



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN**

**APLICACIÓN MÓVIL CON GEOLOCALIZACIÓN PARA
GESTIÓN DE LA INFORMACIÓN DE ATENCIÓN MÉDICA
DOMICILIARIA UTILIZANDO HERRAMIENTAS DEVOPS**

**Trabajo de Titulación para optar al título de Ingeniero en
Tecnologías de la información**

Autor:

Vélez Bastidas Walter Andrés

Tutor:

PhD. Miryan Estela Narváez Vilema

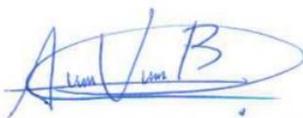
Riobamba, Ecuador. 2023

DECLARATORIA DE AUTORÍA

Yo, Walter Andrés Vélez Bastidas, con cédula de ciudadanía 1719577718, autor del trabajo de investigación titulado: Aplicación móvil con geolocalización para gestión de la información de atención médica domiciliaria utilizando herramientas DevOps, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 07 de noviembre de 2023.



Walter Andrés Vélez Bastidas
C.I: 1719577718



ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 02 días del mes de agosto de 2023, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **WALTER ANDRÉS VÉLEZ BASTIDAS** con CC: **1719577718**, de la carrera de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado “**APLICACIÓN MÓVIL CON GEOLOCALIZACIÓN PARA GESTIÓN DE LA INFORMACIÓN DE ATENCIÓN MÉDICA DOMICILIARIA UTILIZANDO HERRAMIENTAS DEVOPS**”, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



firmado electrónicamente por:
MIRYAN ESTELA
NARVAEZ VILEMA

PhD. Miryan Estela Narváez Vilema
TUTORA

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

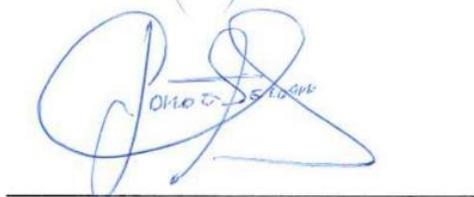
Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Aplicación móvil con geolocalización para gestión de atención médica domiciliaria utilizando herramientas DevOps, presentado por Walter Andrés Vélez Bastidas, con cédula de identidad número 1719577718, bajo la tutoría de la PhD. Miryan Estela Narváez Vilema; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 07 días del mes de noviembre de 2023.

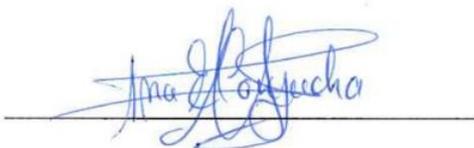
Lady Espinoza, Mgs.
PRESIDENTE DEL TRIBUNAL DE GRADO



Jorge Delgado, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO



Ana Congacha, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO





CERTIFICACIÓN

Que, **VÉLEZ BASTIDAS WALTER ANDRÉS** con CC: **1719577718**, estudiante de la Carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**APLICACIÓN MÓVIL CON GEOLOCALIZACIÓN PARA GESTIÓN DE LA INFORMACIÓN DE ATENCIÓN MÉDICA DOMICILIARIA UTILIZANDO HERRAMIENTAS DEVOPS**", cumple con el 3 %, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 11 de octubre de 2023



firmado electrónicamente por
**MIRYAN ESTELA
NARVAEZ VILEMA**

PhD. Estela Narváez
TUTOR(A)

DEDICATORIA

A Dios, fuente inagotable de sabiduría y fortaleza, y a mis amados padres, dedico este proyecto de investigación. En cada paso de este camino, su guía, amor incondicional y apoyo constante han sido mi faro y refugio. A ustedes, mi roca eterna y fuente de inspiración, rindo homenaje por inspirarme a alcanzar esta meta.

Walter Andrés Vélez Bastidas

AGRADECIMIENTO

Al finalizar este proyecto, mi corazón se llena de gratitud hacia todos los que han hecho posible esta labor.

A mis padres, Walter y Mary, merecen mi agradecimiento más profundo. Su amor inquebrantable, su paciencia sin límites y su aliento constante han sido la mayor motivación en este viaje. Gracias por creer en mí, incluso cuando cometí errores y enfrenté fracasos.

A mi hermana Gardenia y su esposo Daniel han mostrado una generosidad y hospitalidad inmensas al abrirme las puertas de su hogar. Cada gesto de apoyo ha dejado una huella imborrable en mi ser. Sin su apoyo, este logro no hubiera sido posible.

A mi tutora, la PhD. Estela Narváez, ha sido una guía invaluable durante el desarrollo de mi tesis. Su orientación y correcciones han sido fundamentales para mi crecimiento académico. Aprecio enormemente su paciencia y profesionalismo a lo largo de nuestra colaboración. Su dedicación ha dejado una impresión duradera en mi experiencia de investigación.

A mis profesores, cuya experiencia y orientación me han guiado a través de este proceso, les debo mucho. Sus valiosas enseñanzas y consejos han sido fundamentales para mi crecimiento académico y personal.

A mis amigos y compañeros de estudio, que han compartido conmigo momentos de alegría y apoyo en los desafíos, les agradezco de corazón. Su compañía ha hecho que este viaje sea más significativo.

Finalmente, agradezco a Dios, que ha sido mi guía y fortaleza en cada etapa de este proyecto. Su presencia ha sido un faro de esperanza en los momentos de dificultad y una fuente de inspiración en mi búsqueda de conocimiento.

Walter Andrés Vélez Bastidas

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO ANTIPLAGIO

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

ABSTRACT

CAPÍTULO I. INTRODUCCIÓN.....	17
1.1 Planteamiento y justificación de la investigación.....	19
1.2 Formulación del problema.....	19
1.3 Objetivos.....	19
1.3.1 General.....	19
1.3.2 Específicos.....	19
CAPÍTULO II. MARCO TEÓRICO.....	20
2.1 DevOps	20
2.1.1 Desarrollo y operaciones	20
2.1.2 Concepto.....	20
2.1.3 Aspectos fundamentales	21
Fases	21
Prácticas (Todo continuo).....	22
Herramientas DevOps.....	23
2.2 Aplicaciones móviles.....	26
2.3 Geolocalización	26
2.3.1 Sistema de posicionamiento global (GPS)	26
2.3.2 APIs de geolocalización	27
2.3.2.1 Google Maps	27
2.4 Gestión de la información.....	27

2.5	Atención médica domiciliaria.....	28
2.6	Sistema gestor de base de datos (SGBD)	28
2.6.1	Bases de datos SQL y No SQL.....	28
2.7	Metodología de desarrollo de software	29
2.7.1	Metodologías tradicionales y ágiles	29
2.7.2	Mobile-D	29
2.7.3	Inception Deck.....	30
2.8	Modelo de calidad	30
2.8.1	FURPS	30
CAPÍTULO III. METODOLOGÍA.....		32
3.1	Identificación de variables.....	32
3.1.1	Variable independiente	32
3.1.2	Variable dependiente	32
3.2	Tipo y diseño de la investigación	32
3.2.1	Según la fuente de investigación	32
3.2.2	Según el objeto de estudio	32
3.3	Operacionalización de variables.....	33
3.4	Unidad de análisis.....	34
3.5	Población y muestra	34
3.6	Técnicas e instrumentos de recolección de datos	34
3.6.1	Entrevistas	34
3.7	Técnicas de análisis e interpretación de la información	34
3.8	Desarrollo de la propuesta	34
3.8.1	Exploración.....	35
3.8.1.1	Why are we here?.....	35
3.8.1.1.1	Requerimientos funcionales.....	35
3.8.1.1.2	Requerimientos no funcionales.....	36
3.8.1.2	Elevator Pitch	36
3.8.1.3	Product Box.....	37
3.8.1.4	Not List.....	37
3.8.1.5	Meet Your Neighbors.....	38
3.8.2	Inicialización	38
3.8.2.1	Show your Solution.....	38

Arquitectura del sistema	38
Arquitectura de implementación	39
Diagrama de base de datos	39
3.8.2.2 What Keeps us Up at Night?	39
3.8.2.3 Size it up.....	40
3.8.2.4 What’s Going to give?.....	41
3.8.2.5 What’s Going to Take?	42
3.8.2.6 Herramientas de desarrollo.....	42
3.8.3 Producción y Estabilización	43
3.8.3.1 Entorno DevOps	43
Docker	43
Jenkins	45
Git y GitHub	46
SonarQube	47
3.8.3.2 Desarrollo	48
3.8.3.2.1 Diseño	48
3.8.4 Pruebas	53
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN	54
Parámetros de evaluación	54
Dispositivo utilizado para la evaluación.....	54
4.1 Resultados.....	54
4.1.1 Simulación en el dispositivo móvil	55
4.1.2 Análisis de los resultados	55
Comparación de valores obtenidos y establecidos en el modelo FURPS	55
4.1.2.1 Eficacia.....	56
4.1.2.2 Tiempo de respuesta.....	56
4.1.2.3 Utilización de recursos	57
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.	60
5.1 Conclusiones.....	60
5.2 Recomendaciones	60
BIBLIOGRAFÍA	62
ANEXOS	66
Anexo 1 Decoradores para establecer relaciones entre entidades en TypeORM.....	66

Anexo 2 Documentación de Api Rest en Nestjs con Swagger.....	66
Anexo 3 Principales pantallas de la aplicación móvil.	67
Anexo 4 Pruebas de JMeter para la aplicación móvil.	68

ÍNDICE DE TABLAS

Tabla 1. Definición de las fases del ciclo de vida de DevOps.....	21
Tabla 2. Práctica de DevOps; evolución continua.....	23
Tabla 3. Categorías de herramientas DevOps.....	24
Tabla 4. Segmentos del sistema GPS	27
Tabla 5. Visión general de las bases de datos SQL y NoSQL.....	28
Tabla 6. Modelo de calidad FURPS	31
Tabla 7. Operacionalización de variables	33
Tabla 8. Técnicas e instrumentos para la recolección de datos	34
Tabla 9. Requerimientos funcionales	35
Tabla 10. Requisitos no funcionales	36
Tabla 11. Not List.....	37
Tabla 12. Meet your neighbors.....	38
Tabla 13. What keeps us up at night?.....	40
Tabla 14. Roadmap de las iteraciones	40
Tabla 15. Ecuador de factores.....	41
Tabla 16. Presupuesto.....	42
Tabla 17. Parámetro de evaluación.....	54
Tabla 18. Características del dispositivo móvil.....	54
Tabla 19. Medición indicadores dispositivo móvil	55
Tabla 20. Resultados obtenidos de la evaluación del dispositivo móvil	55
Tabla 21. Comparación de valores obtenidos con el modelo FURPS	56

ÍNDICE DE FIGURAS

Figura 1. Fases del ciclo de vida de DevOps	21
Figura 2. Herramientas DevOps	24
Figura 3. Proceso de Mobile-D	30
Figura 4. El Inception Deck	30
Figura 5. Product Box	37
Figura 6. Arquitectura del sistema.....	38
Figura 7. Arquitectura de implementación	39
Figura 8. Diagrama de base de datos.....	39
Figura 9. Contenedores utilizados en el desarrollo de la aplicación móvil	44
Figura 10. Archivo Dockerfile	44
Figura 11. Archivo Docker Compose.....	44
Figura 12. Arquitectura de implementación del Backend.....	45
Figura 13. Arquitectura de implementación de Jenkins	45
Figura 14. Configuración Jenkins	46
Figura 15. Arquitectura de implementación de GitHub	46
Figura 16. Repositorio GitHub backend (Nestjs) y frontend (Flutter)	47
Figura 17. GitHub Webhooks y Ngrok	47
Figura 18. Configuración y resultados de SonarQube.....	48
Figura 19. Diseño conceptual de la aplicación.....	48
Figura 20. Diseño de interfaz de la aplicación móvil herramienta NinjaMock y Figma ...	50
Figura 21. Pruebas de la API REST realizadas en Postman	52
Figura 22. Código de implementación framework Nestjs.....	53
Figura 23. Resultado de eficacia	56
Figura 24. Resultado de tiempo de respuesta	57
Figura 25. Resultado CPU.....	57
Figura 26. Resultado RAM	58
Figura 27. Resultado Almacenamiento	58
Figura 28. Resultado promedio de utilización de recursos.....	59

RESUMEN

El proyecto de investigación tuvo como objetivo desarrollar una aplicación móvil para profesionales sanitarios que atienden a pacientes en sus domicilios, permitiéndoles registrar y consultar información médica relevante, como historiales clínicos y ubicación geográfica.

Para el desarrollo de la aplicación móvil, se utilizó un entorno DevOps con herramientas y prácticas que facilitan la automatización, agilidad y mejora continua del ciclo de vida del software, optimizando el tiempo de desarrollo y evitando tareas repetitivas. Entre las herramientas se encuentran Git, GitHub, Jenkins, Docker y SonarQube. Además, se emplearon tecnologías y herramientas como JWT (JSON Web Token) para la autenticación y autorización de usuarios, el framework Nest para el desarrollo del backend y el framework Flutter para el desarrollo del frontend. Estas tecnologías permitieron crear una aplicación segura, escalable y con una interfaz de usuario atractiva y fácil de usar.

El proyecto se basó en la metodología ágil Mobile-D, diseñada específicamente para el desarrollo de aplicaciones móviles. Esta metodología proporcionó una estructura flexible y adaptable para el desarrollador, permitiendo una gestión eficiente de los requisitos y una rápida respuesta a posibles cambios.

Además, se aplicó el modelo de calidad FURPS para evaluar aspectos fundamentales del software, como el rendimiento, considerando indicadores como la eficacia, tiempo de respuesta y consumo de recursos (CPU, RAM y almacenamiento). Para obtener esta evaluación, se realizó simulaciones con peticiones en tiempo real desde un dispositivo móvil Android mediante el software Apache JMeter.

Los resultados de las simulaciones mostraron que la aplicación móvil obtuvo una alta puntuación según el modelo de calidad FURPS, convirtiéndose en un software altamente recomendable por su excelente rendimiento y cumplimiento con los estándares de calidad establecidos.

Palabras claves: Aplicación móvil, profesionales sanitarios, entorno DevOps, Mobile-D, modelo de calidad FURPS

Abstract

This research project aimed to develop a mobile application for health professionals who provide home care to patients, enabling them to record and access relevant medical information, such as medical histories and geographic location. A DevOps environment was used to develop the mobile application, with tools and practices that facilitate automation, agility, and continuous improvement of the software life cycle, optimizing the development time and avoiding repetitive tasks. The tools are Git, GitHub, Jenkins, Docker, and SonarQube. In addition, technologies, and tools such as JWT (JSON Web Token) for user authentication and authorization, the Nest framework for backend development, and the Flutter framework for frontend development were used. These technologies allowed the creation of a secure, scalable, and user-friendly application with an attractive user interface. The project was based on the agile methodology Mobile-D, explicitly designed for mobile application development. This methodology provided the developer with a flexible and adaptable structure, efficiently managing requirements and quickly responding to possible changes. In addition, the FURPS quality model was applied to evaluate fundamental aspects of the software, such as performance, considering indicators such as effectiveness, response time, and resource consumption (CPU, RAM, and storage). Simulations were performed with real-time requests from an Android mobile device using the Apache JMeter software to obtain this evaluation. The results of the simulations showed that the mobile application obtained a high score according to the FURPS quality model, becoming a highly recommended software for its excellent performance and compliance with the established quality standards.

Keywords: Mobile application, health professionals, DevOps environment, Mobile-D, FURPS quality model



Reviewed by:

Lic. Jenny Freire Rivera

ENGLISH PROFESSOR

C.C. 0604235036

CAPÍTULO I. INTRODUCCIÓN

Las tecnologías digitales han experimentado un gran avance en la última década, facilitando su integración en diversos ámbitos de la sociedad, como educación, economía, negocios, salud, entre otros. En el ámbito de la salud, las tecnologías digitales han generado un cambio significativo en la forma de gestionar procesos relacionados con el acceso a la información médica, el diagnóstico y tratamiento de las enfermedades. Esto implica que se puede brindar una atención médica básica y de calidad a las personas sin que sea necesario que se desplacen a un centro de salud, por ejemplo, mediante el uso de telesalud para consultas virtuales. La Organización Mundial de la Salud reconoce que aprovechar el potencial de las tecnologías digitales es fundamental para lograr la cobertura sanitaria universal y para estar preparados ante futuras pandemias. Un ejemplo de cómo las tecnologías digitales están transformando el mundo con mayor rapidez y profundidad que cualquier otra innovación es la tecnología móvil. Entre sus aplicaciones, se destacan aquellas que se centran en el monitoreo y manejo de enfermedades, el apoyo a la toma de decisiones clínicas, la educación sanitaria, etc. Estas aplicaciones han demostrado ser efectivas en la comunicación entre pacientes y profesionales de la salud, mejorando la calidad y la eficiencia de los servicios sanitarios. Por ello, es importante implementar estrategias para acelerar y fomentar la adopción y soluciones de salud digital [1].

La salud electrónica (e-Salud) es un concepto amplio que implica soluciones digitales para sistemas de vigilancia, prevención, promoción y atención de la salud. Dentro de este concepto se encuentra la salud móvil (m-Salud), que se refiere al uso de dispositivos móviles como teléfonos inteligentes o tabletas para apoyar los procesos sanitarios [2]. Sin embargo, las aplicaciones móviles en salud enfocadas al agendamiento de citas médicas no han tenido una gran acogida en los países de América Latina como Ecuador. Esto se debe a que algunas de estas aplicaciones carecen de características que mejoren la interacción con el usuario, como una interfaz amigable, una navegación sencilla, una personalización adecuada, etc. Esto representa un problema que limita el aprovechamiento de las ventajas que ofrecen las aplicaciones móviles en salud, como una mayor accesibilidad, comodidad, rapidez y seguridad [3].

La presente investigación propone el desarrollo de una aplicación móvil para gestionar la información de atención médica domiciliaria. Esta aplicación se desarrollará utilizando herramientas DevOps como Git, GitHub, Jenkins, Docker y SonarQube. Estos recursos proporcionan un entorno de desarrollo eficiente y colaborativo, permitiendo un control de versiones efectivo, una integración continua y automatizada y un análisis de código estático y mejora de la calidad. La aplicación móvil permitirá a los profesionales de la salud registrar y consultar la información médica de los pacientes que reciben atención médica domiciliaria. Cabe mencionar que no existe una buena difusión de aplicativos para visitas médicas que reúnan los datos de historia clínica de los pacientes desde la perspectiva del profesional de la salud. Aunque hay aplicaciones móviles que se enfocan en el seguimiento de tratamientos médicos, estas se limitan a enfermedades específicas y no ofrecen la opción de crear y

completar la historia clínica del paciente con opciones de seguimiento y evolución. Además, estas aplicaciones presentan desafíos en su utilización, sin tener en cuenta las limitaciones de tiempo o dificultades que los profesionales de la salud puedan tener para el registro de datos de los pacientes. Por esta razón, se busca ayudar tanto a los médicos como a los pacientes para una atención médica domiciliaria de calidad.

El trabajo está organizado con la siguiente estructura: El Capítulo I describe el planteamiento del problema. El Capítulo II presenta el marco teórico, que permite conocer los conceptos relacionados con la temática abordada. El Capítulo III explica la metodología utilizada y las herramientas DevOps aplicadas en el desarrollo de la aplicación móvil. El Capítulo IV muestra los resultados obtenidos al evaluar el rendimiento del aplicativo mediante el modelo de calidad FURPS. Finalmente, el Capítulo V expone las conclusiones y recomendaciones derivadas de la investigación.

1.1 Planteamiento y justificación de la investigación

El país demanda de servicios de salud a domicilio ya sea en personas que requieran asistencia como consecuencia del COVID-19, así como también cualquier otro tipo de padecimiento que pueda sufrir el ser humano.

En Ecuador se emitió un documento de estandarización con lineamientos para la atención ambulatoria y domiciliaria de pacientes con COVID-19. Este documento es de cumplimiento obligatorio para todos los profesionales del Ministerio de Salud Pública en todos los niveles de atención. El objetivo es reducir la demanda de atención presencial en los establecimientos de salud de primer, segundo y tercer nivel [4].

Los problemas identificados en el proceso de visitas domiciliarias que realizan los médicos a los pacientes se centran en el registro de horas de atención, geolocalización de la vivienda, agendamiento de citas e ingreso y actualización del historial clínico, esto conlleva a un mal uso del recurso tiempo, poco orden de los registros de pacientes y mayor carga de trabajo para el profesional de la salud, porque la gestión de esta información la realiza manualmente.

Se da una solución tecnológica mediante el desarrollo de una aplicación móvil para la gestión de la información de atención médica domiciliaria, utilizando herramientas DevOps. Además, en esta aplicación móvil se integra Google Maps API para una mejor gestión de la geolocalización, permitiendo a los profesionales de la salud agilizar los problemas anteriormente identificados, en el proceso de gestión de la información de sus pacientes al realizar la atención médica domiciliaria.

1.2 Formulación del problema

En base al modelo de calidad FURPS, ¿en qué medida, una aplicación móvil con geolocalización mejoraría la gestión de información de atención médica domiciliaria?

1.3 Objetivos

1.3.1 General

Desarrollar una aplicación móvil con geolocalización para gestión de la información de atención médica domiciliaria utilizando herramientas DevOps.

1.3.2 Específicos

- Investigar herramientas DevOps para el desarrollo de aplicaciones móviles.
- Diseñar una aplicación móvil con geolocalización para la gestión de información de atención médica domiciliaria.
- Evaluar el rendimiento de la aplicación móvil utilizando el modelo de calidad FURPS.

CAPÍTULO II. MARCO TEÓRICO

2.1 DevOps

2.1.1 Desarrollo y operaciones

En primer lugar, es importante considerar cómo se organizan los equipos en los diferentes departamentos de tecnología de una empresa que aún no ha implementado DevOps. Un estudio realizado por Lawakatara et al. [5], ofrece una descripción detallada de cómo se implementa DevOps en la práctica, incluyendo el caso de una empresa europea de ciberseguridad. El líder del equipo profesional de esta empresa donde se implementó DevOps afirmó lo siguiente:

Se originó hace unos cuatro años. Teníamos un equipo de operaciones y otro de desarrollo de sistemas. Esos equipos trabajaban muy estrechamente entre sí, pero también tenían objetivos y metas diferentes. El éxito y el fracaso se medían de distintas maneras. En operaciones se trataba más de mantener los sistemas en funcionamiento y no perturbar nada, y en desarrollo, de acelerar el desarrollo y poner las cosas en producción. Eran objetivos contrapuestos, y eso creaba fricciones. Así que empezamos a romperlas [5].

La principal fuerza impulsora detrás de esta noción, que Patrick Debois empleó por primera vez en su conferencia sobre "Infraestructura y operaciones ágiles" en 2008, fue la necesidad de abordar las ineficiencias en las fases tradicionales de desarrollo de software [6][7]. Para Hrusto et al. [8], DevOps ha evolucionado recientemente hasta convertirse en una solución ampliamente aceptada para eliminar las barreras entre los equipos de desarrollo y operaciones.

2.1.2 Concepto

Como señala Azad y Hyrynsalmi [9], es importante tener en cuenta que este término se ha interpretado y descrito de diversas maneras debido a la ambigüedad percibida de su significado para diferentes autores (filosofía, movimiento, cultura, enfoque etc.) e indica que la forma más efectiva de comprender DevOps es como un paradigma que abarca un método, un conjunto de principios y prácticas que fomentan la comunicación y colaboración entre los miembros de un equipo, facilitando así el trabajo efectivo entre desarrolladores y operadores de TI.

De acuerdo con Pennington [10], "DevOps es un conjunto de herramientas y prácticas que ayudan a las organizaciones a construir, probar e implementar software de manera más confiable y a un ritmo más rápido."

Leite et al. [11], combina las definiciones más citadas de DevOps y elabora su propia definición: "DevOps es un esfuerzo colaborativo y multidisciplinar dentro de una organización para automatizar la entrega continua de nuevas versiones de software garantizando su corrección y fiabilidad."

Se debe considerar que esta última definición introduce otro nuevo concepto llamado “entrega continua” que se detallará más adelante.

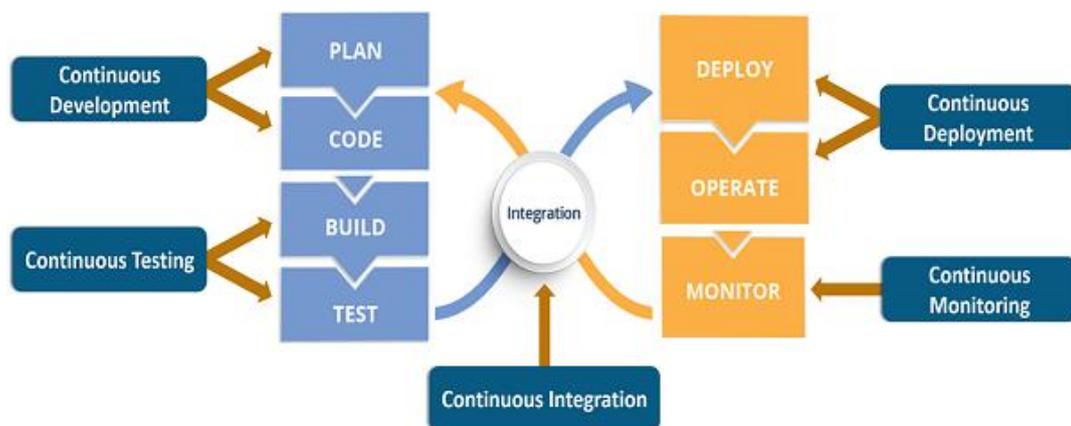
2.1.3 Aspectos fundamentales

El principal objetivo de DevOps es reducir el tiempo de desarrollo de las aplicaciones y acelerar la publicación de nuevas funciones solicitadas por los clientes, para ello se constituye de fases y herramientas que fomentan su adopción [12].

Fases

Cuando se trata de DevOps, es útil dividir el proceso en fases que conforman una canalización de DevOps. Esto permite desglosar y describir las herramientas y procesos utilizados en cada fase (ver Figura 1). Aunque diferentes personas pueden dividir la canalización en distintos puntos, el resultado generalmente describe el mismo proceso. Lo esencial es mantener una terminología consistente dentro de la organización para que todos estén al tanto [13].

Figura 1. Fases del ciclo de vida de DevOps



Fuente: <https://www.edureka.co/blog/devops-lifecycle/#whatisdevops lifecycle>

La Tabla 1 muestra las definiciones de cada una de las fases:

Tabla 1. Definición de las fases del ciclo de vida de DevOps

Equipo de desarrollo	
Planificación	Se definen los objetivos y las especificaciones de creación del software, así como el calendario preliminar de versiones y actualizaciones a lo largo de varias iteraciones.
Codificación	Los desarrolladores utilizan herramientas y complementos en sus entornos de desarrollo para promover buenas prácticas de codificación y prevenir fallas de seguridad y anti-patrones de código, reduciendo el número de compilaciones fallidas y problemas en las pruebas.

Compilación	Los desarrolladores confirman su código en un repositorio compartido y otro desarrollador lo revisa. Un proceso automatizado construye el código y ejecuta pruebas para identificar regresiones. Si hay errores, se notifica al desarrollador. Este proceso continuo minimiza los problemas de integración y resalta los errores temprano.
Pruebas	Después de compilar con éxito, la aplicación se prueba manual y automáticamente en un entorno de ensayo. Las pruebas manuales incluyen pruebas de aceptación del usuario y las automatizadas pueden incluir análisis de seguridad y rendimiento. Esto permite agregar nuevas pruebas sin afectar el flujo de trabajo ni el entorno de producción.
Equipo de operaciones	
Despliegue	Se centra en la reimplantación del software en el entorno de producción cuando una compilación está lista para ser lanzada, existen herramientas y procesos que pueden automatizar el proceso para garantizar una implementación confiable y sin interrupciones.
Funcionamiento	Después del despliegue, se gestiona la configuración de la aplicación y el entorno se ajusta automáticamente para manejar cambios en la demanda. Se recopilan comentarios de los clientes para mejorar el desarrollo del producto. Este ciclo de retroalimentación es valioso para mejorar el producto.
Monitoreo	Se monitorea el entorno y se recopila información sobre el comportamiento, rendimiento y errores del cliente y la canalización de DevOps para mejorar el producto y el proceso de desarrollo. El ciclo de DevOps es un proceso continuo que termina cuando el producto ya no es necesario.

Fuente: <https://doi.org/10.29057/esh.v10i19.8121>

<https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba>

<https://ieeexplore.ieee.org/document/9373305>

Es importante no caer en el error de pensar en DevOps como un ciclo en cascada, donde las etapas están separadas por límites rígidos y solo se puede comenzar una etapa después de que la anterior haya finalizado por completo [14].

Prácticas (Todo continuo)

En el contexto de DevOps, es común escuchar términos relacionados con la continuidad, esto se debe a que la continuidad es un elemento clave en el proceso de DevOps. Cada uno de estos términos está relacionado con las diferentes fases de la canalización de DevOps y es importante entender cómo se relacionan entre sí como se muestra en la Tabla 2.

Tabla 2. Práctica de DevOps; evolución continua

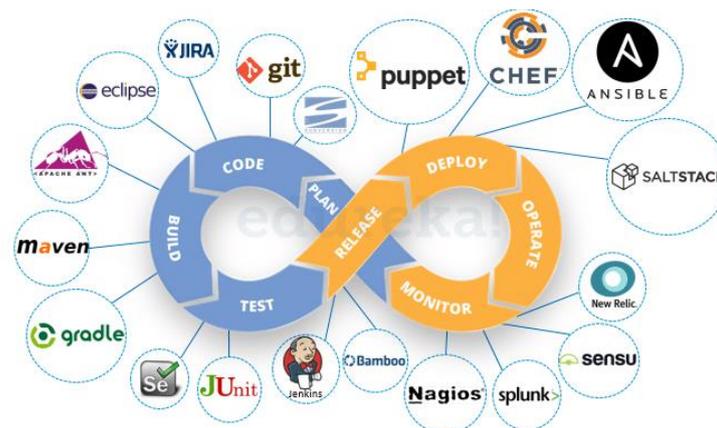
Evolución continua	
Integración continua	Combina varios pasos, como la compilación de código y la validación de la cobertura de código, para proporcionar retroalimentación rápida a los desarrolladores al detectar errores de integración. Los desarrolladores integran su trabajo diariamente en un repositorio común y cada integración se prueba para garantizar que el sistema siga funcionando correctamente.
Entrega continua	Es una extensión de la integración continua que automatiza el proceso de implementación de nuevas versiones en producción. Se alinea con las fases de prueba y lanzamiento de la canalización de DevOps y permite a las organizaciones controlar manualmente cuándo se lanzan nuevas versiones.
Desarrollo continuo	Combina la planificación y la codificación. Durante esta fase, se establece el objetivo del proyecto y los desarrolladores comienzan a escribir código para la aplicación utilizando herramientas.
Pruebas continuas	La aplicación desarrollada se somete a pruebas continuas para detectar errores. Existen herramientas de automatización para facilitar este proceso. Las pruebas pueden realizarse durante todo el ciclo de vida del proyecto o adaptarse y ejecutarse continuamente según los requisitos específicos.
Supervisión continua	La supervisión del rendimiento de una aplicación es de vital importancia para los desarrolladores de aplicaciones. En esta fase, los desarrolladores registran datos mediante el uso de la aplicación y monitorean continuamente cada función.
Despliegue continuo	Es una versión avanzada de la entrega continua que elimina el paso manual de aprobar nuevas versiones en producción. Cada compilación que supera todas las comprobaciones se implementa automáticamente en producción.

Fuente: <https://www.sciencedirect.com/science/article/pii/S0950584923000046>
<https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba>
<https://ijarsct.co.in/Paper8368.pdf>

Herramientas DevOps

Son fundamentales para adoptar DevOps y son útiles en todo el proceso de desarrollo (ver Figura 2) y en la canalización de CI/CD. Son esenciales para automatizar tareas repetitivas y están en constante evolución para satisfacer las demandas de modernización de aplicaciones y transición a la nube [15] [16].

Figura 2. Herramientas DevOps



Fuente: <https://www.edureka.co/blog/devops-tools>

Se utilizan diferentes herramientas para apoyar diferentes aspectos del proceso de desarrollo. La elección de estas herramientas depende principalmente de las características del producto y de las necesidades del cliente o usuario, como se muestra en la Tabla 3 [17].

Tabla 3. Categorías de herramientas DevOps

Herramientas	Definición	Ejemplos
Gestión de código fuente (SCM)	Es un conjunto de prácticas para rastrear cambios en el código fuente, crear versiones y permitir la colaboración entre equipos. SCM puede usarse para monitorear cambios en cualquier elemento que pueda cambiar durante el desarrollo de software. Las herramientas SCM permiten compartir código y trabajar desde diferentes ubicaciones.	GitHub es la herramienta más popular para el control de versiones debido a su soporte para sistemas distribuidos y su modelo freemium y de código abierto. Otras herramientas populares para la gestión de código fuente incluyen Subversion, Mercurial y Bitbucket.
Compilación	Compilar es el proceso de crear un programa ejecutable a partir de archivos de código fuente. Esto puede incluir la compilación del código en instrucciones de máquina y el manejo de dependencias, como el uso de bibliotecas externas.	Ant, Maven, Gradle y MS Build. Gradle es la herramienta más utilizada ya que combina características de otras herramientas.
Integración continua (CI)	Los desarrolladores usan herramientas automatizadas para integrar y fusionar código. La elección de herramientas depende	Algunas herramientas populares en esta categoría son Travis CI, Jenkins, TeamCity y Codeship,

	del producto y del cliente. Durante el proceso, el código se prueba y se dan comentarios inmediatos en caso de errores.	siendo Jenkins una de las más utilizadas.
Gestión de la configuración	Mantiene la coherencia de los productos de software. Esto implica el seguimiento de cambios en artefactos y versiones. Los desarrolladores deben seguir el progreso de nuevas características desde su solicitud hasta su implementación.	Hay muchas herramientas disponibles para la gestión de la configuración, como Chef, Puppet y Ansible.
Basadas en la nube	Son utilizadas a menudo por profesionales de DevOps porque facilitan la colaboración y la implementación, apoyando las prácticas de DevOps.	Algunas herramientas populares en la nube incluyen los servicios de Microsoft Azure e IBM, entre otros. Amazon Web Services también ofrece una gama de servicios para DevOps, como Amazon Beanstalk, que permite la implementación continua.
Pruebas automatizadas	Las pruebas son una parte crucial de la canalización automatizada de DevOps y se combinan con el software en la nube para formar lo que se conoce como Pruebas como Servicio. Esto mejora la colaboración y la calidad del producto de software.	DevOps siempre realiza pruebas continuas en combinación con la automatización, utilizando herramientas como Cucumber, Selenium y JMeter.
Contenedores	Son utilizados para desarrollar plataformas y desplegar aplicaciones en la infraestructura, reduciendo el tiempo entre el desarrollo y la producción. Son fáciles de implementar y mantener para los profesionales de DevOps.	Docker Containers y Kubernetes son las herramientas más importantes.
Implementación	Estas herramientas están diseñadas para gestionar la distribución automatizada de nuevas versiones. Después de cada modificación del código fuente, se realizan pruebas y se distribuye una versión actualizada sin necesidad de intervención manual.	La implementación utiliza programas como Capistrano, Jenkins y Ansible.

Monitoreo	Estas herramientas vigilan el uso de la memoria, la RAM y la carga de la CPU en un esfuerzo por abordar los problemas de infraestructura que pueden repercutir en las soluciones empresariales.	Una de las herramientas de monitorización más populares es Nagios. Cacti es una herramienta de monitorización que también se utiliza además de Nagios y NewRelic, Graphite y New Relic.
Colaboración	La base de DevOps es principalmente la confianza, la comunicación abierta y el trabajo en equipo eficaz. Se anima a los equipos a compartir objetivos, ideas y responsabilidades en el marco de DevOps.	Herramientas como Jira y Slack se utilizan sobre todo para la colaboración.

Fuente: <https://bth.diva-portal.org/smash/record.jsf?pid=diva2%3A1333712&dswid=-5346>

Teniendo en cuenta a David y Daniels [18], citado por Delgado [19], menciona que es importante recordar que DevOps no es solo una colección de herramientas. En lugar de las características esenciales de las propias herramientas, es la forma en que se utilizan lo que las define como "DevOps". Los valores, tradiciones y conocimientos que conforman esta cultura son tan cruciales como los instrumentos que se emplean.

2.2 Aplicaciones móviles

Son herramientas de software creadas en distintos lenguajes de programación para cumplir una función específica en dispositivos tecnológicos como teléfonos inteligentes, tabletas, relojes inteligentes, televisores etc. Se caracterizan por ser prácticas, dinámicas y fáciles de instalar y usar [20].

2.3 Geolocalización

Es una solución informática que permite determinar la ubicación de un objeto en un entorno físico (geoespacial) o virtual (Internet). Esta tecnología utiliza dos tipos de datos: información activa basada en el usuario o dispositivo y búsqueda pasiva basada en el servidor o correlación de datos. Estos datos se combinan para proporcionar un resultado de geolocalización más preciso [21].

2.3.1 Sistema de posicionamiento global (GPS)

Es un sistema de radionavegación basado en el espacio. Consta de tres segmentos como se muestra en la Tabla 4:

Tabla 4. Segmentos del sistema GPS

Espacial	Control	Usuarios
La constelación de satélites GPS consta de más de 30 satélites operativos, cada uno equipado con relojes atómicos redundantes. Cada satélite transmite su posición y hora a intervalos regulares.	Los satélites son rastreados por una red de control terrestre. Usan un radar para escanear y confirmar que los satélites están realmente donde se cree que están.	Los receptores GPS interceptan las señales transmitidas por los satélites y determinan su posición calculando cuánto tiempo tardaron las señales en llegar a ellos.

Fuente: https://www.nasa.gov/directorates/heo/scan/communications/policy/what_is_gps

La mayoría de las personas se refieren al sistema que ayuda a determinar una ubicación como ‘GPS’, pero existen otros sistemas de posicionamiento global conocidos colectivamente como ‘Sistemas de Navegación por Satélite Global’ (GNSS). Estados Unidos tiene el sistema GPS, Rusia tiene GLONASS, China tiene BeiDou y la Unión Europea tiene Galileo. Cada uno de estos sistemas es una versión propia y funciona de manera similar [22].

2.3.2 APIs de geolocalización

Son herramientas que permiten localizar dispositivos con alta precisión en cualquier parte del mundo. El desarrollo de tecnologías como el GPS, Wifi, redes móviles inalámbricas y técnicas de identificación de ubicación IP ha llevado a la creación de una amplia variedad de aplicaciones que aprovechan las ventajas de la geolocalización, además, pueden calcular la mejor ruta entre diferentes puntos considerando factores como el tiempo y la distancia, entre otras características [23] [24].

2.3.2.1 Google Maps

Es una de las aplicaciones más populares. Anteriormente era conocida como "Where 2 Technologies", fue adquirida por Google Inc. en 2004 y recibió el nuevo nombre de Google Maps. Esta aplicación utiliza el rastreo GPS civil para localizar personas y lugares. Antes sólo tenía unas pocas funciones que servían únicamente para la navegación, pero hoy en día ofrece un montón de funciones como vista de calles, tiempo estimado de llegada y otras características interesantes. Como resultado, Google Maps es una de las aplicaciones más útiles y fiables [25].

2.4 Gestión de la información

Es la captura, almacenamiento, recuperación y uso adecuado y optimizado de la información. Puede ser de gestión de información personal u organizativa. Su objetivo principal es garantizar que todos los datos pertinentes se almacenen de manera organizada para que sea posible acceder a ellos cuando sea necesario y usarlos para la toma de decisiones [26].

2.5 Atención médica domiciliaria

El Ministerio de Salud Pública de Ecuador [4] la define como: “intervenciones preventivas, de promoción y de atención de salud de las personas, en la que se debe realizar un análisis de la valoración de las necesidades de las familias y su entorno, determinando sus riesgos, fortalezas y relaciones.”

2.6 Sistema gestor de base de datos (SGBD)

Es un programa informático creado para almacenar, recuperar, consultar y gestionar datos. Los datos pueden ser generados, leídos, actualizados y eliminados por entidades autorizadas gracias a interfaces de usuario (UI). Los SGBD son cruciales porque ofrecen a programadores, administradores de bases de datos y usuarios finales una visión consolidada de los datos y liberan a las aplicaciones y usuarios finales de tener que comprender dónde se alojan físicamente los datos [27].

2.6.1 Bases de datos SQL y No SQL

En algunos casos, la información debe estar relacionada entre sí para evitar la duplicación de información y una mejor organización (SQL), en otros casos, para mejorar el rendimiento, algunos desarrolladores no imponen relaciones de datos (No SQL) [28]. En la Tabla 5, se muestran las características de cada base de datos.

Tabla 5. Visión general de las bases de datos SQL y NoSQL

	SQL	NoSQL
Almacenamiento de datos	Almacena datos en forma de tabla, datos representados en un modelo relacional, con filas y columnas.	El término NoSQL engloba la masa de bases de datos, cada una con diversos modelos de almacenamiento de datos. Los predominantes son: documento, columnar, clave-valor y grafo.
Esquemas y flexibilidad	Cada registro sigue un esquema estático, lo que significa que las columnas deben ser absolutas e inaccesibles antes de la entrada de datos y que cada fila debe conservar los datos de cada columna. Esto se puede modificar; sin embargo, implica alterar toda la base de datos.	Los esquemas son dinámicos y los datos se representan regularmente en modo documento, lo que significa que no hay que guardar los datos de cada columna y que se pueden añadir registros fácilmente.
Escalabilidad	En casi todas las circunstancias, las bases de datos SQL son escalables verticalmente, esto quiere decir que podemos cargar	El escalado es horizontal, lo que significa añadir servidores adicionales a una base de datos NoSQL. Esta cantidad de

	datos en un único servidor aumentando el tamaño de los recursos de hardware. En esencia, es factible escalar, pero se trata de un proceso difícil y que requiere mucho tiempo.	servidores puede ser hardware asequible, por lo que resulta mucho más rentable que el escalado vertical.
Cumplimiento de ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)	La gran mayoría de las bases de datos relacionales permiten propiedades ACID.	Varía según la tecnología, pero muchas soluciones NoSQL sacrifican las propiedades ACID en favor de la disponibilidad y la escalabilidad.

Fuente: <http://docs.neu.edu.tr/library/6784727281.pdf>

2.7 Metodología de desarrollo de software

De acuerdo con Rivas et al. [29], es “el estudio y determinación del método más adecuado para incrementar algo, en este caso el software”. Hoy en día, la palabra “desarrollo” se usa con frecuencia para describir tareas que involucran la construcción, producción, actualización o modificación de software.

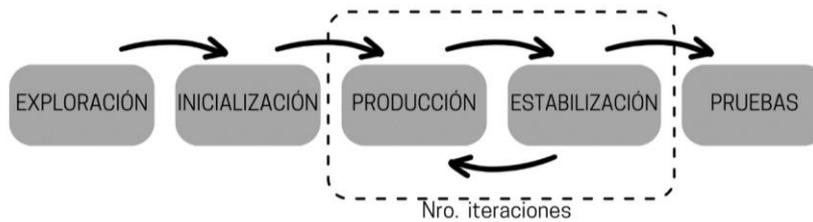
2.7.1 Metodologías tradicionales y ágiles

En el desarrollo de software, las metodologías tradicionales y ágiles presentan dos enfoques opuestos. Las metodologías ágiles enfatizan la participación del cliente en la planificación del proyecto y la singularidad y experiencia del equipo del proyecto, que suele ser más pequeño que en el enfoque tradicional [30]. Por otro lado, la documentación es esencial en la metodología tradicional ya que durante las primeras fases se establecen los requisitos y otros recursos para todo el proyecto y se trabaja de acuerdo con ellos [31].

2.7.2 Mobile-D

Abrahamsson et al. [32], idearon una metodología que citaron Mathur y Satapathy [33], determinaron que el proceso ágil es el mejor para los entornos de desarrollo móvil. Desarrollaron una nueva metodología denominada “Mobile-D”. Esta metodología se basa en la metodología eXtreme Programming (XP), un término de escalabilidad basado en la familia Crystal methodologies y las etapas del ciclo de vida del Rational Unified Process (RUP) y consta de cinco fases (ver Figura 3) [34].

Figura 3. Proceso de Mobile-D

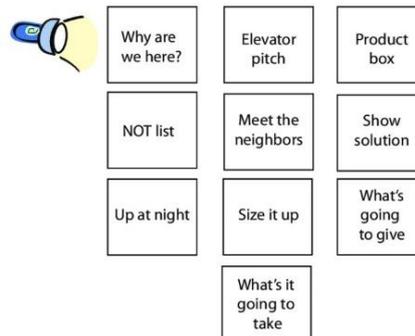


Fuente: <https://doi.org/10.1109/ICCAIS48893.2020.9096751>

2.7.3 Inception Deck

Es una herramienta ágil fundamental en la etapa inicial de un proyecto. Mediante una cuidadosa selección de preguntas y ejercicios (ver Figura 4), su propósito radica en eliminar la incertidumbre y brindar una visión clara del proyecto, estableciendo expectativas precisas y logrando una comprensión compartida entre el equipo y los interesados. Esta herramienta proporciona una manera ágil y eficiente de identificar los aspectos centrales del proyecto y comunicarlos de manera efectiva, fomentando la alineación, la toma de decisiones bien fundamentadas y la colaboración desde el inicio mismo del proyecto [35].

Figura 4. El Inception Deck



Fuente: https://www.oreilly.com/library/view/the-agile-samurai/9781680500066/f_0026.html

2.8 Modelo de calidad

Un modelo de calidad de software contiene varias características (o factores) de calidad y criterios en función de esos factores [36]. Con ellos se puede abordar la calidad de un producto, como por ejemplo el modelo de calidad FURPS.

2.8.1 FURPS

Teniendo en cuenta a Yeomans y Rogers [37], citado por Araujo [38], menciona que, en la actualidad, hay varios modelos disponibles para evaluar la calidad de las herramientas de software. Estos modelos han sido desarrollados por diferentes empresas y organizaciones con el objetivo de determinar si un software cumple con los estándares de calidad, mediante

el análisis de diversas características. Uno de los modelos más destacados debido a su flexibilidad es el modelo FURPS creado por Hewlett Packard. En la Tabla 6 se detalla los criterios asociados a cada factor de calidad que posee este modelo.

Tabla 6. Modelo de calidad FURPS

Atributos	Factores contenidos	Descripción
Funcional	Funcionalidad	El conjunto de características, capacidades y seguridad.
No funcional	Usabilidad	Puede incluir factores humanos, estética, coherencia en la interfaz de usuario, ayuda en línea y sensible al contexto, asistentes y agentes, documentación de usuario y materiales de formación.
	Rendimiento	Condiciones sobre requisitos funcionales como velocidad, eficiencia, disponibilidad, precisión, rendimiento, tiempo de respuesta, tiempo de recuperación y uso de recursos.
	Fiabilidad/ confiabilidad	Frecuencia y gravedad de los fallos, recuperabilidad, previsibilidad, precisión y tiempo medio entre fallos.
	Apoyabilidad/ capacidad de soporte	Testabilidad, extensibilidad, adaptabilidad, mantenibilidad, compatibilidad, configurabilidad, mantenibilidad, instalabilidad, localizabilidad.

Fuente: <https://www.tandfonline.com/doi/full/10.1080/20476965.2022.2162445>

El modelo FURPS no solo proporciona métricas de calidad, sino que también puede adaptarse como una herramienta de clasificación de requisitos de software. FURPS establece una puntuación en tres niveles:

- **Nivel alto (high):** Este nivel se alcanza cuando el software cumple entre el 95% y el 100% de los requisitos evaluados. En este caso, el software es altamente recomendado debido a su excelente rendimiento y cumplimiento de los estándares de calidad.
- **Nivel moderado (moderate):** En este nivel, el software cumple entre el 80% y el 95% de los requisitos. Aunque existen ciertos factores que deben corregirse, el riesgo asociado es controlable. Se recomienda realizar las mejoras necesarias para optimizar el rendimiento y la calidad del software.
- **Nivel bajo (low):** Cuando el cumplimiento del software es inferior al 80%, se encuentra en este nivel. En esta situación, el software no cumple con los factores evaluados y su uso conlleva un alto riesgo. Es fundamental abordar las deficiencias identificadas y mejorar la calidad del producto antes de su implementación.

CAPÍTULO III. METODOLOGÍA

La investigación se centra en realizar un análisis cuantitativo, con el propósito de obtener datos mensurables acerca del rendimiento de la aplicación móvil. Esto será posible gracias a la implementación del modelo de calidad FURPS.

3.1 Identificación de variables

3.1.1 Variable independiente

Aplicación móvil con geolocalización

3.1.2 Variable dependiente

Evaluar el rendimiento de la aplicación móvil para gestión de la información de atención médica domiciliaria.

3.2 Tipo y diseño de la investigación

3.2.1 Según la fuente de investigación

Esta investigación es de tipo bibliográfica, ya que se llevó a cabo una búsqueda exhaustiva de información en revistas, libros, bases de datos científicas, tesis y otras fuentes relacionadas con el tema de investigación. Esto permitió comprender mejor los conceptos relacionados con aplicaciones móviles y herramientas DevOps, brindando una visión más amplia del tema.

3.2.2 Según el objeto de estudio

Es una investigación es de **tipo aplicada**, porque se identifican problemas específicos y concretos en la atención médica domiciliaria que realizan los profesionales de la salud.

Es una **investigación descriptiva** porque se recopilará información cuantificable para analizar y describir los resultados obtenidos tras evaluar el rendimiento de la aplicación con el modelo FURPS.

3.3 Operacionalización de variables

Tabla 7. Operacionalización de variables

Elaborado por: El autor

PREGUNTA DE INVESTIGACIÓN	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACIÓN	DIMENSIÓN	INDICADORES
En base al modelo de calidad FURPS, ¿en qué medida, una aplicación móvil con geolocalización mejoraría la gestión de información de atención médica domiciliaria?	Aplicación móvil con geolocalización para gestión de la información en la atención médica domiciliaria utilizando herramientas DevOps.	<p>Objetivo general</p> <p>Desarrollar una aplicación móvil con geolocalización para gestión de la información de atención médica domiciliaria utilizando herramientas DevOps.</p> <p>Objetivos específicos</p> <ul style="list-style-type: none"> • Investigar herramientas DevOps para el desarrollo de aplicaciones móviles en el área de la salud. • Diseñar una aplicación móvil con geolocalización para la gestión de la información de atención médica domiciliaria. • Evaluar el rendimiento de la aplicación móvil utilizando el modelo de calidad FURPS. 	<p>Independiente:</p> <p>Aplicación móvil con geolocalización.</p> <p>Dependiente:</p> <p>Evaluar el rendimiento de la aplicación móvil para gestión de la información de atención médica domiciliaria.</p>	<p>Aplicación de software que se instala en dispositivos móviles o tabletas para ayudar al usuario en una labor concreta.</p> <p>El rendimiento es la medición objetiva y la experiencia percibida por el usuario del tiempo de carga y el tiempo de ejecución, y en el grado de fluidez del contenido durante las interacciones del usuario.</p>	<p>Desarrollo de la aplicación móvil con geolocalización.</p> <p>Rendimiento al gestionar la información de atención médica domiciliaria utilizando la herramienta Apache JMeter.</p>	<p>Independiente:</p> <p>Numero de módulos. Número de formularios de historia clínica con el estándar CIE-10.</p> <p>Dependiente:</p> <ul style="list-style-type: none"> • Criterios de rendimiento del modelo de calidad FURPS: • Velocidad de procesamiento. • Tiempo de respuesta. • % de consumo de recursos (RAM, procesador).

3.4 Unidad de análisis

El estudio se desarrolló con médicos que laboran en el Hospital Clínica Metropolitana Riobamba.

3.5 Población y muestra

Población: Según el enfoque de investigación planteado, se considera una población infinita, ya que se obtendrán datos de diversas mediciones utilizando JMeter.

Muestra: No aplica debido al tamaño de población que se utiliza para este caso de estudio.

3.6 Técnicas e instrumentos de recolección de datos

La técnica e instrumento que se ha utilizado en la investigación para recopilar datos se detallan en la Tabla 8.

Tabla 8. Técnicas e instrumentos para la recolección de datos

TÉCNICA	INSTRUMENTO
Herramienta	Apache JMeter

Fuente: El autor

3.6.1 Entrevistas

Con el objetivo de recolectar información sobre la gestión de la información de atención médica domiciliaria y los requerimientos funcionales y no funcionales para la aplicación móvil, se ha realizado una entrevista informal a los directivos del Hospital General Clínica Metropolitana Riobamba.

3.7 Técnicas de análisis e interpretación de la información

Se utilizó la herramienta JMeter, para medir el rendimiento de aplicación móvil en diferentes circunstancias basados en el modelo de calidad FURPS como: peticiones, promedio de tiempo de respuesta y uso de recursos.

3.8 Desarrollo de la propuesta

Se describe detalladamente cómo se implementó la metodología Mobile-D para desarrollar una aplicación móvil utilizando herramientas DevOps. El proceso constó de varias fases, comenzando con la exploración para definir el alcance del proyecto. Posteriormente, durante la fase de inicialización, se tomaron medidas para asegurar el éxito del proyecto. En la fase de producción, se llevó a cabo el desarrollo del producto y en la estabilización se verificó que todo funcionara correctamente. Finalmente, en las pruebas se comprobó que el sistema y los requisitos cumplían con los objetivos establecidos.

3.8.1 Exploración

Durante la etapa de exploración, se utilizó el Inception Deck para establecer los requisitos iniciales y la visión general del producto. Con el fin de recopilar esta información de manera efectiva, se llevó a cabo un taller inicial en el que se recopilaron datos relevantes para lograr una conceptualización más detallada del proyecto. Se dio inicio al taller abordando la primera pregunta del Inception Deck.

3.8.1.1 Why are we here?

En respuesta a la primera pregunta del Inception Deck, “¿Por qué estamos aquí?”, estamos aquí con el fin de desarrollar una aplicación móvil con capacidad de geolocalización que facilite a los médicos el registro de la historia clínica de sus pacientes en el ámbito de la atención médica domiciliaria. Se desea proporcionar a los profesionales de la salud una interfaz intuitiva y sencilla para ingresar la información médica de manera eficiente.

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
Identificación del requerimiento	RF03
Nombre del requerimiento	Ingreso de pacientes
Descripción del requerimiento	Se proporcionará la funcionalidad para que los profesionales de la salud ingresen la información y los datos necesarios de sus pacientes, asegurando su correcta identificación.
Prioridad del requerimiento	Alta
Identificación del requerimiento	RF04

Nombre del requerimiento	Gestión de historia clínica
Descripción del requerimiento	Se permitirá ingresar la información del historial clínico de cada paciente, incluyendo diagnósticos, tratamientos y notas relevantes.
Prioridad del requerimiento:	Alta
Identificación del requerimiento	RF05
Nombre del requerimiento	Evolución clínica de forma cronológica
Descripción del requerimiento	Los médicos podrán acceder a la información de la historia clínica de un paciente y actualizarla de manera cronológica para supervisar y controlar su evolución hasta el alta médica.

Fuente: El autor

3.8.1.1.2 Requerimientos no funcionales

Describen criterios del funcionamiento general del sistema (ver Tabla 10).

Tabla 10. Requisitos no funcionales

Id	Descripción del requisito	Categoría
RNF01	La aplicación móvil es compatible con los principales sistemas operativos móviles.	Compatibilidad
RNF02	La aplicación tiene un diseño modular y sigue buenas prácticas de programación para facilitar su mantenimiento y extensión en el futuro.	Mantenibilidad, escalabilidad
RNF03	La aplicación tiene un tiempo de respuesta rápido para todas las interacciones, proporcionando una experiencia fluida y sin demoras perceptibles.	Rendimiento
RNF04	La interfaz de la aplicación es intuitiva, amigable y fácil de usar, con un diseño atractivo y coherente.	Interfaz
RNF05	La aplicación cumple con los estándares y regulaciones de privacidad y seguridad de datos, garantizando la confidencialidad y protección de la información médica.	Seguridad
RNF06	La aplicación es capaz de gestionar múltiples usuarios concurrentes sin degradar su rendimiento y funcionalidad.	Escalabilidad

Fuente: El autor

3.8.1.2 Elevator Pitch

La siguiente técnica en el Inception Deck es el Elevator Pitch, que implica presentar de forma atractiva, breve y veloz la idea central del proyecto:

“A diario, los profesionales de la salud se enfrentan a desafíos en la gestión manual de la información médica en la atención médica domiciliaria, lo que resulta en un mal uso del tiempo, registros de pacientes desorganizados y una carga de trabajo adicional. Sin embargo,

se ha encontrado una solución que ha transformado esta situación. Se presenta una aplicación móvil que cambia la forma en que se gestionaba la información médica. Con esta app, el profesional de la salud puede optimizar su tiempo, organizar de manera eficiente los registros de sus pacientes y reducir su carga de trabajo. Es el momento de simplificar y mejorar la experiencia del profesional de la salud con esta aplicación móvil especializada.”

3.8.1.3 Product Box

Esta técnica se emplea con el fin de crear de manera ágil una representación visual del proyecto. Los beneficios que ofrece al usuario incluyen características específicas y un slogan conciso. Se ha desarrollado el siguiente Product Box como ejemplo en la Figura 5:

Figura 5. Product Box



Fuente: El autor

3.8.1.4 Not List

Dentro del Inception Deck, también se encuentra el Not List, una sección en la que se registran características y funcionalidades que no se tomarán en cuenta como requisitos del producto. Esta metodología asegura que en el futuro no se añadan funcionalidades que carezcan de valor para el producto. Se han identificado las siguientes características en el Not List como se muestra en la Tabla 11.

Tabla 11. Not List

LO QUE NO ESTÁ AL ALCANCE	SIN RESOLVER
Sistema de chat en tiempo real	Implementación de autenticación biométrica
Integración con redes sociales	Interfaz multilingüe
Característica de diagnóstico automático	Funcionalidad de recordatorio de citas
Opciones de pago integradas	Integración con sistemas de gestión hospitalaria existentes

Fuente: El autor

3.8.1.5 Meet Your Neighbors

El objetivo es reconocer a todas las personas que participan en el proyecto y están relacionadas con él (ver Tabla 12).

Tabla 12. Meet your neighbors

Partes interesadas	Responsabilidades	Rol del proyecto
Médicos	Gestionar historiales clínicos de pacientes	Usuario principal
Desarrollador de la aplicación	Diseñar, desarrollar y mantener la aplicación	Equipo técnico

Fuente: El autor

3.8.2 Inicialización

El objetivo principal de esta fase en Mobile-D es asegurar el éxito del proyecto mediante una definición clara de los recursos necesarios y el tiempo requerido para su ejecución. Siguiendo la secuencia de trabajo establecida en la fase de exploración, se dará paso a la ejecución de las técnicas restantes del Inception Deck.

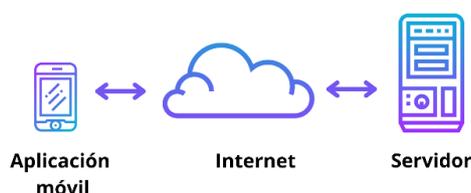
3.8.2.1 Show your Solution

Con esta técnica se presenta la solución al problema planteado de manera clara y concisa. Incluye una descripción de alto nivel de la solución, sus componentes principales y la arquitectura o diseño propuesto.

Arquitectura del sistema

Se empleó una arquitectura cliente-servidor (ver Figura 6), donde el cliente envía solicitudes al servidor y este proporciona respuestas. El sistema incluye un servidor de base de datos que almacena los registros y solicitudes de los usuarios. Los usuarios deben tener acceso a internet para recibir información actualizada.

Figura 6. Arquitectura del sistema

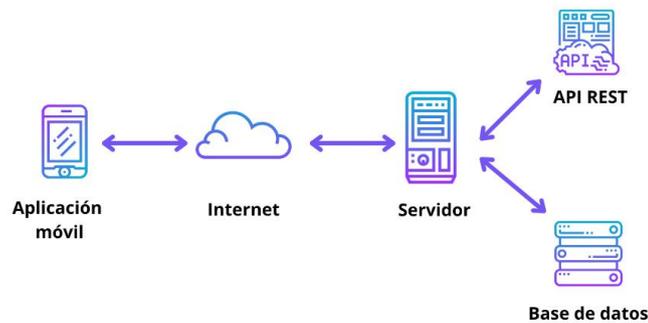


Fuente: El autor

Arquitectura de implementación

Los clientes utilizan dispositivos móviles para acceder a la aplicación móvil, donde esta se comunica con el servidor a través de una API REST. En el servidor, se encuentra alojada la base de datos donde se almacenan los datos relevantes para el funcionamiento de la aplicación como se observa en la Figura 7.

Figura 7. Arquitectura de implementación

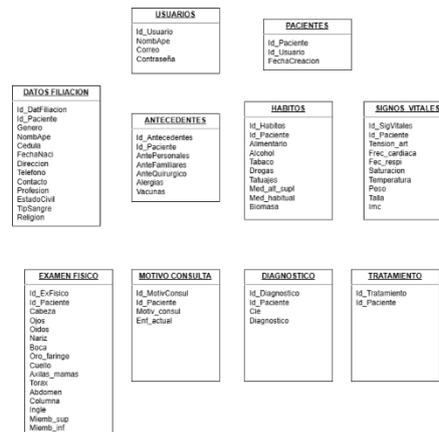


Fuente: El autor

Diagrama de base de datos

Se empleó PostgreSQL como base de datos relacional y TypeORM para establecer relaciones entre entidades (ver Anexo 1). Esto proporcionó un modelo de datos coherente y eficiente, con un almacenamiento sólido y una gestión simplificada, como se muestra en la Figura 8.

Figura 8. Diagrama de base de datos



Fuente: El autor

3.8.2.2 What Keeps us Up at Night?

El propósito de esta técnica es identificar y comprender los riesgos potenciales que pueden afectar al proyecto, priorizando aquellos con un impacto significativo. Se busca tomar medidas preventivas y establecer estrategias de mitigación desde el inicio del proyecto para minimizar los riesgos y asegurar su éxito (ver Tabla 13).

Tabla 13. What keeps us up at night?

Categoría	Riesgos
Riesgos de gestión	Mala planificación del proyecto Falta de formación del desarrollador
Riesgos tecnológicos	Fallas en la aplicación Disponibilidad de los servicios de proveedores Cambios en los servicios de proveedores

Fuente: El autor

En el desarrollo de la aplicación se identificaron los siguientes riesgos:

- **Riesgo de mala planificación del proyecto:** Posibilidad de una planificación inadecuada que genere retrasos y aumento de costos en el desarrollo del producto.
- **Riesgo de falta de formación del desarrollador:** Peligro de que el equipo carezca de los conocimientos necesarios, lo que puede ocasionar retrasos en el proyecto. Se recomienda que el equipo evalúe y mejore sus habilidades mediante cursos y capacitaciones.
- **Riesgo de fallas en la aplicación:** Posibilidad de que ocurran errores o problemas en el funcionamiento de la aplicación. Se espera minimizar este riesgo durante las fases de producción, pruebas y correcciones.
- **Riesgo de disponibilidad de servicios de proveedores:** Riesgo asociado a la dependencia de terceros para servicios como plataformas de pagos, chat y video llamadas, y servidores de alojamiento. La mitigación de este riesgo puede ser compleja debido a la falta de alternativas de respaldo y la dificultad para cambiar de proveedores.
- **Riesgo de cambios en los servicios de proveedores:** Posibilidad de que los proveedores realicen cambios en sus servicios sin previo aviso. Se recomienda mantenerse informado a través de noticias y boletines de actualización para mitigar este riesgo.

3.8.2.3 Size it up

Se busca obtener un Roadmap con las diferentes releases y su respectivo orden de entrega, listando las características obtenidas de los requisitos funcionales (ver Tabla 14). Una Release se define como una entrega de software con funcionalidades completas.

Tabla 14. Roadmap de las iteraciones

Roadmap Viality			
Release 0	Release 1	Release 2	Release 3
DevOps	RF01	RF03	RF04
	RF02		RF05

Fuente: El autor

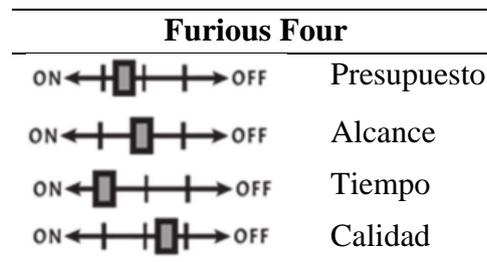
Los objetivos de cada iteración son las siguientes:

- **Release 0:** Configurar entorno de desarrollo con herramientas DevOps.
- **Release 1:** Realizar el registro y autenticación de usuarios.
- **Release 2:** Integrar el ingreso de pacientes.
- **Release 3:** Llevar a cabo la gestión y evolución de historia clínica de cada paciente.

3.8.2.4 What's Going to give?

El objetivo es establecer y negociar con el cliente o stakeholders la prioridad de los factores que pueden influir en el proyecto. Estos factores comúnmente se conocen como los "Furious Four", que incluyen el presupuesto, el alcance, el tiempo y la calidad. Una manera sencilla de visualizar esta negociación de factores es representándola como un ecualizador, donde se busca equilibrar y ajustar cada factor según las necesidades y restricciones del proyecto.

Tabla 15. Ecualizador de factores



Fuente: <https://agilewarrior.wordpress.com/2010/11/06/the-agile-inception-deck/>

A partir de la tabla anterior, se pueden extraer las siguientes conclusiones:

- **Presupuesto:** El presupuesto es flexible gracias al uso de software libre para el desarrollo de la aplicación móvil y las herramientas DevOps, lo que permite reducir costos y ajustar el presupuesto según sea necesario.
- **Alcance:** El alcance del proyecto puede ser modificado si es necesario para adaptarse a cambios en los requisitos o nuevas necesidades que puedan surgir durante el desarrollo.
- **Tiempo:** Se ha fijado una fecha límite para la entrega del proyecto y el plazo es riguroso, lo que significa que es necesario cumplir con los plazos establecidos para asegurar que el proyecto sea entregado a tiempo.
- **Calidad:** La calidad del código es importante y no debe ser comprometida, pero en caso de falta de tiempo, se pueden hacer ajustes para garantizar la calidad del código teniendo en cuenta el plazo de entrega del proyecto.

Los valores de los factores pueden cambiar, pero cualquier modificación afectará a otros. Por ejemplo, si el presupuesto cambia, se debe evaluar su impacto en el alcance o el tiempo.

Hay interdependencia entre los factores, por lo que cualquier ajuste tendrá implicaciones en los demás.

3.8.2.5 What's Going to Take?

Finalmente, se puede estimar el coste del proyecto en función del tiempo, desarrollador y servicios tecnológicos necesarios (ver Tabla 16).

Tabla 16. Presupuesto

Cantidad	Concepto	Cobro Mensual	Total 4 Meses
1	Personal	0 \$	0 \$
2	Servidores	0 \$	0 \$
	Totales	0 \$	0 \$

Fuente: El autor

Al finalizar las herramientas del Inception Deck, se ha alcanzado la meta para la fase de Inicialización. Se tiene una clara comprensión de la infraestructura necesaria, se ha definido y mitigado el riesgo, se ha establecido un roadmap de iteraciones y funcionalidades, y se ha determinado la prioridad de los factores que pueden tener impacto en el proyecto. Además, se ha detallado el presupuesto con el que contará el proyecto.

3.8.2.6 Herramientas de desarrollo

Se utilizaron las siguientes herramientas en el desarrollo del aplicativo móvil:

- **Visual Studio Code:** Popular y versátil editor de código fuente con una amplia gama de características y extensiones para desarrolladores.
- **Node.js:** Entorno de tiempo de ejecución de JavaScript que permite ejecutar código en el lado del servidor y construir aplicaciones web escalables y de alto rendimiento.
- **Flutter:** Framework de desarrollo de aplicaciones móviles de código abierto creado por Google, que permite crear aplicaciones nativas para iOS y Android utilizando un solo código base en el lenguaje de programación Dart.
- **PostgreSQL:** Sistema de gestión de bases de datos relacional de código abierto y altamente escalable que ofrece confiabilidad, flexibilidad y características avanzadas.
- **Docker:** Plataforma de virtualización a nivel de sistema operativo que permite empaquetar, distribuir y ejecutar aplicaciones y sus dependencias en contenedores independientes y portátiles.
- **Git:** Sistema de control de versiones distribuido ampliamente utilizado para el seguimiento de cambios en el código fuente y la colaboración en proyectos de desarrollo de software.

- **GitHub:** Plataforma en línea basada en Git que permite alojar, colaborar y gestionar el control de versiones de proyectos de software, facilitando la colaboración entre desarrolladores.
- **Jenkins:** Herramienta de automatización de código abierto utilizada para la integración y entrega continuas (CI/CD), que automatiza el proceso de compilación, prueba y despliegue de aplicaciones.
- **SonarQube:** Plataforma de análisis estático de código fuente que permite identificar problemas de calidad del código, realizar un seguimiento de métricas y facilitar la revisión continua de la calidad del software.
- **JMeter:** Herramienta de código abierto de Apache para pruebas de carga y evaluación de rendimiento en aplicaciones web, permitiendo simular escenarios de carga y medir tiempo de respuesta y utilización de recursos.
- **Postman:** Plataforma de colaboración para el desarrollo de API que facilita el envío de solicitudes HTTP, la visualización de respuestas, la creación de pruebas automatizadas y la generación de documentación interactiva.
- **TablePlus:** Herramienta de gestión de bases de datos que proporciona una interfaz gráfica fácil de usar para interactuar con diversas bases de datos.

3.8.3 Producción y Estabilización

El propósito de la fase de Producción y Estabilización es implementar la funcionalidad requerida en el producto mediante un ciclo de desarrollo iterativo e incremental. Durante esta etapa, se mostrarán las características previamente establecidas en el roadmap del proyecto

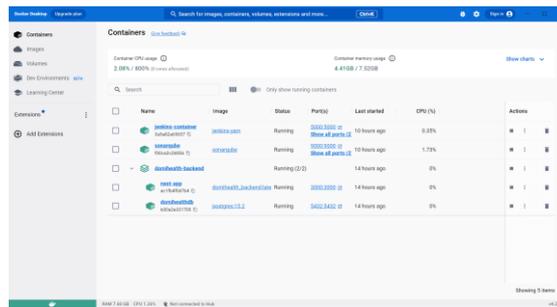
3.8.3.1 Entorno DevOps

La implementación del entorno de herramientas DevOps es esencial para el proyecto, ya que facilita el inicio del desarrollo de la aplicación móvil en el Backend y el Frontend. A continuación, se detallarán las herramientas utilizadas para implementar el entorno DevOps y sus respectivas funciones en el proyecto.

Docker

Para descargar la imagen y el contenedor de la base de datos PostgreSQL, el servidor de integración continua Jenkins y la herramienta de análisis de código SonarQube, se utilizó Docker. Además, Docker permitió empaquetar la API REST en Nestjs y crear un contenedor backend para la API REST junto a la base de datos PostgreSQL, garantizando así un despliegue consistente y portátil. En la Figura 9, se pueden observar los contenedores corriendo en la aplicación Docker Desktop de Windows. La configuración se llevó a cabo de la siguiente manera:

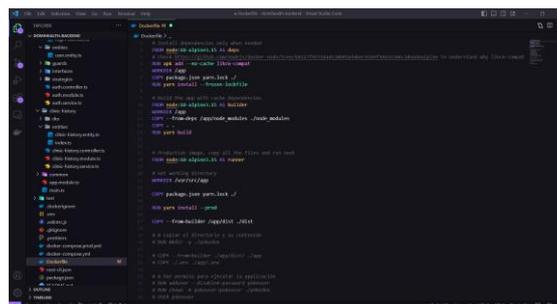
Figura 9. Contenedores utilizados en el desarrollo de la aplicación móvil



Fuente: El autor

- Para el contenedor de PostgreSQL, Jenkins y SonarQube, se descargó la imagen predefinida desde Docker Hub.
- Una vez finalizado el desarrollo de la API REST utilizando el framework Nestjs, se creó un archivo Dockerfile en la raíz del proyecto (ver Figura 10), se definieron los pasos para construir la imagen del contenedor, que incluía la instalación de dependencias y la configuración del entorno necesario para su ejecución.

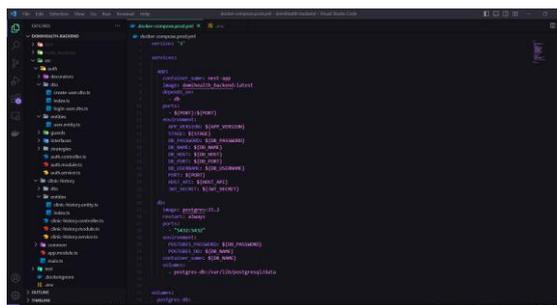
Figura 10. Archivo Dockerfile



Fuente: El autor

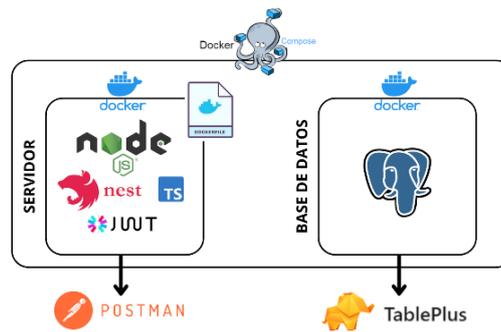
- Como se muestra en la Figura 11, mediante Docker Compose, se unió la imagen creada junto a la imagen de la base de datos y se estableció la comunicación entre los contenedores. Esto permitió crear un entorno de desarrollo cohesivo y reproducible, donde la base de datos PostgreSQL y la API REST de Nestjs trabajaban de manera conjunta como se observa en la Figura 12.

Figura 11. Archivo Docker Compose



Fuente: El autor

Figura 12. Arquitectura de implementación del Backend

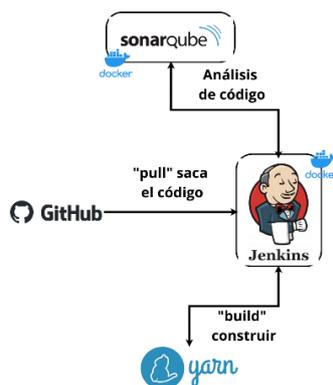


Fuente: El autor

Jenkins

Se utilizó como servidor de integración continua para automatizar las tareas de compilación y pruebas como se muestra en la Figura 13. La configuración se realizó de la siguiente manera:

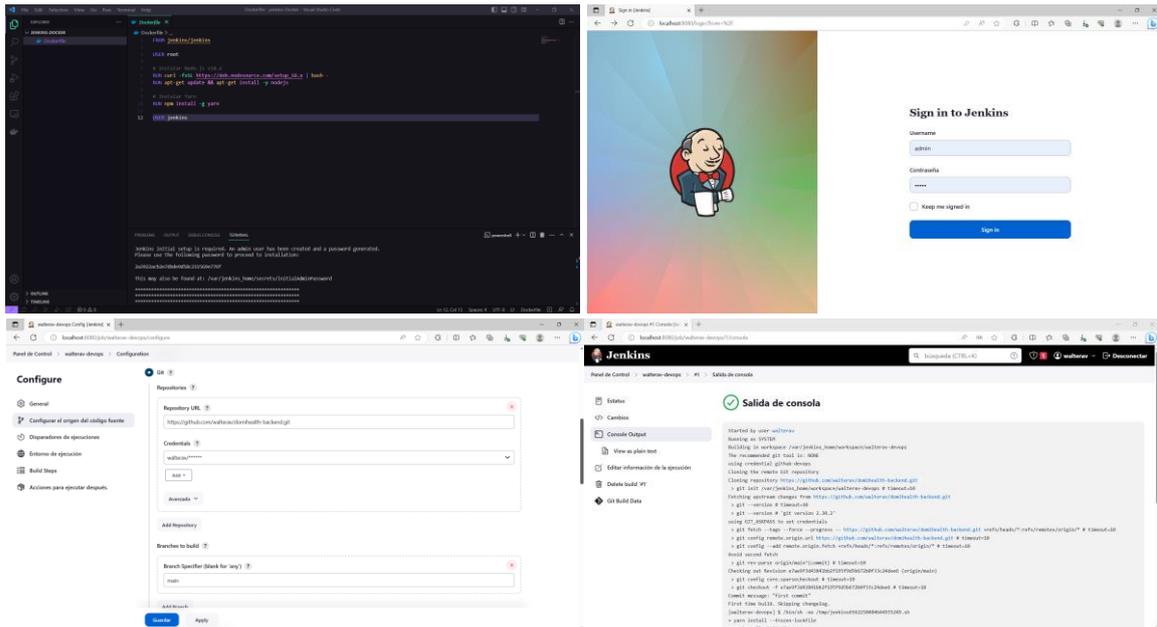
Figura 13. Arquitectura de implementación de Jenkins



Fuente: El autor

- Se configuró el pipeline de Jenkins, donde se definieron los pasos necesarios para cada etapa del ciclo de vida del desarrollo a través de Yarn, Git y GitHub.
- Jenkins fue configurado como servidor de Integración Continua, de tal manera que cada vez que se realizara un “git push” al repositorio de GitHub, Jenkins descargaría automáticamente el código y ejecutaría el proceso de construcción y pruebas. Esto se puede observar en la Figura 14.

Figura 14. Configuración Jenkins

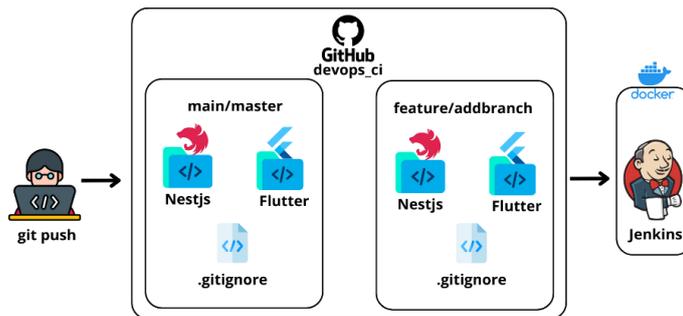


Fuente: El autor

Git y GitHub

Se utilizó Git como sistema de control de versiones usando el flujo de trabajo Gitflow, y GitHub como plataforma para alojar y gestionar los repositorios del proyecto. La arquitectura de implementación se puede observar en la Figura 15. La configuración se llevó a cabo de la siguiente manera:

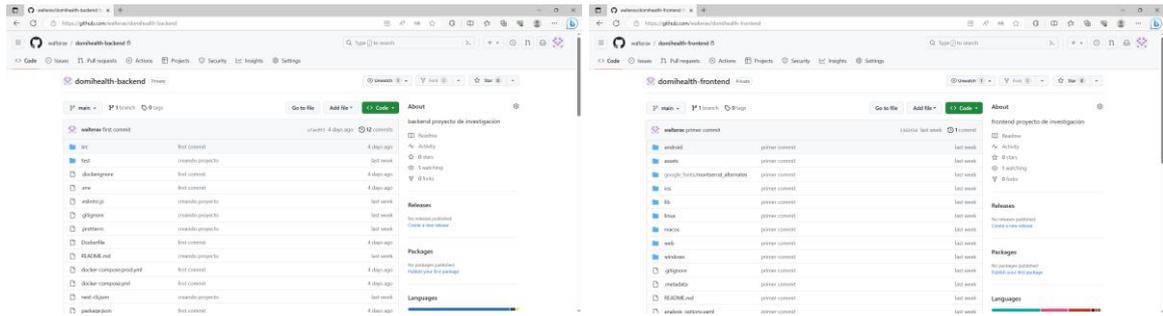
Figura 15. Arquitectura de implementación de GitHub



Fuente: El autor

- Como se muestra en la Figura 16, se creó un repositorio en GitHub para el Backend y Frontend del proyecto y se clonó en el entorno local de desarrollo de Jenkins.

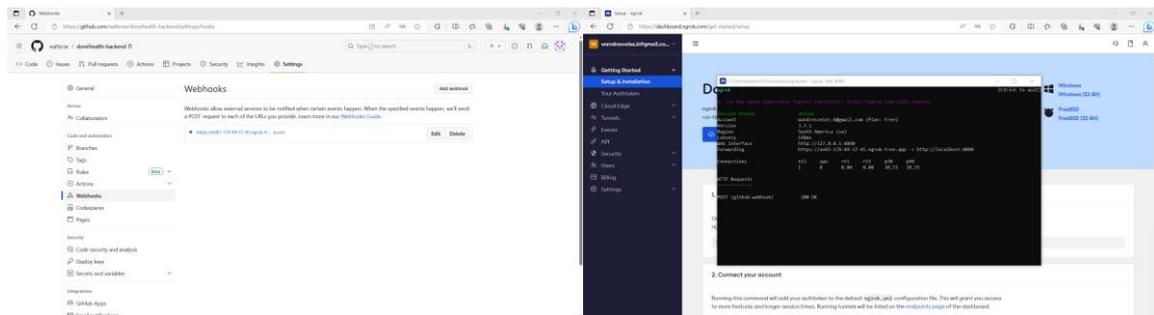
Figura 16. Repositorio GitHub backend (Nestjs) y frontend (Flutter)



Fuente: El autor

- Se usaron Webhooks en GitHub para gestionar eventos específicos. Al hacer un “git push”, Jenkins analiza el código con SonarQube. Ngrok crea túneles seguros para redirigir solicitudes a Jenkins. Así, GitHub envía solicitudes POST a través del webhook, conectando de forma segura los eventos en GitHub y Jenkins, como muestra la Figura 17.

Figura 17. GitHub Webhooks y Ngrok



Fuente: El autor

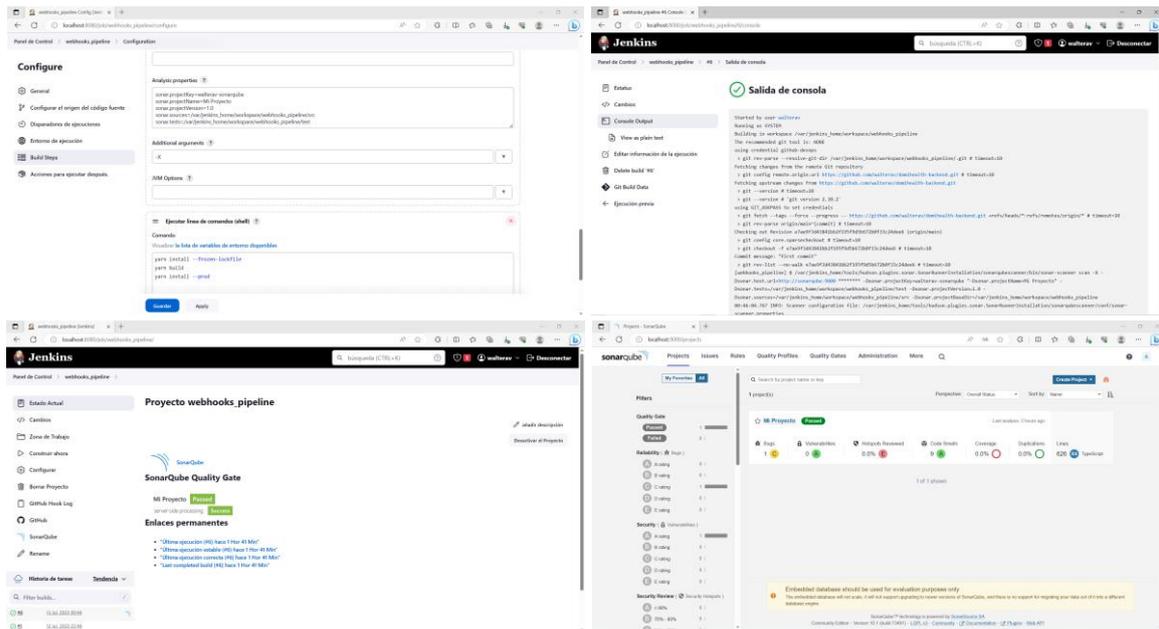
En algunos casos establecieron políticas de revisión de código en GitHub, utilizando pull requests para revisar y aprobar los cambios antes de su incorporación al repositorio principal.

SonarQube

Se utilizó para realizar análisis estáticos de código y garantizar la calidad del software. La configuración se realizó de la siguiente manera:

- Se integró SonarQube en el servidor dedicado Jenkins a través de un plugin y se configuraron las reglas de análisis estático según los estándares y mejores prácticas definidas, de modo que después de cada “git push”, Jenkins descargara el código de GitHub y realizaba el análisis utilizando SonarQube.
- Los informes y métricas de calidad se generaban en SonarQube, lo que permitía identificar posibles problemas y realizar mejoras en el código durante el desarrollo como se muestra en la Figura 18.

Figura 18. Configuración y resultados de SonarQube



Fuente: El autor

3.8.3.2 Desarrollo

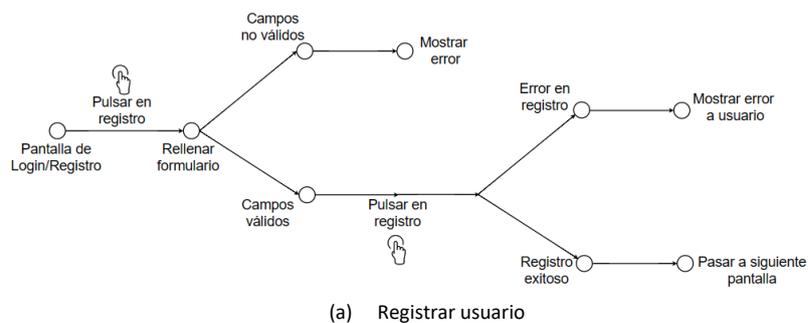
Durante la fase de desarrollo, se llevaron a cabo diversas tareas y actividades con el propósito de crear el producto. En consecuencia, se presentan a continuación las tareas detalladas.

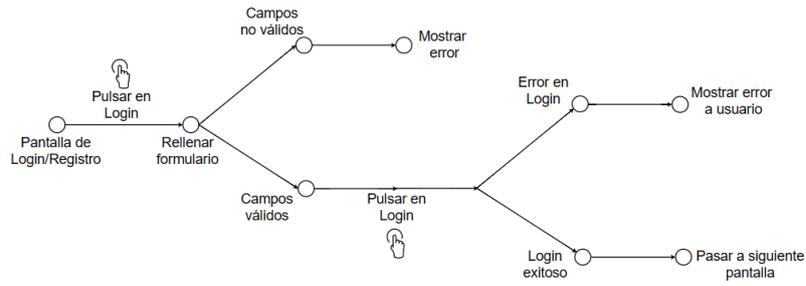
3.8.3.2.1 Diseño

Diseño conceptual

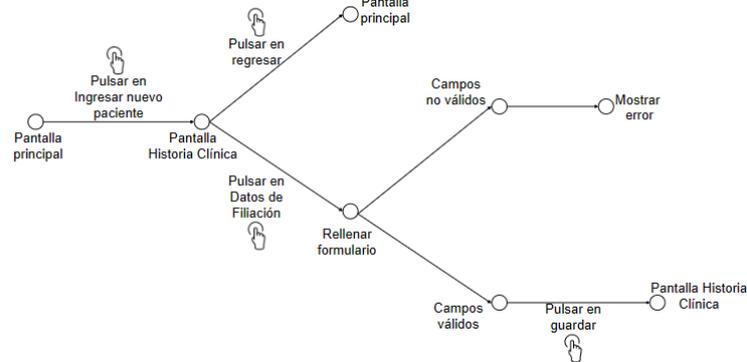
Como parte del proceso de diseño de la aplicación, se utilizaron mapas mentales para conceptualizar las diferentes pantallas del aplicativo móvil. Se prestó especial atención al diseño conceptual de la pantalla de ingreso de pacientes, la gestión de la historia clínica y la pantalla de notas de evolución clínica. En cada caso, se enfatizó en proporcionar una representación clara de los elementos y flujos de interacción, así como en ofrecer una forma eficiente y segura para que los usuarios realicen las operaciones necesarias como se detalla en la Figura 19.

Figura 19. Diseño conceptual de la aplicación

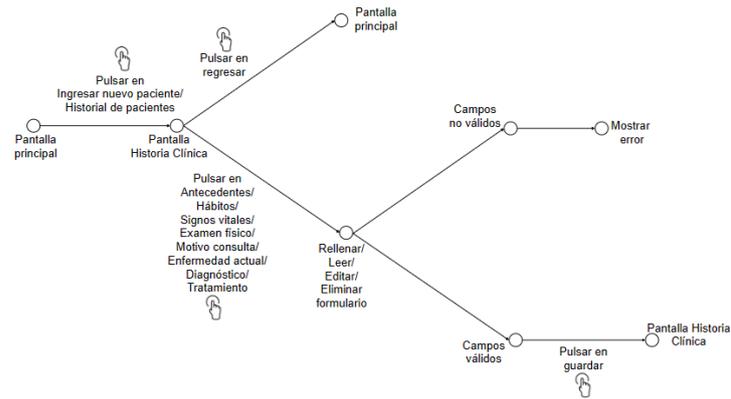




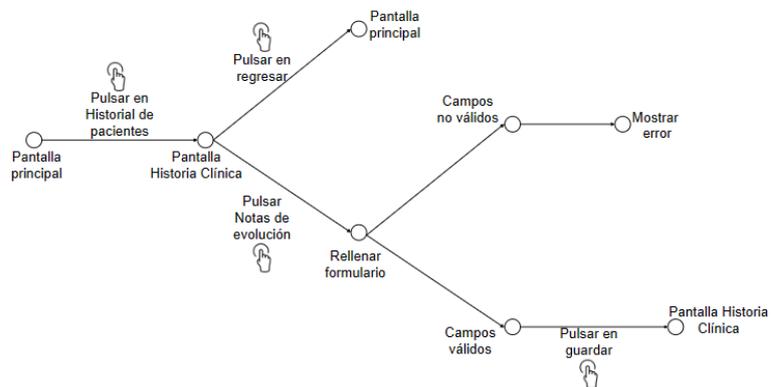
(b) Ingreso de usuario



(c) Historia clínica (ingreso de paciente)



(d) Gestión de información de historia clínica



(e) Notas de evolución clínica

Fuente: El autor

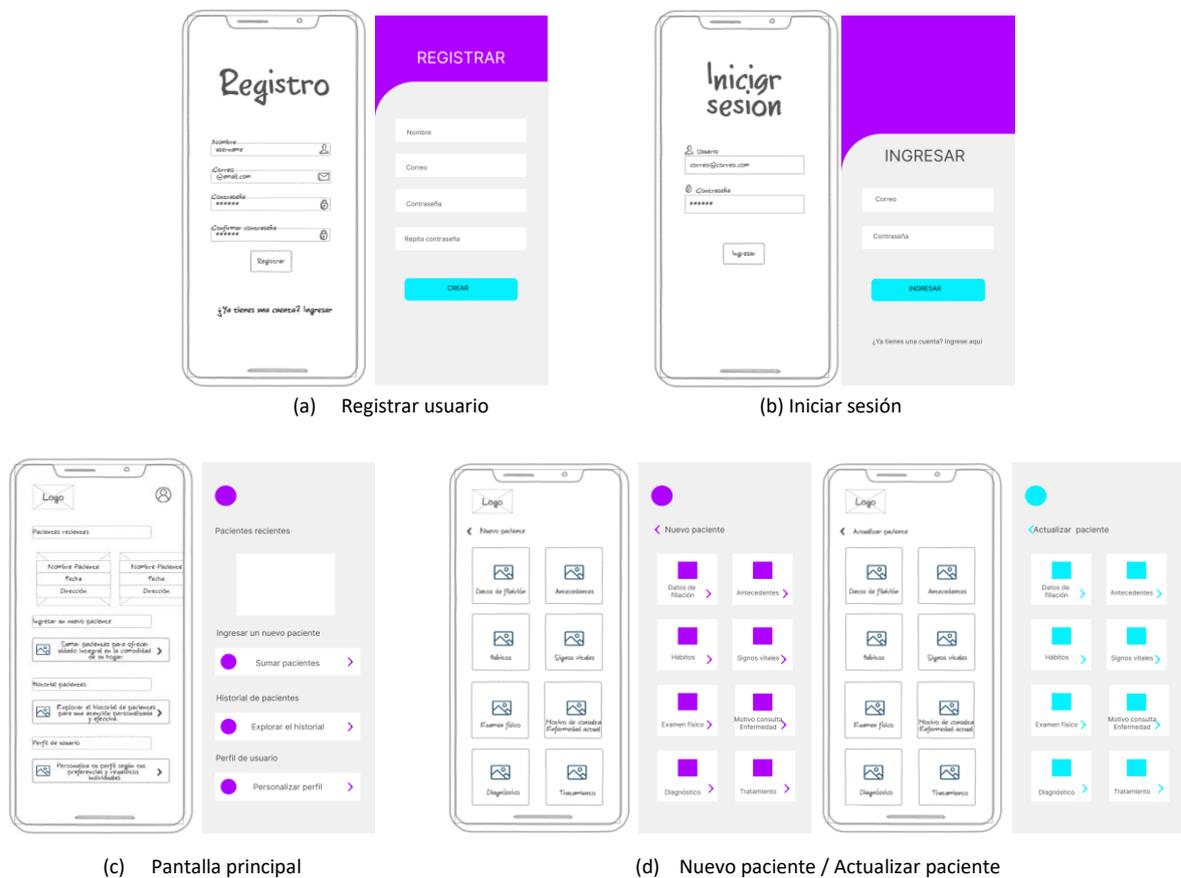
Diseño de interfaz

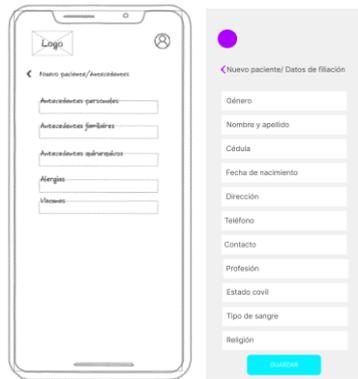
Basándose en el diseño conceptual, se crearon los diseños de interfaz para las diferentes pantallas de la aplicación utilizando las herramientas NinjaMock y Figma (ver Figura 20). El objetivo fue crear interfaces atractivas y bien estructuradas que faciliten el acceso y la interacción de los usuarios con la aplicación.

En el caso de la interfaz de la historia clínica, se buscó mejorar la experiencia del usuario mediante elementos bien ubicados y una interacción sencilla. Para ingresar y gestionar los datos del paciente, se emplearon formularios simples basados en los datos de filiación y otras opciones disponibles.

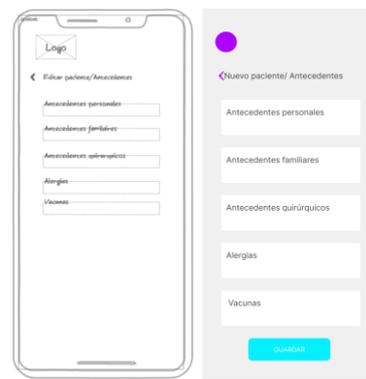
Por último, el diseño de la interfaz para las notas de evolución en la historia clínica se completó cuando la historia clínica se llenó con todos los datos necesarios de forma correcta. Esto permitió a los usuarios registrar y acceder a las notas de evolución de sus pacientes de manera segura y sencilla.

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

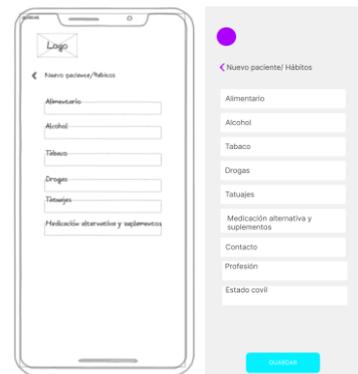




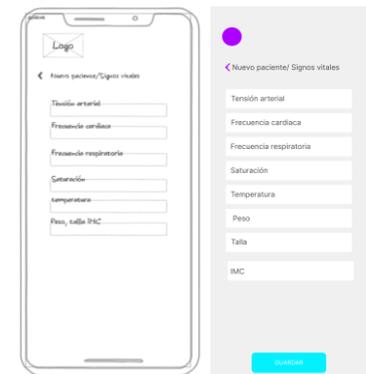
(e) Crear datos de filiación/Editar datos de filiación



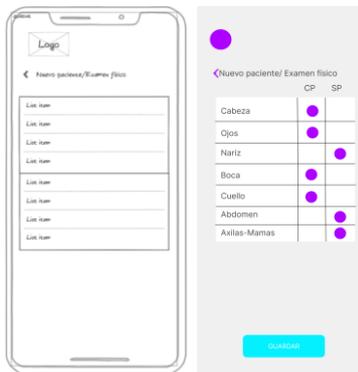
(f) Antecedentes/Editar antecedentes



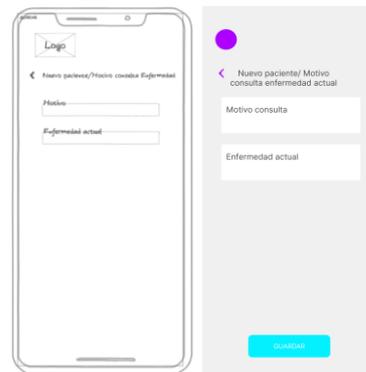
(g) Crear hábitos/Editar hábitos



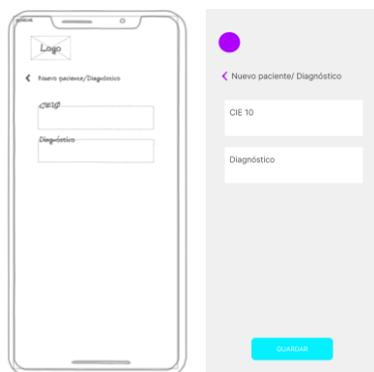
(h) Crear signos vitales/Actualizar signos vitales



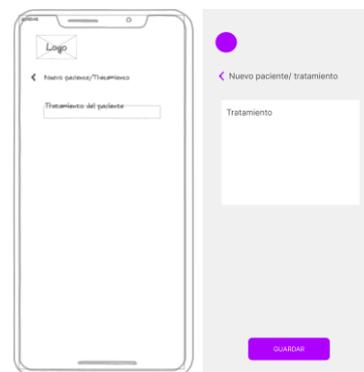
(i) Crear examen Físico/ Actualizar examen físico



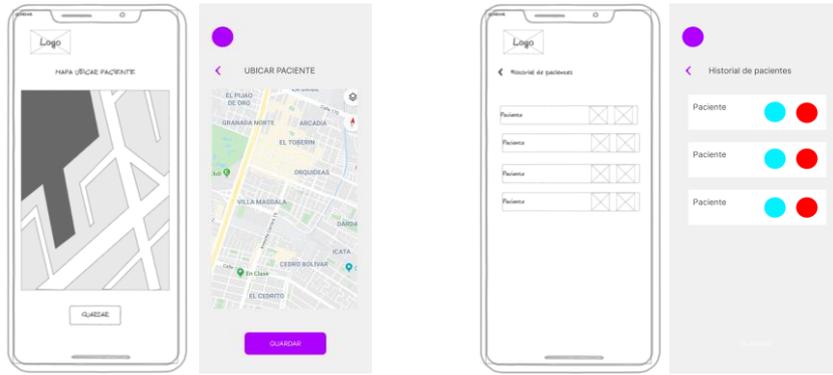
(j) Crear motivo consulta/ Enfermedad actual /Actualizar



(k) Crear diagnóstico/ Actualizar diagnóstico

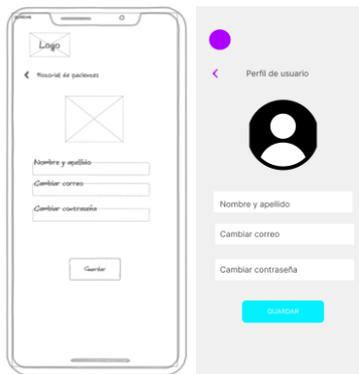


(l) Crear tratamiento/notas de evolución /Actualizar



(m) Ubicar paciente en el mapa

(n) Historial de pacientes



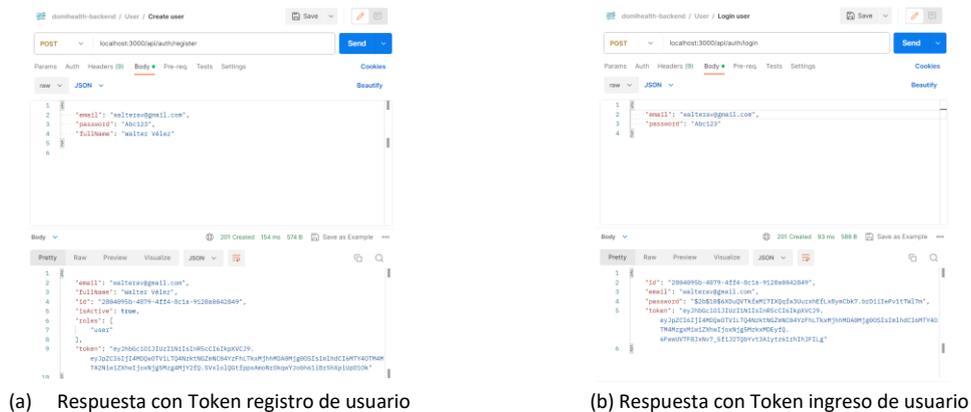
(ñ) Perfil de usuario

Fuente: El autor

Backend

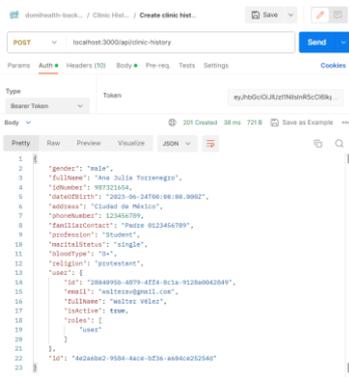
Se utilizó Postman para realizar pruebas y comprobar las rutas del backend desarrollado en Nestjs, como se muestra en la Figura 21. Además, se utilizó Swagger para documentar la API REST y facilitar su uso y comprensión, como se visualiza en el Anexo 2. En la Figura 22 se muestra el código del backend utilizando arquitectura limpia.

Figura 21. Pruebas de la API REST realizadas en Postman

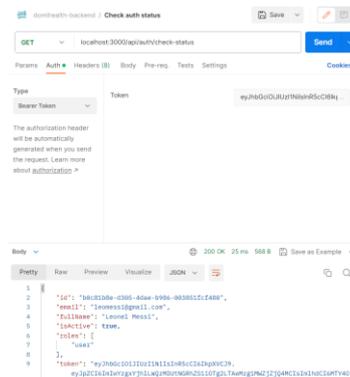


(a) Respuesta con Token registro de usuario

(b) Respuesta con Token ingreso de usuario



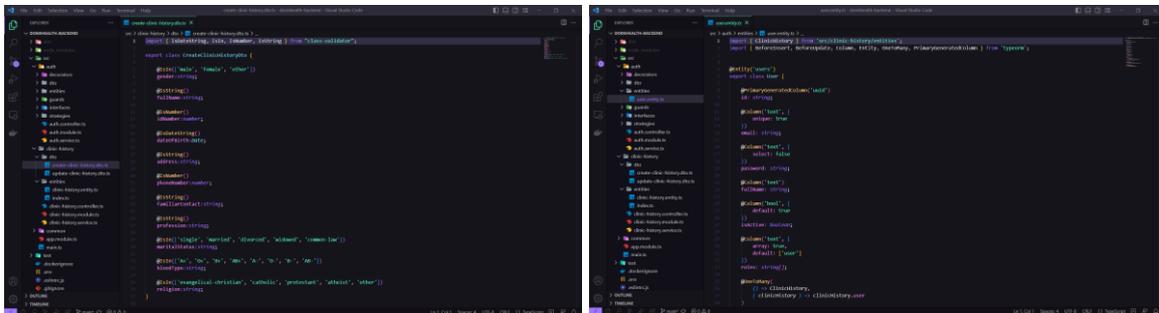
(c) Respuesta con Bearer Token Datos Filiación



(d) Respuesta con Bearer Token Check Status

Fuente: El autor

Figura 22. Código de implementación framework Nestjs



Fuente: El autor

Lanzamiento

Después de finalizar la aplicación móvil, se realizaron pruebas específicas para el sistema operativo Android. Para ello, se generó un archivo APK que incluye todos los elementos esenciales para la instalación en dispositivos Android. Posteriormente, llevamos a cabo la instalación en un dispositivo Android utilizando el archivo APK, asegurándonos de validar el correcto funcionamiento y la compatibilidad de la aplicación en dicho entorno. Con estos pasos completados, la aplicación se encontraba lista para su lanzamiento y para ser utilizada por los usuarios de dispositivos Android.

3.8.4 Pruebas

Se llevó a cabo un análisis preliminar de los indicadores de la variable independiente, ya que el rendimiento puede depender de este análisis. Luego, se desarrolló una aplicación móvil con funcionalidades como autenticación, registro de pacientes, eliminación y modificación de información y visualización de datos. Posteriormente, se evaluó el rendimiento utilizando JMeter, llevando a cabo un total de 10 solicitudes con diferentes finalidades durante el intervalo de 1 segundo durante la simulación. La aplicación se utilizó en un dispositivo móvil Android con características específicas. Finalmente, se recopiló, tabuló y analizó la información y se representó mediante ilustraciones estadísticas y se comparó con los criterios del modelo de calidad FURPS.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

En el marco del proyecto de investigación, se ha completado el desarrollo de una aplicación móvil diseñada para mejorar la gestión de la atención médica domiciliaria (ver Anexo 3). Como parte del tercer objetivo del proyecto, cuyo propósito era medir el rendimiento de la aplicación, se llevaron a cabo pruebas específicas.

Parámetros de evaluación

El análisis de los indicadores es esencial para evaluar y mejorar la calidad y el desempeño de la aplicación. En la Tabla 17 se presentan los indicadores clave que deben ser considerados. Estos indicadores cumplen una función primordial en la garantía de un rendimiento óptimo y una experiencia de usuario satisfactoria.

Tabla 17. Parámetro de evaluación

Tipo de requerimiento	Indicadores
Rendimiento	Eficacia Tiempo de respuesta Utilización de recursos: <ul style="list-style-type: none">• Uso CPU• Uso memoria RAM• Uso almacenamiento

Fuente: El autor

Dispositivo utilizado para la evaluación

En la Tabla 18 se observan las características del dispositivo móvil utilizado.

Tabla 18. Características del dispositivo móvil

Descripción	Dispositivo móvil Android
Marca	Samsung Galaxy
Modelo	SM-A325M/DS
Procesador	Helio G80
Versión sistema operativo	Android 11
RAM	4 GB
Almacenamiento	128 GB

Fuente: El autor

4.1 Resultados

Los resultados de cada parámetro de evaluación fueron obtenidos utilizando Apache JMeter, una herramienta versátil para simular pruebas de carga (ver Anexo 4). Durante estas pruebas, se realizaron un total de 10 solicitudes simuladas por segundo. Posteriormente, se llevó a cabo el cálculo de la media ponderada en porcentaje de cada indicador, siguiendo las

directrices establecidas por el modelo de calidad FURPS. Esto permitió obtener una evaluación más precisa de los resultados.

4.1.1 Simulación en el dispositivo móvil

La Tabla 19 muestra los resultados obtenidos al ejecutar la simulación en un dispositivo Android.

Tabla 19. Medición indicadores dispositivo móvil

N	Tiempo de respuesta (s)	CPU%	RAM%	Almacenamiento%
1	0,368	10	6	2
2	0,520	9	5	3
3	0,540	9	6	1
4	0,508	8	7	3
5	0,506	7	9	2
6	0,516	8	8	1
7	0,501	7	8	2
8	0,509	7	8	3
9	0,542	8	7	1
10	0,536	9	10	2

Fuente: El autor

4.1.2 Análisis de los resultados

Tras finalizar las pruebas, se registraron valores cuantitativos en los resultados. La Tabla 20 muestra el promedio de estos valores.

Tabla 20. Resultados obtenidos de la evaluación del dispositivo móvil

Indicadores	Dispositivo móvil
Eficacia	100%
Tiempo de respuesta	0,5046s
CPU	8,2%
RAM	7,4%
Almacenamiento	2%

Fuente: El autor

Comparación de valores obtenidos y establecidos en el modelo FURPS

Evaluar los resultados obtenidos en las pruebas, en relación a los valores establecidos por el modelo de calidad FURPS, es de vital importancia. La Tabla 21 proporciona una visión clara para determinar si la aplicación cumple con los estándares de rendimiento.

Tabla 21. Comparación de valores obtenidos con el modelo FURPS

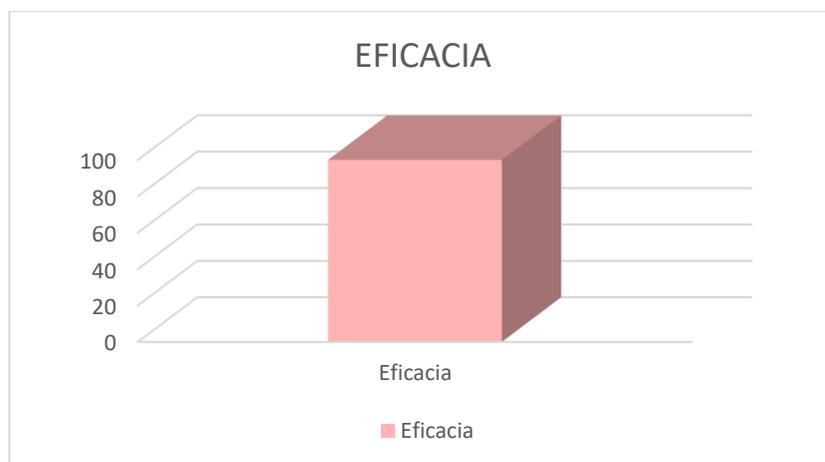
Indicador	FURPS	Valor deseado	Valor obtenido
Eficacia	95%	95%	100%
Tiempo de respuesta	5s	5s	0,5046 s
Utilización de recursos	25%	25%	5,86%

Fuente: El autor

4.1.2.1 Eficacia

La aplicación móvil ha demostrado un alto nivel de eficacia, con un valor del 100%. En la Figura 23 se observa el resultado del indicador mencionado anteriormente.

Figura 23. Resultado de eficacia

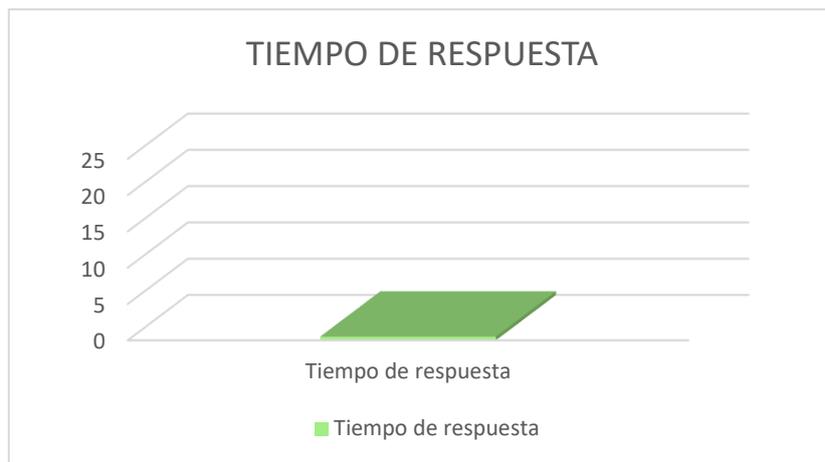


Fuente: El autor

4.1.2.2 Tiempo de respuesta

Como se observa en la Figura 24, se ha logrado un excelente resultado porque se obtuvo el valor de 0,5046 segundos. Esta cifra es inferior al valor deseado de 5 segundos establecido por el modelo de calidad FURPS, indicando que la aplicación responde rápidamente a las interacciones del usuario.

Figura 24. Resultado de tiempo de respuesta



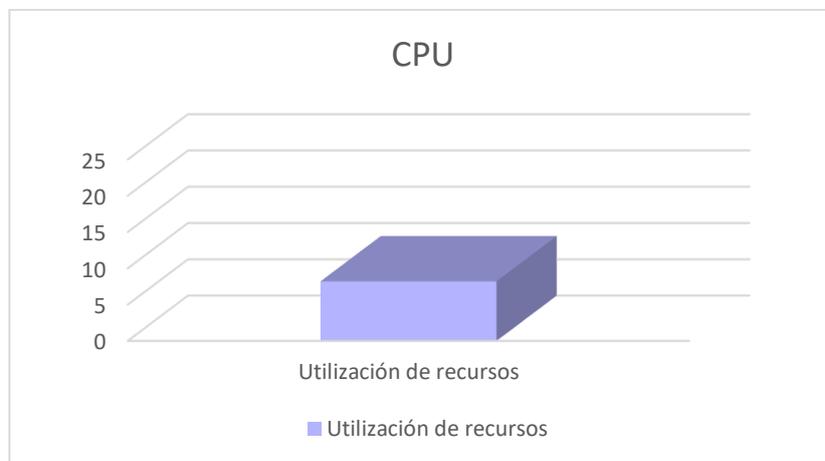
Fuente: El autor

4.1.2.3 Utilización de recursos

CPU

Según los datos presentados en la Figura 25, la aplicación utilizó en promedio el 8,2% de la CPU durante las pruebas. Esta información indica que la aplicación está optimizando el uso del procesador y no está generando una carga excesiva en la CPU.

Figura 25. Resultado CPU

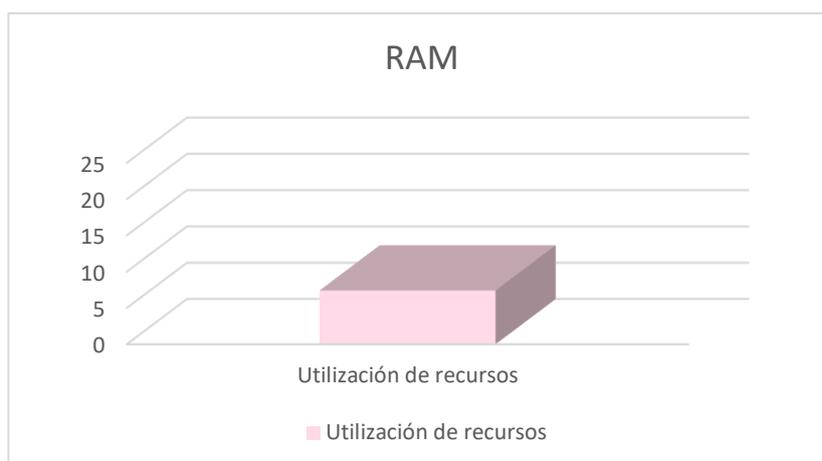


Fuente: El autor

RAM

En la Figura 26 se observa que la aplicación consumió en promedio el 7,4% de la RAM. Este valor indica que la aplicación está gestionando adecuadamente la memoria y no está provocando un uso elevado esta, favoreciendo un rendimiento óptimo.

Figura 26. Resultado RAM

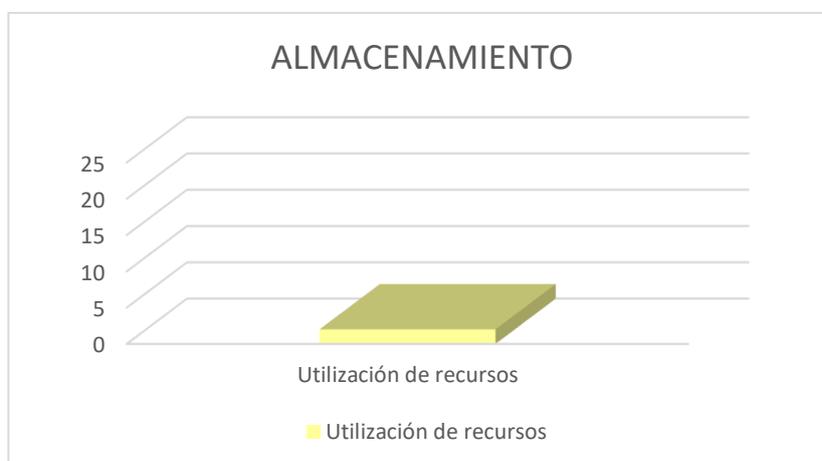


Fuente: El autor

Almacenamiento

De acuerdo con los datos que se muestra en la Figura 27, la aplicación ocupó en promedio el 2% del almacenamiento del dispositivo. Este valor señala que la aplicación está aprovechando el espacio de manera eficiente y no está consumiendo un espacio innecesario en el almacenamiento.

Figura 27. Resultado Almacenamiento

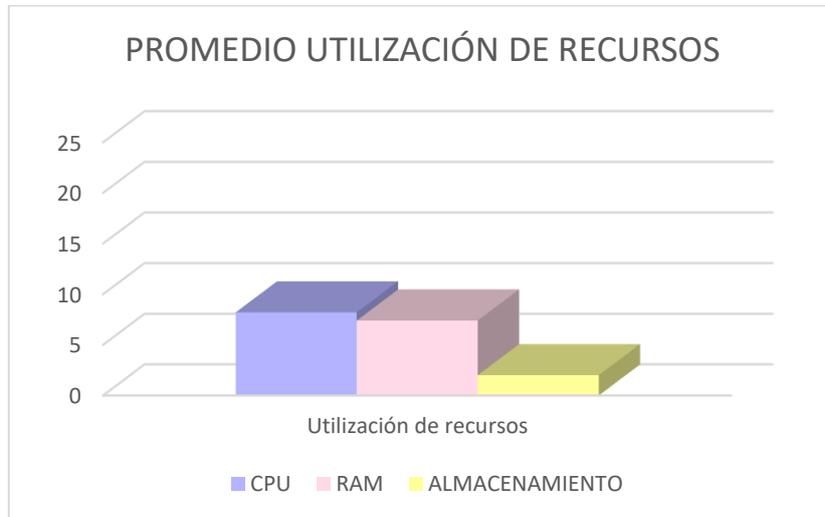


Fuente: El autor

Promedio utilización de recursos

El promedio de consumo de recursos que se registró fue de 5,86% como se observa en la Figura 28. Este resultado indica una eficiente gestión de los recursos del dispositivo móvil por parte de la aplicación. Siendo el valor deseado también del 25%, la aplicación se encuentra utilizando los recursos de manera óptima, contribuyendo a un mejor rendimiento y una experiencia fluida para los usuarios.

Figura 28. Resultado promedio de utilización de recursos



Fuente: El autor

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La investigación y selección de herramientas DevOps como Git, GitHub, Jenkins, Docker y SonarQube ha demostrado ser acertada para el desarrollo de aplicaciones móviles. Estas herramientas proporcionaron un entorno de desarrollo eficiente y colaborativo, permitiendo un control de versiones efectivo a través de Git y GitHub, una integración continua y automatizada con Jenkins, y un análisis de código estático y mejora de la calidad con SonarQube. La implementación exitosa de estas herramientas ha fortalecido el proceso de desarrollo de la aplicación móvil.
- El diseño de la aplicación móvil para la gestión de información de atención médica domiciliaria ha aprovechado las ventajas de Docker. La utilización de Docker ha permitido empaquetar la API REST en Nestjs y crear un contenedor backend junto con la base de datos PostgreSQL. Esto ha asegurado un despliegue consistente y portátil, facilitando la implementación de la aplicación en diferentes entornos sin preocuparse por las dependencias y configuraciones específicas. Docker ha contribuido significativamente a la eficiencia y portabilidad del despliegue de la aplicación móvil.
- La evaluación del rendimiento de la aplicación móvil utilizando el modelo de calidad FURPS ha sido efectuada con éxito. Se ha empleado JMeter para llevar a cabo pruebas de rendimiento exhaustivas, evaluando aspectos críticos como la eficacia, tiempo de respuesta y utilización de recursos. Estas pruebas han permitido identificar y abordar posibles cuellos de botella y optimizar el rendimiento general de la aplicación móvil. El enfoque basado en el modelo FURPS ha proporcionado un marco estructurado y completo para la evaluación de la calidad de la aplicación móvil, asegurando que cumpla con los requisitos establecidos.

5.2 Recomendaciones

- Ampliar la automatización del proceso de desarrollo mediante la exploración e integración de herramientas DevOps adicionales. Se sugiere considerar la incorporación de herramientas como Ansible o Kubernetes para mejorar la gestión de la infraestructura y facilitar la implementación y escalabilidad de la aplicación móvil en entornos de producción. Esto permitirá lograr una mayor eficiencia y flexibilidad en el despliegue y actualización de la aplicación, reduciendo el tiempo y los costos asociados al proceso.
- Implementar prácticas de seguridad durante el desarrollo y despliegue de la aplicación móvil. Se recomienda considerar la integración de herramientas de análisis de vulnerabilidades, como OWASP ZAP, para identificar posibles brechas de seguridad en el código. Además, es importante aplicar buenas prácticas de autenticación, autorización y cifrado de datos, protegiendo así la información

sensible relacionada con la atención médica domiciliaria. Esto aumentará la confianza del usuario en la aplicación y reducirá el riesgo de incidentes de seguridad.

- Establecer un proceso de monitorización y análisis continuo del rendimiento de la aplicación móvil. Se sugiere utilizar herramientas como Prometheus o ELK Stack para recopilar y analizar métricas de rendimiento en tiempo real. Esto permitirá identificar cuellos de botella, optimizar la eficiencia y la experiencia del usuario, y responder de manera proactiva a posibles problemas de rendimiento. Además, se podrán tomar decisiones informadas sobre futuras mejoras en la aplicación basadas en datos objetivos.

BIBLIOGRAFÍA

- [1] World Health Organization, “Proyecto de estrategia mundial sobre salud digital 2020–2025”. Consultado: el 17 de julio de 2023. [En línea]. Disponible en: https://www.paho.org/hq/index.php?option=com_docman&view=download&category_slug=56-directing-council-spanish-
- [2] C. Díaz, “Salud electrónica (e-Salud): un marco conceptual de implementación en servicios de salud”, *Gac Med Mex*, vol. 155, núm. 2, pp. 176–183, 2019, doi: 10.24875/GMM.18003788.
- [3] A. J. Reinoso González y C. D. Zhirzhan Cabrera, “Desarrollo de una aplicación móvil para el agendamiento de citas de consultas médicas utilizando técnicas de procesamiento de lenguaje natural aplicadas a un asistente virtual”, Universidad Politécnica Salesiana, Cuenca, 2022. Consultado: el 17 de julio de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/22064>
- [4] K. Sandoval, S. Pozo, y F. Andrade, “Atención ambulatoria y domiciliaria a pacientes con la COVID-19 en el Primer Nivel de Atención”, ene. 2022. Consultado: el 10 de mayo de 2023. [En línea]. Disponible en: <https://www.salud.gob.ec/wp-content/uploads/2022/01/LINEAMIENTO-MANEJO-DOMICILIARIO-COVID-08-Enero-2022-FI.pdf>
- [5] L. E. Lwakatare *et al.*, “DevOps in practice: A multiple case study of five companies”, *Inf Softw Technol*, vol. 114, pp. 217–230, oct. 2019, doi: 10.1016/J.INFSOF.2019.06.010.
- [6] P. Debois, “Agile infrastructure and operations: How infra-gile are you?”, *Proceedings - Agile 2008 Conference*, pp. 202–207, 2008, doi: 10.1109/AGILE.2008.42.
- [7] P. Debois, “Agile 2008 Toronto: Agile Infrastructure and Operations Presentation”, *Personal website of Patrick Debois*, el 9 de octubre de 2008. <https://www.jedi.be/blog/2008/10/09/agile-2008-toronto-agile-infrastructure-and-operations-presentation/> (consultado el 17 de mayo de 2023).
- [8] A. Hrusto, E. Engström, y P. Runeson, “Towards optimization of anomaly detection in DevOps”, *Inf Softw Technol*, vol. 160, p. 107241, ago. 2023, doi: 10.1016/J.INFSOF.2023.107241.
- [9] N. Azad y S. Hyrynsalmi, “DevOps critical success factors - A systematic literature review”, *Inf Softw Technol*, vol. 157, p. 107150, may 2023, doi: 10.1016/J.INFSOF.2023.107150.
- [10] J. Pennington, “What is DevOps?”, *Taptu*, el 18 de julio de 2019. <https://medium.com/taptuit/what-is-devops-fb3d044ef659> (consultado el 18 de mayo de 2023).
- [11] L. Leite, C. Rocha, F. Kon, D. Milojicic, y P. Meirelles, “A Survey of DevOps Concepts and Challenges”, *ACM Computing Surveys (CSUR)*, vol. 52, núm. 6, nov. 2019, doi: 10.1145/3359981.

- [12] F. R. María Ana y N. C. Felipe De Jesús, “DevOps: un vistazo rápido”, *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla*, vol. 10, núm. 19, pp. 35–40, ene. 2022, doi: 10.29057/ESH.V10I19.8121.
- [13] J. Pennington, “The Eight Phases of a DevOps Pipeline”, *Taptu*, el 18 de julio de 2019. <https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba> (consultado el 18 de mayo de 2023).
- [14] J. Quijano, “DevOps, fases del ciclo iterativo de esta nuevo marco de desarrollo”, *Genbeta*, el 1 de febrero de 2018. <https://www.genbeta.com/desarrollo/el-ciclo-de-devops-una-guia-para-iniciarse-en-las-fases-que-lo-componen> (consultado el 18 de mayo de 2023).
- [15] U. Hamza, M. Syed-Mohamad Sharifah, y N. Lee Abdullah, “DevOps Adoption Guidelines, Challenges, and Benefits: A Systematic Literature Review”, *ICRRD Journal*, núm. 1, pp. 2773–5958, 2023, doi: 10.53272/icrrd.
- [16] M. Gokarna, “DevOps phases across Software Development Lifecycle”, *TechRxiv*, ene. 2021, doi: 10.36227/TECHRXIV.13207796.V2.
- [17] S. Bheri y S. Vummenthala, “An Introduction to the DevOps Tool Related Challenges”, Master’s Thesis, Blekinge Institute of Technology, Blekinge, 2019. Consultado: el 21 de mayo de 2023. [En línea]. Disponible en: <https://urn.kb.se/resolve?urn=urn:nbn:se:bth-18344>
- [18] J. Davis y K. Daniels, “Effective DevOps”, *O’Reilly Media, Inc.*, 2016, Consultado: el 16 de mayo de 2023. [En línea]. Disponible en: <https://www.oreilly.com/library/view/effective-devops/9781491926291/>
- [19] M. D. Delgado Basilio, “El paradigma devops y su implementación en el desarrollo de software / The devops paradigm and its imlementation in the software development”, *Universidad & Ciencia*, vol. 9, núm. 3, pp. 134–142, sep. 2020, Consultado: el 16 de mayo de 2023. [En línea]. Disponible en: <https://revistas.unica.cu/index.php/uciencia/article/view/1604/html>
- [20] L. Calvo, “¿Qué es una app, para qué se utiliza y qué tipos existen?”, *GoDaddy*, el 16 de diciembre de 2022. <https://es.godaddy.com/blog/que-es-una-app-y-para-que-se-utiliza/> (consultado el 4 de mayo de 2023).
- [21] Z. Rashid, C. S. Mon, y R. Kolandaisamy, “Proposing a development of geolocation mobile application for airport pickup of international students pickup”, *ACM International Conference Proceeding Series*, vol. Part F147956, pp. 229–232, 2019, doi: 10.1145/3316615.3316639.
- [22] J. Bowler, “Explainer: How does GPS actually work?”, *Cosmos Magazine*, el 5 de mayo de 2023. <https://cosmosmagazine.com/technology/gps-glonass-gnss-maps-military/> (consultado el 8 de mayo de 2023).
- [23] V. Trafaniuc, “¿Qué es una API de geolocalización? 4 aplicaciones y 7 ventajas”, *Maplink Blog*, el 11 de enero de 2022. <https://maplink.global/blog/es/api-de-geolocalizacion/> (consultado el 18 de mayo de 2023).
- [24] J. Frankenfield, “Geolocalización: Qué es, Cómo funciona, Ejemplos”, *Investopedia*, el 25 de enero de 2021. <https://www.investopedia.com/terms/g/geolocation.asp> (consultado el 5 de mayo de 2023).

- [25] H. Mehta, P. Kanani, y P. Lande, “Google Maps”, *Google Maps Article in International Journal of Computer Applications*, vol. 178, núm. 8, pp. 975–8887, 2019, doi: 10.5120/ijca2019918791.
- [26] A. Malak, “What is Information Management? The Complete Guide”, *The ECM Consultant*, el 2 de mayo de 2023. <https://theecmconsultant.com/what-is-information-management/> (consultado el 10 de mayo de 2023).
- [27] M. Rouse, “What is a Database Management System (DBMS)?”, *Techopedia*, el 28 de abril de 2023. <https://www.techopedia.com/definition/24361/database-management-systems-dbms> (consultado el 10 de mayo de 2023).
- [28] V. Valverde, N. Portalanza, y P. Mora, “Análisis descriptivo de base de datos relacional y no relacional”, *Atlante Cuadernos de Educación y Desarrollo*, núm. junio, jun. 2019, Consultado: el 11 de mayo de 2023. [En línea]. Disponible en: <https://www.eumed.net/rev/atlante/2019/06/base-datos-relacional.html>
- [29] C. I. Rivas, V. P. Corona, J. F. Gutiérrez, y L. Hernández, “Metodologías actuales de desarrollo de software”, *Revista Tecnología e Innovación*, vol. 2, núm. 5, pp. 980–986, dic. 2015, Consultado: el 11 de mayo de 2023. [En línea]. Disponible en: www.ecorfan.org/bolivia
- [30] K. Bugarová y J. Šimíčková, “Risk management in traditional and agile project management”, *Transportation Research Procedia*, vol. 40, pp. 986–993, ene. 2019, doi: 10.1016/J.TRPRO.2019.07.138.
- [31] S. Shaikh y S. Abro, “Comparison of Traditional & Agile Software Development Methodology: A Short Survey”, *International Journal of Software Engineering and Computer Systems*, vol. 5, núm. 2, pp. 1–14, 2019, doi: 10.15282/ijsecs.5.2.2019.1.0057.
- [32] P. Abrahamsson *et al.*, “Mobile-D: An agile approach for mobile application development”, *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA*, pp. 174–175, 2004, doi: 10.1145/1028664.1028736.
- [33] B. Mathur y S. M. Satapathy, “An analytical comparison of mobile application development using agile methodologies”, *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, vol. 2019-April, pp. 1147–1152, abr. 2019, doi: 10.1109/ICOEI.2019.8862532.
- [34] A. Aljuhani y A. Alhubaishy, “Incorporating a Decision Support Approach within the Agile Mobile Application Development Process”, *ICCAIS 2020 - 3rd International Conference on Computer Applications and Information Security*, mar. 2020, doi: 10.1109/ICCAIS48893.2020.9096751.
- [35] Jonathan. Rasmusson, “The Agile Samurai: How Agile Masters Deliver Great Software”, *The Pragmatic Bookshelf*, p. 262, 2010.
- [36] E. Ronchieri y M. Canaparo, “Assessing the impact of software quality models in healthcare software systems”, *Health Systems*, vol. 12, pp. 85–97, 2023, doi: 10.1080/20476965.2022.2162445.
- [37] D. C. Yeomans y P. Rogers, *Project Management Made Simple and Effective*. Dog Ear Publishing, 2017.

- [38] Orlen Araujo, “MODELO FURPS APLICADO AL ANÁLISIS DE CALIDAD DE UN SOFTWARE DESARROLLADO CON SENCHAS EXT JS”, *Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas*, vol. 5, núm. 1, ago. 2020, Consultado: el 17 de julio de 2023. [En línea]. Disponible en: [//reddi.unlam.edu.ar/index.php/ReDDi/article/view/121/224](http://reddi.unlam.edu.ar/index.php/ReDDi/article/view/121/224)

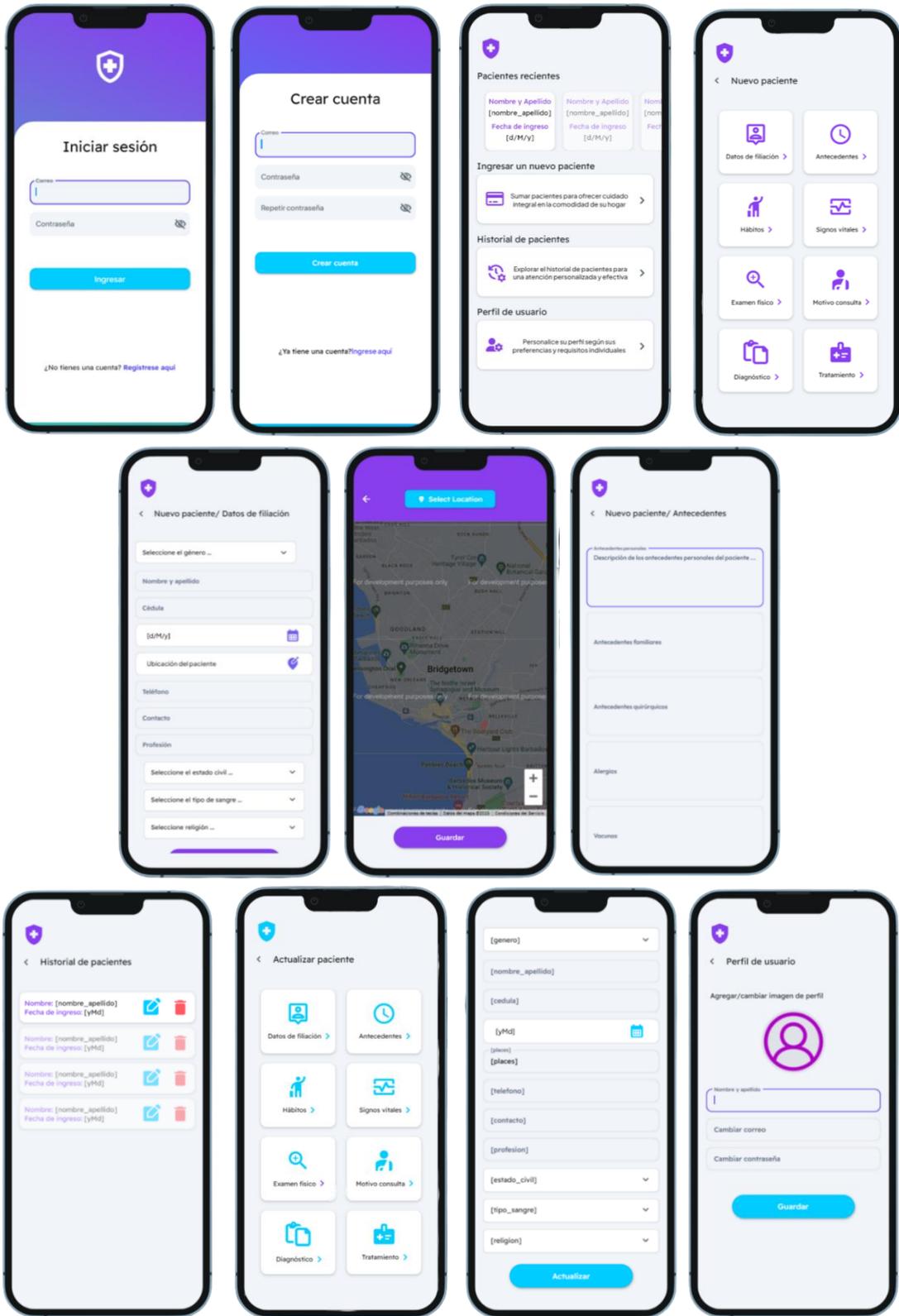
ANEXOS

Anexo 1 Decoradores para establecer relaciones entre entidades en TypeORM.

```
src > clinic-history > entities > clinic-history.entity.ts > ...
34
35 @Column("text", {
36 })
37 familiarContact:string;
38
39 @Column("text", {
40 })
41 profession:string;
42
43 @Column("text", {
44 })
45 maritalStatus:string;
46
47 @Column("text", {
48 })
49 bloodType:string;
50
51 @Column("text", {
52 })
53 religion:string;
54
55 @ManyToOne(
56 () => User,
57 user => user.clinicHistory,
58 { eager: true }
59 )
60 user: User;
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661

```

Anexo 3 Principales pantallas de la aplicación móvil.



Anexo 4 Pruebas de JMeter para la aplicación móvil.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	17:16:23.575	Thread Group 1-1	-3	441	✓	8115	165	441	4
2	17:16:23.574	Thread Group 1-1	REGISTRAR USUA...	441	✓	8115	165	441	4
3	17:16:23.674	Thread Group 1-2	-3	366	✓	8115	165	366	6
4	17:16:23.672	Thread Group 1-2	REGISTRAR USUA...	366	✓	8115	165	366	6
5	17:16:23.768	Thread Group 1-3	-3	307	✓	8115	165	307	7
6	17:16:23.767	Thread Group 1-3	REGISTRAR USUA...	307	✓	8115	165	307	7
7	17:16:23.869	Thread Group 1-4	-3	258	✓	8115	165	257	4
8	17:16:23.868	Thread Group 1-4	REGISTRAR USUA...	258	✓	8115	165	257	4
9	17:16:23.961	Thread Group 1-5	-3	213	✓	8115	165	212	4
10	17:16:23.960	Thread Group 1-5	REGISTRAR USUA...	213	✓	8115	165	212	4
11	17:16:24.018	Thread Group 1-1	-5	205	✓	8115	165	204	3
12	17:16:24.018	Thread Group 1-1	INICIAR SESIÓN	205	✓	8115	165	204	3
13	17:16:24.041	Thread Group 1-2	-5	242	✓	8115	165	241	2
14	17:16:24.040	Thread Group 1-2	INICIAR SESIÓN	242	✓	8115	165	241	2
15	17:16:24.076	Thread Group 1-3	-5	218	✓	8115	165	217	2
16	17:16:24.076	Thread Group 1-3	INICIAR SESIÓN	218	✓	8115	165	217	2
17	17:16:24.128	Thread Group 1-4	-5	192	✓	8115	165	192	6
18	17:16:24.127	Thread Group 1-4	INICIAR SESIÓN	192	✓	8115	165	192	6
19	17:16:24.168	Thread Group 1-7	-3	190	✓	8115	165	190	4
20	17:16:24.167	Thread Group 1-7	REGISTRAR USUA...	190	✓	8115	165	190	4
21	17:16:24.223	Thread Group 1-1	-6	173	✓	8115	165	173	2
22	17:16:24.223	Thread Group 1-1	PANTALLA PRINC...	173	✓	8115	165	173	2
23	17:16:24.266	Thread Group 1-8	-3	165	✓	8115	165	164	11
24	17:16:24.265	Thread Group 1-8	REGISTRAR USUA...	165	✓	8115	165	164	11
25	17:16:24.285	Thread Group 1-2	-6	179	✓	8115	165	179	5

Scroll automatically? Child samples? No of Samples: 130 Latest Sample: 58 Average: 194 Response: 714

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configure

Search: Case sensitive Regular exp. Search Reset

Text Sampler result Request Response data

- 7 Thread Name: Thread Group 1-8
- 5 Sample Start: 2023-07-29 17:16:24 ECT
- 7 Load time: 170
- 7 Connect Time: 2
- 7 Latency: 170
- 5 Size in bytes: 8115
- 5 Sent bytes: 165
- 8 Headers size in bytes: 0
- 8 Body size in bytes: 0
- 6 Sample Count: 1
- 6 Error Count: 0
- 8 Data type ("text"["bin"]):
- 8 Response code: 200
- 8 Response message: Number of samples in transaction : 1, number of failing samples : 0

SampleResult fields:

- 9 ContentType:
- 9 DataEncoding: null

Raw Parsed