

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

ANÁLISIS, MODELADO Y DISEÑO DE UNA APLICACIÓN mHEALTH COMO
APOYO EN EL AUTOCUIDADO EN PACIENTES RENALES CRÓNICOS.

INTEGRANTES:

NEFTHALY ABREGO

DEYBI MOJICA

ASESOR

DR. JUAN JOSÉ SALDAÑA

TRABAJO DE GRADUACIÓN

PARA OPTAR POR EL TÍTULO DE LICENCIATURA EN INGENIERÍA DE
SISTEMAS DE INFORMACIÓN

REPÚBLICA DE PANAMÁ

AÑO

2024

RESUMEN

El presente trabajo de investigación titulado "Análisis, modelado y diseño de una aplicación mHEALTH como apoyo en el autocuidado en pacientes renales crónicos." describe el desarrollo de un sistema software para apoyar a los pacientes en el registro de datos de su condición para facilitar el seguimiento del tratamiento. El objetivo principal de la investigación fue analizar, modelar y diseñar la aplicación mHealth con una arquitectura adecuada, una interfaz fácil de usar y alineado a los requerimientos críticos.

Para alcanzar este objetivo, se implementó la metodología de desarrollo de software Agile OpenUP que permite un ciclo de vida ágil e iterativo centrado en el usuario, esta metodología es flexible y completa. Se dedicaron iteraciones para cada etapa de la metodología las cuales permitieron mejorar la comprensión del contexto, levantamiento de arquitectura del sistema alineada al usuario y construcción. La arquitectura del sistema se diseñó utilizando modelos de UML para comprender el dominio de discurso del contexto de salud, de los pacientes que luego fueron abstraídos los conceptos para estructura el sistema. Al definir los modelos de entidades y relaciones se mapearon a los recursos de HL7 FHIR para permitir transformaciones de mapeos que permitan la interoperabilidad.

Esto permitió la implementación fácil utilizando tecnologías web como Laravel para un enfoque multiplataforma que permite entregar datos a la aplicación móvil y estructurar los modelos de datos a los formatos para la interoperabilidad de HL7.

El sistema software desarrollado permite a los usuarios registrar información para facilitar la gestión de horarios, medicación, tratamiento, consumos de líquidos, síntomas. Los resultados de las pruebas realizadas indican que el sistema es fácil de usar para los usuarios, valido para los procesos de registro de datos de la condición del paciente. El sistema software es funcional adecuado para los requerimientos y necesidades de registro del paciente.

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

Resumen	13
Dedicatoria.....	14
Agradecimientos	15
Índice General	16
Índice de Figuras	19
Índice de Tablas	20
Introducción	21
CAPÍTULO 1: GENERALIDADES DEL PROYECTO	22
1.1 Planteamiento del problema.....	22
1.2 Justificación.....	24
1.2.1 Relevancia de la investigación en el contexto de los Objetivos de Desarrollo Sostenible (ODS)	25
1.3 Objetivos	26
1.3.1 Objetivo general	26
1.3.2 Objetivos específicos	26
1.4 Alcance y limitaciones	26
1.5 Revisión de la literatura o antecedentes	27
1.5.1 Antecedentes nacionales.	27
1.5.2 Antecedentes internacionales	28
1.6 Preguntas de investigación	29
1.7 Definición de términos utilizados.....	29
Capítulo 2: Marco teórico y conceptual.....	30
2.1 Marco conceptual.....	30
2.1.1 Salud Digital y e-Health.....	30
2.1.2 mHealth: Salud Móvil	30
2.1.3 Tecnologías, plataformas, arquitecturas y herramientas de Desarrollo en el contexto de la salud.	30
2.1.4 HEALTHCARE DATA FORMATS AND STANDARDS	31
2.2 Marco teórico.....	32
2.2.1 Ingeniería de software.....	32
2.2.2 Ingeniería de requisitos	32

2.2.3	Metodologías de desarrollo de software.....	33
2.2.4	HL7 cMHAFf.....	33
2.2.5	Patrones arquitectónicos de FHIR.....	34
2.2.6	Mecanismos de transformación de formatos de datos	34
2.3	Diseño Centrado en el Usuario	34
2.4	Metodologías ágiles	34
CAPITULO 4: MARCO METODOLÓGICO		34
4.1	Materiales y métodos	34
4.2	Recopilación del conocimiento asociado al contexto del sistema.	34
4.2.1	HL7.....	36
4.2.2	Metodología de desarrollo de software OpenUP	38
4.2.3	Principios de OpenUP	39
4.2.4	Fases de OpenUP	41
4.2.5	RoadMap de OpenUP	45
4.2.6	Roles	45
4.2.7	Tareas organizadas por Disciplinas de OpenUP.....	47
4.2.8	Artefactos organizados por dominios	49
Capitulo 5: metodología de desarrollo de software.....		51
4.3	Incepción.....	51
4.3.1	Definición de la visión del proyecto	51
4.3.2	Definición de los requisitos de alto nivel de solución.....	52
4.3.3	Documento de especificación de requerimientos.....	52
4.3.4	Catálogo de requerimientos inicial	53
4.3.5	Establecimiento de fundamentos del proyecto.....	53
Estimación del tiempo y esfuerzo.		54
4.4	Desarrollo de la arquitectura candidata del sistema.....	62
4.5	Elaboración	72
4.5.1	Refinamiento de la declaración de la visión y requisitos	72
4.5.2	Documento de análisis del sistema software	76
4.5.3	Elaboración de documento de arquitectura del sistema.....	76
4.6	Construcción	79

4.6.1	Iteración I	79
4.6.2	Iteración II	80
4.7	Transición.....	81
CAPÍTULO 6: RESULTADOS.....		82
5.1	Componentes diseñados / Borrador.....	82
5.1.1	Modulo Interfaz de gestión de perfil	82
5.1.2	Módulo de registro de signos vitales	82
5.1.3	Modulo Interfaz de seguimiento de tratamiento.....	82
5.1.4	Submódulo gestión de turnos horarios de sesión.....	82
5.1.5	Submódulo registro de control de pesos	82
5.1.6	82	
5.1.7	Modulo Interfaz de registro de consumos	82
5.1.8	Submódulo registro de seguimiento de líquidos.....	82
5.1.9	Submódulo registro de seguimiento de alimentos.....	82
5.1.10	82	
5.1.11	Modulo Interfaz de registro de medicamentos e insumos ...	82
Conclusiones		14
Recomendaciones		14
Referencias bibliográficas.....		14
Anexos.....		17

ÍNDICE DE FIGURAS

*Ilustración 1: Indicar el título de la figura.....**¡Error! Marcador no definido.***

ÍNDICE DE TABLAS

Tabla 1.1: Título de la tabla.¡Error! Marcador no definido.

INTRODUCCIÓN

CAPÍTULO 1: GENERALIDADES

DEL PROYECTO

1.1 Planteamiento del problema

La enfermedad renal crónica (ERC) ha emergido como una de las preocupaciones globales de salud pública a nivel mundial, según (Kovesdy, 2022) se estima más del 10% de la población mundial afectando a más de 800 millones de personas en todo el mundo. Este problema de salud tiene una carga desproporcionadamente alta en los países de bajos y medianos ingresos. La ERC se ha convertido en una de las principales causas de mortalidad en todo el mundo ocupando la octava posición en la región Panamericana(Pan American Health Organization, 2021), según la Organización Panamericana de la Salud (PAHO) ha tenido un aumento significativo en las muertes asociadas en las últimas dos décadas.(Pan American Health Organization, 2021)

La enfermedad renal crónica (ERC) puede ser causada por diversas razones como la glomerulonefritis aguda, la hipertensión arterial, la diabetes y otras causas como enfermedades crónicas o autoinmunes, estilo de vida. Estas producen una pérdida progresiva de la función renal. (National Kidney Foundation, 2024)

La gestión adecuada de estas condiciones críticas de salud requiere una atención continua, seguimiento y adherencia a tratamientos para garantizar el bienestar del paciente. El empoderamiento del paciente es algo necesario para que adquiera el conocimiento, habilidades y la confianza para organizar y dar seguimiento constante a su condición.(Lightfoot et al., 2022)

En este contexto como menciona (Alwakeel & Lano, 2022) en la última década se han incrementado el desarrollo de aplicaciones enfocadas al cuidado de la salud(mHealth) que permiten apoyar el control, seguimiento y cuidado, lo que propone una oportunidad para mejorar el bienestar y seguimiento continuo de condiciones de salud en contextos específicos.

1.1.1 Enfermedad renal crónica en Panamá

La enfermedad renal crónica (ERC) es uno de los problemas de salud en Panamá, con una prevalencia del 12.6% estas asociadas a comorbilidades comunes como hipertensión y diabetes causante del 2.8% de muertes anuales en la población adulta.

Los pacientes con ERC enfrentan múltiples desafíos, incluyendo hospitalizaciones frecuentes debido a complicaciones durante su condición entre las más comunes infecciones, eventos cardiovasculares y tromboembolismos venosos en la investigación de (Rueda Borrero et al., 2023) realizada en Complejo Hospitalario Dr. Arnulfo Arias Madrid, afectado la calidad de vida del paciente.

Ante esta realidad, es crucial brindar apoyo a los pacientes renales para que puedan llevar a cabo un adecuado autocuidado y mejorar su calidad de vida. En este contexto, el desarrollo de una aplicación móvil que les proporcione de forma integral adecuada a los requerimientos propone una solución innovadora para apoyar a los pacientes en el seguimiento de su condición. Para ello es fundamental que el diseño de estas soluciones siga un enfoque centrado en el usuario, analizando, modelando las necesidades de información para garantizar así la funcionalidad, usabilidad, escalabilidad y de la aplicación.

Al proporcionar a los pacientes con ERC una herramienta práctica y accesible para el seguimiento de procesos de autocuidado, se busca empoderarlos en la gestión de su enfermedad, fomentando la adherencia al tratamiento, la prevención de complicaciones de su condición para mejorar la calidad de vida.

1.2 Justificación

El creciente uso de las tecnologías de información en el contexto de la salud ha traído múltiples herramientas de software que facilitan el control y apoyo en el cuidado y manejo de enfermedades, digitalización de los procesos clínicos, así como mejora en la atención de manera personalizada. Las aplicaciones mHealth son un subconjunto de e-Health que aprovecha los dispositivos móviles para gestionar la salud, haciéndolas una poderosa herramienta para apoyo a objetivos de salud.(mHealth: New horizons for health through mobile technologie | WHO | Regional Office for Africa, s/f). El ODS 3: "Garantizar una vida saludable y promover el bienestar para todos en todas las edades busca abordar la salud global, incluida la gestión de enfermedades crónicas, y proporciona un contexto relevante para la exploración de soluciones mHealth para la autogestión de pacientes en diferentes contextos.

Los dispositivos móviles pueden ser una poderosa herramienta ya que estos poseen elementos integradas como sensores, herramientas de accesibilidad y tecnologías de comunicación permiten un amplio panorama de casos de uso para propuestas de desarrollo que promuevan hacia el desarrollo de soluciones enfocadas en los objetivos de desarrollo sostenible. En el contexto de salud; como lo establece (Rowland et al., s/f) el uso de aplicaciones móviles apoyar los diagnósticos clínicos, mejorar el enfoque para guiar el comportamiento del paciente, la adherencia al tratamiento y su cumplimiento, así como también apoyo en la educación continua de su condición.

La salud a través de dispositivos móviles(mHealth) encuentran desafíos de diseño y adopción debido a la falta de características de usabilidad y accesibilidad. (Vélez Bastidas, 2023)

1.2.1 Relevancia de la investigación en el contexto de los Objetivos de Desarrollo Sostenible (ODS)

El proyecto se encuentra asociado con múltiples Objetivos de Desarrollo Sostenible (ODS), destacando especialmente el ODS 3, el cual se centra en asegurar una vida saludable y fomentar el bienestar para toda la población. Al proponer una nueva manera para mejorar la calidad de vida de los pacientes que sufren de enfermedades renales crónicas a través de la herramienta, se fomenta cambios positivos en el comportamiento así como la concientización de la importancia de la buena salud.(Naciones Unidas, s/f-e) También apoya el ODS 4, que busca garantizar una educación inclusiva, equitativa y de calidad, esto porque promueve a nuevas oportunidades de aprendizaje durante la vida, lo cual es esencial para cambiar la forma en la que el paciente da seguimiento a su condición.(Naciones Unidas, s/f-b)

Al proporcionar una herramienta de apoyo accesible y fácil de usar que se adapta a las necesidades específicas del contexto nacional y prioriza los requisitos del grupo social objetivo, se contribuye a la reducción de las desigualdades, como lo establece el ODS 10 que busca disminuir las brechas sociales que amenazan el desarrollo sostenible.(Naciones Unidas, s/f-d)

Además, este enfoque fortalece la sostenibilidad de las ciudades y comunidades, en línea con el ODS 11, al fomentar el autocuidado a través de una plataforma que brinda el servicio para el seguimiento de la condición, al permitir una gestión más efectiva de la enfermedad se empodera a los pacientes, se promueve un entorno más saludable y sostenible en el que las comunidades pueden prosperar mejorando como este grupo se desenvuelve.(Naciones Unidas, s/f-a)

Por otro lado, esta investigación también impulsa la innovación en la industria de la salud y la infraestructura, como lo establece el ODS 9.(Naciones Unidas, s/f-c) al modelar el sistema con estándares de interoperabilidad internacionales, se sientan las bases para la integración de datos con otras plataformas de salud en el futuro, lo que amplía aún más el impacto positivo de

esta iniciativa en la mejora de la atención médica y la calidad de vida de los pacientes.

1.3 Objetivos

1.3.1 Objetivo general

Analizar, modelar y diseñar una plataforma mHealth para el apoyo en el registro de datos de autocuidado en pacientes crónicos renales.

1.3.2 Objetivos específicos

- Evaluar e identificar la ontología o modelo E-R del contexto del software.
- Analizar las necesidades de información y registro de información del contexto.
- Analizar y recopilar los requerimientos funcionales y no funcionales de la aplicación.
- Establecer iteraciones para modelar y diseñar un prototipo mínimo funcional priorizando los requerimientos críticos.
- Modelar la estructura del sistema, bases de datos y arquitectura de despliegue.
- Diseñar el esquema de navegación e interfaces de usuario.
- Construcción del prototipo mínimo funcional aplicando iteraciones de construcción AgileOpenUp.

1.4 Alcance y limitaciones

El proyecto se centra en el proceso de desarrollo de una plataforma mHealth modelando, diseñando la arquitectura, sus funcionalidades priorizadas siguiendo la metodología Agile OpenUP para permitir el registro de datos de en el proceso de autocuidado de los pacientes con enfermedades renales crónicas.

Las limitaciones del proyecto son las restricciones de tiempo esto plantea el modelado y priorización de los requisitos críticos para el desarrollo de un prototipo mínimo que cumpla en funcionalidad que apoye a los procesos de autocuidado, necesidades de registro de datos que permitan una mejora en el seguimiento de adherencia al tratamiento.

Para ello se realizan diagramas UML para modelar arquitectónicamente el sistema para el registro de datos, estructuras de almacenamiento con modelado de datos.

1.5 Revisión de la literatura o antecedentes

Se han identificado proyectos similares a nivel nacional e internacional, lo que proporciona información relevante para evaluar propuestas similares en el diseño y desarrollo de la solución.

1.5.1 Antecedentes nacionales.

Según la revisión de literatura, a nivel nacional los investigadores (Villarreal & Samudio, s/f) establecieron una plataforma para la autogestión de datos de pacientes hipertensos. La plataforma permite a los pacientes controlar su enfermedad de manera más efectiva y recibir alertas y recomendaciones personalizadas. La plataforma está construida sobre el framework Laravel y utiliza el patrón de arquitectura Modelo-Vista-Controlador (MVC). Esta plataforma permite generar estadísticas del comportamiento de la hipertensión según diferentes indicadores, lo que es relevante para el área de salud y permite el desarrollo de planes de prevención.

Así mismo en el contexto de salud renal los investigadores de la Universidad de Panamá, Centro Regional de Veraguas (Duan et al., 2020; Gisela et al., 2024) desarrollaron una aplicación móvil llamada ACIL, para brindar una herramienta para el control de la ingesta de líquidos en personas con nefropatía crónica. La aplicación permite acceder de manera rápida y sencilla a información sobre la cantidad de líquidos consumidos de frutas y legumbres. Sin embargo, los

investigadores identificaron desafíos en la precisión del seguimiento debido al ingreso manuales del algoritmo afectando la confiabilidad, así como mayores la necesidad de pruebas de accesibilidad y usabilidad de la interfaz para una buena experiencia de usuario. En el proceso de desarrollo de la aplicación se levantaron los requerimientos, seguido de las fases de diseño y desarrollo. Se consideraron tanto los requisitos funcionales como los no funcionales, incluyendo aspectos como el rendimiento, la disponibilidad, la seguridad y la usabilidad. Entre los requerimientos principales se consideraron registro de la ingesta de líquidos para el seguimiento, establecimiento de metas personalizadas y recordatorios. ACIL se presenta como una herramienta prometedora para concientizar sobre el consumo de líquidos en personas con nefropatía crónica. Sin embargo, se recomienda realizar pruebas de usabilidad en diferentes regiones del país para adaptar la aplicación a las preferencias y necesidades de los usuarios en cada región.

1.5.2 Antecedentes internacionales

A nivel internacional en la investigación realizada por (Duan et al., 2020) se presenta el desarrollo de una aplicación móvil para mejorar el autocuidado del paciente con hipertensión. El estudio utilizó el diseño centrado en el usuario. Se realizaron entrevistas cualitativas y cuestionarios para recopilar información de los pacientes y se diseñaron seis módulos funcionales para la aplicación. La aplicación se sometió a pruebas de usabilidad y se evaluó en un estudio con 143 pacientes durante dos meses. Los resultados mostraron una mejora significativa en el control del paciente. El autor concluye que el uso de aplicaciones móviles para el autocuidado son una herramienta eficaz para mejorar la condición y adherencia al tratamiento.

Los datos obtenidos la investigación se basará en a través de, así como, encuestas a pacientes en el contexto de la propuesta y análisis de casos de satisfacción de la funcionalidad relevante en el ámbito de mHEALTH. La información recopilada se utilizará para fundamentar el diseño del prototipo y

evaluar su eficacia. La información recopilada se utilizará para fundamentar el diseño del prototipo y evaluar su eficacia.

1.6 Preguntas de investigación

¿Qué datos específicos, en formatos específicos, necesitan registrar los pacientes para realizar un seguimiento eficaz del autocuidado a través de mHealth?

¿Cuáles son las entidades clave y sus relaciones en el contexto para modelar y estructurar el sistema?

¿Qué procesos clave (seguimiento, registro de datos) se necesitan para dar soporte al autocuidado a través de mHealth?

¿Qué modelo arquitectónico debe tener el prototipo para abordar eficazmente las necesidades identificadas en el contexto del sistema?

1.7 Definición de términos utilizados

CAPÍTULO 2: MARCO TEÓRICO Y CONCEPTUAL

En este capítulo, se aborda el marco teórico necesario para fundamentar el desarrollo de la aplicación móvil centrada en la autogestión de pacientes con hemodiálisis y diálisis peritoneal. Se exploran diversos aspectos relacionados al contexto, al diseño, el desarrollo móvil, la seguridad y el diseño centrado en el usuario.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9073222/>

2.1 Marco conceptual

2.1.1 Salud Digital y e-Health

- Definición y alcance de la salud digital.
- Concepto de e-Health y su papel en la mejora de los servicios de salud.
- Aplicaciones de la e-Health en la autogestión de enfermedades crónicas como la insuficiencia renal y la diálisis.

2.1.2 mHealth: Salud Móvil

- Significado y aplicación de la mHealth en el contexto de la atención médica.
- Importancia de las aplicaciones móviles en el monitoreo y gestión de enfermedades crónicas.
- Rol de la mHealth en la facilitación de la autogestión de pacientes con hemodiálisis y diálisis peritoneal.

2.1.3 Tecnologías, plataformas, arquitecturas y herramientas de Desarrollo en el contexto de la salud.

- Descripción de las tecnologías utilizadas en el desarrollo de aplicaciones móviles.
- UML

- Frameworks y entornos de desarrollo más relevantes para el proyecto.
- Consideraciones sobre la accesibilidad móvil y su importancia para garantizar la inclusión de todos los usuarios.

2.1.4 HEALTHCARE DATA FORMATS AND STANDARDS

Los estándares

[Healthcare Data Formats | HealthShare Health Connect 2023.1 \(intersystems.com\)EE](https://healthcare.dataformats.org.uk/HealthShare/HealthConnect/2023.1/)

El principal objetivo de FHIR es solucionar la comunicación de sistema-a-sistema (B2B) y sistema-a-aplicación (B2C), sin hacer ninguna asunción sobre los sistemas. La parte B2B es básicamente la aproximación de la mensajería para el siglo 21; La parte de B2C supone la creación de unas APIs adecuadas para los programas y aborda una necesidad primordial – hacer sencillo construir aplicaciones modernas sanitarias. El reto está en que lo segundo claramente hace asunciones sobre la lógica de negocio y el contenido que incluyen los sistemas subyacentes.

La intención principal de openEHR es resolver el reto de los de salud de los pacientes – un historial de paciente longevo, versionado, distribuido y computable – y hacerlo de un modo robusto a los cambios futuros, exponiendo la semántica usando una arquitectura de plataforma. Vemos la interoperabilidad como una consecuencia técnica natural de una plataforma basada en una semántica formalizada. Por supuesto, openEHR en si misma solo propone algunos elementos de dicha plataforma – necesita ser usado en conjunción con terminologías, ontologías, bases de datos de medicamentos, interfaces de servicio y demás.

OpenEHR:

Es un enfoque en la interoperabilidad semántica que proporciona un modelo de datos altamente estructurado y semánticamente rico llamados arquetipos, lo que puede ser beneficioso para capturar datos clínicos detallados y específicos en el contexto de atención médica. Los arquetipos proporcionan

atributos con gran granularidad y flexibilidad lo que permite una representación detallada y personalizable de la información relevante para la gestión.

Este enfoque por arquetipos define un lenguaje de consulta denominado AQL(Architype Query Language) el cual permite para soportar las operaciones transaccionales para la gestión de datos, lo que significa que puede ser una opción sólida si la aplicación necesita mantener un registro extenso y duradero de los datos de salud de los pacientes.

HL7 FHIR

El estándar se basa en la búsqueda de la interoperabilidad técnica en el contexto médico que permite intercambiar datos de salud entre sistemas. Este estándar define recursos que son accedidos mediante estándares web como JSON y RESTful APIs, lo que facilita la integración con aplicaciones móviles y otros sistemas de TI de salud, puede ser utilizado para el modelado de datos a través del mapeo de los recursos a ontologías que representen las diferentes entidades, atributos y relaciones en un contexto. Es un estándar que está siendo ampliamente adoptado en la industria de la salud y lo respaldan organizaciones importantes, por lo que hay muchas herramientas y recursos disponibles para su implementación y uso.

Este estándar para el contexto de proyecto propuesto permite una estructura de modelamiento para representar la estructura interna del sistema que permite capturar y gestionar datos clínicos específicos y detallados sobre la condición del paciente, a diferencia de OpenEHR propone un enfoque de interoperabilidad semántica con mayor granularidad en los datos asociando además otros códigos y estándares como LOIC, SNOMED.

2.2 Marco teórico

2.2.1 Ingeniería de software

2.2.2 Ingeniería de requisitos

En el contexto del desarrollo de software, la ingeniería de requisitos (IR) se encarga de comprender las necesidades de los usuarios y stakeholders, y de traducirlas en especificaciones que puedan ser utilizadas por los equipos de

desarrollo para construir el software. La ingeniería de ontologías (IO) se centra en la creación y gestión de ontologías, que son representaciones formales de conocimiento conceptual dentro de un dominio específico. Las ontologías ofrecen una representación formal del conocimiento del dominio, lo que las convierte en una herramienta poderosa para la captura y gestión de requisitos en el ámbito de la salud. Las metodologías de IO, como METHONTOLOGY o NeOn, se basan en la construcción de ontologías para facilitar la comunicación entre los diferentes stakeholders y para asegurar la calidad de los requisitos.

Según (Otero Arrascue Danny Frank Asesor & Mejia Cabrera Heber Ivan, 2020), en su tesis relacionada a la comparación de mapas conceptuales basados en ontologías y UML para mejorar el proceso de elicitation y comprensión de requerimientos, concluyen que la identificación del dominio de la aplicación a través de el modelado de ontologías permite aumentar el valor del producto final, el proceso de modelado de ontología en el dominio de la aplicación permite la especificación formal de los conceptos abstractos clave, propiedades y relaciones entre entidades en un mismo contexto de la aplicación.

Una ontología permite definir los conceptos fundamentales de un área de conocimiento o contexto en particular, esto permite mejorar la comunicación y entendimiento entre los involucrados en el proyecto, permitir identificar conceptos y abstraer el conocimiento para reutilización de la información o datos, esto permite facilitar la interpretación común a través de un modelo estructurado de la información que permiten mejorar la calidad del software y la interoperabilidad de los sistemas.

<https://www.mdpi.com/2079-9292/10/9/1060>

<https://content.iospress.com/articles/applied-ontology/ao230278>

2.2.3 Metodologías de desarrollo de software

2.2.4 HL7 cMHAFF

https://www.hl7.org/implement/standards/product_brief.cfm?product_id=47

2.2.5 Patrones arquitectónicos de FHIR

[Patrones arquitectónicos FHIR. FHIR abre la puerta a un nuevo ... | por Steve Munini | Helios Software](#)

2.2.6 Mecanismos de transformación de formatos de datos

2.3 Diseño Centrado en el Usuario

- Principios del diseño centrado en el usuario y su relevancia en aplicaciones de salud.
- Métodos y técnicas para la investigación y diseño de experiencias de usuario en el contexto de la salud.
- Importancia de la usabilidad y la accesibilidad en la creación de una experiencia de usuario satisfactoria y efectiva.

2.4 Metodologías ágiles

CAPITULO 4: MARCO METODOLÓGICO

4.1 Materiales y métodos

4.2 Recopilación del conocimiento asociado al contexto del sistema.

Se aplica un modelado de los conceptos clave abstrayendo a entidades, atributos y relaciones que permiten diseñar la estructura del sistema propuesto, para la mejor comprensión del contexto operativo de aplicación, esto permite abstraer ideas principales para identificar las entidades y sus atributos que permiten modelar las relaciones para estructurar el modelo de almacenamiento, transformación para interoperabilidad y estructuras lógicas del sistema.

Recopilación de conocimiento

Para este proceso se evalúan documentos de requisitos en el contexto de la solución para identificar los conceptos involucrados.

Fuentes de conocimiento:

- Expertos técnicos en salud.
- Pacientes

Definición de entidades principales:

- **Paciente e historia clínica resumida:** Documentación que recoge información fundamental como antecedentes familiares. Es una lista enumerada de las condiciones o problemas de salud del paciente.
 - **Tratamiento:** Describe los datos que se generan en los procesos de atención al paciente, se requieren fechas de turnos en las que se producen las sesiones con posibles anotaciones por parte del paciente.
 - **Medicamento y dosis:** Describe la lista de medicamentos, horarios de medicación y dosis.
 - **Insumos médicos:** Describe todos los insumos requeridos por el paciente como cantidades de insumos que son utilizados para las sesiones del tratamiento.
 - **Síntomas:** Involucra el registro de datos para permitir el rastreo de síntomas debidos a complicaciones en el seguimiento del tratamiento o plan de medicación.
- Signos vitales:** Registros de signos vitales para el monitoreo y seguimiento.
- **Consumos y dieta:** Régimen alimentario adecuado para los pacientes del contexto, requieren el registro de consumo de alimentos y limitación de cantidades de líquidos.
 - **Recordatorios**
 - **Horarios**
 - **Laboratorios**

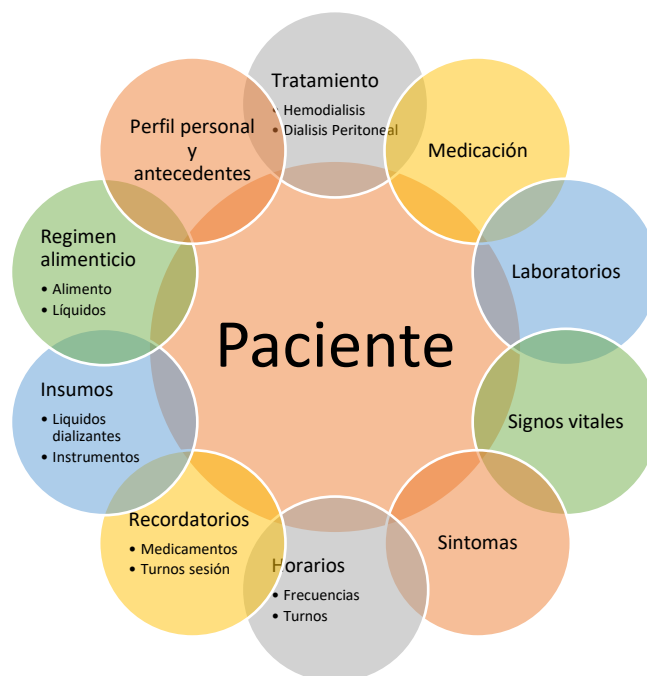


Figura n. Delimitación del contexto operativo del sistema.

4.2.1 HL7

Para el proceso de modelado de la arquitectura se aplica un enfoque para identificar los elementos considerados en el contexto del software propuesto, esto permite identificar entidades u objetos que permiten mapear entidades del dominio del sistema a recursos de HL7.

Figura n. Modelos de entidades y relaciones del dominio de discurso.

Para esta etapa se realiza un mapeo del modelo de la ontología a los recursos de HL7 relacionando las entidades, relaciones y atributos, esto facilita modelar los diagramas de clases con sus relaciones y estructura del almacenamiento del sistema con el estándar HL7.

Elemento de datos contexto paciente	Recurso HL7	Descripción
Paciente	Patient	Persona que desempeña el rol de paciente con sus datos de perfil
Médico o enfermera	Practitioner	Persona que desempeña el rol de médico, nefrólogo.

Familiar	Related Person	Persona que desempeña el rol de familiar del paciente
Signos vitales	Observation(Observación-vitalsignos: FHIR v6.0.0-cibuild)	Signos vitales como la presión arterial, la temperatura , pulso.
Laboratorios	Observation(Observación: FHIR v6.0.0-cibuild)	Resultados de pruebas de laboratorio relevantes para la ERC por ejemplo creatinina, potasio, fosforo, electrolitos.
Medicamento	Medicamento	Definir un medicamento como diuréticos, medicamentos para la presión arterial, etc.
Medicamento planificado	Solicitud de medicación	Prescripción de medicación para el paciente
Medicamentos tomados	Declaración de medicación	Lista de medicamentos actuales del paciente
Ubicación del paciente	Ubicación	Ubicación del paciente
Dieta	NutriciónOrden	Describir la dieta ordenada (p. ej., baja en sodio, baja en potasio)
Ingesta de líquidos	Observation	Cantidad de líquidos consumidos por el paciente
Salida de orina	Observation	Cantidad de orina excretada por el paciente
Historial social	Observation	Historial médico del paciente (p. ej., diabetes, hipertensión)
Síntoma del paciente	Condition	Síntomas persistentes del paciente que necesitan un manejo a largo plazo (p. ej., fatiga, hinchazón)
Resultado de la prueba de laboratorio	Observación	Resultados adicionales de pruebas de laboratorio relevantes (p. ej., hemoglobina, calcio, fósforo)
Cita	Cita	Visitas programadas con los proveedores de atención médica
Paciente-médico	Encuentro	Interacción entre un paciente y un proveedor de atención médica (p. ej., visita a la clínica, consulta de telemedicina)

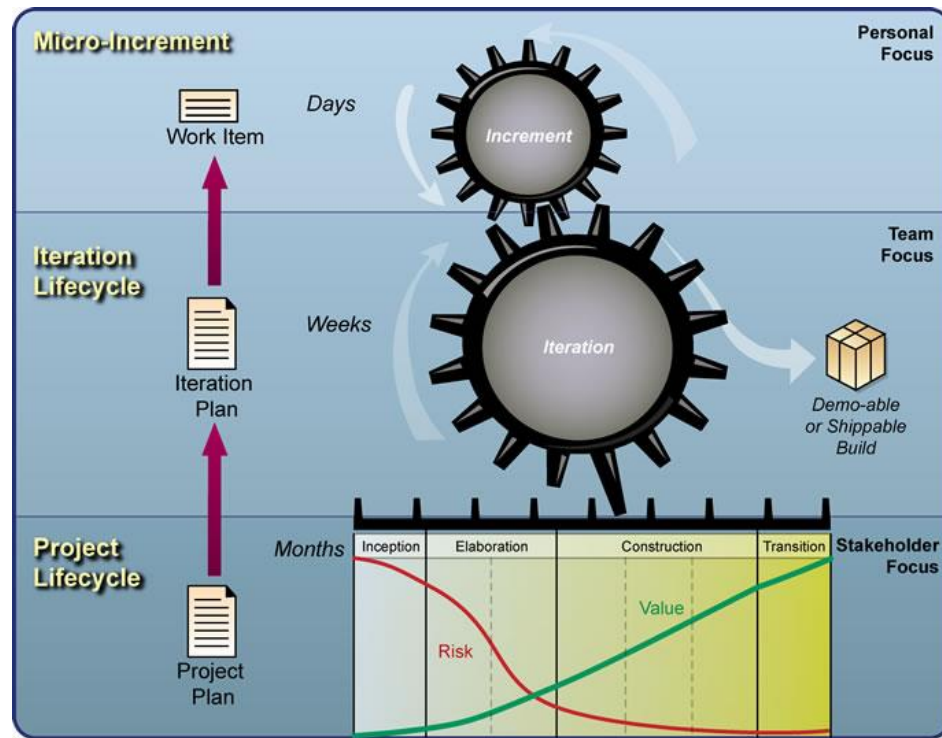
4.2.2 Metodología de desarrollo de software OpenUP

La metodología AgilOpenUp, parte del proyecto Eclipse Process Framework (EPF), es una versión simplificada y flexible del RUP, este involucra un proceso pragmático ágil con iteraciones incrementales dentro del ciclo de vida, tiene una naturaleza colaborativa y define roles basados en la filosofía ágil. Es un proceso de desarrollo de software iterativo con iteraciones de semanas, aplicando el concepto de micro incrementos como unidades de trabajo que aportan valor al progreso del proyecto permitiendo que el producto se acerque en cada iteración al valor esperado por los involucrados.

Esta se basa en aplicar solo el contenido fundamental para proyectos pequeños o medianos que ofrecen software como un producto principal, permitiendo ser extensible para agregar o adaptar procesos. Además, se adapta bien a los proyectos en los que las partes interesadas están involucradas en el proyecto significativamente.

Las disciplinas RUP, como el modelado de negocios, el entorno o la implementación, no se incluyen, ya que se consideraron como Partes adecuadas del proceso de desarrollo solo para proyectos grandes. Además, la abertura podría no ser un proceso adecuado para proyectos que requieran un alto nivel de formalidad, no incluyan a las partes interesadas como roles activos en el proceso y entregan otros artefactos que el software.

La abertura se adapta de RUP algunos elementos básicos, como: DENICIÓN DE LA ESTRUCTURA METHOLOGÍA (es decir, fases, roles, artefactos, tareas), iteración, principios de un enfoque incremental y centrado en la arquitectura. También considera la gerencia de riesgos a través de todo el proceso. OpenUP se une a muchas prácticas que ayudan a los equipos a ser más efectivos en el desarrollo del software, como: Filosofía ágil que se centra en la naturaleza colaborativa del desarrollo de software, la extensibilidad que permite que el proceso se extienda o adapte a las necesidades de un proyecto u Organización a través de elementos de contenido y proceso reutilizables identificados. También es un proceso de herramientas-agnóstico y de baja ceremonia que tiene una aplicación en una amplia variedad de tipos de proyectos.



4.2.3 Principios de OpenUP

Colaborar para sincronizar intereses y compartir conocimiento, este principio busca promover prácticas que impulsen a un ambiente de equipo saludable que faciliten la colaboración y desarrollo de conocimiento compartido del proyecto.

Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto, este principio busca aplicar prácticas que permiten a los participantes de los proyectos desarrollar una solución que maximice los beneficios obtenidos por los participantes y que cumplan con los requisitos y restricciones del proyecto.

Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo, esto permite definir una arquitectura candidata permitiendo ver desde temprano la viabilidad de las soluciones planteadas.

Desarrollo evolutivo para llevar a cabo retroalimentación y una mejora continua, promoviendo las prácticas que permiten a los equipos de desarrollo obtener retroalimentación temprana y continua de los participantes del proyecto,

permitiendo la entrega de demostraciones en incrementos progresivos de la funcionalidad.

Ciclo de Vida del proyecto (Project Life Cycle): El ciclo de vida en del proyecto en OpenUP se refiere a las fases o etapas que atraviesa un proyecto desde su inicio hasta su finalización, este permite tener un enfoque de amplia visibilidad de todo el proyecto.

Estas fases representan los diferentes estados del proyecto y pueden incluir etapas como la incepción, elaboración, construcción, transición. Cada fase del ciclo de vida tiene sus propios objetivos, actividades y entregables asociados que aportan valor al producto. El ciclo de vida en OpenUP proporciona una estructura general para el desarrollo y gestión del proyecto. En las primeras dos etapas se responden a preguntas para abordar los riesgos y retos de la solución tecnológica a desarrollar para el cliente, esto permite abordar estratégicamente los retos para crear valor en el producto a través de la correcta identificación y solución de estos en la dos últimas etapas en donde se construyen los componentes arquitectónicos.

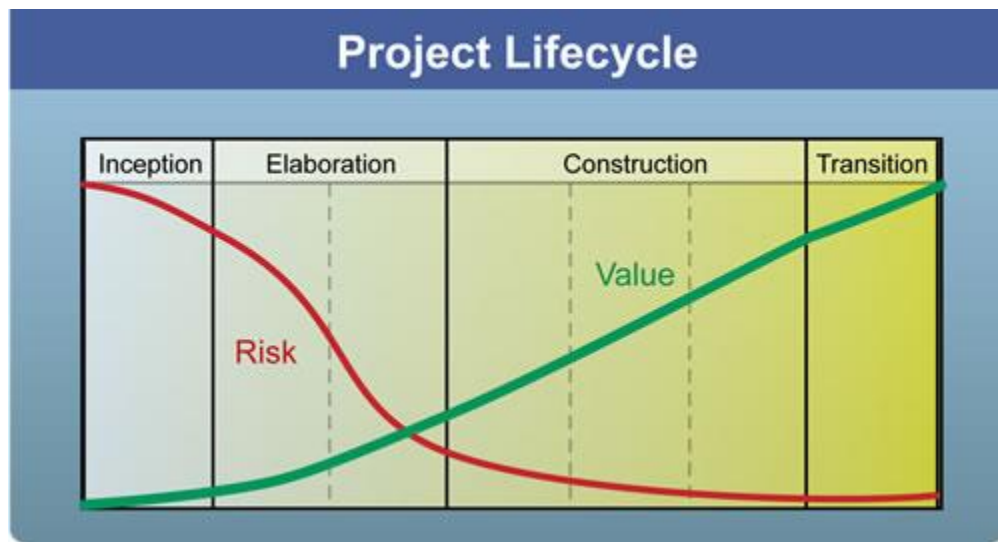


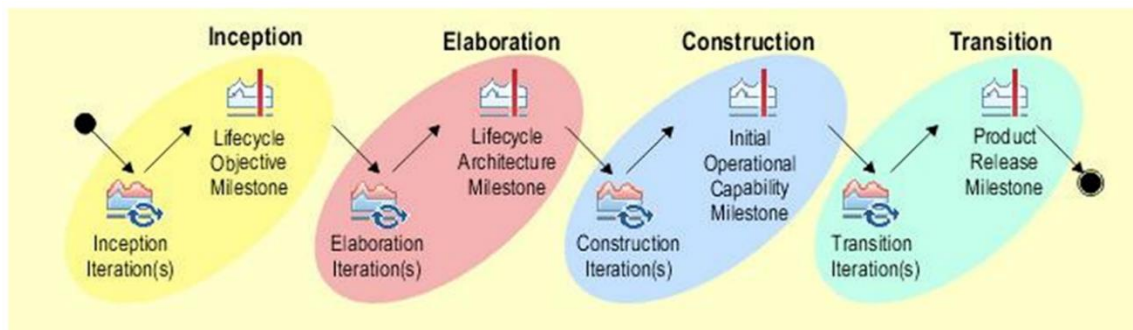
Figura: Reducción de riesgos (curva roja) y creación de valor (curva verde) durante el ciclo de vida del proyecto.

Iteration LifeCycle: El ciclo de vida de la iteración proporciona prácticas para el equipo que permiten aprovechar las iteraciones permitiendo que este se

enfoque en obtener y ofrecer valor de manera incremental a los interesados del proyecto.

Micro incrementos: Los micro incrementos contribuciones que permiten alcanzar objetivos en las iteraciones, pueden ser resultados del trabajo de horas o días en colaboración con el equipo para alcanzar metas. Por ejemplo, identificación de partes interesadas, definición de un documento de visión, el diseño, desarrollo e implementación de un caso de uso o escenario pueden dividirse en pequeñas tareas que pueden ser acordadas y resueltas en pocas horas dando como resultado contribuciones que alimentan al proyecto.

Estos micro incrementos bien definidos cambian los resultados de los documentos del proyecto haciendo que este incremente en valor y disminuyan los riesgos por la obtención de resultados en cada micro incremento.



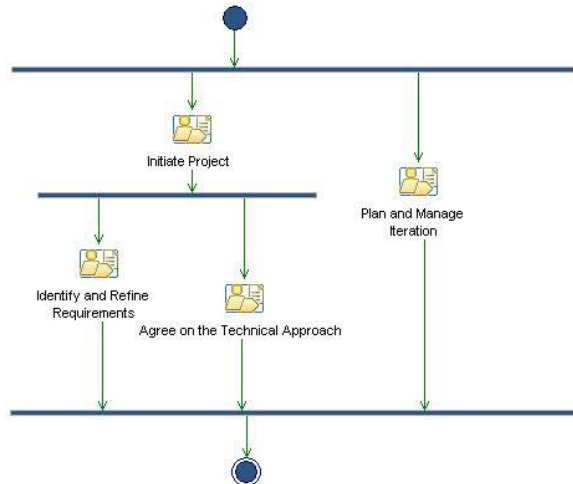
4.2.4 Fases de OpenUP

OpenUP divide el ciclo de vida del proyecto en cuatro fases principales: Inicio, Elaboración, Construcción y Transición. Cada fase tiene objetivos específicos y actividades asociadas. Dentro de cada fase, se realizan iteraciones cortas y frecuentes. Cada iteración produce un incremento funcional del producto.

Inception

La primera fase del proceso de desarrollo, donde las partes interesadas y los miembros del equipo colaboran para determinar el alcance y los objetivos del

proyecto. El propósito en esta fase es tomar la decisión de si el proyecto debe proceder. En esta fase, el alcance del proyecto debe aclararse, así como los objetivos del proyecto y la viabilidad de la solución prevista con sus límites.



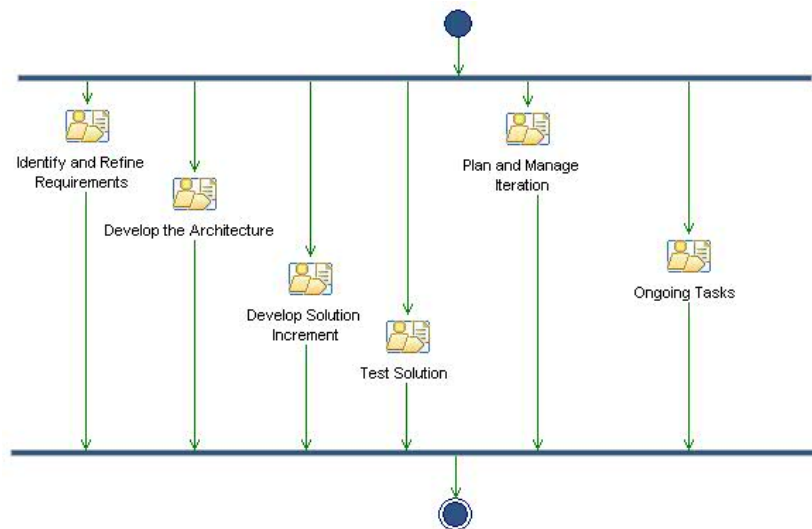
Al finalizar la última iteración de esta fase, debe estar claro quién está interesado en este sistema y por qué, cuáles son las funcionalidades clave del sistema y cuáles de estas son las más críticas. También se debe identificar una posible solución junto con una arquitectura candidata. Todo este primer proceso de definición de actividades ayuda al gerente del proyecto a estimar el costo del proyecto, planificar su cronograma y comprender los riesgos asociados.

Si existen más iteraciones en esta fase debido a las múltiples partes interesadas y nivel de complejidad en sus relaciones puede ser necesaria la creación de un prototipo o pruebas de concepto.

Elaboración

La segunda fase en el ciclo de vida del proyecto busca abordar los riesgos técnicos arquitectónicamente importantes esto a través la comprensión detallada de los requisitos, en particular los requisitos críticos, esto permite establecer una arquitectura como línea base para asegurar la calidad en el posterior esfuerzo, esta línea base de la arquitectura puede ser validada juntamente con los

involucrados permitiendo vislumbrar y disminuir los riesgos técnicos a través de pruebas y casos de uso críticos que permiten validar la solución y alinearla a las expectativas del usuario.



Durante esta fase respondemos a las cuestiones de adecuación de la arquitectura para desarrollar la aplicación, muchos riesgos técnicos se abordan como resultado de detallar los requisitos y de diseñar, implementar y probar la arquitectura candidata validándola con los interesados.

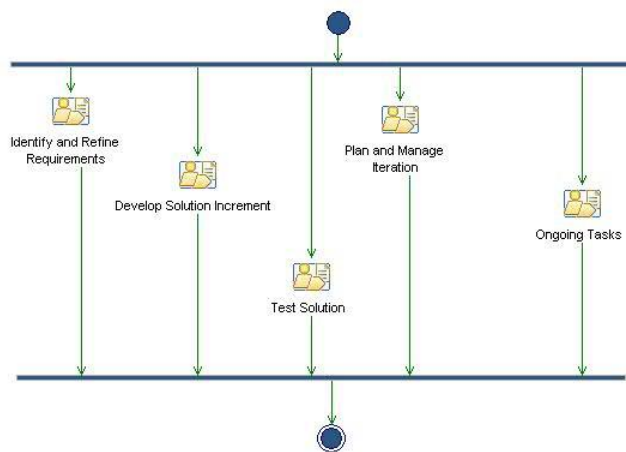
Los riesgos no técnicos involucran aspectos no relacionados con la tecnología, sino con cuestiones legales, regulatorias, recursos humanos, ambientales, culturales entre otras. Esto involucra por ejemplo licencias de uso de componentes en la arquitectura o de código abierto. La identificación de estos riesgos es esencial lo que nos permite producir un cronograma preciso de un plan de proyecto de alto nivel y estimaciones de costos conociendo la complejidad potencial del proyecto, los riesgos asociados, así como la tecnología implicada.

Cada iteración en esta fase contribuye a obtener una amplia comprensión del esfuerzo que va a ser necesario a través de validaciones con usuarios con la priorización de los escenarios de caso de uso críticos que cubre el proyecto, arreglos a la línea base tecnológica de la arquitectura de la solución, así como la solución a problemas no técnicos y los recursos o soluciones necesarias requeridos para mitigar los riesgos identificados como altos.

Construcción

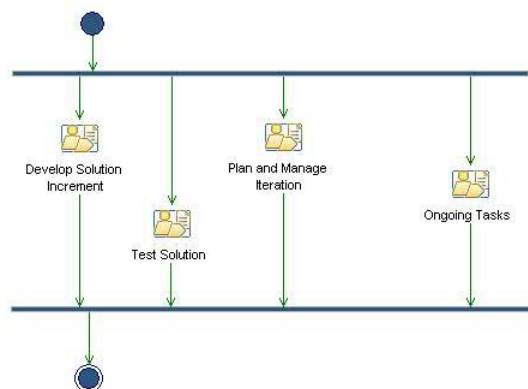
Esta es la tercera fase del proceso conteniendo mayores iteraciones que las anteriores. Se centra en el desarrollo funcional de los requisitos, diseños y arquitectura establecidas, así como la implementación y pruebas de las características funcionales del software.

El propósito en esta fase es desarrollar iterativamente el producto de tal manera que, en cada iteración se obtiene una versión más completa de características entorno a los requerimientos levantados que solucionan el problema identificado en el entorno del usuario.



El producto se desarrolla de manera iterativa, lo que significa que la versión final dependerá del número de iteraciones y la última iteración realizada es aquella que más se aproxima a la solución total esperada por el usuario.

Transición



Esta es la última fase del proceso, que se centra en la transición del software al entorno de los usuarios finales en donde las partes interesadas ven el desarrollo del producto completo y listo para la entrar en el contexto operativo de los usuarios. En esta se valida la funcionalidad, cumplimiento de expectativas, verificación de pruebas y errores que se realizan para la validación del producto.

4.2.5 RoadMap de OpenUP

OpenUP permite que los equipos apliquen un ciclo de vida de mucha interacción con reuniones regulares donde se abordan los riesgos y la mitigación a estos. Todo el trabajo desarrollado en las iteraciones se enumera, rastrea y asigna en listas de elementos de trabajo. Los miembros del equipo utilizan un mismo repositorio único para sincronizar los resultados de las tareas, así como las solicitudes de cambio en los requerimientos de usuario, la notificación de errores y otras solicitudes de las partes interesadas. De este repositorio se obtienen y clasifican los casos de uso para describir los requisitos con buen detalle, realizando validaciones con los interesados. Esto permite abordar los requisitos que afectan a la arquitectura de manera significativa haciendo que, una vez estos estén definidos, en la etapa de elaboración se aborden y estabilicen significativamente reduciendo los riesgos mayormente en esta etapa.

Durante la etapa de construcción se realizan varias pruebas por iteración que satisface la resolución de los requisitos del usuario, se reducen errores y hacen correcciones, estas modificaciones se integran a la base del código en el repositorio.

OpenUP describe un conjunto de roles que realizan tareas y producen o usan artefactos.

4.2.6 Roles

Estos son los Analistas, Arquitectos, Diseñadores, Desarrolladores, y Testers. Cada rol tiene responsabilidades específicas dentro del proyecto.

Partes (Stakeholders):	Interesadas	Grupos de interés cuyas necesidades deben ser satisfechas por
---------------------------	-------------	--

	el proyecto. Cualquier persona que esté materialmente afectada por los resultados del proyecto puede desempeñar este papel.
Gerente de Proyecto (Project Manager):	Lidera la planificación del proyecto en colaboración con los interesados y el equipo, coordina las interacciones con las partes interesadas y mantiene al equipo enfocado en los objetivos del proyecto.
Analista (Analyst):	Representa las preocupaciones del cliente y del usuario final al recopilar la opinión de las partes interesadas, comprendiendo el problema a resolver y estableciendo prioridades para los requisitos.
Arquitecto (Architect):	Responsable de diseñar la arquitectura de software, tomando decisiones técnicas clave que influyen en el diseño y la implementación del proyecto.
Desarrollador (Developer):	Responsable de desarrollar una parte del sistema, incluido el diseño de acuerdo con la arquitectura establecida y la implementación, pruebas unitarias e integración de componentes.
Tester (Tester):	Responsable de actividades centrales de pruebas, como identificar, determinar, implementar y realizar pruebas necesarias, así como registrar

y analizar los resultados de las pruebas.

4.2.7 Tareas organizadas por Disciplinas de OpenUP

OpenUP dentro del contenido del método define un conjunto de 6 disciplinas que son áreas de preocupación que permiten trabajar en simultaneo. organizando las tareas por disciplinas que están relacionadas y que cumplen objetivos en el proceso iterativo de desarrollo del proyecto, estas al agruparse permiten una colaboración ordenada entre categorías de actividades que permiten estructurar ordenadamente el proceso iterativo de desarrollo, estas son: requisitos, arquitectura, desarrollo, pruebas, gestión de proyectos y configuración.

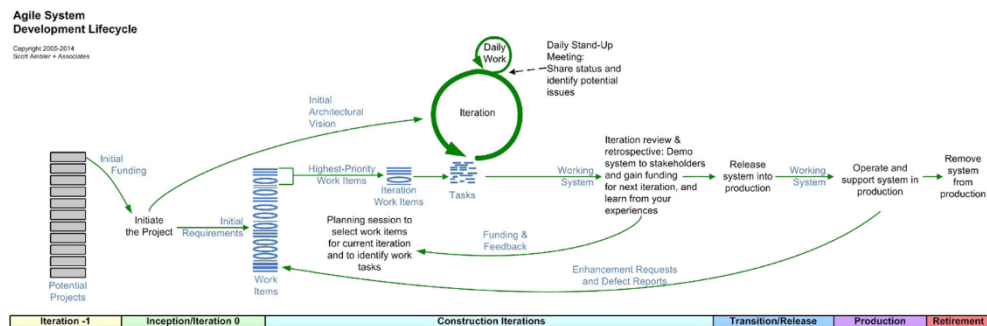
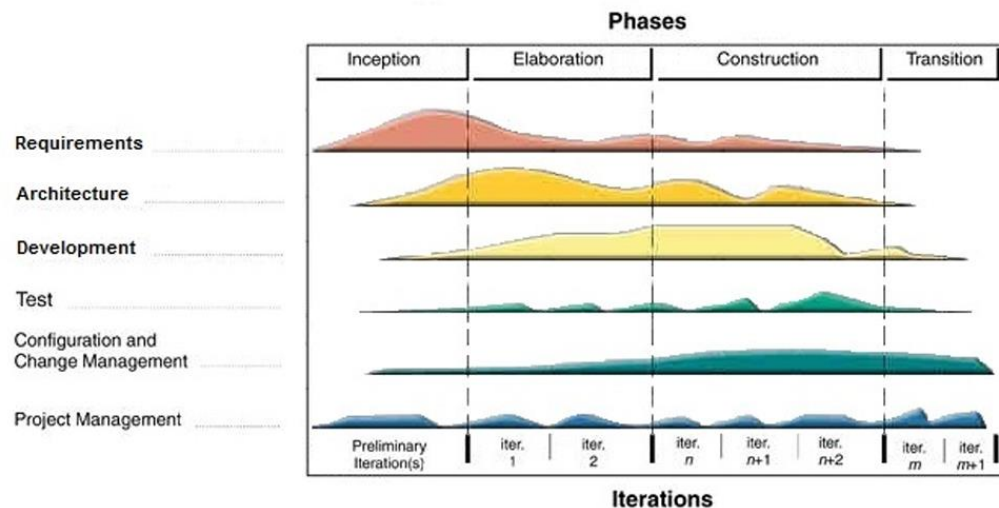


Fig. 2 The Agile System Development Lifecycle, courtesy of Ambler 2008 (ref³²)

Requisitos:

Esta disciplina implica la elicitación, análisis, especificación de los requisitos, como validarlos y gestionar los requerimientos para el sistema a desarrollar. Esta disciplina aporta a las de arquitectura y desarrollo.

El propósito es comprender el problema y las necesidades de los usuarios, definir lo que el sistema debe hacer e interacciones con interfaces externas y sus límites, identificar restricciones técnicas y proporcionar la base para la planificación y estimación de costos.

Arquitectura

- Ofrecer pautas para crear una estructura sólida basada en requisitos clave.
- Evolucionar una arquitectura resistente y definir su representación.

Desarrollo

- Transformar los requisitos en un diseño adaptable al entorno de implementación.
- Construir el sistema gradualmente y verificar que las partes técnicas funcionen según lo previsto.

Pruebas

Esta disciplina valida el sistema con los requisitos.

- Proporcionar retroalimentación temprana sobre el cumplimiento de los requisitos.
- Identificar problemas y garantizar la calidad del sistema, mejorando la eficiencia.

Gestión de configuración y cambios

Esta disciplina administra los cambios de los requisitos del sistema.

- Mantener consistencia en los productos de trabajo mientras evolucionan.
- Adaptarse a cambios y problemas de manera eficiente, proporcionando datos para medir el progreso.

Gestión de proyectos

- Priorizar tareas en consenso con los interesados.

- Estimular la colaboración del equipo y mantener un flujo continuo de trabajo y evaluación.
- Proporcionar un marco para gestionar el riesgo y adaptarse al cambio de manera efectiva.

4.2.8 Artefactos organizados por dominios

Se utilizan una variedad de artefactos, como modelos UML, documentos de requisitos, diagramas de casos de uso, entre otros, para capturar y comunicar la información del proyecto. Las actividades están diseñadas para producir estos artefactos de manera eficiente.

Arquitectura

Notebook de Arquitectura: Describe el contexto para el desarrollo de software, incluyendo decisiones, fundamentos, suposiciones, explicaciones e implicaciones de formar la arquitectura.

Requerimientos

Visión de Requerimientos: Define la visión de los interesados sobre el producto a desarrollar, especificando las necesidades clave y características.

Especificación de Requerimientos de Soporte: Captura los requerimientos a nivel del sistema que no están cubiertos en escenarios o casos de uso, incluyendo requerimientos de atributos de calidad y requerimientos funcionales globales.

Diseño de Desarrollo

Diseño de Desarrollo: Describe la funcionalidad del sistema requerida en términos de componentes, sirviendo como una abstracción del código fuente.

Implementación

Implementación: Representa archivos de código de software, archivos de datos y archivos de soporte como archivos de ayuda en línea que representan las partes crudas de un sistema que pueden ser construidas.

Pruebas

Caso de Prueba: Especifica un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito de evaluar algún aspecto particular de un escenario.

Registro de Pruebas: Recopila la salida en bruto capturada durante una ejecución única de una o más pruebas para un ciclo de prueba único.

Guion de Prueba: Contiene las instrucciones paso a paso que realizan una prueba.

Gestión de Proyectos

Plan de Proyecto: Recopila toda la información necesaria para gestionar el proyecto, incluyendo una descripción de las fases e hitos del proyecto.

Plan de Iteración: Describe los objetivos, asignaciones de trabajo y criterios de evaluación para la iteración.

Lista de Riesgos: Enumera los riesgos conocidos y abiertos del proyecto, ordenados por importancia y asociados con acciones específicas de mitigación o contingencia.

Lista de Elementos de Trabajo: Contiene una lista de todo el trabajo programado que se debe realizar dentro del proyecto, así como trabajo propuesto que puede afectar al producto en este o en futuros proyectos; se utiliza con frecuencia para estimaciones del proyecto.

CAPITULO 5: METODOLOGÍA DE DESARROLLO DE SOFTWARE

4.3 Incepción

En la fase de incepción del proyecto, se establecen los fundamentos esenciales y se define la visión general del proyecto, identificando los objetivos, requisitos de alto nivel conjunto con los stakeholders involucrados. Con esto se logra un comprender general en el equipo del proyecto.

4.3.1 Definición de la visión del proyecto

Objetivos y requisitos de alto nivel:

El principal objetivo del proyecto es desarrollar un prototipo mHealth, dirigida a pacientes renales en diálisis. Esta aplicación tiene como finalidad facilitar el proceso de registro de información de la enfermedad renal crónica para la adherencia al tratamiento y mejora al autocuidado, permitiendo a los usuarios registrar y monitorear información relevante sobre su salud, identificados en la sección 2 de este trabajo de investigación, esto permite resolver problemas relacionados a falta de adherencia al tratamiento, perdida de registros de seguimiento de salud, este enfoque integral de registro permite apoyar el proceso de autocuidado.

Declaración de Visión:

La aplicación busca ser una herramienta integral para el apoyo a la gestión de la enfermedad renal crónica, proporcionando a los pacientes una plataforma fácil de usar y accesible que les permita el registro de datos en su contexto.

Identificación de stakeholders:

Usuarios Finales:

Tipo de usuario	Paciente en tratamiento de diálisis.
Habilidades	Conocimientos mínimos de dispositivos móviles y uso de plataformas.
Actividades	Registrar información en la aplicación para el seguimiento de su condición.

Criterios de Éxito:

La aplicación debe ser intuitiva y fácil de usar.

4.3.2 Definición de los requisitos de alto nivel de solución.

En esta etapa, se detallan los requisitos de alto nivel que guiarán el proceso de análisis, modelado, diseño y desarrollo de la solución propuesta. Esto incluye el levantamiento de los requerimientos críticos principales de funcionamiento en el contexto, para garantizar la alineación de la solución a las necesidades de registros de los pacientes.

Los requisitos de alto nivel de la solución incluirán funcionalidades importantes para el seguimiento de la condición, esto incluye el seguimiento de la ingesta de líquidos, el registro de insumos médicos como medicamentos, la gestión de la configuración del perfil, seguimiento de sesiones de tratamiento y registros de datos asociados.

4.3.3 Documento de especificación de requerimientos.

En esta fase, se detalla el documento de especificación de requerimientos del producto con todas las funcionalidades y características recopiladas que se deben desarrollar en el sistema. Se negocian a priorizar las tareas y mantener un seguimiento del progreso del proyecto. En el backlog del producto de, se incluirán todas las características y funcionalidades necesarias para la aplicación. Se listan tareas como la creación de formularios de registro de datos, el desarrollo de

algoritmos para el seguimiento de síntomas y la integración de notificaciones de recordatorio de medicamentos.

Durante esta fase, se crea un backlog del producto que contiene todas las funcionalidades y características que se deben desarrollar en la aplicación. Esto ayudará a priorizar las tareas y mantener un seguimiento del progreso del proyecto.

En el backlog del proyecto software, se incluirán todas las características necesarias para la aplicación, como:

Formularios de registro de datos.

Desarrollo de algoritmos para el seguimiento de síntomas.

Integración de notificaciones de recordatorio de medicamentos.

4.3.4 Catálogo de requerimientos inicial

El catálogo de requerimientos inicial proporciona una lista detallada de todos los requisitos funcionales y no funcionales de la aplicación. Esto incluye aspectos como la seguridad de los datos del paciente, el rendimiento de la aplicación en dispositivos móviles y la capacidad de sincronización de datos entre dispositivos.

Por ejemplo, entre los requisitos funcionales se incluirá la capacidad de sincronizar datos entre dispositivos y la posibilidad de generar informes de progreso. Los requisitos no funcionales podrían abordar aspectos como la seguridad de los datos del paciente y el rendimiento de la aplicación en dispositivos móviles.

4.3.5 Establecimiento de fundamentos del proyecto

Criterios de Éxito:

La aplicación debe ser intuitiva y fácil de usar para los pacientes renales, mejorando su capacidad para gestionar su enfermedad.

En esta sección se establecerán criterios de éxito y se identificarán riesgos y supuestos clave para el proyecto. Por ejemplo, un criterio de éxito sería que al menos el 80% de los pacientes encuestados encuentren la aplicación fácil de usar y útil. Un riesgo identificado podría ser la falta de adopción por parte de los pacientes debido a la resistencia al cambio o la falta de acceso a dispositivos móviles.

Riesgos y Supuestos:

Riesgo: Falta de adopción por parte de los pacientes debido a problemas de accesibilidad o falta de interés.

Supuesto: Los pacientes tendrán acceso a dispositivos Android compatibles.

Estimación del tiempo y esfuerzo.

En esta sección se realizará una estimación preliminar del tiempo y esfuerzo requeridos para desarrollar la aplicación, junto con un desglose de cronograma y rutas críticas. Esto ayudará a planificar adecuadamente los recursos y garantizar la entrega oportuna del proyecto.

Herramientas ágiles para el modelado de requisitos adaptadas al enfoque por prototipos

Definición de requisitos funcionales y no funcionales basada en la iteración de prototipos.

- Requerimientos funcionales

ID	Caso de Uso	Descripción	Parámetros Importantes
RQ1	Gestionar perfil	Registro de información personal y configuraciones de	- Información personal: Nombre, fecha de

		la aplicación para gestionar los parámetros. Incluyendo datos médicos, alergias y contactos de emergencia.	nacimiento, género, talla. - Información médica: Tipo de tratamiento (hemodiálisis o diálisis peritoneal), fecha de inicio del tratamiento.
RQ2	Gestión de historial medico	Ver y actualizar su información de perfil,	- Datos médicos: Diagnóstico, historial médico, alergias. - Contactos de emergencia: Nombres y números de teléfono.
RQ3	Gestión y Registro de Sesiones de Hemodiálisis	Los pacientes registran las sesiones de diálisis, ingresando datos como fecha, duración y observaciones relevantes. Dependiendo de la configuración del tipo de sesión ya sea hemo o peritoneal.	- Fecha y hora de la sesión. - Duración de la sesión. - Observaciones: Síntomas experimentados durante la sesión.

		Permitir programación de turnos, sesiones, fechas y alarmas.	
RQ4	Visualización de Sesiones de Hemodiálisis	Los pacientes pueden visualizar su historial de sesiones de hemodiálisis.	- Registro de sesiones: Historial de fechas, duraciones y observaciones.
RQ5	Registro de Sesiones de Diálisis Peritoneal	Los pacientes registran las sesiones de diálisis peritoneal, ingresando datos como fecha, volumen de líquido utilizado, tiempo de permanencia y observaciones relevantes.	<ul style="list-style-type: none"> - Fecha y hora de la sesión. - Volumen de líquido utilizado. - Tiempo de permanencia del líquido en la cavidad peritoneal. - Cantidad de líquido drenado. - Observaciones: Cambios en la apariencia del efluente.
R	Visualización de Sesiones de Diálisis Peritoneal	Los pacientes pueden visualizar su historial de sesiones de diálisis peritoneal.	- Registro de sesiones: Historial de fechas, volúmenes utilizados, tiempos de permanencia y observaciones.

	Recordatorio de Sesiones de Hemodiálisis y Diálisis Peritoneal	La aplicación envía recordatorios automáticos para las próximas sesiones de hemodiálisis y diálisis peritoneal al paciente.	<ul style="list-style-type: none"> - Recordatorio con fecha y hora. - Tipo de sesión (hemodiálisis o diálisis peritoneal).
	Seguimiento de Síntomas y Efectos Secundarios	Los pacientes registran síntomas y efectos secundarios después de las sesiones de hemodiálisis y diálisis peritoneal.	<ul style="list-style-type: none"> -Lista de síntomas seleccionables. - Grado de intensidad de los síntomas. - Notas adicionales.
	Gestión de Medicamentos	Los pacientes registran y gestionan la medicación relacionada con la hemodiálisis y diálisis peritoneal, estableciendo recordatorios y recibiendo información sobre medicamentos.	<ul style="list-style-type: none"> - Nombre del medicamento. - Dosificación y frecuencia. - Recordatorios personalizables.
	Gestión de citas	Permitir programar citas y realizar anotaciones de las citas.	

	Visualización de Informes y Resultados	Los pacientes pueden acceder a informes y resultados de laboratorio relacionados con la hemodiálisis y diálisis peritoneal.	<ul style="list-style-type: none"> - Informes claros y comprensibles. - Gráficos y tendencias.
	Registro de Parámetros Específicos de Diálisis Peritoneal	Los pacientes registran parámetros específicos de diálisis peritoneal, como presión intraabdominal o color del efluente.	<ul style="list-style-type: none"> -Lista de parámetros específicos. -Mediciones y observaciones.
	Configuración de Notificaciones y Preferencias	Los pacientes pueden personalizar las notificaciones y ajustar las preferencias de la aplicación según sus necesidades.	<ul style="list-style-type: none"> - Configuración de recordatorios. - Preferencias de visualización.
	Registro de Consumo de Líquidos	Los pacientes registran su consumo de líquidos durante el día.	<ul style="list-style-type: none"> -Tipo de líquido consumido (agua, bebidas específicas). -Cantidad de líquido consumido. -Fecha y hora del registro.

- Requerimientos no funcionales

Usabilidad	La interfaz de usuario debe ser intuitiva y fácil de usar, especialmente para pacientes que pueden no tener experiencia técnica. Se deben minimizar los pasos necesarios para registrar información o acceder a recursos educativos.
Compatibilidad	La aplicación debe ser compatible con una variedad de dispositivos y navegadores web para garantizar que los pacientes puedan acceder a ella desde diferentes plataformas, como smartphones, tabletas y computadoras de escritorio.
Confiabilidad	La aplicación debe ser confiable y libre de errores, especialmente en lo que respecta a la precisión de los registros médicos y la entrega de recordatorios de sesiones de diálisis. Debe tener mecanismos de copia de seguridad para proteger los datos del paciente contra la pérdida.

Ejemplo

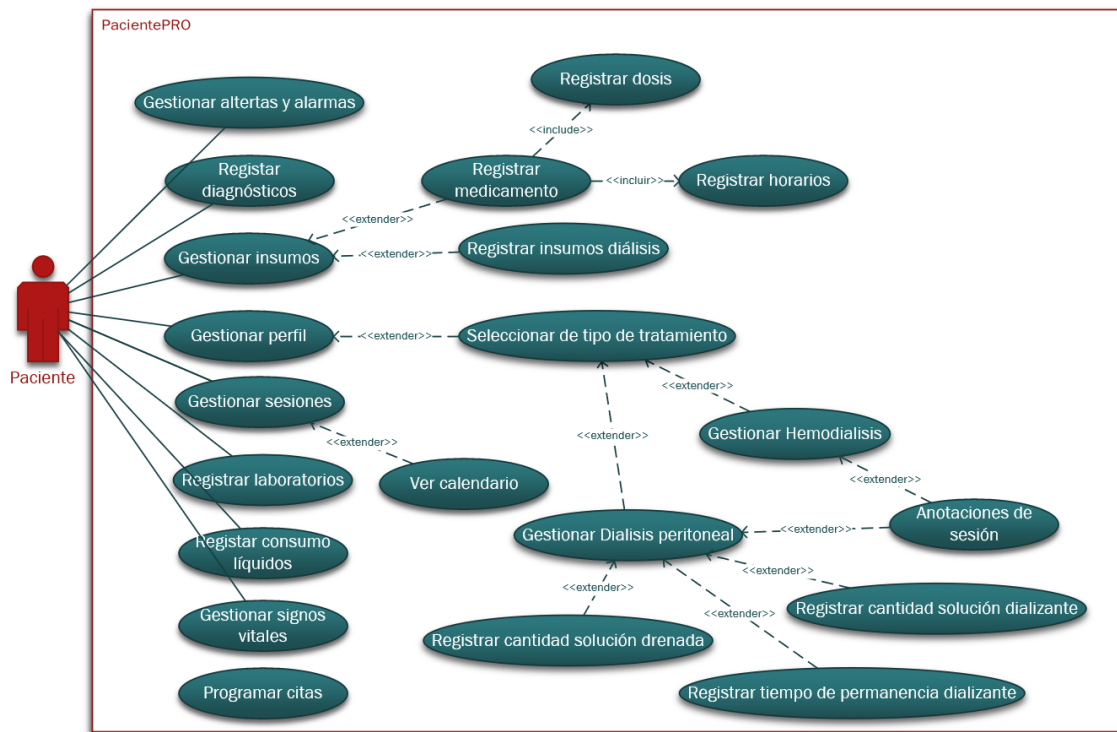
3.8.1.1.1 Requerimientos funcionales

Están determinados por las funcionalidades que requiere la aplicación. Se decidieron en las entrevistas informales con los directivos del Hospital Clínica Metropolitana Riobamba como se muestra en la Tabla 9.

Tabla 9. Requerimientos funcionales

Identificación del requerimiento	RF01
Nombre del requerimiento	Registro de usuario
Descripción del requerimiento	Los profesionales de la salud podrán crear una cuenta en la aplicación proporcionando su nombre, apellido, dirección de correo electrónico y contraseña.
Prioridad del requerimiento	Alta
Identificación del requerimiento	RF02
Nombre del requerimiento	Autenticación de usuario
Descripción del requerimiento	Los usuarios registrados podrán iniciar sesión en la aplicación utilizando su correo electrónico y contraseña.
Prioridad del requerimiento	Alta

Casos de uso



Planificación del proyecto

Gestión de riesgos técnicos y no técnicos del proyecto

Metodología aplicada para la identificación de riesgos asociados al proyecto, Identificación de los riesgos, evaluación a través de calificación el impacto y la probabilidad de cada riesgo, planteamiento de estrategias de mitigación específicas para cada riesgo y priorización para poner atención a los riesgos de mayor impacto y probabilidad.

Riesgo	Descripción	Impacto	Probabilidad	Estrategias de Mitigación	Prioridad
Problemas de usabilidad y experiencia de usuario	Diseño de interfaces inadecuados, navegación poco intuitiva.	Alta	Alta	1. Involucrar al cliente definiendo esquemas de navegación preliminar. 2. Diseño preliminar de la interfaz para evaluación y retroalimentación.	Alta

Fallos en la Integración con Base de Datos y APIs	Problemas técnicos en la integración y sincronización de datos.	Moderada	Alta	1. Pruebas de integración continuas. 2. Mecanismos de redundancia local para datos en sqlite.	Media
Brechas de Seguridad en la Aplicación	Vulnerabilidades en las tecnologías utilizadas.	Moderada	Moderada	1. Evaluar versiones de tecnología y sus CVE. 2. Implementar medidas de seguridad estándar.	Media
Incompatibilidad con Dispositivos Móviles	Limitación del acceso para usuarios con ciertos dispositivos.	Moderada	Moderada	1. Uso de tecnologías web services multiplataforma. 2. Pruebas de visualización en diferentes dispositivos. 3. Diseño adaptable a diferentes resoluciones de pantalla.	Media
Requerimientos inmaduros	Definición incompleta o ambigua de los requisitos.	Moderada a Alta	Moderada	1. Aplicación de metodologías de modelado para estructurar el conocimiento del dominio. 2. Análisis exhaustivo de los requisitos con el usuario. 3. Documentación detallada y actualizada de los requisitos.	Alta

- Requerimientos de infraestructura y despliegue
- Proceso de modelado UML para representar la estructura y relaciones de los requisitos
 - Diagramas de casos de uso para identificar y describir las interacciones con el sistema
 - Diagramas de clases para definir las entidades clave y sus relaciones
 - Diagramas de actividad para representar flujos de trabajo y procesos

Hora	Día 1	Día 2-9	Día 10
1			
2	Planificación de historias. El gestor de proyecto presenta las historias.	Reunión de situación (15 min)	Reunión de situación (15 min)
3	Planificación: Interacciones, definir tareas y estimar historias.		
4			
5			
6			
7			Demo
			Retrospectiva
8	Reagruparse, Negociar alcance y llegar a acuerdos.	Presentación preliminar solo una vez a mitad de la iteración.	

Tabla 5: Ejemplo de planificación de iteración

Como <rol>, podré <actividad> de tal modo que <valor de negocio>.

Con este guión, el equipo aprende a centrarse tanto en los roles de usuario y en los beneficios del negocio de las nuevas funcionalidades propuestas.

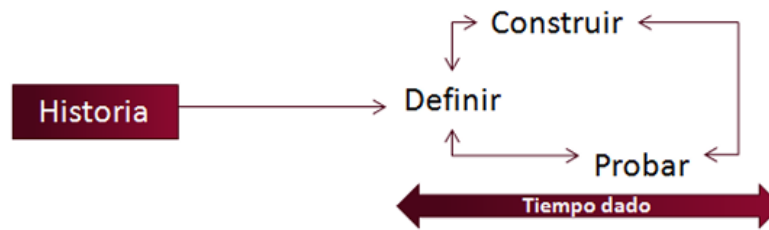


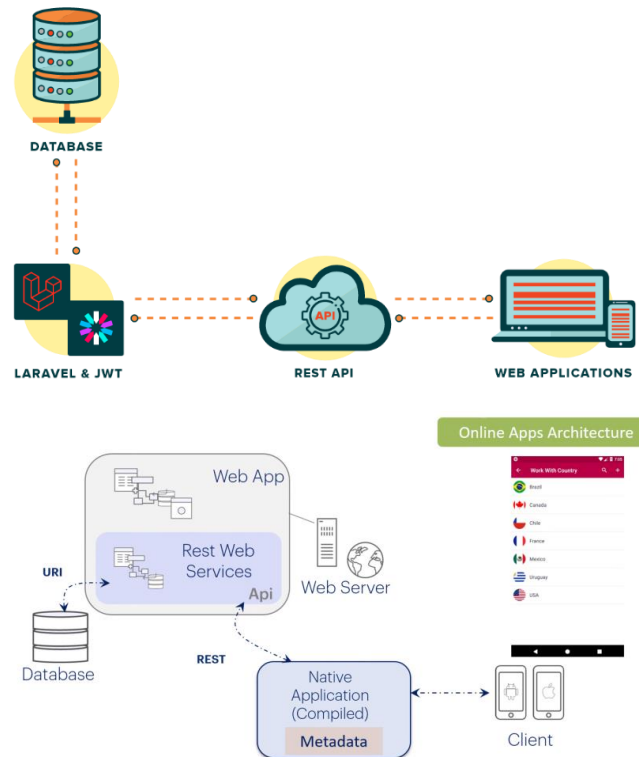
Figura 14: Diagrama Historia de Usuario

Como <rol>, podré <actividad> de tal modo que <valor de negocio>

Dentro del contexto de e-banking planteado, podríamos tener por ejemplo las siguientes historias de usuario:

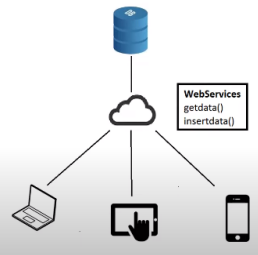
- Como <usuario (cliente del banco dado de alta)>, podré <ver mi gasto diario> de tal modo que <pueda controlar mi gasto>.
- Como <administrador>, podré <establecer la política de seguridad de permisos> de tal modo que <pueda controlar y mantener el sistema seguro>.
- Como <usuario>, podré <elegir un tema colores e iconos al visualizar la web> de tal modo que <la visualización sea más agradable para los clientes>.
- Como <utilidad>, podré <enviar mensajes a la bandeja de entrada de mis usuarios> de tal modo que <conozcan sus gastos en el momento y verifiquen su gasto>.

4.4 Desarrollo de la arquitectura candidata del sistema



Servicio Web

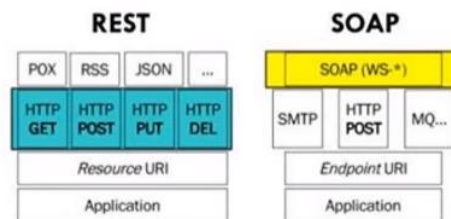
- Un servicio web (en inglés, *web service* o *web services*) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- Los servicios web suelen ser considerados como APIs Web que pueden ser accedido dentro de una red (principalmente internet) y ejecutados en el sistema que los aloja.



Consideradas como una interfaz expuesta a través de una API

Servicio Web

Protocol Layering



- REST (Representational State Transfer): arquitectura que, haciendo uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE).

- Web Services Protocol Stack: conjunto de servicios y protocolos de los servicios web.
- XML, SOAP, HTTP, FTP o SMTP.
- WSDL (Web Services Description Language)
- UDDI (Universal Description, Discovery and Integration)
- WS-Security (Web Service Security)

Arquitectura SOA

¿Cuándo usarlo?

- Construcción de aplicaciones con múltiples servicios y una interfaz única.
- Construcción de aplicaciones en la nube.
- Comunicación basada en mensajes.
- Funcionalidad con independencia de plataformas.
- Necesidad de establecer servicios federados.
- Exposición de servicios con independencia del conocimiento por parte del cliente de su interfaz.
- Adecuado para escenarios de interoperabilidad e integración.

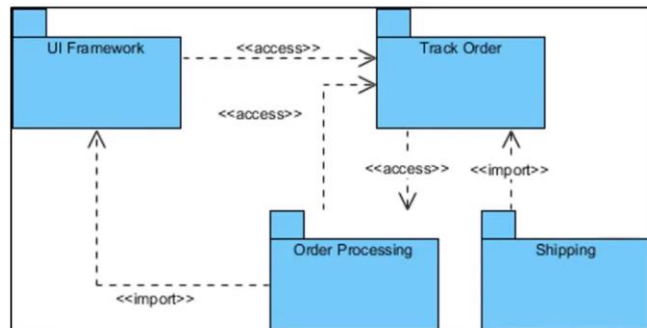
Componentes arquitectónicos del sistema

Diagrama de clases

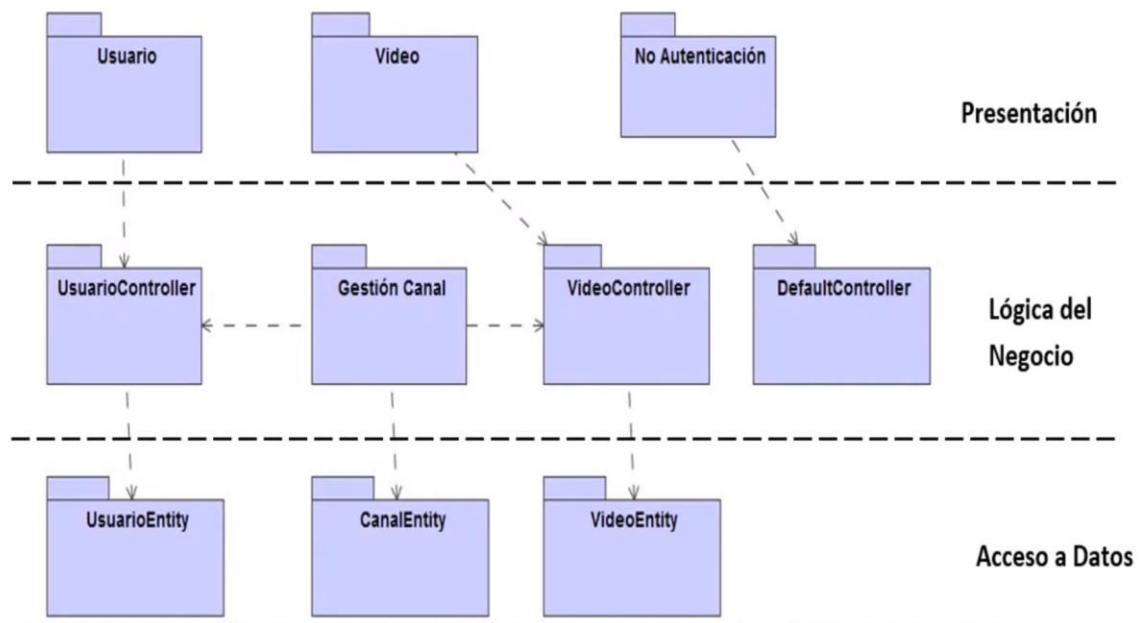
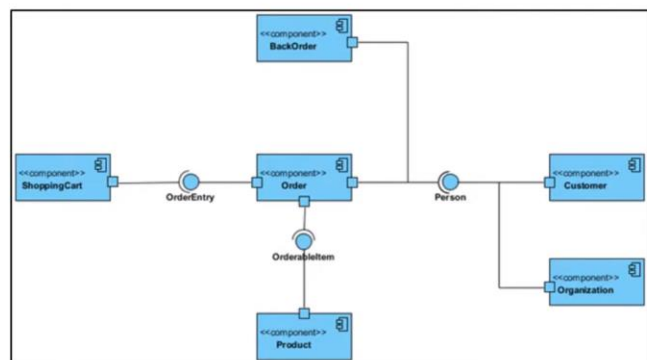
Diagrama de actividad

Diagrama de paquetes

- Paquetes
- Subsistemas



- Componentes



Modelo de la base de datos (Modificar plagio)

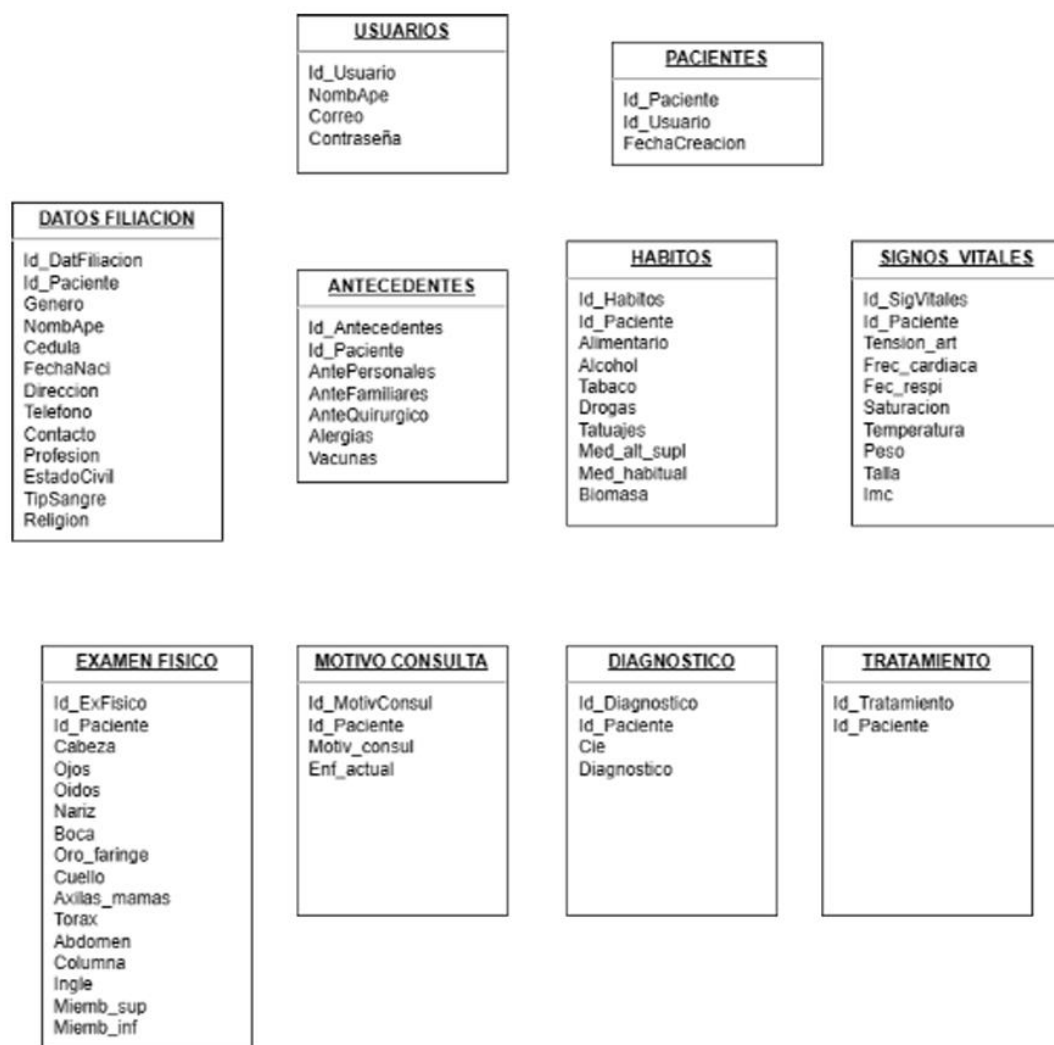
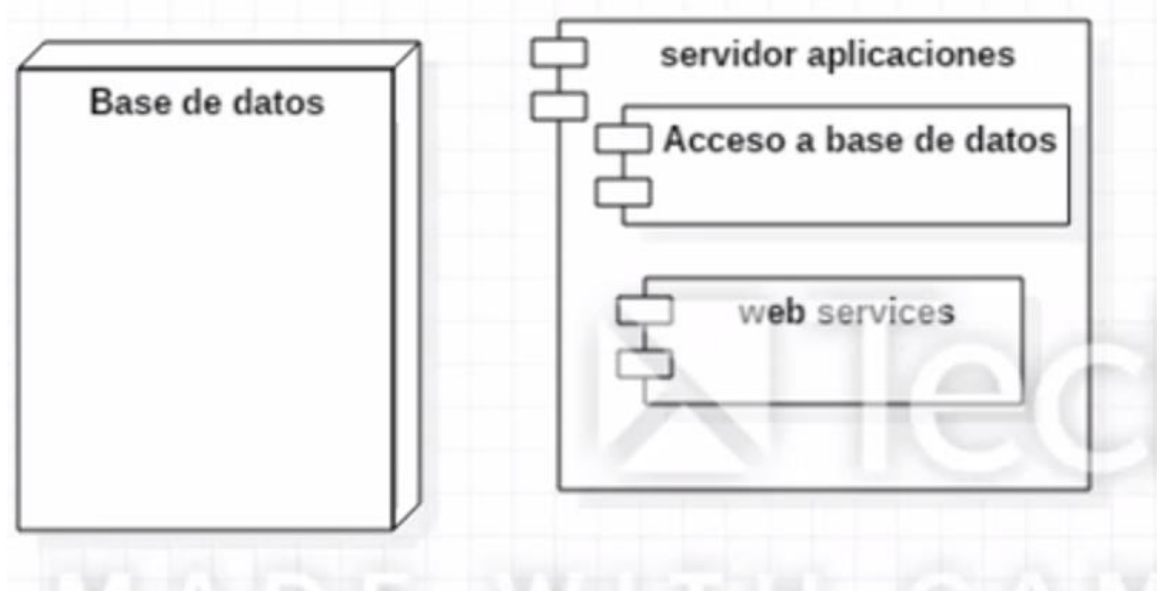
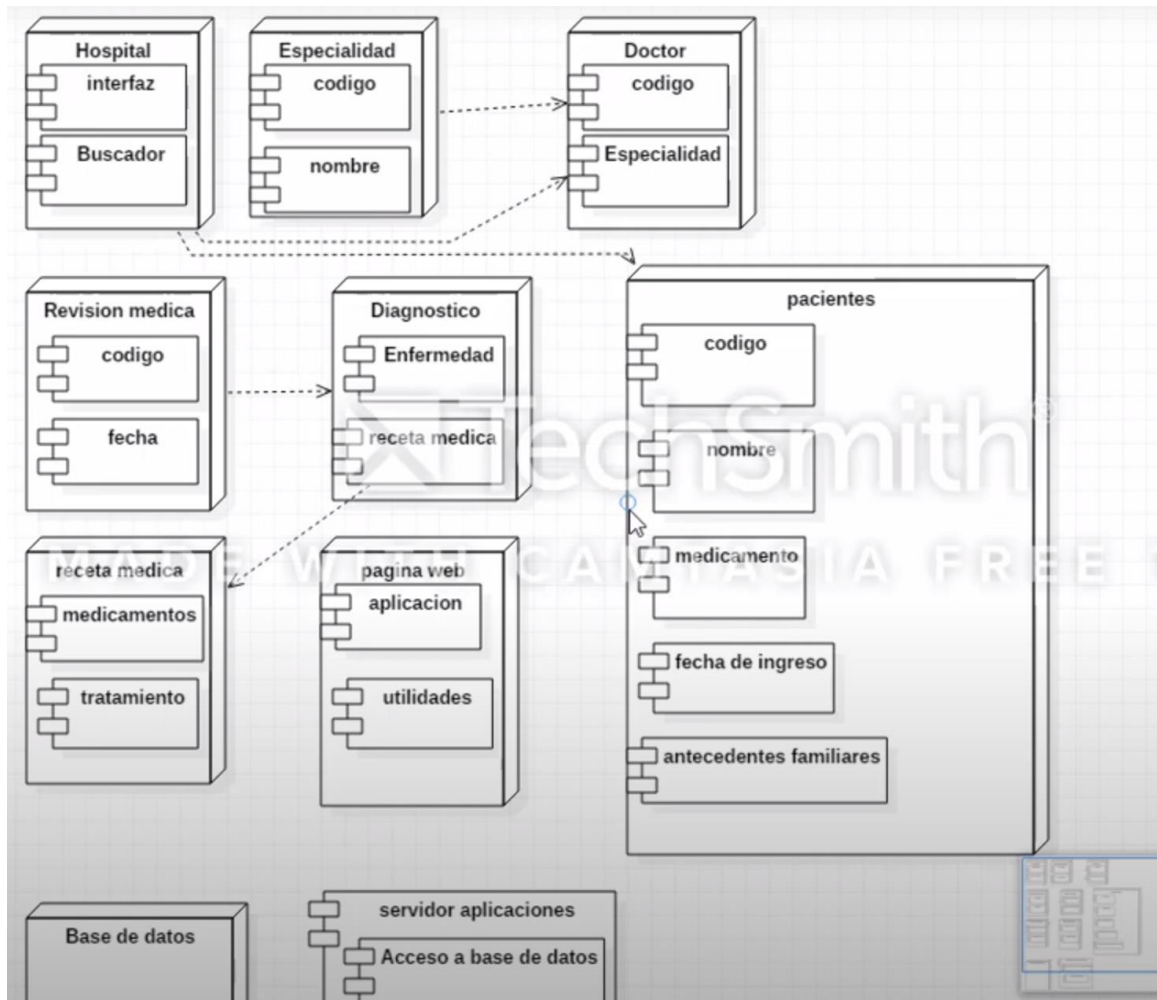
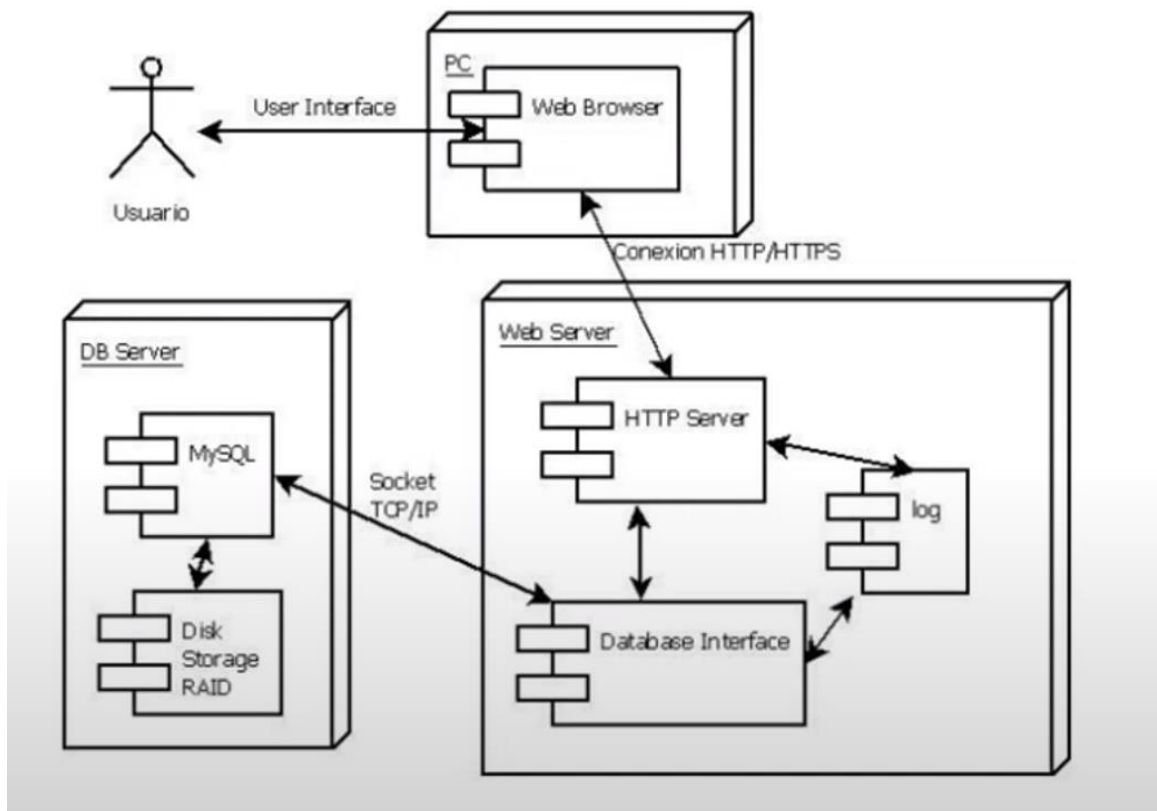
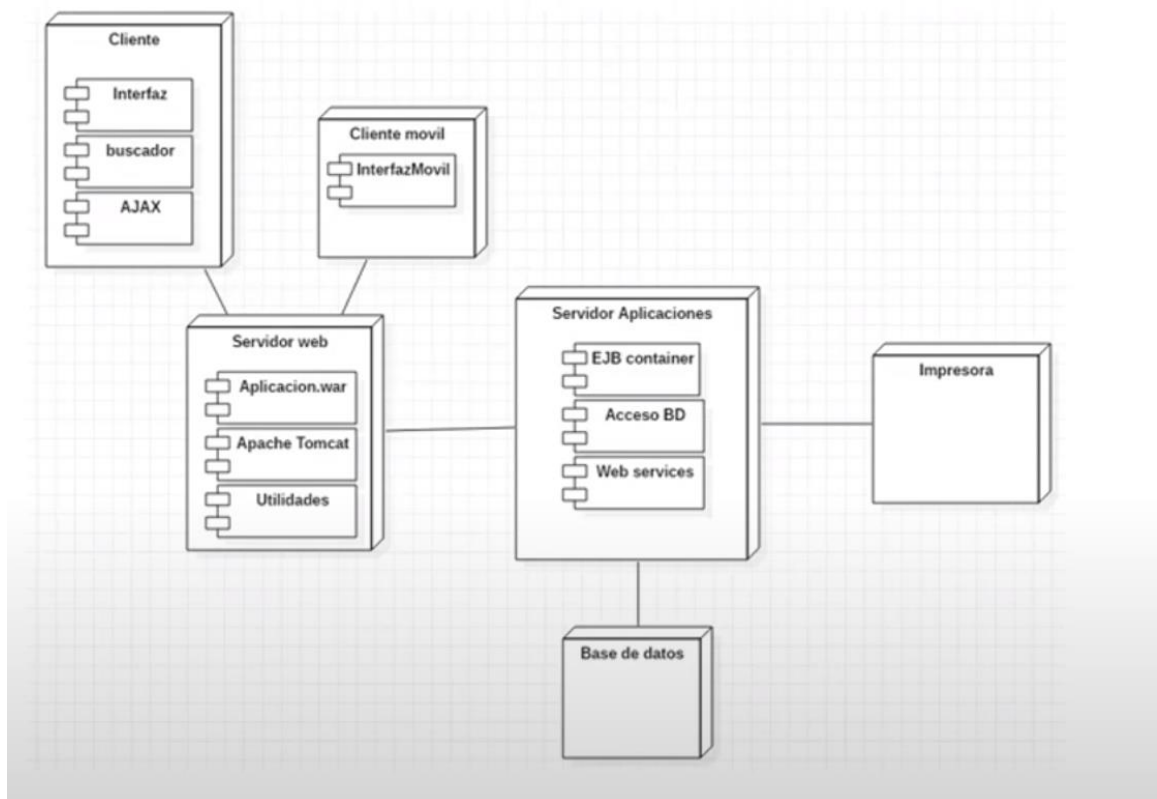


Diagrama de despliegue



Alternativa 2



Diseño de interfaz

Esquema de navegación

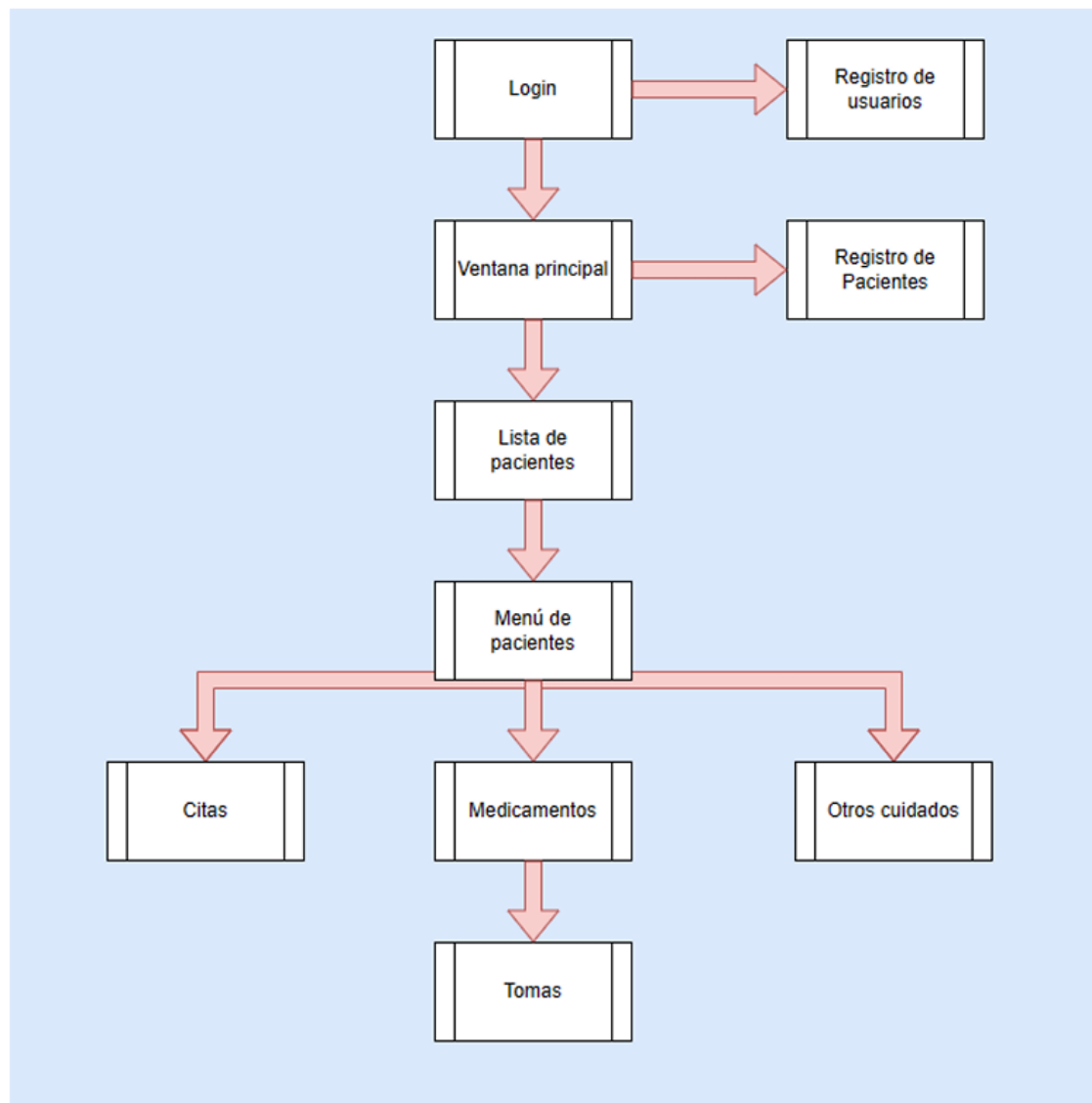


Ilustración 6. Diagramas de vistas de la aplicación.

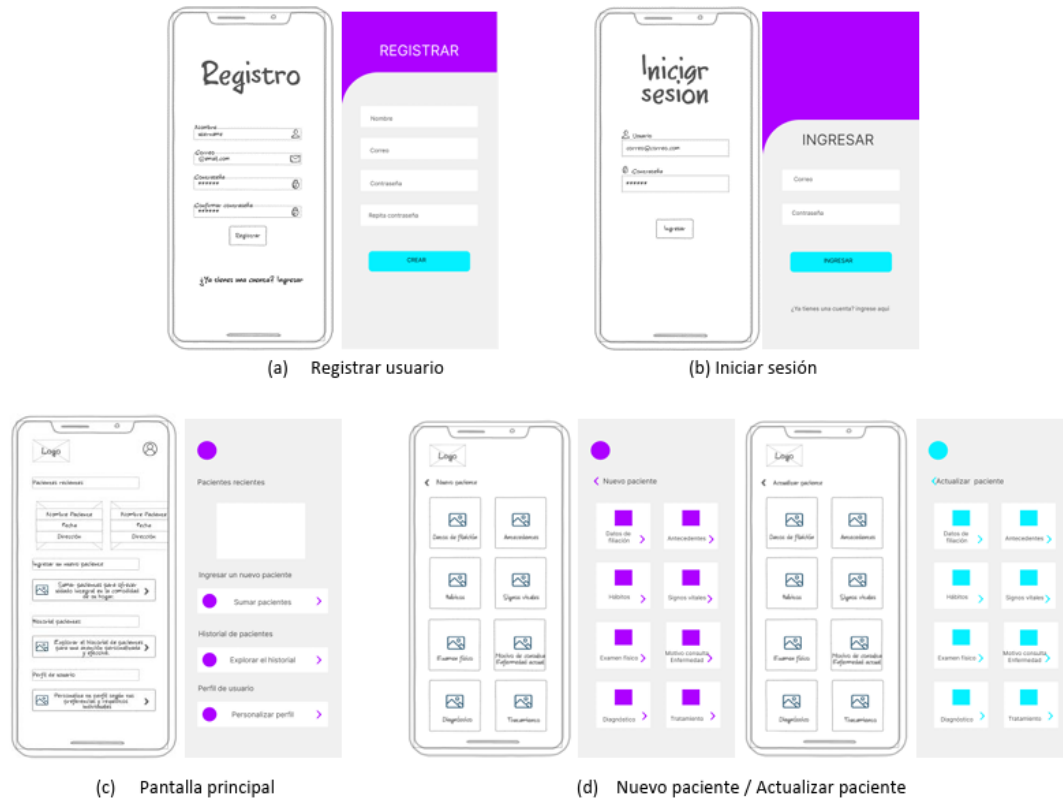
Mockups

Wireframes

Prototipo front-end aplicación cliente

Ejemplo

Figura 20. Diseño de interfaz de la aplicación móvil herramienta NinjaMock y Figma



Nuevo paciente/ Antecedentes

Antecedentes personales

Antecedentes familiares

Antecedentes quirúrgicos

Alergias

Vacunas

Nuevo paciente/ Datos de filiación

Género

Nombre y apellido

Cédula

Fecha de nacimiento

Dirección

Teléfono

Contacto

Profesión

Estado civil

Tipo de sangre

Religión

Guardar

(e) Crear datos de filiación/Editar datos de filiación

Nuevo paciente/ Antecedentes

Antecedentes personales

Antecedentes familiares

Antecedentes quirúrgicos

Alergias

Vacunas

Guardar

(f) Antecedentes/Editar antecedentes

Nuevo paciente/ Hábitos

Alimentario

Alcohol

Tabaco

Drogas

Tatuajes

Medicación alternativa y suplementos

Contacto

Profesión

Estado civil

Guardar

(g) Crear hábitos/Editar hábitos

Nuevo paciente/ Signos vitales

Tensión arterial

Frecuencia cardíaca

Frecuencia respiratoria

Saturación

Temperatura

Peso

Talla

IMC

Guardar

(h) Crear signos vitales/Actualizar signos vitales

Nuevo paciente/ Examen físico

CP SP

Cabeza

Ojos

Nariz

Boca

Cuello

Abdomen

Axilas-Mamas

Guardar

(i) Crear examen Físico/ Actualizar examen físico

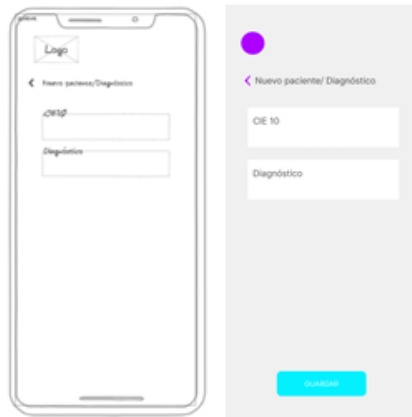
Nuevo paciente/ Motivo consulta enfermedad actual

Motivo consulta

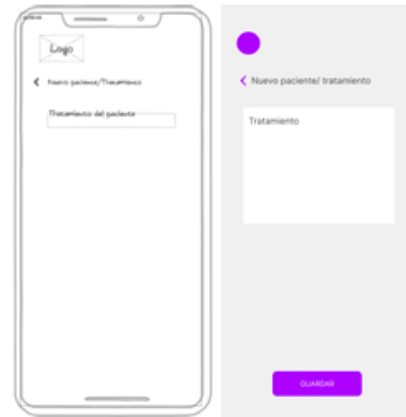
Enfermedad actual

Guardar

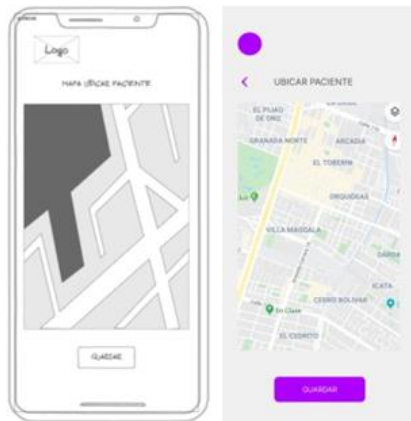
(j) Crear motivo consulta/ Enfermedad actual /Actualizar



(k) Crear diagnóstico/ Actualizar diagnóstico



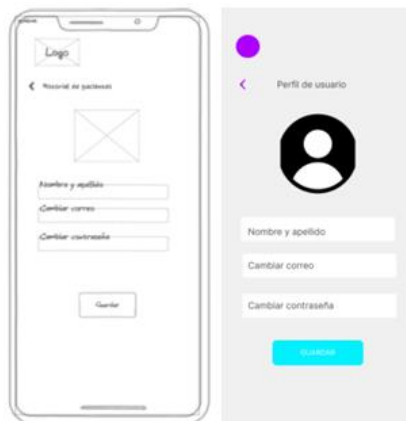
(l) Crear tratamiento/notas de evolución /Actualizar



(m) Ubicar paciente en el mapa



(n) Historial de pacientes



(ñ) Perfil de usuario

Fuente: El autor

Tecnologías
AndroidStudio

Kotlin
SQLite
Laravel
PHP
XML, JSON

HL7: El estándar HL7 (Health Level Seven) es un agrupador de estándares de información clínica que permite el intercambio de información entre diferentes Hospitales, centros de salud o demás sistemas de salud.

Dentro de este estándar los más comunes y utilizados en la actualidad son HL7 V2, HL7 V3, FHIR y CDA.

4.5 Elaboración

En esta fase, nos enfocaremos en refinar la visión del proyecto, elaborar la arquitectura del sistema y comenzar la construcción de la aplicación.

4.5.1 Refinamiento de la declaración de la visión y requisitos

Refinamiento de la Declaración de Visión:

La aplicación tiene como objetivo principal mejorar la calidad de vida de los pacientes renales al proporcionarles herramientas específicas para el seguimiento y la gestión de su enfermedad. Entre los objetivos específicos se incluyen:

Seguimiento de la ingesta de líquidos.

Monitorización de la presión arterial.

Gestión de la dieta y nutrición.

Además, se establecerán métricas de éxito, como el número de descargas de la aplicación en el primer mes, la tasa de retención de usuarios después de tres meses de uso y la satisfacción del usuario medida a través de encuestas periódicas.

4.3.4. Requisitos Funcionales

Identificador	RF01
Nombre	Registro con correo
Introducción	El usuario podrá registrarse en la aplicación introduciendo unas credenciales válidas.
Entrada	El usuario introducirá un correo válido y una contraseña que debe cumplir los siguientes requisitos: una longitud mayor de 6 para registrarse, un carácter mayúscula y otra minúscula.
Proceso	Se comprueba que el correo y contraseña cumplen el formato correcto y se comprueba con la base de datos si el correo no existe previamente.
Salida	Se almacenan los datos del usuario en la base de datos y se permite al acceso a la aplicación, almacenando los datos en la caché del dispositivo para su uso.

Identificador	RF02
Nombre	Registro con servicios de terceros

Ejemplo de especificación de requerimientos

Otros requerimientos

Portabilidad (*Portability*)

La capacidad de ejecutarse en diferentes plataformas computacionales. En el sistema Holytime se han identificado los siguientes requisitos:

PORT1: El sistema deberá ofrecer una interfaz similar, tanto para la aplicación móvil IOS como para Android.

PORT2: La aplicación para móviles debe desarrollarse para ambos sistemas operativos, Android y IOS. Por ello, se limita el uso de la aplicación a todo aquel dispositivo que integren dichos SO.

Requisito	Atributo de portabilidad	KPI(<i>Key Performance Indicator</i>)
PORT1	Interfaz para móviles	Aplicación móvil
PORT2	Aplicación disponible	IOS y Android

Recuperabilidad (*Recoverability*)

RECU1: En caso de caída del sistema se recuperarán los datos de los usuarios en menos de 45 minutos.

RECU2: Deben existir sistemas de respaldo en caso de un funcionamiento fallido.

RECU3: Se realizará un *backup* de los datos del sistema con periodicidad semanal.

Requisito	Atributo de recuperabilidad	KPI(<i>Key Performance Indicator</i>)
RECU1	Recuperar datos	< 45 minutos
RECU2	Debe haber	Sistema de respaldo
RECU3	RTO	Semanal

Seguridad (*Security*) e Integridad (*Integrity*)

SEC1: Las comunicaciones serán cifradas garantizando autenticidad, integridad y confidencialidad.

SEC2: El sistema contará con un módulo de verificación y autenticación de usuario.

SEC3: Los datos almacenados en las BBDD deben estar cifrados.

42

Requisito	Atributo de seguridad	KPI (<i>Key Performance Indicator</i>)
SEC1	Tipo comunicaciones	Cifradas
SEC2	Autenticación	<i>Tokens</i>
SEC3	Cifrado	Todos los datos

Ejemplo de descripción de casos de uso

❖ Consultar tiempo atmosférico

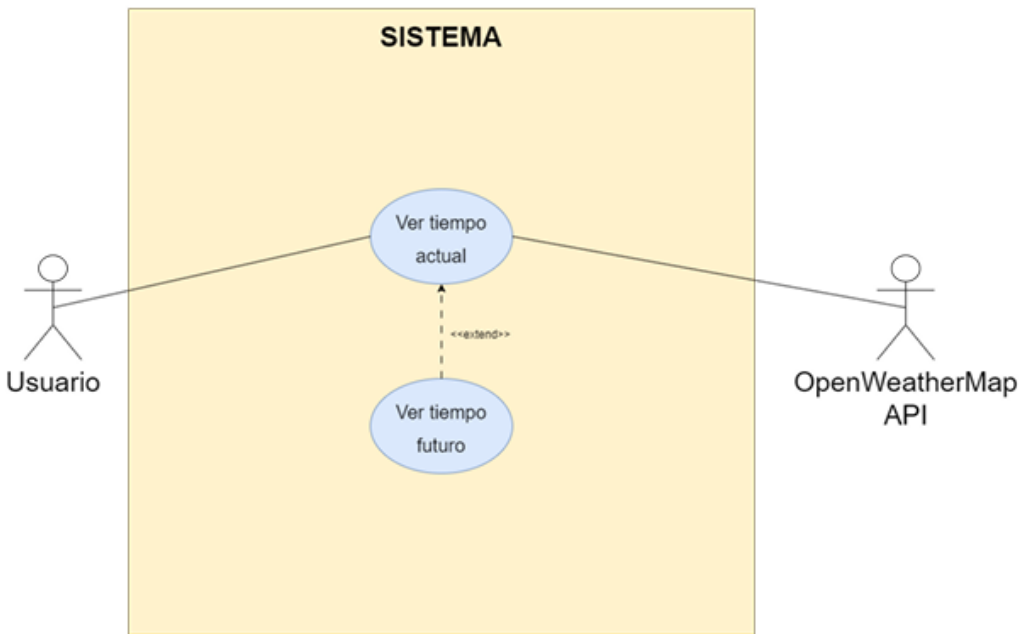


Figura 13. Caso de uso tiempo atmosférico

		Id	C.U. 8
		Nombre	Ver tiempo actual
		Actores	Usuario, API de terceros
		Precondiciones	El usuario debe estar autenticado
Curso normal de eventos			
USUARIO	SISTEMA	API de terceros	
1. El usuario selecciona la opción de tiempo atmosférico en el menú principal.	2. El sistema solicita la información a la API de terceros, enviándole la ubicación y hora de la ciudad.	3. Envía la información solicitada al sistema	

4.5.2 Documento de análisis del sistema software

En esta fase, se elaborará un documento detallado que describa los requisitos funcionales y no funcionales de la aplicación. Esto incluirá los flujos de trabajo para el registro de datos de salud, la visualización de informes de progreso y otras funcionalidades clave. Por ejemplo:

Documento de Análisis del Sistema Software:

Se detallarán los requisitos funcionales, como la capacidad de los usuarios para registrar datos de salud, configurar recordatorios de medicamentos y acceder a recursos educativos. También se especificarán los requisitos no funcionales, como la seguridad de los datos del paciente y el rendimiento de la aplicación en diferentes dispositivos Android.

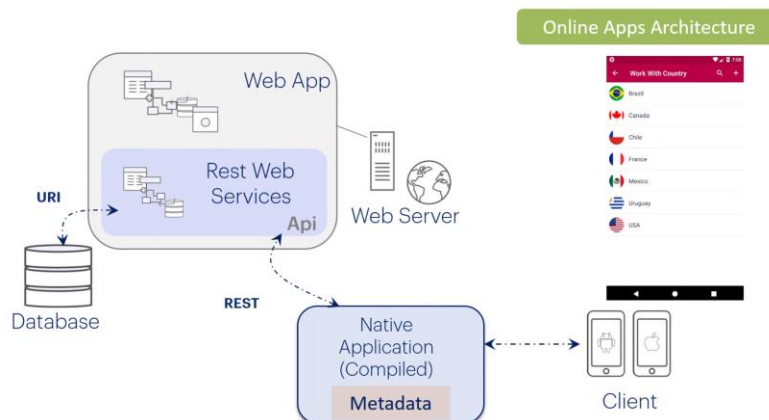
4.5.3 Elaboración de documento de arquitectura del sistema

Identificación de Arquitectura:

Se decidirá si la aplicación utilizará una arquitectura cliente-servidor o si se almacenarán los datos localmente en el dispositivo del usuario. Se establecerán los estándares, paquetes y librerías que se utilizarán en el desarrollo de la aplicación, así como el esquema de navegación y el diseño de la interfaz gráfica.

Diseño de Arquitectura de Alto Nivel:

Se definirá la arquitectura de la aplicación, incluyendo la estructura de la base de datos, la lógica de negocio y la interfaz de usuario. Se elaborará el diseño del sistema, especificando el esquema de navegación, la base de datos del sistema y los estándares de desarrollo a seguir. Además, se diseñará una interfaz de usuario intuitiva y atractiva que cumpla con las necesidades y expectativas de los usuarios finales.



Elaboración del diseño del sistema esquema de navegación

El esquema de navegación de la aplicación debe ser intuitivo y fácil de usar para los pacientes renales. Para ello, proponemos una estructura de navegación clara y organizada, que permita a los usuarios acceder rápidamente a las diferentes secciones y funcionalidades de la aplicación. Por ejemplo:

Página de inicio: Desde aquí, los usuarios podrán acceder a las funciones principales de la aplicación, como el registro de datos de salud, la visualización de informes y la configuración de recordatorios.

Sección de registro de datos: Permite a los usuarios ingresar información relevante sobre su salud, como la presión arterial, niveles de azúcar en la sangre y síntomas experimentados.

Sección de informes: Aquí los usuarios podrán visualizar informes detallados sobre su progreso y estado de salud, incluyendo gráficos y estadísticas.

Sección de configuración: Permite a los usuarios personalizar la aplicación según sus preferencias, como configurar recordatorios de medicamentos y ajustar la interfaz de usuario.

Elaboración de base de datos del sistema

Para la base de datos del sistema, proponemos un enfoque estructurado que permita almacenar y gestionar eficientemente los datos de los usuarios. Se

utilizará una base de datos relacional para garantizar la integridad de los datos y facilitar su manipulación. Por ejemplo:

Tabla de usuarios: Almacena la información de los usuarios, como nombres, correos electrónicos y preferencias de configuración.

Tabla de registros de salud: Contiene los datos ingresados por los usuarios, como presión arterial, niveles de azúcar en la sangre y síntomas.

Tabla de informes: Guarda los informes generados a partir de los datos de salud de los usuarios, incluyendo gráficos y estadísticas.

Elaboración de estándares, paquetes y librerías

Para mantener un desarrollo consistente y eficiente, se utilizarán estándares de codificación y buenas prácticas de desarrollo de software. Además, se emplearán paquetes y librerías de código abierto que faciliten el desarrollo y mejoren la calidad del producto final. Por ejemplo:

Estándares de codificación: Se seguirán las convenciones de codificación establecidas por la comunidad de desarrollo de Android para garantizar la legibilidad y mantenibilidad del código.

Paquetes y librerías: Se utilizarán bibliotecas como Retrofit para realizar llamadas a API, Glide para la carga de imágenes y Room para la gestión de la base de datos local.

Elaboración de interfaz grafica

La interfaz gráfica de la aplicación se diseñará teniendo en cuenta las necesidades y preferencias de los usuarios finales. Se utilizarán elementos visuales claros y atractivos, con una paleta de colores suaves y un diseño limpio y minimalista. Por ejemplo:

Diseño intuitivo: Se priorizará la usabilidad y la accesibilidad, utilizando elementos familiares para los usuarios y evitando la sobrecarga de información.

Navegación coherente: Se mantendrá una estructura de navegación consistente en toda la aplicación, con botones y menús ubicados de forma intuitiva para facilitar la interacción.

Feedback visual: Se proporcionarán indicadores visuales claros para informar a los usuarios sobre el estado de la aplicación, como mensajes de éxito o error al realizar una acción.

4.6 Construcción

4.6.1 Iteración I

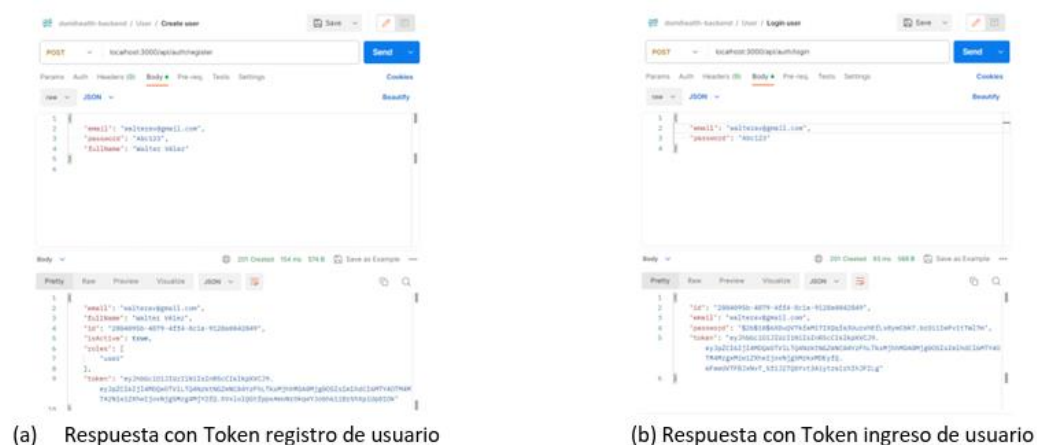
Implementación de Funcionalidades:

Durante la primera iteración, se desarrollarán las funcionalidades básicas de la aplicación, como el registro de datos de salud y la configuración de recordatorios de medicamentos. Por ejemplo, se implementará un formulario de ingreso de datos que permita a los usuarios registrar su presión arterial y niveles de azúcar en la sangre.

- **Desarrollo de Funcionalidades Específicas:**
 - Implementar características como registro de medicamentos, seguimiento de síntomas, programación de citas y acceso a recursos educativos.

Pruebas de Unidad y Pruebas de Integración:

Figura 21. Pruebas de la API REST realizadas en Postman



- **Pruebas de Unidad:**

- Verificar el correcto funcionamiento de cada componente de la aplicación, como el registro de datos de salud y la generación de informes.

- **Pruebas de Integración:**

- Probar la interoperabilidad entre los diferentes módulos de la aplicación, como la integración de la base de datos con la interfaz de usuario.

Desarrollo de Interfaces de Usuario:

- **Diseño de Interfaz Intuitiva:**

- Crear una interfaz de usuario atractiva y fácil de usar que permita a los pacientes renales acceder rápidamente a las funciones principales de la aplicación.

4.6.2 Iteración II

Implementación de Funcionalidades:

En la segunda iteración, se añadirán funcionalidades adicionales y se realizarán mejoras basadas en la retroalimentación de la primera iteración. Por ejemplo, se podría agregar una función de chat para que los usuarios puedan comunicarse con su médico o con otros pacientes.

- **Desarrollo de Funcionalidades Específicas:**

- Implementar características como registro de medicamentos, seguimiento de síntomas, programación de citas y acceso a recursos educativos.

Pruebas de Unidad y Pruebas de Integración:

- **Pruebas de Unidad:**

- Verificar el correcto funcionamiento de cada componente de la aplicación, como el registro de datos de salud y la generación de informes.

- **Pruebas de Integración:**

- Probar la interoperabilidad entre los diferentes módulos de la aplicación, como la integración de la base de datos con la interfaz de usuario.

Desarrollo de Interfaces de Usuario:

- **Diseño de Interfaz Intuitiva:**
 - Crear una interfaz de usuario atractiva y fácil de usar que permita a los pacientes renales acceder rápidamente a las funciones principales de la aplicación.

4.7 Transición

Pruebas de Aceptación del Usuario:

Pruebas con Pacientes Reales:

Solicitar la participación de pacientes renales para probar la aplicación y proporcionar retroalimentación sobre su usabilidad y utilidad.

Preparación para el Despliegue:

Documentación y Configuración:

Preparar guías de usuario y documentación técnica, así como configurar el entorno de producción para el despliegue de la aplicación.

Despliegue y Entrega:

Lanzamiento de la Aplicación:

Publicar la aplicación "RenalCare" en la tienda de aplicaciones de Google Play y realizar la entrega a los pacientes renales y sus cuidadores.

Soporte Inicial:

Proporcionar soporte técnico inicial y estar disponible para resolver cualquier problema que surja durante el uso inicial de la aplicación.

CAPÍTULO 6: RESULTADOS

- 5.1 Componentes diseñados / Borrador**
 - 5.1.1 Modulo Interfaz de gestión de perfil**
 - 5.1.2 Módulo de registro de signos vitales**
 - 5.1.3 Modulo Interfaz de seguimiento de tratamiento**
 - 5.1.4 Submódulo gestión de turnos horarios de sesión**
 - 5.1.5 Submódulo registro de control de pesos**
 - 5.1.6**
 - 5.1.7 Modulo Interfaz de registro de consumos**
 - 5.1.8 Submódulo registro de seguimiento de líquidos**
 - 5.1.9 Submódulo registro de seguimiento de alimentos**
 - 5.1.10**
 - 5.1.11 Modulo Interfaz de registro de medicamentos e insumos**

CONCLUSIONES

RECOMENDACIONES

REFERENCIAS BIBLIOGRÁFICAS

- Alwakeel, L., & Lano, K. (2022). Functional and Technical Aspects of Self-management mHealth Apps: Systematic App Search and Literature Review. *JMIR human factors*, 9(2), e29767. <https://doi.org/10.2196/29767>
- Duan, H., Wang, Z., Ji, Y., Ma, L., Liu, F., Chi, M., Deng, N., & An, J. (2020). Using Goal-Directed Design to Create a Mobile Health App to Improve Patient Compliance With Hypertension Self-Management: Development and Deployment. *JMIR mHealth and uHealth*, 8(2). <https://doi.org/10.2196/14466>
- Gisela, B., Tejedor, S., González, R., Omar, E., Calderón, C., Tejedor, E., León, D., Luis, J., & Ramirez, M. (2024). Proceso de elaboración de una aplicación Android dirigida a personas con nefropatía crónica para el control de la ingesta de líquidos a través el consumo de frutas y legumbres. *REICIT*, 3(2), 74–94. <https://doi.org/10.48204/REICT.V3N2.4678>
- Kovesdy, C. P. (2022). Epidemiology of chronic kidney disease: an update 2022. *Kidney International Supplements*, 12(1), 7. <https://doi.org/10.1016/J.KISU.2021.11.003>
- Lightfoot, C. J. ;, Nair, D. ;, Bennett, P. N. ;, Smith, A. C. ;, Griffin, A. D. ;, Warren, M. ;, Wilkinson, T. J., Müller, R.-U., Greenwood, S. A., Castle, E. M., Lightfoot, C. J., Nair, D., Bennett, P. N., Smith, A. C., Griffin, A. D., Warren, M., & Wilkinson, T. J. (2022). Patient Activation: The Cornerstone of Effective Self-

- Management in Chronic Kidney Disease? *Kidney and Dialysis* 2022, Vol. 2, Pages 91-105, 2(1), 91–105. <https://doi.org/10.3390/KIDNEYDIAL2010012>
- Naciones Unidas. (s/f-a). *Ciudades - Desarrollo Sostenible*. Recuperado el 11 de marzo de 2024, de <https://www.un.org/sustainabledevelopment/es/cities/>
- Naciones Unidas. (s/f-b). *Educación - Desarrollo Sostenible*. Recuperado el 11 de marzo de 2024, de <https://www.un.org/sustainabledevelopment/es/education/>
- Naciones Unidas. (s/f-c). *Infraestructura - Desarrollo Sostenible*. Recuperado el 11 de marzo de 2024, de <https://www.un.org/sustainabledevelopment/es/infrastructure/>
- Naciones Unidas. (s/f-d). *Reducir las desigualdades entre países y dentro de ellos - Desarrollo Sostenible*. Recuperado el 11 de marzo de 2024, de <https://www.un.org/sustainabledevelopment/es/inequality/>
- Naciones Unidas. (s/f-e). *Salud - Desarrollo Sostenible*. Recuperado el 11 de marzo de 2024, de <https://www.un.org/sustainabledevelopment/es/health/>
- National Kidney Foundation. (2024). *Chronic kidney disease (CKD) - Symptoms, causes, treatment*. <https://www.kidney.org/atoz/content/about-chronic-kidney-disease>
- Otero Arrascue Danny Frank Asesor, B., & Mejia Cabrera Heber Ivan, M. (2020). Comparación de mapas conceptuales basado en ontología y lenguaje de modelado unificado, para mejorar la comprensión de los requerimientos en la elaboración del product backlog del marco de gestión ágil Scrum. *Repositorio Institucional - USS*. <http://repositorio.uss.edu.pe//handle/20.500.12802/6753>
- Rowland, S. P., Fitzgerald, J. E., Holme, T., Powell, J., & McGregor, A. (s/f). *PERSPECTIVE What is the clinical value of mHealth for patients?* <https://doi.org/10.1038/s41746-019-0206-x>
- Rueda Borrero, E., Moreno Ordoñez, R., & Quintero, J. (2023). CAUSAS DE ADMISIÓN EN ADULTOS CON ENFERMEDAD RENAL CRÓNICA EN HOSPITAL DE REFERENCIA DE LA CIUDAD DE PANAMÁ. *Revista Médico Científica*, 35, 4–12. <https://doi.org/10.37416/rmc.v35i2.716>

- Pan American Health Organization. (2021). *The burden of kidney diseases in the Region of the Americas, 2000-2019*. <https://www.paho.org/en/enlace/burden-kidney-diseases>
- Vélez Bastidas, W. A. (2023). *APLICACIÓN MÓVIL CON GEOLOCALIZACIÓN PARA GESTIÓN DE LA INFORMACIÓN DE ATENCIÓN MÉDICA DOMICILIARIA UTILIZANDO HERRAMIENTAS DEVOPS*. UNIVERSIDAD NACIONAL DE CHIMBORAZO .
- Villarreal, V., & Samudio, M. (s/f). *Plataforma para la autogestión de datos de pacientes hipertensos en Panamá Platform for the data self-control of patients with arterial hypertension in Panamá*. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

ANEXOS

Etapas de inyección:

Disciplina de requerimientos.

Checklists	CK1: Qualities of Good Requirements CK2: Vision Use-Case Model Supporting Requirements Use Case
Concepts	Supporting Requirements Use Case Requirement Attributes Requirements Traceability Use-Case Model Actor
Guidelines	Detail Use Cases and Scenarios Effective Requirement Reviews Find and Outline Actors and Use Cases Requirements Gathering Techniques Requirement Pitfalls Supporting Requirements Use Case Formats Writing Good Requirements Use-Case Model

Disciplina de arquitectura

Checklists	Architecture Notebook
Concepts	Architectural Mechanism Architecturally Significant Requirements Business Pattern Component Pattern Visual Modeling Executable Architecture Software Architecture
Guidelines	Abstract Away Complexity Architectural View Develop the Architecture Layering Outline the Architecture Architectural Mechanisms Determine Architecturally Significant Requirements Entity-Control-Boundary Pattern

Disciplina de desarrollo

Checklists	Design Implementation
Concepts	Continuous Integration Developer Testing Refactoring Coding Standard Design

	<u>Test-first Design</u>
Guidelines	<u>Transforming design into implementation</u> <u>Continuous Integration</u> <u>Promoting Changes</u> <u>Designing Visually</u> <u>Use-Cases Realizations</u> <u>Refactoring</u> <u>Software Reuse</u> <u>Analyze the Design</u> <u>Evolve the Design</u> <u>Test-first Design</u>

Disciplina de test

Checklists	<ul style="list-style-type: none"> • <u>Test Case</u> • <u>Test Script</u>
Concepts	<ul style="list-style-type: none"> • <u>Types of Test</u> • <u>Failure Analysis and Report Creation</u>
Guidelines	<ul style="list-style-type: none"> • <u>Maintaining Automated Test Suite</u> • <u>Test Suite</u> • <u>Programming Automated Tests</u>