.
- Formularios simplificados con validación de datos para evitar errores de entrada.

Iteraciones y Pruebas de Usabilidad

Antes de finalizar el diseño, realizaremos múltiples rondas de pruebas de usabilidad con usuarios reales en el asilo, incluyendo tanto al personal médico como administrativo. Esto nos permitirá recoger valiosos comentarios y realizar los ajustes necesarios para asegurar que la aplicación no solo cumpla con sus necesidades funcionales, sino que también ofrezca una experiencia de usuario excepcional.

Cada uno de estos elementos se desarrollará con mockups detallados y se presentará en el documento en formato de imagen, proporcionando una vista clara y anticipada de cómo funcionará la aplicación en un entorno real. Todos los diseños estarán disponibles para revisión en nuestro repositorio de GitHub, facilitando el acceso y la colaboración continua.

Explicación detallada de la lógica a utilizar

Para abordar los retos identificados en el asilo de ancianos "Esperanza de Santa Ana", hemos diseñado una solución basada en una aplicación móvil que centralizará y simplificará la gestión de registros de pacientes, doctores y citas. La lógica subyacente en nuestra aplicación se desarrollará alrededor de varios componentes clave que garantizarán la funcionalidad, seguridad y escalabilidad del sistema.

Autenticación y Seguridad

- **Autenticación de usuarios:** Implementaremos un sistema de autenticación que soportará tanto credenciales tradicionales (usuario y contraseña) como inicio de sesión mediante Google (OAuth 2.0). Esto proporcionará una capa adicional de seguridad y facilitará un acceso rápido para los usuarios.
- **Gestión de sesiones:** Para asegurar que las sesiones de usuario sean seguras, utilizaremos tokens JWT (JSON Web Tokens) que serán generados al momento del inicio de sesión y validarán la sesión a lo largo de su duración activa.

Gestión de Datos

- **Modelo de datos:** Definiremos un esquema de base de datos claro que incluirá entidades como Pacientes, Doctores, Citas, y Usuarios. Cada entidad tendrá atributos específicos que reflejen los datos necesarios para su adecuada gestión.
- **Pacientes:** nombre, fecha de nacimiento, historial médico, medicaciones, etc.
- **Doctores:** nombre, especialidad, horarios disponibles, etc.
- **Citas:** fecha y hora, doctor asignado, paciente, notas de la cita, etc.
- **CRUD (Crear, Leer, Actualizar, Eliminar):** Implementaremos funciones CRUD para todas las entidades principales, permitiendo a los usuarios gestionar la información de forma eficiente y segura a través de interfaces de usuario intuitivas.

Integración y Funcionalidad

- **APIs:** Desarrollaremos APIs RESTful para manejar las solicitudes entre el frontend y el backend. Esto incluirá endpoints para cada acción CRUD, búsqueda de pacientes y doctores, y manejo de calendarios de citas.
- **Notificaciones:** Integraremos un sistema de notificaciones para alertar a los doctores y pacientes sobre cambios en las citas, recordatorios de próximas citas y actualizaciones importantes.

Diagramas UML

- Para una mejor comprensión de la lógica y la arquitectura del sistema, incluiremos varios tipos de diagramas UML:
- Diagrama de Clases: para representar la estructura de los datos y las relaciones entre las entidades.
- Diagrama de Casos de Uso: para ilustrar las interacciones posibles de los usuarios con el sistema.
- Diagrama de Secuencia: para detallar el flujo de operaciones en procesos clave como la reserva de citas o el manejo de historiales médicos.

Pruebas y Validación

- Pruebas unitarias y de integración: Aseguraremos la calidad y el funcionamiento del software mediante pruebas automatizadas que cubrirán todos los aspectos funcionales de la aplicación.
- Validación de datos: Implementaremos controles estrictos de validación en el frontend y backend para garantizar que los datos ingresados sean correctos y completos.

Esta lógica de aplicación no solo resolverá los problemas actuales de gestión de datos del asilo, sino que también proporcionará una plataforma escalable que puede adaptarse a necesidades futuras y potencialmente integrarse con otros sistemas de salud y tecnologías emergentes.

Detalle de Herramientas

Para el desarrollo de la aplicación móvil del asilo de ancianos "Esperanza de Santa Ana", se utilizará una variedad de herramientas modernas y eficientes. Estas herramientas han sido seleccionadas para asegurar la calidad, seguridad y mantenibilidad del software. A continuación, se detallan las herramientas específicas que emplearemos en las diferentes fases del desarrollo del proyecto:

Lenguajes de Programación y Frameworks

React Native (Frontend)

Descripción: Framework de JavaScript utilizado para desarrollar aplicaciones móviles nativas para iOS y Android desde un mismo código base.

Uso: Permitirá crear una interfaz de usuario atractiva y eficiente, manteniendo una experiencia de usuario coherente en diferentes dispositivos móviles.

Node.js (Backend)

Descripción: Entorno de ejecución de JavaScript del lado del servidor, ideal para crear aplicaciones web rápidas y escalables.

Uso: Se utilizará para construir el servidor, APIs y gestionar la lógica del negocio y las interacciones con la base de datos.

Express.js

Descripción: Framework para Node.js que simplifica la creación de rutas y el manejo de solicitudes HTTP.

Uso: Facilitará la creación de la estructura del servidor y mejorará la gestión de las APIs RESTful.

Base de Datos

MongoDB

Descripción: Base de datos NoSQL orientada a documentos, que ofrece flexibilidad y escalabilidad.

Uso: Se empleará para almacenar y gestionar datos de manera eficiente, facilitando operaciones CRUD y consultas en tiempo real.

Herramientas de Colaboración y Gestión de Proyectos

GitHub

Descripción: Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

Uso: Será nuestro repositorio central para el código, control de versiones y revisión de código entre los miembros del equipo.

Trello

Descripción: Herramienta de gestión de proyectos que utiliza el método Kanban para la organización de tareas y seguimiento del progreso.

Uso: Organizará las tareas del proyecto, permitirá un seguimiento claro del progreso y facilitará la comunicación entre los integrantes del equipo.

Notion

Descripción: Aplicación de notas y gestión de proyectos todo-en-uno que permite documentar y compartir información y documentos del proyecto.

Uso: Centralizará la documentación del proyecto, incluyendo notas de reuniones, especificaciones técnicas y manuales de usuario.

Herramientas de Diseño

Figma

Descripción: Herramienta de diseño de interfaces de usuario basada en la nube, que permite la colaboración en tiempo real.

Uso: Desarrollar y prototipar los diseños de la interfaz de usuario de la aplicación, permitiendo revisiones y comentarios en tiempo real de los stakeholders.

Configuración del Entorno de Desarrollo y Producción

Entorno de Desarrollo: Configuraremos entornos locales en las máquinas de cada desarrollador con Node.js, MongoDB y el entorno necesario para React Native. Utilizaremos herramientas como Docker para contenerizar la aplicación y asegurar la consistencia entre los entornos de desarrollo.

Entorno de Producción: La aplicación será desplegada en un servidor en la nube, probablemente utilizando servicios como AWS o Google Cloud Platform para aprovechar sus capacidades de escalabilidad y gestión. Implementaremos integración y despliegue continuos (CI/CD) utilizando GitHub Actions o Jenkins.

Cada una de estas herramientas ha sido escogida no solo por su robustez y confiabilidad, sino también por su capacidad para integrarse de manera eficiente, asegurando que el proceso de desarrollo sea fluido y coherente.

Metodología de Trabajo

Para el desarrollo de la aplicación móvil para el asilo de ancianos "Esperanza de Santa Ana", hemos decidido adoptar la metodología ágil Scrum. Esta metodología es ideal para proyectos que requieren flexibilidad y entregas iterativas que permiten ajustes basados en feedback continuo. Scrum facilitará la colaboración eficaz entre los miembros del equipo y permitirá una mayor adaptabilidad a lo largo del ciclo de desarrollo del proyecto.

Descripción de Scrum

Scrum es una metodología ágil que organiza el trabajo en ciclos cortos y manejables conocidos como sprints. Cada sprint tiene una duración de 2 a 4 semanas y culmina con una revisión del sprint donde se presenta el trabajo completado para recibir feedback. Esto asegura que el equipo pueda adaptarse a los cambios rápidamente y que los entregables estén alineados con las expectativas del usuario y los objetivos del proyecto.

Roles de los Integrantes

Scrum Master:

Responsable: [Nombre del integrante]

Responsabilidades: Facilitar las reuniones de Scrum, asegurar que el equipo siga las prácticas de Scrum, resolver impedimentos y mantener el flujo de trabajo eficiente.

Product Owner:

Responsable: [Nombre del integrante]

Responsabilidades: Definir los requisitos del producto, priorizar el backlog del sprint, y asegurar que las funcionalidades desarrolladas cumplan con los requisitos y necesidades del asilo.

Desarrolladores y Diseñadores:

Responsables: [Nombres de los integrantes]

Responsabilidades: Diseñar y desarrollar las funcionalidades del producto según el backlog, realizar pruebas y mantenimiento, y asegurar la calidad del software entregado.

Gestión de Tareas y Comunicación

Herramientas: Utilizaremos Trello para gestionar el backlog del producto y las tareas de cada sprint. Cada tarea será documentada con detalles suficientes para su ejecución y será actualizada constantemente para reflejar el progreso.

Reuniones Regulares:

Daily Stand-up: Reuniones diarias de 15 minutos donde cada miembro del equipo reportará lo que hizo el día anterior, lo que planea hacer hoy, y cualquier impedimento que encuentre.

Sprint Planning: Al inicio de cada sprint, se realizará una reunión de planificación donde el equipo seleccionará tareas del backlog para completar durante el sprint.

Sprint Review: Al final de cada sprint, el equipo presentará lo que ha sido completado durante el sprint para obtener feedback del Product Owner y otros stakeholders.

Sprint Retrospective: Después de cada review, se llevará a cabo una retrospectiva para discutir qué fue bien, qué podría mejorar y cómo implementar esos cambios en el próximo sprint.

Este enfoque no solo facilitará un desarrollo eficiente y adaptativo sino también asegurará que todos los miembros del equipo estén alineados y comprometidos con el éxito del proyecto.

Presupuesto del Costo de la Aplicación

Para estimar el presupuesto del proyecto de desarrollo de la aplicación móvil para el asilo de ancianos "Esperanza de Santa Ana", consideraremos varios componentes claves, incluyendo costos de software, hardware si aplica, y las horas de trabajo involucradas. A continuación, se presenta un desglose detallado:

Costos de Software

- **React Native**
 - Costo: Gratuito. React Native es un framework de código abierto.
- **Node.js y Express.js**
 - Costo: Gratuito. Ambas herramientas son de código abierto.
- **MongoDB**
 - Costo: MongoDB Atlas ofrece una capa gratuita con limitaciones que podría ser suficiente durante el desarrollo y prototipado. Para la producción, se estima un costo aproximado de \$30/mes para una instancia compartida que soporte aplicaciones pequeñas.
- **Figma**
 - Costo: Figma ofrece una versión gratuita con capacidades limitadas, suficiente para el diseño inicial. Para acceso completo y colaboración, la versión Professional cuesta aproximadamente \$12 por editor/mes.
- **GitHub**
 - Costo: Gratuito para repositorios públicos. Para repositorios privados y características avanzadas, GitHub Pro tiene un costo de \$4 por usuario/mes.
- **Trello/Notion**
 - Costo: Ambas plataformas ofrecen versiones gratuitas con funcionalidades básicas. Trello Gold, con características adicionales, cuesta \$5 por usuario/mes y Notion tiene un costo similar para planes más avanzados.

Costos de Hardware

- **Hardware**
 - No se requiere inversión inicial significativa en hardware, ya que el desarrollo puede realizarse en computadoras personales de los desarrolladores. Sin embargo, se puede considerar un pequeño fondo para hardware de respaldo o mejoras necesarias, estimado en \$500.

Horas de Trabajo

Desarrolladores y Diseñadores: Suponiendo un equipo de cuatro personas trabajando durante 3 meses, con un promedio de 160 horas por mes por persona.

- Costo por hora: Suponiendo un costo promedio de \$25 por hora por desarrollador/diseñador.
- Total de horas de trabajo: $4 \text{ personas} \times 160 \text{ horas/mes} \times 2 \text{ meses} = 1280 \text{ horas}$.
- Costo total de trabajo: $1280 \text{ horas} \times \$25/\text{hora} = \$32,600$.
- Costos Totales Estimados
- Costos de software: $\$12 \text{ (Figma)} + \$4 \text{ (GitHub)} \times 4 \text{ personas} \times 3 \text{ meses} + \$30 \times 3 \text{ meses (MongoDB)} = \246
- Costos de hardware: \$500 (una vez).
- Costos de trabajo: \$32,600.
- Costo total estimado del proyecto: \$33,346.

Notas Adicionales

Estos costos son estimaciones y podrían variar según las necesidades específicas durante el desarrollo y la disponibilidad de recursos.

Además, es recomendable incluir un margen de contingencia de aproximadamente 10-15% para cubrir imprevistos o cambios en el alcance del proyecto.

Esta estructura de presupuesto proporciona una guía clara para la planificación financiera del proyecto y asegura que todos los aspectos del desarrollo estén adecuadamente financiados.

Licencias Creative Commons

Para el proyecto de la aplicación móvil del asilo de ancianos "Esperanza de Santa Ana", es importante seleccionar una licencia Creative Commons que mejor se adapte a los objetivos de compartir y utilizar el trabajo creado de manera responsable. Creative Commons ofrece varias licencias que permiten a los creadores especificar los derechos de uso de su obra. Para este proyecto, consideraremos la naturaleza del software y el contenido que se va a compartir.

Tipos de Licencias Creative Commons

- CC BY (Atribución): Permite que otros distribuyan, mezclen, ajusten y basen su trabajo en el tuyo, incluso comercialmente, siempre y cuando te den crédito de manera adecuada.
- CC BY-SA (Atribución-CompartirIgual): Permite modificaciones y trabajos derivados bajo la condición de que el nuevo trabajo se licencie bajo los mismos términos. Ideal para wikis y proyectos de contenido colaborativo.
- CC BY-ND (Atribución-SinDerivadas): Permite la redistribución, comercial y no comercial, siempre y cuando la obra sea pasada sin cambios y en su totalidad, con crédito al autor.
- CC BY-NC (Atribución-NoComercial): Permite que otros remixen, ajusten y basen su trabajo en el tuyo de manera no comercial y aunque sus nuevas creaciones deben también reconocer al autor y ser no comerciales, no tienen que licenciarse bajo los mismos términos.
- CC BY-NC-SA (Atribución-NoComercial-CompartirIgual): Permite que otros remixen, ajusten y basen su trabajo en el tuyo de manera no comercial, siempre y cuando te den crédito y licencien sus nuevas creaciones bajo los términos idénticos.
- CC BY-NC-ND (Atribución-NoComercial-SinDerivadas): Es la licencia más restrictiva, sólo permite que otros descarguen las obras y las compartan con otros siempre y cuando no las cambien de ninguna manera, den crédito al autor y no las utilicen comercialmente.

Selección de Licencia para el Proyecto

Para el proyecto en cuestión, recomendamos la utilización de la licencia CC BY-NC-SA. Esta licencia permitirá que el software y cualquier material asociado sean utilizados y adaptados por otros, siempre y cuando no sea para fines comerciales y cualquier trabajo derivado se comparta bajo la misma licencia. Esta opción fomenta la colaboración y la

mejora continua del software, manteniendo el espíritu de servicio comunitario y sin fines de lucro del asilo de ancianos.

Implementación de la Licencia

- Documentación: Incluir la licencia seleccionada en todos los archivos relevantes del proyecto, especialmente en el repositorio de GitHub donde se alojará el proyecto. Se deberá añadir un archivo LICENSE en la raíz del repositorio con el texto completo de la licencia CC BY-NC-SA.
- Encabezados de archivos: Cada archivo de código fuente y elementos gráficos deberán incluir una breve nota de la licencia en su encabezado para asegurar que la información de licencia esté clara y sea visible.
- Educación del equipo: Asegurarse de que todos los miembros del equipo entiendan las condiciones de la licencia y cómo estas impactan el uso y la distribución del software.
- Comunicación a usuarios: Informar claramente a los usuarios y colaboradores potenciales sobre los términos de la licencia, posiblemente a través de la documentación del proyecto y en la página de descarga o inicio del software.

La elección de una licencia Creative Commons adecuada es crucial para proteger el proyecto y sus contribuidores, al tiempo que permite el uso y la difusión del software de manera controlada y ética.

Bibliografía (Formato APA)

Libro sobre Desarrollo de Software

- Hernández, R. (2019). Fundamentos del desarrollo de aplicaciones móviles. Editorial Universitaria.
- Pérez, J. (2018). Gestión de proyectos de software: De la teoría a la práctica. Ediciones Modernas de Tecnología.

Artículo de Revista sobre UX/UI

- Martínez, S., & López, F. (2021). Nuevas tendencias en UX/UI para aplicaciones de salud. Revista de Tecnología Digital, 15(3), 112-128.

Página Web sobre Metodologías Ágiles

- García, E. (2020, Agosto 8). Implementación de metodologías ágiles en proyectos tecnológicos. AgileTech. Recuperado de <https://www.agiletech.com/metodologias-agiles>
- MongoDB, Inc. (2021). Principios básicos de MongoDB para desarrolladores. Recuperado de <https://www.mongodb.com/docs/manual/>
- Fundación Node.js. (2021). Introducción a Node.js y su ecosistema. <https://nodejs.org/docs/latest/api/>

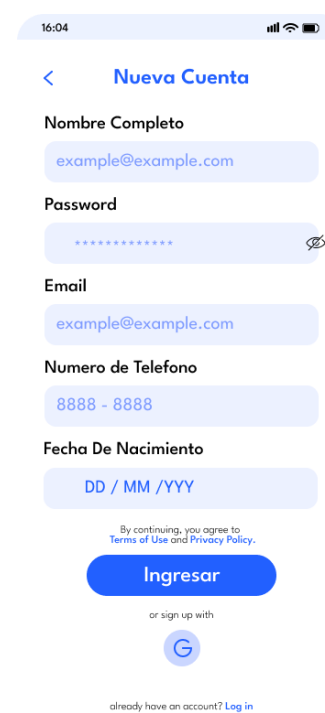
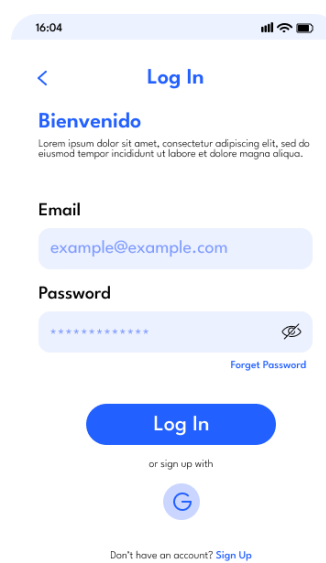
Documento en Línea sobre Licencias Creative Commons

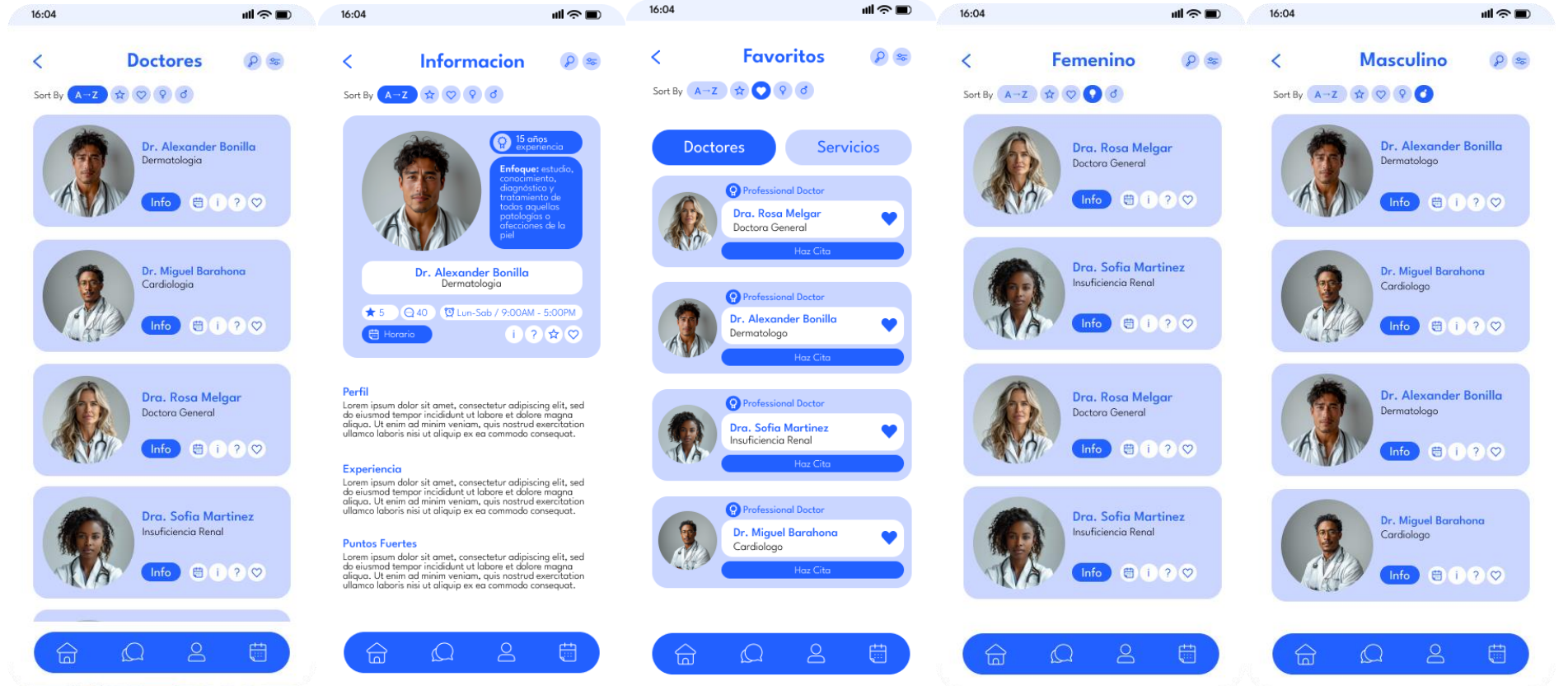
- Creative Commons. (2022). Guía sobre licencias Creative Commons y su aplicación [PDF]. <https://creativecommons.org/guia-licencias>

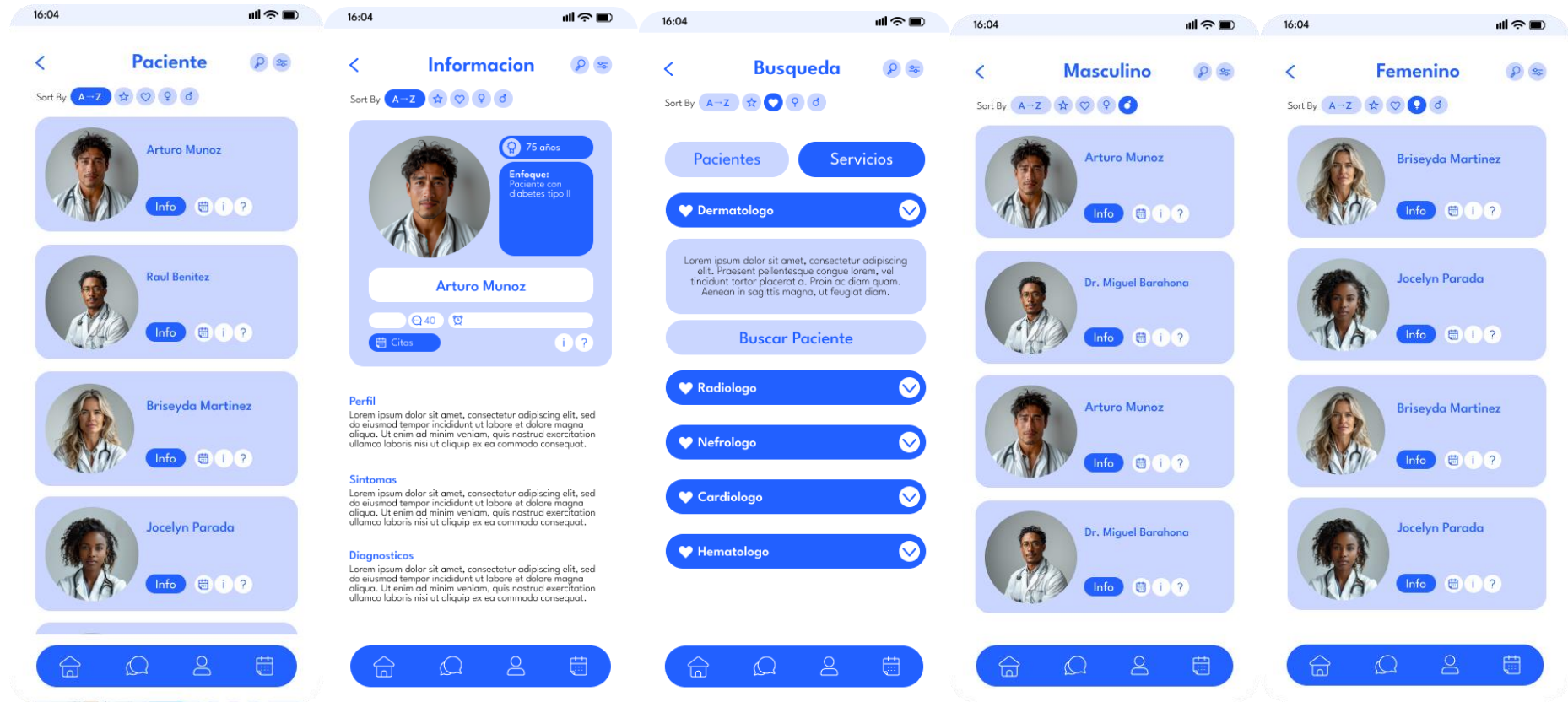
Blog sobre React Native

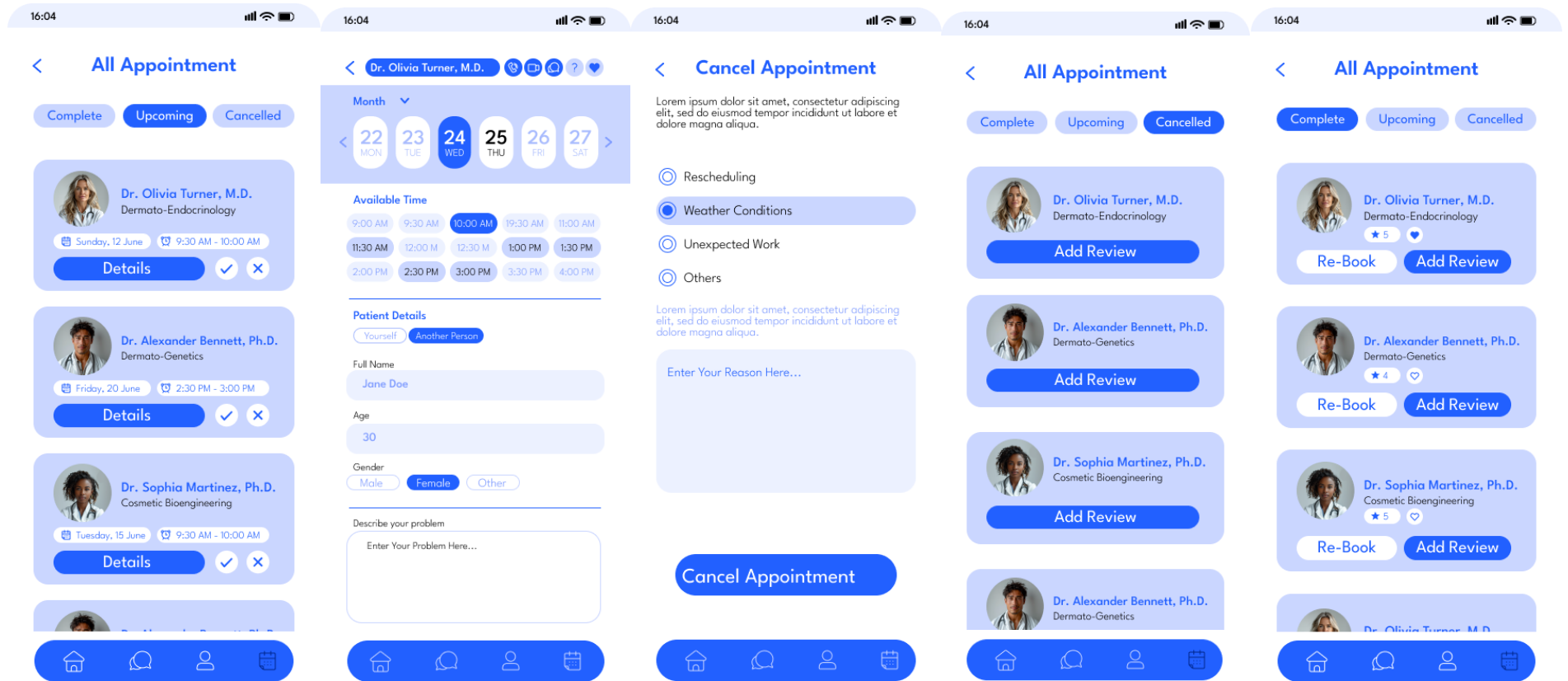
- Paulo Garza (2024, Enero 09). Optimizando el rendimiento en aplicaciones con React Native. Blog de Desarrollo Móvil. Recuperado de <https://paulogalarza.com/optimiza-el-rendimiento-de-las-aplicaciones-de-react-native/>
- Moreno, L., & Reyes, S. (2019). Aplicaciones móviles con React Native: Casos de éxito y desafíos. En Actas de la Conferencia Internacional de Desarrollo Móvil (pp. 134-145).

ANEXOS









1 Splash Screen



2 Bienvenida



3 Logeo, Crear Cuenta (Authentication)



4 Inicio



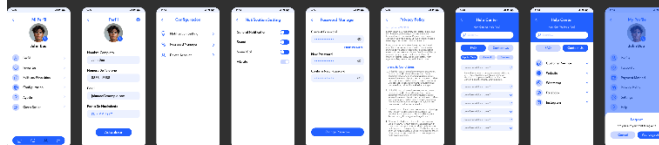
5 Doctores, Favoritos E Informacion



6 Pacientes E Informacion



7 Pantalla De Perfiles



7 Notificaciones, Chat



8 Agendar Cita



9 Citas

