**Universidad Nacional Mayor de San Marcos**

Universidad del Perú. Decana de América

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Software**

****

**CURITA SAN MARCOS**

**Trabajo como parte del curso de Taller de Construcción de Software Web**

**AUTORES**

Fiorella Patricia MIRANO SURQUISLLA  
Roger NEIRA CARQUIN  
Jose Alessandro QUISPE CABELLO  
Jatziry Fernanda SANCHEZ WONG  
Jocelyn Estrella SOTELO ARCE

Mariano VILLANUEVA CHIRITO

Sebastian LANDEO CUENTAS

**Asesor:**

Anita Marlene REYES HUAMAN

**LIMA – PERÚ**

**2024**

**Índice General**

[**1. CAPÍTULO I: INTRODUCCIÓN 3**](#_vrm32p81deae)

[**2. CAPÍTULO II: FASE DE PLANIFICACIÓN 5**](#_cffz52re62jv)

[2.1. Contexto del Proyecto 5](#_qjbi66ruf78j)

[2.1.1. Análisis FODA 5](#_37vc2l7z5dpi)

[Fortalezas: 6](#_aecsv9biiacl)

[Oportunidades: 6](#_p24fpj8gvp0g)

[Debilidades: 6](#_ffny8vnvvjya)

[Amenazas: 6](#_dv1xl3c8m3wa)

[2.1.2. Objetivos del proyecto 8](#_4z4xqe89q4za)

[Objetivo General: 8](#_hdz1pjbthcpo)

[Objetivos Específicos: 9](#_6ya6z5jfs5wf)

[2.2. Modelo de Negocio 9](#_alptuba66s8n)

[2.2.1. Identificación de procesos 9](#_vkd8rhm1jbae)

[2.2.2. Segmentación de mercado / usuarios 10](#_phb3ltguqcf)

[2.3. Análisis de Factibilidad 12](#_kgiw9a8l0xd0)

[2.4. Resultados esperados 12](#_xfyrkxp686k2)

[2.4.1. Corto Plazo 12](#_tefjel4x0ngs)

[2.4.2. Largo Plazo 12](#_3gzrqa82maeo)

[2.5. Gestión de Stakeholders 12](#_fcdyvai2btb7)

[2.5.1. Establecimiento de roles 12](#_skezv3wptu1s)

[2.5.2. Asignación de actividades 12](#_1pr5k11oe9v3)

[2.6. Gestión del Alcance 12](#_nmkanoqqy6by)

[2.7. Gestión del tiempo 12](#_m5pj5qcw00am)

[**3. CAPÍTULO III: FASE DE DISEÑO 12**](#_ezvtbynli0vi)

[3.1. Objetivos 12](#_ogwuk58rhwnr)

[3.2. ¿Quienes intervienen? 12](#_oqw3yapiyxdf)

[3.3. Actividades en la fase de diseño 12](#_ilgu5votqkek)

[3.4. Análisis de requerimientos 12](#_ufzoi8ft5940)

[3.4.1. Reconocimiento y evaluación de problemas 12](#_vrbv1alccqxp)

[3.5. Historias de Usuarios 12](#_qvpx0qu7ph3)

[3.6. Arquitectura del proyecto 12](#_jwnofoyu0aw8)

[3.6.1. Arquitectura de tecnologías 12](#_k2zfmotjo2my)

[3.6.2. Arquitectura de software 12](#_79m2bzumx7x7)

[3.6.3. Diagrama de Componentes 12](#_ktg30hw50dzg)

[3.7. Prototipo de Proyecto 12](#_ns5f15oujs)

[**4. CAPÍTULO IV: FASE DE EJECUCIÓN 13**](#_9r5sgehzbkox)

[**5. CAPÍTULO V: FASE DE PRUEBAS 13**](#_4hcll6nic04)

[**6. CAPÍTULO VI: FASE DE LANZAMIENTO 13**](#_ifpq3p7cqxkk)

# 

# **Capítulo 1**

# **Introducción**

**1.1 Resumen**

**1.2 Modelo Mobile Sprint (MMS)**

El Modelo Mobile Sprint (MMS) es una metodología ágil para el desarrollo de aplicaciones móviles, basada en ciclos iterativos cortos. Permite la implementación progresiva de funcionalidades, adaptándose a cambios y mejoras de manera eficiente.

La aplicación del Modelo Mobile Sprint (MMS) en el desarrollo de Curita San Marcos permitiría una ejecución estructurada y flexible del proyecto, basada en ciclos iterativos enfocados en la mejora continua. Cada sprint se centraría en la implementación y optimización de funcionalidades clave, como la gestión de disponibilidad de citas y la integración de notificaciones de confirmación de pago. A través de la recopilación de retroalimentación por parte de los usuarios y stakeholders en cada iteración, se realizarían ajustes tanto a nivel técnico como de usabilidad, garantizando una evolución progresiva de la aplicación. Esta metodología favorece la adaptación ágil a nuevos requerimientos sin comprometer la estabilidad del sistema, optimizando los tiempos de desarrollo y asegurando una experiencia eficiente e intuitiva para los usuarios.

**1.3 Fases del MMS**

El desarrollo de la aplicación de citas médicas seguirá la metodología MMS (Modelo Mobile Sprint) que cuenta con un conjunto de fases estructuradas para garantizar su calidad y alineación con las necesidades de los usuarios.

* **Planificación:** En esta fase, se definirán las metas del proyecto, se recogerán los requisitos de los usuarios y se establecerá el cronograma de trabajo para asegurar que la aplicación cumpla con las expectativas y necesidades de los pacientes y el personal médico.
* **Arquitectura de la aplicación:** Aquí se organizará la estructura de la app, definiendo las capas o componentes esenciales que permitirán la interacción eficiente entre el sistema de reservas, los usuarios y la base de datos. Además, se asegura que la arquitectura sea escalable y segura.
* **Diseño:** En esta etapa, se diseña la interfaz de usuario teniendo en cuenta la usabilidad y accesibilidad, así como las necesidades específicas de los pacientes al reservar citas médicas. También se definen los requisitos de rendimiento y las pruebas que garantizarán su correcto funcionamiento.
* **Desarrollo:** Durante esta fase, se lleva a cabo la implementación de la funcionalidad de la aplicación. Se desarrollarán las características que permitirán a los usuarios ver disponibilidad, seleccionar horarios, y confirmar citas médicas de manera intuitiva.
* **Pruebas:** En esta fase, se realizarán diversas pruebas para asegurar que la aplicación sea confiable, rápida y fácil de usar. Se probarán la funcionalidad general, la compatibilidad entre dispositivos y el rendimiento bajo condiciones de uso real.

# **Capítulo 2**

# **Fase de Planificación**

## **Contexto del Proyecto**

La Clínica Universitaria de la Universidad Nacional Mayor de San Marcos (UNMSM) desempeña un rol fundamental al ofrecer servicios médicos a estudiantes, docentes, personal administrativo y, en ciertos casos, a personas externas. Sin embargo, los procesos actuales de gestión de citas presentan varias deficiencias, como tiempos de espera excesivos, largas filas y falta de un sistema digital que optimice la experiencia del usuario.

En este contexto, se plantea el desarrollo de una aplicación móvil llamada Curita San Marcos, que tiene como propósito principal mejorar la experiencia de los usuarios al reservar citas médicas en la clínica universitaria. Este proyecto busca resolver los problemas actuales mediante la implementación de un sistema de reservas eficiente y accesible, que además incorpore herramientas tecnológicas avanzadas, como la inteligencia artificial, para facilitar la interacción y optimizar la gestión interna.

El proyecto abarca tanto a los usuarios finales (estudiantes, docentes, personal administrativo y externos) como al personal médico y administrativo de la clínica, quienes serán beneficiarios directos de las mejoras en los procesos de programación y gestión de citas.

### **Análisis FODA**

**Figura 1.**

*Análisis FODA del proyecto*

|  |  |
| --- | --- |
| **Fortalezas:**  1. La clínica universitaria cuenta con una amplia base de usuarios que requieren sus servicios médicos, garantizando una demanda constante. 2. Uso de tecnología avanzada, como inteligencia artificial, para mejorar la experiencia del usuario. 3. La universidad cuenta con infraestructura tecnológica y personal capacitado que facilita la implementación del proyecto. | **Oportunidades:**  1. Reducción de las largas filas y tiempos de espera mediante un sistema eficiente de reservas. 2. Incremento en la satisfacción de los usuarios al optimizar el proceso de citas médicas. 3. Posibilidad de integrar futuras funcionalidades, como la telemedicina, para expandir los servicios de la clínica. 4. Mejor aprovechamiento del personal médico y administrativo al contar con herramientas para organizar su tiempo de manera efectiva. |
| **Debilidades:**  1. Resistencia al cambio por parte de los usuarios acostumbrados a los procesos manuales actuales. 2. Dependencia de una conexión a internet estable para el correcto funcionamiento de la aplicación. 3. Posible curva de aprendizaje para algunos usuarios en el uso de la aplicación. | **Amenazas:**  1. Riesgo de saturación en los horarios de atención si la demanda supera la capacidad disponible. 2. Competencia indirecta de aplicaciones de salud externas que ofrecen servicios similares. 3. Dificultades en la actualización constante de la base de datos de médicos, especialidades y horarios disponibles. |

***Nota.* Elaboración propia**

En la Figura 1 se evidencia que las fortalezas del proyecto radican en su capacidad para aprovechar los recursos existentes dentro de la Universidad Nacional Mayor de San Marcos (UNMSM). La alta demanda de los servicios de la clínica universitaria asegura una base sólida de usuarios, lo que respalda la viabilidad del proyecto. Asimismo, la implementación de tecnología avanzada, como la inteligencia artificial, no solo mejora la experiencia del usuario, sino que también posiciona a la clínica como una entidad innovadora en la gestión de servicios médicos. La infraestructura tecnológica y el talento humano disponible en la universidad representan un respaldo clave para la ejecución exitosa de esta iniciativa.

Por otro lado, como se muestra en la Figura 1, las oportunidades del proyecto se concentran en su capacidad para abordar las ineficiencias actuales en la gestión de citas médicas. La reducción de tiempos de espera y la eliminación de largas filas son beneficios directos que incrementarán la satisfacción de los usuarios. Además, la digitalización abre las puertas a la integración de funcionalidades futuras, como la telemedicina, lo que podría ampliar el alcance de los servicios ofrecidos. Por otro lado, el personal médico y administrativo también se verá beneficiado, ya que el sistema permitirá gestionar de forma más eficiente los horarios y recursos, optimizando su tiempo y esfuerzo.

Se colige también que una de las principales debilidades del proyecto es la resistencia al cambio, especialmente entre los usuarios que están acostumbrados a los procesos manuales. Esto podría ralentizar la adopción inicial de la aplicación. Además, el correcto funcionamiento del sistema depende de una conexión a internet estable, lo que podría representar un desafío en ciertas áreas. Finalmente, existe la posibilidad de que algunos usuarios enfrenten una curva de aprendizaje al interactuar con la aplicación, lo que requerirá estrategias de capacitación y soporte técnico para minimizar estos inconvenientes.

Por último, de la figura se desprende que entre las amenazas principales se encuentra la saturación de horarios de atención si la demanda supera la capacidad operativa de la clínica, lo que podría generar insatisfacción entre los usuarios. Asimismo, existen aplicaciones de salud externas que podrían representar una competencia indirecta al ofrecer servicios similares. Finalmente, el proyecto dependerá de la actualización constante de los datos relacionados con médicos, especialidades y horarios disponibles; cualquier retraso o error en este proceso podría impactar negativamente en la confianza de los usuarios.

### **Objetivos del proyecto**

#### **Objetivo General:**

Desarrollar una aplicación móvil que permita a los usuarios de la Clínica Universitaria de la UNMSM gestionar citas médicas de forma eficiente y accesible, mejorando su experiencia y optimizando los procesos internos de la clínica.

#### **Objetivos Específicos:**

* Diseñar e implementar un sistema digital para la reserva, cancelación y reprogramación de citas médicas, adaptado a las necesidades de los usuarios.
* Incorporar notificaciones automáticas y recordatorios de citas para reducir la tasa de inasistencia.
* Crear un módulo de administración para el personal de la clínica que permita gestionar horarios y disponibilidad de manera centralizada.
* Implementar un chatbot con inteligencia artificial para resolver dudas frecuentes de los usuarios, optimizando la atención al cliente.
* Enviar correos electrónicos con detalles de las citas confirmadas, mejorando la comunicación entre usuarios y personal administrativo.

## **Modelo de Negocio**

### **Identificación de procesos**

La identificación de procesos en el proyecto Curita San Marcos es esencial para garantizar que las funciones principales de la aplicación se alineen con las necesidades de los usuarios y el personal de la clínica. Por ende, se definieron los siguientes procesos:

**Tabla 1.**

*Identificación de Procesos*

|  |  |  |
| --- | --- | --- |
| **Proceso** | **Descripción** | **Actor Principal** |
| **Reservar citas** | Los usuarios seleccionan especialidades, médicos y horarios disponibles para agendar citas médicas. | Usuarios Finales |
| **Gestión de disponibilidad** | El personal de la clínica configura los horarios y días disponibles de los médicos y recursos. | Personal Administrativo |
| **Soporte mediante chatbot** | Implementar un asistente virtual que resuelva consultas frecuentes de los usuarios. | Usuarios Finales |
| **Consulta y seguimiento** | Los usuarios visualizan el historial de citas y realizan seguimientos de sus atenciones médicas. | Usuarios Finales |
| **Pago de citas médicas** | Los usuarios serán capaces de realizar el pago de la consulta | Usuarios Finales |

***Nota. Elaboración propia***

### **Segmentación de mercado / usuarios**

La segmentación de mercado de Curita San Marcos se divide en dos grupos principales. En primer lugar, se encuentran los usuarios finales, que incluyen a estudiantes, docentes, personal administrativo y pacientes externos. Los estudiantes de la UNMSM representan una parte significativa de la demanda de servicios médicos en la clínica universitaria, ya que dependen de estos servicios como parte de su bienestar integral durante su formación académica. Por su parte, los docentes requieren un acceso ágil y eficiente a los servicios médicos debido a sus ajustados horarios laborales. El personal administrativo de la universidad también forma parte importante de este grupo, buscando una experiencia fluida y cómoda en la gestión de su atención médica. Asimismo, los pacientes externos que ocasionalmente acceden a los servicios médicos de la clínica universitaria completan este segmento de usuarios finales.

El segundo grupo está compuesto por el personal de la clínica, quienes también son beneficiarios clave del sistema. Este grupo incluye a los médicos, quienes necesitan una herramienta eficiente para administrar sus horarios y brindar una atención organizada y puntual a sus pacientes. También incluye al personal administrativo, encargado de la coordinación y gestión operativa de las citas médicas. Este grupo se beneficiará de un sistema centralizado y digitalizado, diseñado para optimizar su carga laboral y mejorar la eficiencia en la prestación de los servicios de salud.

Esta segmentación permite que el desarrollo de Curita San Marcos esté enfocado en cubrir las necesidades específicas de cada grupo, asegurando que tanto los usuarios finales como el personal de la clínica tengan una experiencia positiva y eficiente.

## **Análisis de Factibilidad**

**Tabla 2.**

*Análisis de factibilidad*

|  |  |  |  |
| --- | --- | --- | --- |
| **RECURSOS HUMANOS** | | | |
| **Rol/Profesión** | **Cantidad** | **Costo Individual** | **Costo Total** |
| Programador UX/UI | 02 | 2000 | 4000 |
| Programador Front End | 01 | 2200 | 2200 |
| Programador Full Stack | 01 | 2800 | 2800 |
| Diseñadora UX/UI | 01 | 2000 | 4000 |
| Programador Back End | 02 | 2500 | 5000 |
| Scrum Master | 01 | 3000 | 3000 |
| **COSTOS OPERATIVOS** | | | |
| Material | Cantidad | Costo Individual | Costo Total |
| Energía eléctrica | 07 | 100 | 700 |
| **RECURSOS TECNOLÓGICOS** | | | |
| **HARDWARE** | | | |
| **Material** | **Cantidad** | **Costo Individual** | **Costo Total** |
| Computadoras/Laptops | 07 | 2000 | 14000 |
| **SOFTWARE** | | | |
| **Material** | **Cantidad** | **Costo Individual** | **Costo Total** |
| JavaScript | 01 | - | - |
| Baileys | 01 | - | - |
| Gemini | 01 | - | - |
| Docker | 01 | - | - |
| Google Firebase | 01 | - | - |

## **Resultados esperados**

### **Corto Plazo**

* Obtener la atención de los estudiantes, docentes y personal administrativo de la UNMSM sobre nuestro aplicativo.
* Conseguir al menos **50 usuarios registrados** en la primera fase de implementación.
* Reducir los tiempos de espera en la gestión de citas en al menos un **30%** en comparación con el sistema actual.
* Detectar y corregir la mayor cantidad de errores posibles en la primera versión del aplicativo.

### **Largo Plazo**

* Finalizar el desarrollo del aplicativo y publicarlo en **Google Play Store** y **GitHub** en el apartado *release* del repositorio, permitiendo que cualquier usuario interesado pueda instalarlo.
* Mejorar continuamente la aplicación, incorporando nuevas funcionalidades basadas en la retroalimentación de los usuarios.
* **Automatizar el 80%** de los procesos administrativos relacionados con la gestión de citas médicas en la clínica universitaria.
* Integrar herramientas de inteligencia artificial para mejorar la asistencia virtual y optimizar la asignación de citas.

## **Gestión de Stakeholders**

### **Establecimiento de roles**

Para el desarrollo del sistema de gestión de citas médicas en la UNMSM, se ha definido un equipo de trabajo con roles específicos que garantizarán el cumplimiento de los objetivos del proyecto. Cada rol dentro del equipo tiene responsabilidades clave que contribuyen a la planificación, desarrollo, implementación y mantenimiento de la aplicación. La correcta asignación de estos roles permitirá una ejecución eficiente del proyecto, asegurando que cada etapa se lleve a cabo con los estándares requeridos.

**Tabla 3.**

*Establecimiento de Roles*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MATRIZ GENERAL DE STAKEHOLDERS** | | | | |
| **ID** | **Interesado** | **Cargo** | **Rol** | **Responsabilidades** |
| DEV001 | Neira Carquin. Roger | Estudiante de la FISI - UNMSM | Programador UX/UI | Diseñar y optimizar la experiencia del usuario, asegurando que la interfaz sea intuitiva y accesible. |
| DEV002 | Mirano Surquislla, Fiorella Patricia | Estudiante de la FISI - UNMSM | Programador Front End | Implementar la interfaz gráfica, asegurando una presentación atractiva y funcional. |
| DEV003 | Sotelo Arce, Jocelyn Estrella | Estudiante de la FISI - UNMSM | Programador UX/UI | Diseñar y optimizar la experiencia del usuario, asegurando que la interfaz sea intuitiva y accesible. |
| DEV004 | Quispe Cabello, Jose Alessandro | Estudiante de la FISI - UNMSM | Programador Back End | Desarrollar y mantener la lógica del servidor y la base de datos para un rendimiento eficiente. |
| DEV005 | Sanchez Wong, Jatziry Fernanda | Estudiante de la FISI - UNMSM | Programador Full Stack | Integrar el desarrollo del front-end y back-end, asegurando una comunicación fluida entre ambos. |
| DEV006 | Landeo Cuentas, Sebastian | Estudiante de la FISI - UNMSM | Diseño UX/UI | Se encargará de garantizar que la satisfacción del usuario sea la correcta. |
| DEV007 | Villanueva Chirito, Mariano | Estudiante de la FISI - UNMSM | Programador Back End | Desarrollar y mantener la lógica del servidor y la base de datos para un rendimiento eficiente. |
| SCR001 | Sanchez Wong, Jatziry Fernanda | Estudiante de la FISI - UNMSM | Scrum Master | Facilitar el desarrollo ágil, asegurando que el equipo cumpla con las metodologías y plazos establecidos. |
| SPN001 | Reyes Huamana, Anita Marlene | Profesora del Taller de Construcción de Software Móvil | Patrocinador | Brindar apoyo y supervisión académica para la correcta ejecución del proyecto. |
| SPN002 | Facultad de Sistemas e Informatica | Facultad de Sistemas e Informatica | Patrocinador | Proveer recursos y aval académico para la ejecución y evaluación del proyecto. |
| USR001 | Estudiantes de la Universidad Nacional Mayor de San Marcos | Estudiantes de la Universidad Nacional Mayor de San Marcos | Usuarios finales | Utilizar el aplicativo para la gestión de citas médicas en la clínica universitaria. |

### 

### **Asignación de actividades**

Se han distribuido las actividades entre los miembros del equipo de manera organizada. Cada actividad ha sido asignada con base en las competencias y responsabilidades de cada rol, asegurando que el proyecto avance según el cronograma establecido.

**Tabla 4.**

*Matriz de Actividades*

|  |  |  |
| --- | --- | --- |
| **MATRIZ DE ACTIVIDADES** | | |
| **ID** | **Actividad** | **Descripción** |
| A01 | Selección del tema | El equipo define el enfoque y objetivo principal del proyecto |
| A02 | Planteamiento de la solución | Se analiza la problemática y se define la solución tecnológica a implementar |
| A03 | Definición de requerimientos funcionales | Se identifican las funcionalidades clave que debe cumplir el sistema. |
| A04 | Definición de requerimientos no funcionales | Se establecen aspectos de seguridad, rendimiento y usabilidad del sistema. |
| A05 | Diseño de la base de datos | Se define la estructura lógica de la base de datos basada en los requerimientos. |
| A06 | Modelado de base de datos | Se desarrolla el modelo físico de la base de datos, incluyendo relaciones y restricciones. |
| A07 | Implementación en Base de Datos | Se implementa la base de datos en el entorno elegido |
| A08 | Arquitectura del sistema | Se diseña la estructura general del software, incluyendo capas y tecnologías. |
| A09 | Prototipado de UI | Se crean prototipos interactivos de la interfaz de usuario en Figma u otra herramienta. |
| A10 | Creación del repositorio | Se configura el repositorio en GitHub para la gestión del código y documentación. |
| **Desarrollo del software** | | |
| A11 | Página de inicio | Se desarrolla la interfaz principal del sistema. |
| A12 | Desarrollo del Inicio de sesión | Implementación del sistema de autenticación para usuarios. |
| A13 | Registro de nuevo usuario | Permite seleccionar el tipo de usuario (alumno, personal, externo) y verificar con código. Se consultará a la profesora sobre el funcionamiento de los códigos. |
| A14 | Ver especialidades y médicos | Permite consultar los médicos disponibles y sus especialidades. |
| A15 | Ver fechas disponibles | Implementación de un calendario para seleccionar fechas de atención. |
| A16 | Reservar cita | Se desarrolla el sistema de agendamiento de citas médicas. |
| A17 | Realizar pago | Implementación de pagos con tarjeta (simulador) y generación de código de pago presencial (pendiente de confirmación). Se envía un correo de confirmación. |
| A18 | Ver citas pagadas | Los usuarios pueden revisar su historial de citas confirmadas. |
| A19 | Llenar historia clínica | Luego de una cita, el usuario completa su historia clínica. Solo se permite una historia clínica por usuario. Se generará un PDF. |
| A20 | |  | | --- | | Implementación de IA |  |  | | --- | |  | | Se integrarán algoritmos de inteligencia artificial para mejorar la experiencia del usuario. |

***Fuente:*** *elaboración propia.*

Como resultado de esta primera etapa se obtiene un acta de constitución de proyecto en el que sintetiza lo antes estudiado.

**Tabla 5.**

*Acta de constitución*

|  |  |  |
| --- | --- | --- |
| **ACTA DE CONSTITUCIÓN DEL PROYECTO** | | |
| Curita San Marcos | | |
| **Empresa / Organización** | Grupo 5 | |
| **Proyecto** | Curita San Marcos | |
| **Fecha de inicio** | 15/01/25 | |
| **Fecha de entrega** | 08/03/25 | |
| **Cliente** | UNMSM – Clínica Universitaria | |
| **Patrocinador principal** | Facultad de Ingeniería de Sistemas e Informática | |
| **Gerente de proyecto** | Sanchez Wong, Jatziry Fernanda | |
| **Lista de Patrocinadores** | | |
| **Nombre** | **Cargo** | **Departamento / División** |
| Reyes Huamán, Anita Marlene | Asesora | Profesora FISI UNMSM |
| **Justificación del Proyecto** | | |
| **Justificación teórica**  El proyecto tiene como objetivo principal aplicar los conocimientos adquiridos en el curso de taller de construcción de software móvil y en cursos previos para planificar y desarrollar una aplicación móvil orientada a la gestión eficiente de citas médicas en un entorno universitario. Desde un punto de vista teórico, este proyecto es esencial, ya que contribuye al conocimiento existente sobre cómo las aplicaciones móviles pueden optimizar el acceso a los servicios de salud, mejorar la organización de las citas médicas y reducir tiempos de espera a través de la digitalización de procesos. Además, la integración de herramientas como la inteligencia artificial permite mejorar la experiencia del usuario, agilizando la asignación de turnos y el seguimiento de historiales clínicos.  **Justificación práctica**  El proyecto busca facilitar el acceso de los estudiantes, personal universitario y usuarios externos a los servicios médicos mediante una plataforma digital intuitiva. La aplicación permitirá la reserva de citas médicas, la consulta de especialidades y médicos disponibles, la gestión de pagos, el almacenamiento seguro de historiales clínicos y la implementación de recordatorios para mejorar la asistencia a consultas. La importancia de este proyecto radica en la creciente demanda de soluciones tecnológicas en el sector salud, la necesidad de optimizar la gestión de citas y la digitalización de procesos administrativos que tradicionalmente son engorrosos. Al centralizar estos servicios en una aplicación, se espera mejorar la eficiencia del sistema de salud universitario y brindar una experiencia más cómoda y accesible a los usuarios. | | |
| **Lista de Stakeholders** | | |
| **Nombre** | **Cargo** | **Rol** |
| Neira Carquin. Roger | Estudiante de la FISI - UNMSM | Programador UX/UI |
| Mirano Surquislla, Fiorella Patricia | Estudiante de la FISI - UNMSM | Programador Front End |
| Sotelo Arce, Jocelyn Estrella | Estudiante de la FISI - UNMSM | Programador UX/UI |
| Quispe Cabello, Jose Alessandro | Estudiante de la FISI - UNMSM | Programador Back End |
| Sanchez Wong, Jatziry Fernanda | Estudiante de la FISI - UNMSM | Programador Full Stack |
| Landeo Cuentas, Sebastian | Estudiante de la FISI - UNMSM | Diseño UX/UI |
| Villanueva Chirito, Mariano | Estudiante de la FISI - UNMSM | Programador Back End |
| Sanchez Wong, Jatziry Fernanda | Estudiante de la FISI - UNMSM | Scrum Master |
| Reyes Huamana, Anita Marlene | Profesora del Taller de Construcción de Software Móvil | Patrocinador |
| Facultad de Sistemas e Informatica | Facultad de Sistemas e Informatica | Patrocinador |
| Estudiantes de la Universidad Nacional Mayor de San Marcos | Estudiantes de la Universidad Nacional Mayor de San Marcos | Usuarios finales |
| **Premisas y restricciones** | | |
| **Premisas**  Se pretende entregar la aplicación el 08/03/25 mediante un archivo apk para que los interesados puedan hacer uso de ella y brindar feedback para mejoras en posteriores actualizaciones.  **Restricciones**  El proyecto no contempla la publicación de la aplicación en tiendas como Google Play Store debido a los costos asociados al registro de desarrolladores y las políticas establecidas, que podrían extender el tiempo de desarrollo. Además, el sistema estará diseñado para uso exclusivo de la comunidad universitaria, sin integración con servicios externos de salud en esta primera fase. | | |
| **Presupuesto Inicial** | | |
| El proyecto cuenta con un presupuesto de 35,700 soles, destinado a cubrir los costos de recursos humanos, materiales y tecnológicos necesarios para el desarrollo de la aplicación. | | |
| **Requisitos de aprobación del proyecto** | | |
| * La aplicación deberá ser entregada como fecha final 06/03/23 en formato APK para la prueba de los distintos interesados en el proyecto * El presupuesto asignado **no podrá ser ampliado** para el desarrollo de la aplicación. * Se entregará la documentación completa del proyecto al cliente, para los fines que este estime necesarios. | | |
| **Aprobaciones** | | |
| **Interesados / Patrocinadores** | **Fecha** | **Firma** |
| Sanchez Wong, Jatziry Fernanda | 01/02/25 |  |

***Fuente:*** *elaboración propia.*

## **Gestión del Alcance**

* El desarrollo está previsto para 5 semanas hábiles tras la finalización de la planificación.
* La aplicación está dirigida a la comunidad de la UNMSM, utilizando como credenciales de acceso los códigos institucionales de estudiantes, personal administrativo y externos.
* La aplicación será desarrollada para dispositivos con sistema operativo Android y tendrá un enfoque open source.

### **Naturaleza de la aplicación**

La aplicación será de naturaleza nativa, ya que está dirigida exclusivamente a dispositivos con sistema operativo Android, debido a su amplia adopción dentro de la comunidad universitaria de la UNMSM. Se ha descartado la opción de una aplicación web, ya que el enfoque principal es proporcionar una experiencia fluida y optimizada para la gestión de citas médicas desde dispositivos móviles.

Uno de los aspectos clave del desarrollo es garantizar un rendimiento óptimo, permitiendo a los usuarios realizar tareas como la reserva de citas, el pago de servicios y la consulta de historial médico sin interrupciones. Para lograr esto, se aprovecharán al máximo las capacidades del hardware de los dispositivos Android, incluyendo el uso de notificaciones push para recordar citas y almacenamiento local para mejorar la accesibilidad a la información.

El desarrollo de la aplicación se realizará en Java y Kotlin, lenguajes ampliamente utilizados en la comunidad de Android y que permiten una mejor integración con los servicios del sistema. En cuanto al backend y la gestión de datos, se utilizará Firebase, lo que permitirá una autenticación segura, almacenamiento en la nube y sincronización en tiempo real, facilitando la administración de citas médicas y la comunicación con los usuarios.

* + 1. **Cumplimiento de Accesibilidad**

El desarrollo de la aplicación considera principios de accesibilidad basados en las guías WCAG 2.0, asegurando que pueda ser utilizada por la mayor cantidad de personas posible, incluyendo aquellas con discapacidades o limitaciones físicas. Para lograr esto, se implementarán diversas características que mejorarán la percepción, operación, comprensión y robustez del sistema.

* + - 1. **Perceptible**

En cuanto a la percepción, la interfaz de usuario estará optimizada para pantallas pequeñas, mostrando únicamente la información esencial para evitar sobrecarga visual. Además, se implementará la opción de zoom y ajuste de texto, permitiendo a los usuarios ampliar o reducir el contenido según sus necesidades, lo que mejorará la legibilidad de la aplicación.

* + - 1. **Operable**

Respecto a la operabilidad, los botones y elementos interactivos estarán ubicados estratégicamente para facilitar su acceso, especialmente en dispositivos con pantallas grandes. De esta manera, se garantizará que la navegación sea fluida e intuitiva, evitando dificultades al interactuar con la aplicación.

* + - 1. **Comprensible**

En términos de comprensión, la aplicación se diseñará para adaptarse automáticamente a distintos tamaños de pantalla y orientaciones, permitiendo la variación entre modo vertical y horizontal. Para ello, se utilizarán componentes nativos de Android que aseguren compatibilidad con una amplia gama de dispositivos.

Asimismo, el diseño seguirá las pautas de Material Design, lo que proporcionará una experiencia visual consistente y familiar para los usuarios.

* + - 1. **Robusto**

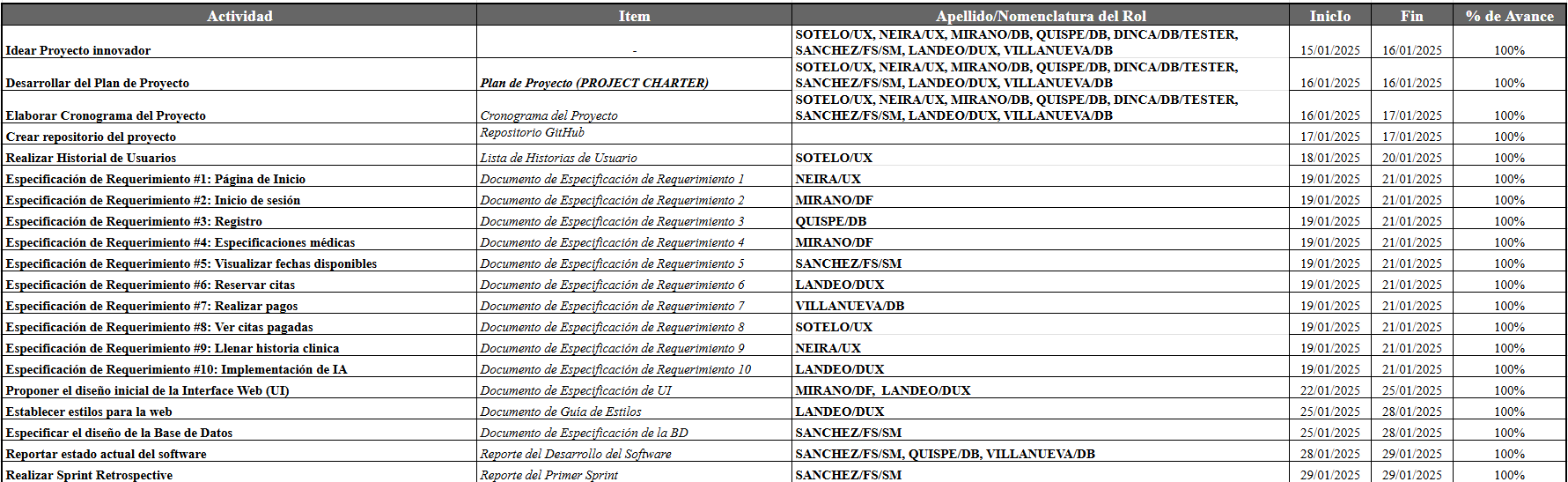
Finalmente, la aplicación será robusta, respondiendo adecuadamente ante eventos no previstos, como el ingreso de credenciales incorrectas o la pérdida de conexión a internet. Se implementarán mensajes de error claros y procesos de recuperación para que el usuario pueda continuar con su experiencia sin frustraciones.

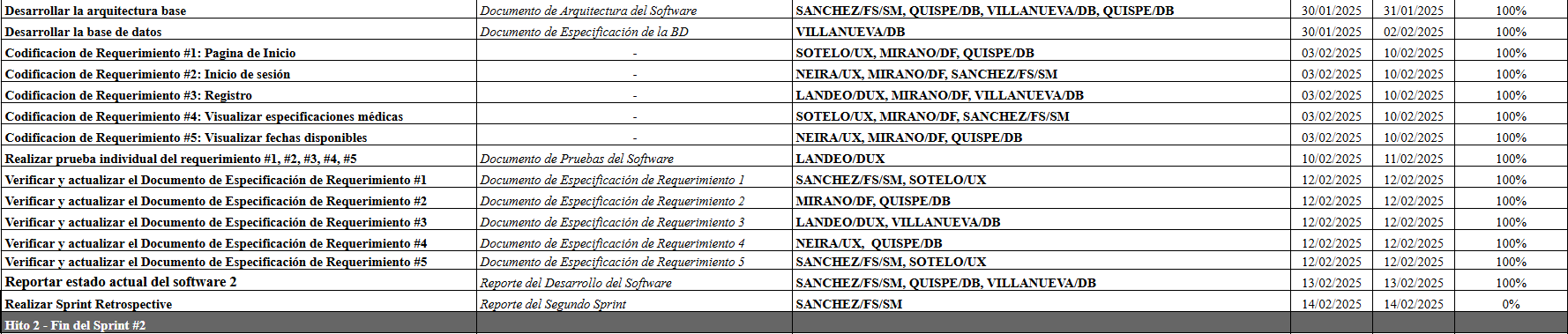
## **Gestión del tiempo**

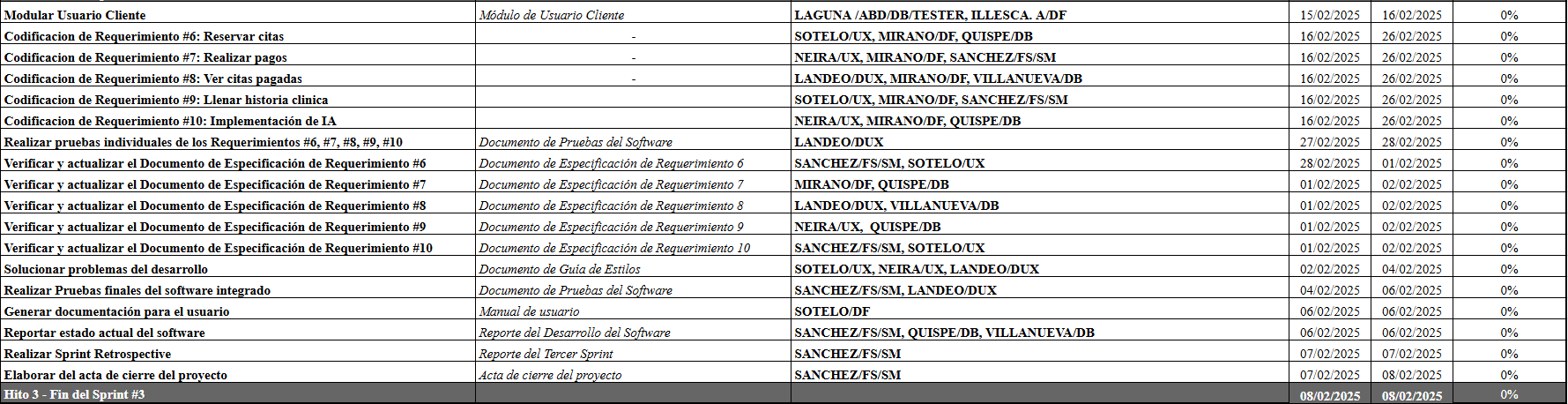
* + 1. **Cronograma**

**Figura 2.**

*Cronograma del Proyecto*







***Fuente:*** *elaboración propia.*

* 1. **Gestión de Costos**

Para el desarrollo de la aplicación orientada a la gestión de citas médicas en la UNMSM, se ha estimado un presupuesto total que abarca los costos esenciales en recurso humano, tecnología, operaciones e imprevistos. Este presupuesto permitirá cubrir todas las fases del proyecto, desde la planificación hasta la implementación y prueba del sistema.

A continuación, se detalla la distribución del presupuesto:

**Tabla 6.**

*Presupuesto Total*

|  |  |
| --- | --- |
| **DETERMINACIÓN DE PRESUPUESTO** | |
| **Recurso Humano** | S/. 21000 |
| **Recurso Tecnológico** | S/. 14000 |
| **Costos Operativos** | S7. 700 |
| **Imprevistos ($)** | 5% |
| **PRESUPUESTO** | S/. 4410 |

***Fuente:*** *elaboración propia.*

Se ha considerado un 5% del presupuesto para imprevistos, garantizando la capacidad de respuesta ante cualquier inconveniente que pueda surgir durante el desarrollo del proyecto

* 1. **Gestión de Cambios**
     1. **Formato de solicitud de cambios**

Para formalizar cualquier cambio dentro del sistema, se utilizará el siguiente formato de solicitud, el cual deberá ser completado por el responsable de la solicitud y evaluado por el equipo técnico:

**Tabla 7.**

*Formato de solicitud de cambios*

|  |  |
| --- | --- |
| **Campo** | **Descripción** |
| **Código de Solicitud** | Identificador único del cambio en el formato SC-[AÑO]-[NÚMERO]. Permite llevar un control ordenado de las solicitudes. |
| **Fecha de solicitud** | Fecha exacta en la que se genera la solicitud del cambio, para su correcto seguimiento y registro histórico. |
| **Solicitante** | Nombre completo y cargo de la persona o área que solicita el cambio. Esto ayuda a determinar la fuente y justificación del requerimiento. |
| **Área responsable** | Departamento o equipo encargado de evaluar, aprobar e implementar el cambio (Ejemplo: Desarrollo, QA, Seguridad, Infraestructura). |
| **Prioridad** | Clasificación del nivel de urgencia del cambio: Alta (crítico para el funcionamiento), Media (importante pero no urgente), Baja (mejoras o ajustes menores). |
| **Título del Cambio** | Nombre breve y claro que describa la naturaleza del cambio solicitado. |
| **Descripción Detallada** | |  | | --- | |  |  |  | | --- | | Explicación completa del cambio solicitado, indicando qué se debe modificar, agregar o eliminar en el sistema. Debe incluir detalles técnicos si es necesario. | |
| **Motivo de cambio** | Razón específica por la cual se solicita el cambio, como la corrección de un error, mejora de funcionalidad, cumplimiento de normativas o adaptación a nuevos requerimientos del usuario. |
| **Beneficios esperados** | Impacto positivo que generará el cambio en el sistema o en la experiencia del usuario, como mayor eficiencia, mejor rendimiento, seguridad mejorada, etc. |
| **Áreas afectadas** | |  | | --- | |  |   Componentes del sistema que se verán modificados por el cambio, como el Frontend, Backend, Base de Datos, Seguridad, UX/UI o cualquier otra parte relevante. |
| **Tiempo Estimado de Implementación** | Cálculo aproximado del tiempo necesario para realizar el cambio, expresado en horas, días o semanas, considerando fases como desarrollo, pruebas y despliegue. |
| **Recursos Necesarios** | Identificación del personal (desarrolladores, testers, analistas), herramientas, software o infraestructura adicional requerida para implementar el cambio. |
| **Impacto en el Proyecto** | Evaluación de cómo el cambio puede afectar el cronograma, presupuesto o alcance del proyecto, y si se requieren ajustes en la planificación. |
| **Estado Actual** | Situación en la que se encuentra la solicitud: En Revisión, Aprobado, Rechazado o Implementado. Esto permite dar seguimiento al avance. |
| **Fecha de Resolución Estimada** | Fecha límite en la que se espera tomar una decisión sobre la solicitud, dependiendo de la complejidad del cambio. |
| **Responsable de Evaluación** | Nombre y cargo de la persona encargada de analizar la solicitud y decidir si se aprueba o rechaza. Puede ser un líder de proyecto, arquitecto de software, etc. |
| **Observaciones/Recomendaciones** | Comentarios adicionales sobre la viabilidad del cambio, posibles riesgos o recomendaciones para su implementación. |
| **Aprobación** | |  | | --- | |  |  |  | | --- | | Espacio para la firma del responsable de aprobación, indicando su cargo y la fecha en la que se toma la decisión final. | |

***Fuente:*** *elaboración propia.*

* + 1. **Lista de estados de solicitud de cambios**

Para gestionar correctamente las solicitudes de cambio en el proyecto, se ha definido una lista de estados que reflejan el ciclo de vida de cada solicitud. Estos estados permiten un seguimiento estructurado desde su creación hasta su implementación o rechazo.

**Tabla 8.**

*Lista de estados*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Estado** | **Descripción** | **Fase** |
| **ST01** | **Registrado** | La solicitud ha sido formalmente ingresada en el sistema y está pendiente de revisión por el comité de cambios. | Cierre |
| **ST02** | **En evaluación** | Se está analizando el impacto del cambio en términos de viabilidad técnica, costos, riesgos y alineación con los objetivos del proyecto. | Análisis |
| **ST03** | **Aprobado** | La solicitud ha sido validada y autorizada para su implementación tras cumplir con los criterios establecidos. | Planificación |
| **ST04** | **Rechazado** | |  | | --- | |  |  |  | | --- | | El cambio no ha sido aprobado debido a su inviabilidad, falta de alineación con los objetivos o impacto negativo en el proyecto. | | Cierre |
| **ST05** | **En Desarrollo** | Se ha iniciado la implementación del cambio conforme a los requisitos y especificaciones aprobadas. | Ejecución |
| **ST06** | **En pruebas** | La modificación ha sido desarrollada y se encuentra en fase de validación para verificar su correcto funcionamiento. | Validación |
| **ST07** | **Pendiente de Implementación** | El cambio ha superado las pruebas y está programado para su despliegue en el entorno productivo. | Implementación |
| **ST08** | **Implementado** | La solicitud ha sido aplicada exitosamente en el sistema y se encuentra operativa. | Cierre |
| **ST09** | **Cancelado** | La solicitud ha sido anulada antes de su implementación debido a modificaciones en los requisitos o cambios estratégicos en el proyecto. | Cierre |

***Fuente:*** *elaboración propia.*

Este esquema garantiza que cada solicitud de cambio sea gestionada de manera meticulosa y alineada con las necesidades del proyecto, minimizando riesgos y optimizando los procesos de toma de decisiones.

* + 1. **Ejemplo de solicitud de cambios**

**Tabla 9.**

*Ejemplo de solicitud de cambios*

|  |  |
| --- | --- |
| **Código de Solicitud** | SC-2025-002 |
| **Fecha de solicitud** | 10/02/2025 |
| **Solicitante** | Jocelyn Sotelo – Líder de UX/UI |
| **Área responsable** | Desarrollo y UX/UI |
| **Prioridad** | Alta |
| **Título del Cambio** | Optimización del diseño responsivo en dispositivos móviles |
| **Descripción Detallada** | |  | | --- | |  |   Se solicita mejorar la adaptabilidad del sistema en pantallas de distintos tamaños, optimizando los estilos CSS y la carga de imágenes. |
| **Motivo de cambio** | Se han detectado problemas de usabilidad en dispositivos móviles, lo que afecta la experiencia del usuario y la tasa de conversión. |
| **Beneficios esperados** | Mejora de la accesibilidad, aumento del tiempo de permanencia en la plataforma y reducción de la tasa de rebote en dispositivos móviles. |
| **Áreas afectadas** | Frontend, UX/UI |
| **Tiempo Estimado de Implementación** | 10 días hábiles |
| **Recursos Necesarios** | Desarrolladores Frontend, Diseñadores UX/UI, herramientas de pruebas responsivas. |
| **Impacto en el Proyecto** | Se requiere ajustar el sprint actual y reprogramar algunas tareas de desarrollo. No se prevé un impacto en costos. |
| **Estado Actual** | En Evaluación |
| **Fecha de Resolución Estimada** | 15/02/2025 |
| **Responsable de Evaluación** | Jatziri Sanchez – Jefe de Desarrollo |
| **Observaciones/Recomendaciones** | Se recomienda realizar pruebas con usuarios antes del despliegue final para asegurar la correcta implementación del cambio. |
| **Aprobación** | |  | | --- | |  |  |  | | --- | | **Firma:** \_\_\_\_\_\_\_\_\_\_\_\_  **Cargo:** \_\_\_\_\_\_\_\_\_\_\_\_  **Fecha:** \_\_\_\_\_\_\_\_\_\_\_\_ | |

***Fuente:*** *elaboración propia.*

# **Capítulo 3**

# **Fase de Diseño**

## **Objetivos**

* Definir la arquitectura técnica y funcional de la aplicación Curita San Marcos.
* Establecer los requerimientos específicos en función de las necesidades de los usuarios.
* Diseñar una interfaz intuitiva y accesible que optimice la gestión de citas médicas.
* Garantizar la escalabilidad y mantenibilidad del sistema.
* Implementar medidas de seguridad para la protección de datos sensibles.

## **¿Quiénes intervienen?**

En la fase de diseño, los principales actores son:

***Equipo de desarrollo:***

* Arquitectos de software
* Diseñadores UX/UI
* Desarrolladores frontend y backend
* Especialistas en seguridad

***Stakeholders (en teoría):***

* Personal administrativo de la clínica de la UNMSM
* Personal médico de la clínica de la UNMSM
* Representantes estudiantiles
* Coordinadores académicos

## **Actividades en la fase de diseño**

1. Análisis detallado de requerimientos
2. Diseño de la arquitectura del sistema
3. Creación de prototipos de interfaz
4. Definición de flujos de usuarios
5. Establecimiento de estándares de diseño
6. Validación con stakeholders
7. Documentación técnica
8. Planificación de pruebas

## **Análisis de requerimientos**

## **Reconocimiento y evaluación de problemas**

Problemas detectados:

1. Gestión manual ineficiente de citas.
2. Largos tiempos de espera.
3. Falta de seguimiento de citas.
4. Comunicación limitada entre pacientes y personal médico.
5. Dificultad en la gestión de horarios médicos.

Soluciones propuestas:

1. Implementación de un sistema automatizado de reservas.
2. Notificaciones en tiempo real para recordatorios y actualizaciones.
3. Sistema de mensajería integrado entre pacientes y personal médico.
4. Calendario dinámico para la gestión eficiente de horarios médicos.

## **Historias de Usuarios**

|  |  |
| --- | --- |
| **Nombre del Proyecto** | Curita San Marcos |
| **Fecha de Preparación** | 22/01/2025 |
| **Versión** | 1.0 |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 1: INICIAR SESIÓN** | |
| **Descripción de la historia** | |
| Como usuario (estudiante, profesor o externo), quiero iniciar sesión con mi correo electrónico y contraseña para acceder a mis funcionalidades personalizadas dentro del sistema. | |
| **Identificador:** | HU001 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | Jocelyn Estrella Sotelo Arce |
| **Requerimiento funcional** | |
| * El sistema debe validar el correo y la contraseña ingresados contra la base de datos. * En caso de error en las credenciales, debe mostrar un mensaje de error: "Contraseña o correo incorrectos". * Implementar opción de recuperación de contraseña mediante correo electrónico. | |
| **Observaciones** | |
| * Los correos deben estar en formato válido. * Los estudiantes y profesores deben usar correos del dominio UNMSM. | |
| **Bosquejos prototipos** | |
| * Pantalla de inicio de sesión con campos para correo y contraseña. * Botón para recuperar contraseña. | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 2: REGISTRO** | |
| **Descripción de la historia** | |
| Como nuevo usuario, quiero registrarme en el sistema indicando si soy estudiante, profesor o externo, para poder acceder a los servicios que ofrece el sistema. | |
| **Identificador:** | HU002 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | Jocelyn Estrella Sotelo Arce |
| **Requerimiento funcional** | |
| * El sistema debe permitir seleccionar el tipo de usuario: estudiante, profesor o externo. * **Estudiantes y profesores:** * Validar que el correo pertenezca al dominio UNMSM. * Verificar en la base de datos el código de estudiante o profesor asociado al correo. * Si la validación es exitosa, enviar un correo de confirmación para crear la cuenta. * **Externos** * Permitir el registro con cualquier correo válido. * Enviar un correo de confirmación para validar la cuenta. | |
| **Observaciones** | |
| * Si el correo o código no coinciden, mostrar el mensaje de error correspondiente. * Los estudiantes y profesores deben registrar su código asociado durante el proceso. | |
| **Bosquejos prototipos** | |
| * Pantalla de selección de tipo de usuario. * Formularios diferenciados para estudiantes, profesores y externos. * Botón de envío con mensaje de validación posterior. | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 3: AGREGAR A CARRITO** | |
| **Descripción de la historia** | |
| Como usuario (estudiante, profesor o externo), quiero poder agregar productos o servicios al carrito de compras para guardarlos temporalmente antes de finalizar la reserva. | |
| **Identificador:** | HU003 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | fiorella patricia mirano surquislla |
| **Requerimiento funcional** | |
| * El sistema debe permitir a los usuarios (estudiantes, profesores o externos) seleccionar y agregar una cita médica al carrito antes de confirmar la reserva. * Debe validar la disponibilidad del médico y la franja horaria antes de agregar la cita. * En caso de que la franja horaria ya esté ocupada, debe mostrar un mensaje de error: *"Esta franja horaria ya no está disponible. Por favor, selecciona otra."* * Implementar opción para visualizar las citas agregadas al carrito antes de proceder a la confirmación de la reserva. | |
| **Observaciones** | |
| * Los usuarios pueden agregar múltiples citas al carrito antes de completar el pago. * Las citas en el carrito deben mantenerse guardadas en la sesión del usuario hasta que se complete el pago o el usuario las elimine. | |
| **Bosquejos prototipos** | |
| * Pantalla de selección de especialidad y médico con opción de elegir fecha y hora. * Botón *"Agregar cita al carrito"*. * Vista del carrito con lista de citas agregadas, información del médico, especialidad, fecha, hora y precio del servicio. * Botón para proceder con el pago y confirmar la reserva. | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 4: RESERVAR CITA** | |
| **Descripción de la historia** | |
| Los usuarios de la aplicación (estudiantes, personal administrativo o externos) podrán reservar citas médicas en la clínica universitaria. Para hacerlo, deberán seleccionar la especialidad médica, el médico de su preferencia, y un horario disponible. La aplicación mostrará esta información de manera sencilla, permitiendo confirmar la reserva con facilidad. Una vez confirmada, se enviará un correo con los detalles de la cita tanto al usuario como al personal administrativo de la clínica. | |
| **Identificador:** | HU004 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | Jatziry Fernanda Sanchez Wong |
| **Requerimiento funcional** | |
| Este requerimiento funcional tiene como objetivo implementar una funcionalidad dentro de la aplicación que permita al usuario:   * Seleccionar una especialidad médica. * Escoger un médico de la lista de profesionales disponibles según la especialidad seleccionada. * Elegir una fecha y horario que estén disponibles. * Confirmar la cita y agregarla al carrito para proceder al pago (H006 y H007).   Una vez completada la reserva, la cita se registrará en la base de datos y se notificará tanto al usuario como al personal administrativo por correo electrónico. | |
| **Observaciones** | |
| * Se debe validar que los horarios de los médicos no tengan conflictos y que la información esté actualizada en tiempo real. * Es importante mostrar mensajes claros en caso de errores (como horarios ocupados o especialidades sin disponibilidad). * Al finalizar el sprint, se realizarán reuniones con el cliente para evaluar posibles ajustes o mejoras en la funcionalidad. | |
| **Bosquejos prototipos** | |
| **Pantalla de selección de especialidad médica:**  Diseño limpio y organizado para que el usuario pueda explorar las especialidades disponibles.  Botones para seleccionar un médico y sus horarios.  **Pantalla de confirmación de cita:**  Resumen detallado de la especialidad, médico, horario y costo de la cita.  Botón para confirmar y agregar al carrito.  **Pantalla de carrito de citas:**  Lista de citas agregadas.  Opción de modificar o proceder al pago. | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 5: VER ATENCIÓN DISPONIBLE** | |
| **Descripción de la historia** | |
| Los usuarios de la aplicación (estudiantes, personal administrativo o externos) podrán consultar la disponibilidad de atención médica en la clínica universitaria.  El sistema permitirá buscar doctores y horarios de atención filtrando por especialidad médica. Los horarios serán **fijos y predeterminados** con doctores asignados a cada uno. Si un horario ya está tomado, se mostrará un mensaje indicando su indisponibilidad.  La información se presentará de manera clara y organizada, permitiendo a los usuarios identificar rápidamente las opciones disponibles. | |
| **Identificador:** | HU005 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | Jose Alessandro Quispe Cabello |
| **Requerimiento funcional** | |
| Este requerimiento funcional tiene como objetivo implementar una funcionalidad dentro de la aplicación que permita al usuario:   * Filtrar doctores disponibles por especialidad médica. * **Ver la lista de doctores** con sus horarios **fijos** de atención disponibles. * **Mostrar un mensaje si un horario ya está ocupado** para evitar conflictos. | |
| **Observaciones** | |
| * La información debe actualizarse automáticamente si un horario se ocupa o un doctor no está disponible. * Es necesario ofrecer una experiencia de navegación intuitiva y rápida para el usuario. * Se debe validar que las especialidades seleccionadas cuenten con doctores disponibles. * Mostrar mensajes claros en caso de errores (como especialidades sin doctores disponibles o problemas de conexión). | |
| **Bosquejos prototipos** | |
| **Pantalla de selección de especialidad médica:**  Diseño limpio y organizado con una lista o botones que permitan seleccionar la especialidad médica.  Un campo de búsqueda para encontrar especialidades específicas rápidamente.  **Pantalla de lista de doctores y hora disponible:**  Tabla o lista que muestre:   * Nombre del doctor. * Calendario o lista detallada con los horarios disponibles del doctor seleccionado. | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 6: VER CITAS RESERVADAS** | |
| **Descripción de la historia** | |
| Los usuarios de la aplicación (estudiantes, personal administrativo o externos) podrán visualizar las citas médicas que han reservado en la clínica universitaria. La información incluirá el detalle de la cita: especialidad médica, médico, fecha, hora y estado de la cita. Además, podrán cancelar o modificar una cita según las políticas establecidas por la clínica. | |
| **Identificador:** | HU006 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | Jose Alessandro Quispe Cabello |
| **Requerimiento funcional** | |
| Este requerimiento funcional tiene como objetivo implementar una funcionalidad dentro de la aplicación que permita al usuario:   * Visualizar una lista de todas las citas reservadas, organizadas por fecha. * Consultar el detalle de cada cita, incluyendo: * Especialidad médica. * Médico asignado. * Fecha y hora de la cita. * Estado de la cita (confirmada, cancelada, en espera, etc.). * Cancelar una cita desde la misma pantalla, si está permitido. * Modificar una cita (redireccionando al flujo de reserva de citas con datos prellenados). | |
| **Observaciones** | |
| * Las citas deben mostrarse en un formato cronológico, con las más próximas primero. * En caso de no tener citas reservadas, mostrar un mensaje claro, como: "No tienes citas reservadas". * Se debe incluir un botón para regresar al menú principal o iniciar una nueva reserva. * Mostrar notificaciones en tiempo real si una cita cambia de estado (por ejemplo, cancelada por el médico). | |
| **Bosquejos prototipos** | |
| **Pantalla de lista de citas reservadas:**  Diseño con una lista cronológica de citas.  Cada cita debe incluir:   * Especialidad médica. * Nombre del médico. * Fecha y hora. * Botones para "Ver Detalle", "Modificar" o "Cancelar".     **Pantalla de detalle de cita:**  Información completa de la cita:   * Especialidad médica. * Médico. * Fecha y hora. * Estado de la cita. * Ubicación en la clínica.   Botones para "Modificar cita" o "Cancelar cita". | |

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO 7: CONFIRMACIÓN DE CITA** | |
| **Descripción de la historia** | |
| El sistema permitirá a los usuarios realizar la confirmación de sus citas médicas y recibir una confirmación por correo electrónico con los detalles de la cita. | |
| **Identificador:** | HU007 |
| **Actor:** | Estudiante/Profesor/Externo |
| **Prioridad:** | Alta |
| **Sprint:** |  |
| **Responsable:** | [Rogger Kevin Neira Carquin](mailto:roggerneira@bcp.com.pe) |
| **Requerimiento funcional** | |
| * Generar un comprobante de la reserva médica. * Enviar correo electrónico de confirmación con: * Detalles de la cita * Código o número de reserva * Actualizar el estado de la cita a "Reservada" | |
| **Observaciones** | |
| * Manejar posibles errores de reservas * Registro de reserva en base de datos * Notificación en caso de reserva exitosa o fallida * Mostrar notificaciones en tiempo real si una cita cambia de estado (por ejemplo, cancelada por el médico). | |
| **Bosquejos prototipos** | |
| **Flujo de reserva:**   1. Seleccionar cita pendiente de confirmación 2. Mostrar datos de la reserva 3. Editar en caso se necesite 4. Confirmar reserva 5. Enviar correo electrónico     **Validaciones:**   1. Verificar disponibilidad de la cita 2. Comprobar que el usuario tenga no tenga una cita previa en el mismo horario. 3. Controlar que no se pueda reservar en una fecha ya separada por otro usuario.     **Consideraciones técnicas:**   1. Generación de comprobantes digitales 2. Sistema de envío de correos electrónicos | |

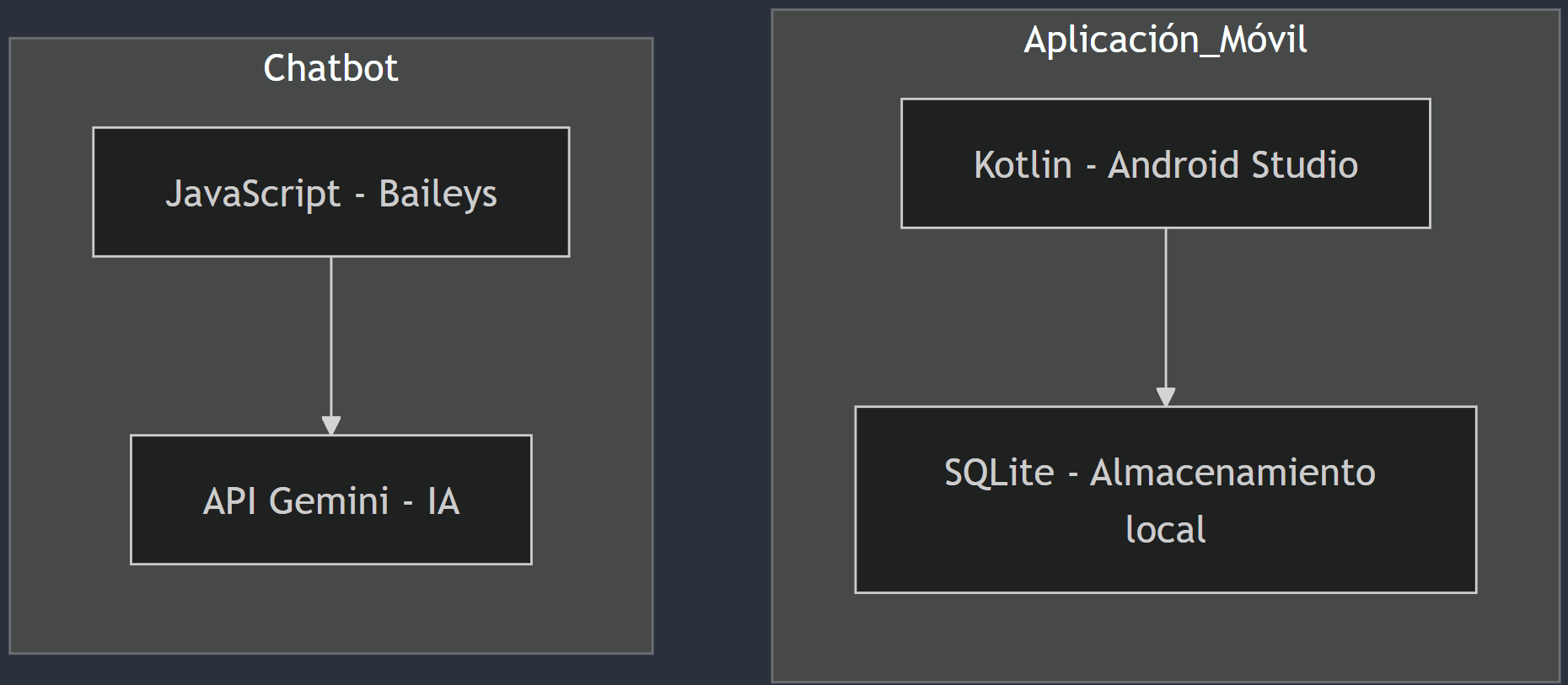
## **Arquitectura del proyecto**

En este bloque se describe la arquitectura de la aplicación móvil y del chatbot, incluyendo las tecnologías utilizadas, la estructura de software y los componentes principales del sistema.

### **Arquitectura de tecnologías**

El proyecto está compuesto por dos sistemas principales:

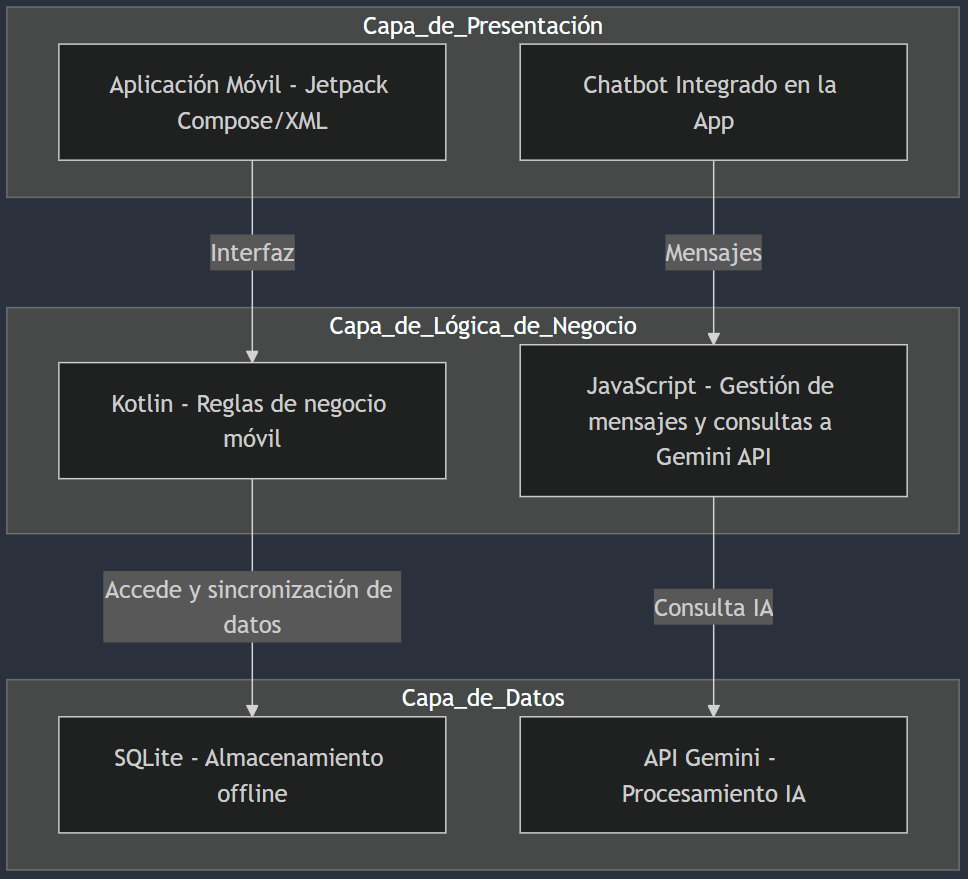
* Aplicación Móvil: Desarrollada en Kotlin con Andorid Studio y utiliza SQLite como base de datos local.
* Chatbot: Implementada en JavaScript con la única API “Gemini” para proporcionar recomendaciones mediante IA. Se ejecuta en un entorno local, por lo que está implementada en la misma app.



### **Arquitectura de software**

El sistema sigue una arquitectura basada en capas:

* **Capa de Presentación:**
  + Apliación móvil en Android con una interfaz basada en Jetpack Compose o XML
  + Chatbot integrada en la App interactuando con el usuario.
* **Capa de Lógica de Negocio:**
  + Implementación de reglas y procesos en Kotlin para la aplicación móvil.
  + Módulos en JavaScript para la gestión de mensajes y consultas a la API de Gemini.
* **Capa de datos:**
  + SQLite para almacenamiento de datos offline en la app.
  + API de Gemini para el procesamiento de IA en el Chatbot.



### **Diagrama de Componentes**

**Aplicación Móvil:**

* Registro e inicio de sesión de usuarios.
* Gestión de citas médicas (crear, modificar, eliminar y visualizar citas)
* Sincronización de datos con SQLite.
* Recomendaciones personalizadas mediante IA Gemini.

**Chatbot:**

* Interacción con usuarios directamente en la app.
* Reserva de citas médicas.
* Consulta de citas pendientes.
* Recomendación de especialidades médicas basadas en IA.
* Interfaz de usuario gráfica

  El contenido generado por IA puede ser incorrecto.

## **Prototipo de Proyecto**

El prototipo incluirá las siguientes pantallas principales creadas en Figma:

1. Inicio/Login

* Autentificación de usuarios
* Registro de nuevos usuarios

1. Reserva de citas:

* Selección de especialidad
* Calendario de disponibilidad
* Confirmación de reserva

1. Carrito de compras:

* Visualización de reservas

Link de prototipos: [Prototipo G05](https://www.figma.com/design/oQExdPuAOgsTn7CLDdSCUO/Curita?node-id=0-1&t=AXZru9LmLrNYKkaT-1)

# **Capítulo IV**

# **Fase de Ejecución**

En el capítulo anterior, se definieron los aspectos clave del diseño y la planificación del sistema, estableciendo la estructura y funcionalidades que se implementarían. En esta fase, se lleva a cabo la ejecución de esas definiciones, transformando los conceptos en una plataforma operativa.

El desarrollo del sistema se enfoca en la integración de los diferentes módulos, la configuración de la base de datos y la implementación de las funciones principales. Se emplean herramientas que facilitan el almacenamiento de información, el control de versiones y la programación de los distintos componentes, asegurando la correcta interacción entre ellos.

## **Objetivos de la fase**

El objetivo de esta fase es materializar el diseño y la planificación en un sistema funcional, implementando los módulos esenciales y asegurando su correcto funcionamiento. Se busca integrar las diferentes herramientas y tecnologías necesarias para garantizar la operatividad del sistema, así como establecer una base sólida que permita su evolución y mejora en etapas posteriores.

## **Herramientas involucradas**

Para la implementación del sistema **Curita San Marcos**, se emplean diversas herramientas tecnológicas que garantizan un desarrollo eficiente, seguro y escalable. Estas herramientas permiten la integración de los distintos módulos del sistema, la administración de datos en tiempo real y la optimización del rendimiento de la aplicación.

1. **GitHub**

GitHub se emplea como plataforma de control de versiones para gestionar el código fuente del proyecto de manera organizada y colaborativa. Su uso permite un seguimiento detallado de los cambios en el desarrollo del software, asegurando un historial estructurado y facilitando la colaboración entre los miembros del equipo.

Dentro del proyecto, GitHub es utilizado para:

* + Gestionar el código mediante un sistema de control de versiones distribuido, asegurando la trazabilidad de cada modificación.
  + Facilitar la integración de nuevas funcionalidades mediante ramas de desarrollo, evitando conflictos en el código principal.
  + Implementar estrategias de integración y entrega continua (CI/CD) para optimizar el proceso de desarrollo.
  + Permitir la documentación del código y la gestión de incidencias a través de GitHub Issues, facilitando la resolución de problemas y el mantenimiento del software.

1. **Firebase**

Firebase es un servicio en la nube utilizado en el proyecto para la administración de datos y la autenticación de usuarios. Su integración permite el acceso rápido y seguro a la información, optimizando la experiencia de los usuarios en la gestión de citas médicas.

Las funcionalidades clave de Firebase en Curita San Marcos incluyen:

* Base de datos en tiempo real para almacenar y sincronizar información sobre citas médicas, usuarios y disponibilidad de médicos.
* Firebase Authentication para la validación y gestión de accesos mediante correo electrónico, Google Sign-In y otros métodos seguros.
* Cloud Functions para la automatización de procesos internos, como la confirmación y actualización de citas.

1. **Kotlin**

Kotlin es el lenguaje principal utilizado en la implementación de la aplicación móvil de Curita San Marcos. Su selección se debe a su compatibilidad con Android, su sintaxis moderna y su capacidad para mejorar el rendimiento de la aplicación.

Dentro del proyecto, Kotlin permite:

* Desarrollar una interfaz intuitiva y amigable para los usuarios, siguiendo los principios de Material Design para garantizar una experiencia fluida.
* Integrar Firebase de manera eficiente, permitiendo la autenticación y el acceso en tiempo real a la información almacenada en la nube.
* Implementar la lógica de negocio relacionada con la gestión de citas médicas, la visualización de disponibilidad y el historial de consultas de los usuarios.
* Optimizar la seguridad y estabilidad de la aplicación mediante el uso de buenas prácticas en el manejo de datos y la gestión de sesiones de usuario.

1. **Java**

Java se utiliza en el desarrollo de funciones avanzadas, especialmente en la implementación del chatbot con inteligencia artificial. Este componente del sistema está diseñado para asistir a los usuarios en la gestión de citas médicas mediante una interfaz conversacional que optimiza la interacción con el sistema.

El chatbot basado en Java permite:

* Responder preguntas frecuentes de los usuarios relacionadas con el proceso de reserva, disponibilidad de médicos y costos de citas.
* Automatizar la programación de citas médicas mediante comandos de texto o selección guiada.
* Implementar procesamiento de lenguaje natural (NLP) para comprender y responder de manera más eficiente a las consultas de los usuarios.
* Integrar servicios de mensajería para mejorar la accesibilidad de la aplicación y reducir la carga de trabajo del personal administrativo de la clínica.

## **Codificación**

* + 1. **Definición de grupos**

Para optimizar el desarrollo del proyecto, se han conformado equipos según la experiencia de cada integrante. El Scrum Master coordina el equipo y la metodología ágil. Los Desarrolladores UI/UX diseñan la interfaz y mejoran la experiencia del usuario, mientras que los Desarrolladores Frontend implementan la interfaz y los Backend gestionan la lógica del servidor y la base de datos. Los Full Stack integran ambos desarrollos, y los Diseñadores UX optimizan la usabilidad. Esta estructura permite una ejecución eficiente y alineada con los objetivos del proyecto.

* + 1. **Definición de entregables**
       1. **Entregable 1: Home**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | Home - Vista principal | | |
| **Descripción** | La pantalla Home es la vista principal de la aplicación y debe ser intuitiva y accesible. Contiene opciones para el registro de usuarios y el inicio de sesión, asegurando una navegación clara desde el primer acceso. | | |
| **Fecha de Inicio** | 19/01/25 | **Fecha de Fin** | 03/02/25 |
| **Actividades** | | | |
| Elección de librerías requeridas | | 19/01/25 | 20/01/25 |
| Programación del Home | | 20/01/23 | 03/02/23 |

* + - 1. **Entregable 2: Módulo de Registro e Inicio de Sesión**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | **Registro e Inicio de Sesión** | | |
| **Descripción** | Este módulo permite a los usuarios registrarse y autenticarse en la aplicación. Se validarán credenciales y se implementará recuperación de contraseña. | | |
| **Fecha de Inicio** | 19/01/25 | **Fecha de Fin** | 03/01/25 |
| **Actividades** | | | |
| Diseño del formulario de registro e inicio de sesión | | 19/01/25 | 20/01/25 |
| Implementación de validaciones y autenticación | | 20/01/23 | 24/01/23 |
| Pruebas de acceso y recuperación de contraseña | | 24/01/25 | 20/01/25 |

* + - 1. **Entregable 1: Módulo de Reserva de Citas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | Reserva de Citas | | |
| **Descripción** | Este módulo permite a los usuarios seleccionar una especialidad médica, elegir un médico y un horario disponible, así como confirmar su cita. | | |
| **Fecha de Inicio** | 19/01/25 | **Fecha de Fin** | 03/01/25 |
| **Actividades** | | | |
| Diseño de la interfaz de selección de especialidad y médico | | 19/01/25 | 20/01/25 |
| Implementación del sistema de reservas y validaciones de disponibilidad | | 20/01/23 | 03/01/23 |
| Pruebas de funcionalidad y ajustes | | 19/01/25 | 20/01/25 |

* + - 1. **Entregable 1: Módulo de Historial de Citas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | Módulo de Historial de Citas | | |
| **Descripción** | Permite a los usuarios visualizar sus citas pasadas y futuras, con detalles sobre el médico, la especialidad y la fecha. | | |
| **Fecha de Inicio** | 16/02/25 | **Fecha de Fin** | 26/02/25 |
| **Actividades** | | | |
| Diseño de la interfaz del historial de citas | | 16/02/23 | 20/02/23 |
| Implementación de consultas a la base de datos y filtrado de citas | | 20/02/23 | 26/02/23 |

* + - 1. **Entregable 1: Implementación de Chatbot con IA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | Home - Vista principal | | |
| **Descripción** | Integración de un asistente virtual para resolver dudas sobre disponibilidad de médicos, especialidades y horarios. | | |
| **Fecha de Inicio** | 16/02/25 | **Fecha de Fin** | 26/02/25 |
| **Actividades** | | | |
| Configuración de API para el chatbot | | 16/02/23 | 20/02/23 |
| Desarrollo de respuestas automatizadas y entrenamiento del modelo | | 20/02/23 | 26/02/23 |

* + 1. **Convenciones para el código**

Para garantizar la claridad y mantenibilidad del código, se establecen buenas prácticas en su estructuración, uso de nombres adecuados para variables y funciones, y una correcta indentación. Estas convenciones permiten que cualquier programador pueda comprender y reutilizar el código de manera eficiente.

* + - 1. **Nombre de variable y funciones**

El código debe seguir un estándar de nomenclatura que facilite su comprensión y evite ambigüedades. Para ello, se aplican las siguientes reglas**:**

|  |  |
| --- | --- |
|  | **RECOMENDACIÓN** |
| **Número de caracteres máximo** | 20 caracteres |
| **Signos que se pueden usar en caso de los nombres** | Se usa **CamelCase** para nombres de variables y funciones (nombreVariableEjemplo) |
| **Si el nombre está conformado por 2 o más palabras** | A partir de la segunda palabra inicial en mayúscula |
| **Para funciones usadas frecuentemente** | Se nombran con verbos en minúscula (cargarCitas(), eliminarCita()) |
| **Clases y objetos** | Se nombran con **PascalCase** (SaludActivity, CitasAdapter) |

**Fuente:** elaboración propia.

* + - 1. **Identación**

La correcta indentación y estructuración del código facilitan la legibilidad, mantenimiento y depuración del sistema. Se establecen las siguientes convenciones para garantizar un código limpio y ordenado:

***Reglas de indentación***

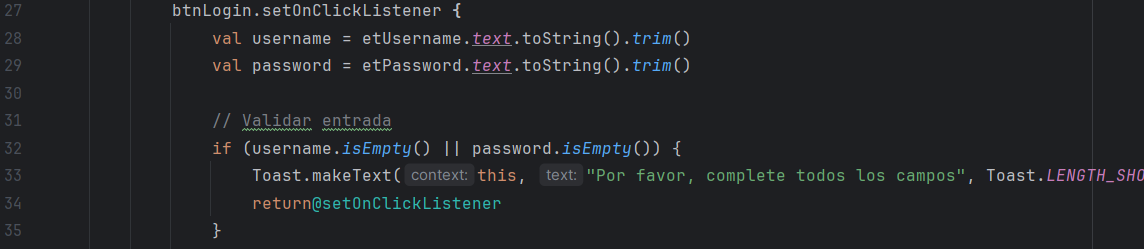
* Se utilizarán 4 espacios por nivel de anidación para mantener uniformidad y evitar el uso de tabuladores.
* Cada bloque de código dentro de estructuras de control (if, for, while, try, catch) debe estar correctamente indentado.
* Se recomienda dejar una línea en blanco entre métodos para mejorar la separación visual.

***Estructura general del código***

El código dentro de una clase debe seguir el siguiente orden lógico:

1. Declaración de paquetes e importaciones
2. Definición de la clase y variables de instancia
3. Métodos del ciclo de vida (onCreate, etc.)
4. Funciones de configuración de la UI y eventos (setOnClickListener, etc.)
5. Métodos auxiliares (operaciones con la base de datos, lógica de negocio, etc.)

***Ejemplificación de código implementado***



* + 1. **Documentación**

La documentación del código y del proyecto es fundamental para facilitar su mantenimiento, escalabilidad y comprensión por parte de otros desarrolladores. Se han establecido estándares de documentación en dos niveles: documentación en el código y documentación externa (README).

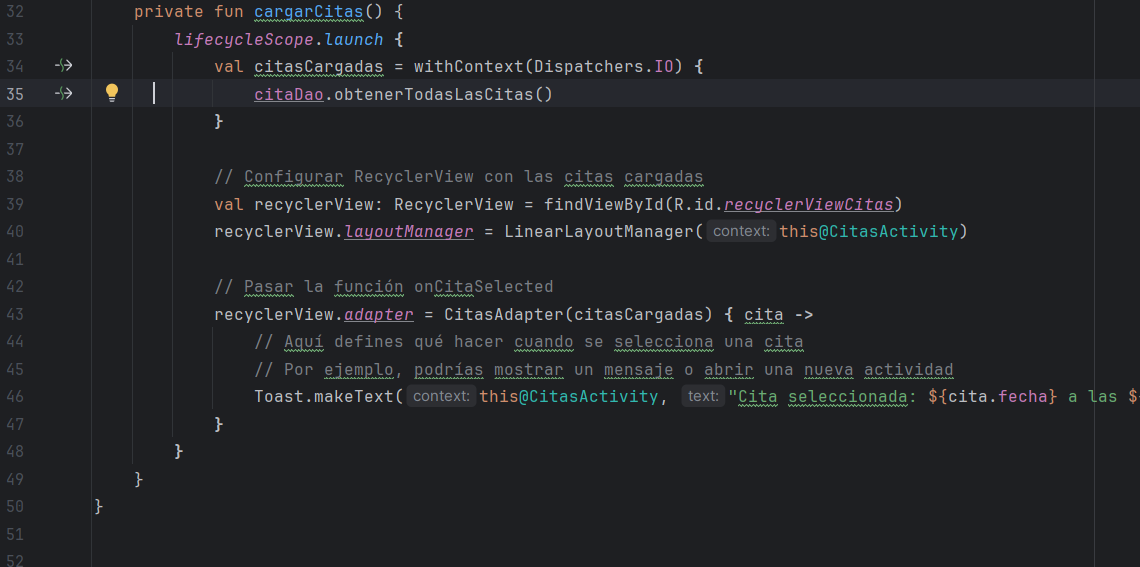
* + - 1. **Documentación en el código**

Cada clase, función y variable debe contar con una descripción clara de su propósito. Se recomienda agregar comentarios explicativos que indiquen qué hace cada bloque de código y, cuando sea necesario, especificar los parámetros de entrada y salida de una función.

Para mejorar la estructura y comprensión del código, se deben documentar los siguientes elementos:

* **Clases:** Cada clase debe incluir un comentario que explique su función dentro del sistema.
* **Métodos y funciones:** Deben tener una descripción clara que indique su propósito y, si aplicable, una lista de los parámetros que reciben y el valor que retornan.
* **Variables importantes:** Aquellas variables que contienen datos clave para el funcionamiento del sistema deben ser explicadas brevemente.
* **Bloques de código complejos:** Si un fragmento de código es difícil de entender a simple vista, se recomienda agregar comentarios para aclarar su lógica.

Además, se debe mantener una organización estructurada del código, evitando comentarios innecesarios o redundantes.



* + - 1. **Documentación externa (README)**

El README es el archivo principal de documentación externa del proyecto. Este archivo proporciona información clara y estructurada sobre la aplicación, facilitando su instalación, uso y mantenimiento.

La documentación en el README debe incluir los siguientes apartados:

* **Descripción del Proyecto:** Se debe indicar el propósito de la aplicación, en este caso, la gestión de citas médicas mediante una interfaz móvil intuitiva. También se pueden mencionar las tecnologías utilizadas, como Kotlin y Android Studio.
* **Características principales:** Se debe describir lo que el usuario puede hacer dentro de la aplicación, como iniciar sesión, programar citas médicas, cancelarlas y consultar el historial de citas. También se debe incluir información sobre el manejo de usuarios y la base de datos.
* **Estructura del Proyecto:** Es importante proporcionar un resumen de la organización del código, especificando qué archivos o módulos corresponden a cada funcionalidad. Se debe mencionar cuáles son las actividades principales, como la pantalla de inicio de sesión, el menú de opciones y los módulos de gestión de citas.
* **Base de Datos:** Se debe incluir una breve explicación de cómo se maneja la base de datos en la aplicación, indicando que se utiliza Room y mencionando las entidades principales, como usuarios y citas. También se deben mencionar los archivos que contienen la lógica de acceso y manipulación de datos.
* **Instalación y Configuración:** Se deben proporcionar los pasos necesarios para ejecutar el proyecto en un entorno de desarrollo. Esto incluye la clonación del repositorio, la apertura del proyecto en Android Studio y su ejecución en un dispositivo o emulador.
* **Requisitos del Sistema:** Se deben especificar las versiones mínimas necesarias del sistema operativo Android y las herramientas de desarrollo, como Android Studio y Kotlin.
* **Autor y Licencia:** Se debe indicar quién ha desarrollado el proyecto y bajo qué tipo de licencia se encuentra, en este caso, la licencia MIT, que permite la modificación y distribución libre del código.

Siguiendo estas prácticas, la documentación del proyecto facilitará su comprensión y permitirá que otros desarrolladores puedan contribuir y mejorar la aplicación en el futuro.

* + 1. **Refactorización**

La refactorización es un proceso esencial en el desarrollo de software que permite mejorar la estructura del código sin alterar su funcionalidad. En el contexto de este proyecto, se aplica para optimizar el rendimiento, reducir la duplicidad y mejorar la legibilidad del código. El enfoque utilizado en el desarrollo de la aplicación sigue las buenas prácticas establecidas en el Modelo Móvil Sprint, evitando problemas como métodos excesivamente largos, estructuras complejas y uso innecesario de parámetros.

Para garantizar un código limpio y eficiente, se han definido dos prácticas clave para la refactorización dentro de este proyecto.

* + - 1. **Comprobar la adaptabilidad**

Antes de realizar cualquier mejora o actualización en el código, es importante evaluar su estructura y adaptabilidad. Si el código es difícil de modificar o expandir, se recomienda realizar una refactorización previa para garantizar que los cambios puedan ser introducidos de manera eficiente y sin afectar otras funcionalidades del sistema.

Este enfoque ayuda a evitar errores inesperados y facilita futuras mejoras en la aplicación. Al aplicar esta estrategia, se logran beneficios en los siguientes aspectos:

* Mantenibilidad: Un código bien estructurado facilita su mantenimiento y permite corregir errores o agregar nuevas funcionalidades sin generar conflictos.
* Reducción de complejidad: Se minimiza el número de dependencias y estructuras innecesarias, haciendo que el código sea más eficiente y fácil de entender.
* Mejor comprensión: Se mejora la legibilidad del código, lo que permite que cualquier desarrollador pueda trabajar en él sin dificultad.
  + - 1. **Omitir el uso de controles y sentencias anidadas**

Para mejorar la claridad del código y evitar redundancias, se recomienda reducir el uso de sentencias anidadas en exceso. En lugar de utilizar múltiples estructuras de control anidadas, se deben aplicar técnicas que optimicen la lógica del programa, como el uso de funciones auxiliares, estructuras de control más compactas o patrones de diseño adecuados.

Por ejemplo, en lugar de utilizar múltiples condicionales anidados, se pueden emplear estrategias como:

* Uso de early return para reducir la profundidad de las estructuras de control.
* Aplicación de polimorfismo en lugar de múltiples sentencias condicionales.
* Organización del código en funciones más pequeñas y reutilizables.

Esto permitirá un código más claro, reduciendo la cantidad de líneas innecesarias y facilitando su mantenimiento a largo plazo.

# **Capítulo V**

# **Fase de Pruebas**

La fase de pruebas es una etapa fundamentar en el desarrollo de la aplicación “Curita San Marcos”, ya que permite garantizar su correcto funcionamiento antes de su despliegue final. Durante esta fase, se identifican y corrigen errores, se validan los requisitos funcionales y no funcionales y se comprueba la experiencia del usuario. El objetico principal es asegurar que la aplicación sea estable, segura y fácil de usar.

* 1. **Tipos de Pruebas**

Para garantizar la calidad del sistema, se han definido distintos tipos de pruebas a realizar:

* + 1. **Pruebas Funcionales**

En estas pruebas se podrá verificar cada módulo de la aplicación para asegurar que cumpla con los requisitos establecidos. Incluyen:

* Pruebas de unidad: Evaluación individual de cada componente o función.
* Pruebas de integración: Verificación del correcto funcionamiento entre módulos.
* Pruebas de sistema: Validación de la funcionalidad completa de la aplicación.
* Pruebas de aceptación: Revisión del cumplimiento de las necesidades del usuario.
  + 1. **Pruebas no Funcionales**

Estas pruebas evalúan características como rendimiento, seguridad y usabilidad:

* Pruebas de rendimiento: Miden la velocidad y eficiencia del sistema bajo diferentes condiciones de carga.
* Pruebas de usabilidad: Analizan la experiencia del usuario en la navegación y uso de la aplicación.
* Pruebas de compatibilidad: Verifican que la aplicación funcione correctamente en distintos dispositivos y versiones de Android.
* Pruebas de seguridad: Evalúan la protección de los datos y la resistencia ante posibles ataques informáticos.
  1. **Plan de Pruebas:**

Para la ejecución de las pruebas, se ha diseñado un plan estructurado que define los escenarios, casos de prueba y criterios de aceptación.

* + 1. **Escenarios de Prueba**
* Los principales escenarios para evaluar incluyen:
* Inicio de sesión y autenticación de usuarios.
* Registro de nuevos usuarios.
* Reserva y cancelación de citas médicas.
* Gestión de disponibilidad por parte del personal administrativo.
* Integración y respuestas del chatbot.
* Notificaciones y recordatorios de citas.
* Procesamiento de pagos.
* Consulta de historial de citas.
  + 1. **Casos de Prueba**

Se han definido casos de prueba detallados para cada funcionalidad. Un ejemplo se presenta en la siguiente tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **FUNCIONALIDAD** | **DESCRIPCIÓN** | **DATOS DE ENTRADA** | **RESULTADO ESPERADO** | **ESTADO** |
| TC01 | Inicio de sesión | Validar que el usuario pueda acceder con credenciales correctas | Correo y contraseña válidos | Acceso exitoso | Pendiente |
| TC02 | Registro de usuario | Comprobar que el registro se realice correctamente | Datos personales y correo válido | Cuenta creada | Pendiente |
| TC03 | Reserva de cita | Verificar que un usuario pueda agendar una cita | Especialidad, médico, fecha y hora | Cita reservada con confirmación | Pendiente |
| TC04 | Cancelación de cita | Comprobar que un usuario pueda cancelar una cita programada | ID de la cita | Cita cancelada exitosamente | Pendiente |
| TC05 | Notificación de cita | Verificar que el sistema envíe recordatorios | Fecha y hora de la cita | Notificación recibida por el usuario | Pendiente |
| C07 | Consulta de historial | Comprobar que el usuario pueda visualizar su historial de citas | ID de usuario | Lista de citas pasadas y futuras | Pendiente |
| TC08 | Chatbot de soporte | Evaluar la capacidad de respuesta del chatbot | Preguntas frecuentes | Respuestas correctas y útiles | Pendiente |
| TC09 | Compatibilidad en dispositivos | Probar la aplicación en diferentes modelos de teléfonos | Múltiples dispositivos y versiones de Android | Funcionamiento estable | Pendiente |
| TC10 | Seguridad y acceso | Validar que no se pueda acceder sin autenticación | Usuario no autenticado | Acceso denegado | Pendiente |

* + 1. **Herramientas de Prueba:**

Para la ejecución y automatización de pruebas, se utilizarán las siguientes herramientas:

* JUnit: Para pruebas unitarias en la aplicación móvil.
* Espresso: Para pruebas de UI en Android.
  1. **Resultados Esperados.**

Se espera que, al finalizar esta fase:

* Se detecten y corrijan errores críticos antes del lanzamiento.
* Se valide el cumplimiento de los requisitos funcionales y no funcionales.
* Se optimice la experiencia del usuario con base en los resultados de pruebas de usabilidad.
* Se garantice la estabilidad y seguridad de la aplicación en producción.

# CAPÍTULO VI: FASE DE LANZAMIENTO