

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

APLICATIVO MÓVIL PARA CONTROL Y MONITOREO DE MEDICAMENTOS ORALES EN PACIENTES CON ENFERMEDADES

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

REINA GAMBOA MIGUEL ESTEBAN

miguel.reina@epn.edu.ec

DIRECTOR: Ing. Adrián Egüez, MSc.

adrian.eguez@epn.edu.ec

CODIRECTOR: Ing. Monserrate Intriago, MSc.

monserrate.intriago@epn.edu.ec

QUITO, DICIEMBRE 2022

DECLARACIÓN

Yo, Reina Gamboa Miguel Esteban, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Reina Gamboa

Miguel Esteban

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Reina Gamboa Miguel Esteban, bajo mi supervisión.

MSc. Adrián Egüez

DIRECTOR DEL PROYECTO

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Reina Gamboa Miguel Esteban, bajo mi supervisión.

MSc. Monserrate Intriago

**CODIRECTOR DEL
PROYECTO**

CONTENIDO

RESUMEN	11
ABSTRACT.....	12
AGRADECIMIENTO.....	13
DEDICATORÍA	14
1. INTRODUCCIÓN	15
1.1. Planteamiento del problema	15
1.2. Objetivos	16
1.2.1. Objetivo general	16
1.2.2. Objetivos específicos.....	16
1.3. Alcance.....	16
1.4. Marco teórico	17
1.4.1. Pacientes dependientes de medicamentos.....	17
1.4.2. Notificaciones push	18
1.4.3. Benchmarking	18
1.4.4. Diseño de interfaces	19
1.4.5. Metodología iterativa incremental.....	20
1.5. Herramientas de desarrollo	21
1.5.1. Flutter	21
1.5.2. Dart.....	21
1.5.3. Firebase	21
1.5.4. Figma	22
1.5.5. Git	22
1.5.6. GitHub.....	22
1.5.7. Android Studio	23
1.5.8. Visual Studio Code.....	23
1.5.9. Postman.....	24
2. METODOLOGÍA	25
2.1. Benchmarking	25
2.1.1. Análisis FODA.....	27
2.1.2. Requerimientos funcionales.....	28
2.1.3. Requerimientos no funcionales.....	29
2.2. Prototipado	29
2.2.1. Fases del prototipado.....	29
2.3. Prototipo móvil.....	31
2.3.1. Iteración 0.....	31

2.3.2.	Iteración 1.....	46
2.3.3.	Iteración 2.....	54
2.3.4.	Iteración 3.....	59
3.	RESULTADOS.....	71
3.1.	Pruebas de funcionalidad.....	71
3.3.	Pruebas de usabilidad.....	78
4.	CONCLUSIONES Y RECOMENDACIONES.....	87
4.1.	Conclusiones.....	87
4.2.	Recomendaciones.....	90
5.	REFERENCIAS BIBLIOGRÁFICAS.....	91
6.	ANEXOS.....	93
6.1.	Anexo 1.....	93
6.2.	Anexo 2.....	93
6.3.	Anexo 3.....	93
6.4.	Anexo 4.....	93
6.5.	Anexo 5.....	93
6.6.	Anexo 6.....	93
6.7.	Anexo 7.....	94
6.8.	Anexo 8.....	94
6.9.	Anexo 9.....	94

Contenido de tablas

Figura 1.	Logotipo de Flutter.	21
Figura 2.	Logotipo de Dart.	21
Figura 3.	Logotipo de Firebase.	22
Figura 4.	Logotipo de Figma.	22
Figura 5.	Logotipo de Git.	22
Figura 6.	Logotipo de GitHub.....	23
Figura 7.	Logotipo de Android Studio.	23
Figura 8.	Logotipo de Visual Studio Code.....	24
Figura 9.	Logotipo de Postman.....	24
Figura 10.	Fases del prototipado.	30
Figura 11.	Arquitectura Cliente-Servidor con Firebase.....	33
Figura 12.	Implementación del patrón provider.	34
Figura 13.	Diagrama de navegación.....	35
Figura 14.	Creación de los mockups en Figma	37
Figura 15.	Base de datos Its time.	37
Figura 16.	Visualización de los documentos de Firebase.....	38
Figura 17.	Pantalla de información del usuario, validación de conexión con la base de datos. .	39
Figura 18.	Servicio de autenticación provisto por Firebase.....	41
Figura 19.	Pantalla para la creación de un nuevo usuario.	42
Figura 20.	Ingreso de información del nuevo usuario.	43
Figura 21.	Alerta de creación de un nuevo usuario.	43
Figura 22.	Formulario para la actualización de información del usuario.....	44
Figura 23.	Ingreso de la información requerida por usuario.....	44
Figura 24.	Validación del nuevo usuario dentro del servicio de autenticación.....	45
Figura 25.	Validación del nuevo usuario dentro del documento usuarios en la base de datos...	45
Figura 26.	Pantalla de inicio.	46
Figura 27.	Prototipo de baja fidelidad del formulario de creación de alarmas.	48
Figura 28.	Formulario para la creación de una alarma.	49
Figura 29.	Formulario con las opciones para la enfermedad “diabetes”.....	49
Figura 30.	Formulario con las opciones para la enfermedad “hipertensión”.	50
Figura 31.	Formulario con la información necesaria para crear una alarma.	50
Figura 32.	Consulta de alarmas creadas mediante Postman.	51
Figura 33.	Pantalla con el listado de alarmas.	53
Figura 34.	Pantalla con el formulario de actualización.	54
Figura 35.	Pantalla de seguidores al iniciar la aplicación.....	57

Figura 36.	Pantalla que contiene todos los usuarios registrados en la aplicación.	57
Figura 37.	Pantalla donde se visualiza los usuarios seguidores.	58
Figura 38.	Visualización de la notificación push en su estado inicial.	61
Figura 39.	Visualización de la notificación push en la vista expandida.	62
Figura 40.	Servicio Cloud Messaging de Firebase.	62
Figura 41.	Dependencias necesarias para el servicio Cloud Messaging.	63
Figura 42.	Ubicación del archivo de servicios de Google.	63
Figura 43.	Función que genera el token por dispositivo.	64
Figura 44.	Estructura del JSON para la notificación push.	65
Figura 45.	Información que se envía desde Postman.	65
Figura 46.	Información de la notificación y datos.	65
Figura 47.	Respuesta del end point.	66
Figura 48.	Visualización de la notificación en el dispositivo.	66
Figura 49.	Estructura del JSON para notificaciones push masivas.	68
Figura 50.	JSON con la información de prueba.	68
Figura 51.	Información de la notificación y datos de notificaciones masivas.	68
Figura 52.	Respuesta del end point.	69
Figura 53.	Dispositivo emulado para las pruebas de envío.	70
Figura 54.	Dispositivo físico con el aplicativo, para las pruebas de envío masivo.	70
Figura 55.	Pregunta 1. Encuesta SUS.	80
Figura 56.	Pregunta 2. Encuesta SUS.	80
Figura 57.	Pregunta 3. Encuesta SUS.	81
Figura 58.	Pregunta 4. Encuesta SUS.	82
Figura 59.	Pregunta 5. Encuesta SUS.	82
Figura 60.	Pregunta 6. Encuesta SUS.	83
Figura 61.	Pregunta 7. Encuesta SUS.	83
Figura 62.	Pregunta 8. Encuesta SUS.	84
Figura 63.	Pregunta 9. Encuesta SUS.	84
Figura 64.	Pregunta 10. Encuesta SUS.	85

Contenido de tablas

Tabla 1.	Detalle de características de las diferentes aplicaciones usadas en el benchmarking ...	27
Tabla 2.	Tabla de análisis FODA	28
Tabla 3.	Requerimientos funcionales.....	28
Tabla 4.	Requerimientos no funcionales.....	29
Tabla 5.	Historias de usuario para el prototipo 0.	32
Tabla 6.	Tiempo estimado para el prototipo 0.	32
Tabla 7.	Modificación de análisis de historias de usuarios del prototipo 0.	39
Tabla 8.	Historias de usuario para el prototipo 1.	40
Tabla 9.	Tiempo estimado para el prototipo 1.	40
Tabla 10.	Modificación de análisis de historias de usuarios del prototipo 1.	46
Tabla 11.	Historias de usuario para el prototipo 2.	47
Tabla 12.	Tiempo estimado para el prototipo 2.	47
Tabla 13.	Modificación de análisis de historias de usuarios del prototipo 2.	51
Tabla 14.	Historias de usuario para el prototipo 3.	52
Tabla 15.	Tiempo estimado para el prototipo 3.	52
Tabla 16.	Modificación de análisis de historias de usuarios del prototipo 3.	54
Tabla 17.	Historias de usuario para el prototipo 4.	55
Tabla 18.	Tiempo estimado para el prototipo 4.	55
Tabla 19.	Modificación de análisis de historias de usuarios del prototipo 4.	58
Tabla 20.	Historias de usuario para el prototipo 5.	59
Tabla 21.	Tiempo estimado para el prototipo 5.	60
Tabla 22.	Modificación de análisis de historias de usuarios del prototipo 5.	66
Tabla 23.	Historias de usuario para el prototipo 6.	67
Tabla 24.	Tiempo estimado para el prototipo 6.	67
Tabla 25.	Modificación de análisis de historias de usuarios del prototipo 6.	70
Tabla 26.	Caso de prueba CPIT001	71
Tabla 27.	Tabla de resultados del caso de prueba CPIT001	71
Tabla 28.	Caso de prueba CPIT002.....	72
Tabla 29.	Tabla de resultados del caso de prueba CPIT002	72
Tabla 30.	Caso de prueba CPIT003	73
Tabla 31.	Tabla de resultados del caso de prueba CPIT003	73
Tabla 32.	Caso de prueba CPIT004.....	73
Tabla 33.	Tabla de resultados del caso de prueba CPIT004	73
Tabla 34.	Caso de prueba CPIT005	74
Tabla 35.	Tabla de resultados del caso de prueba CPIT005	74

Tabla 36.	Caso de prueba CPIT006.....	75
Tabla 37.	Tabla de resultados del caso de prueba CPIT006	75
Tabla 38.	Caso de prueba CPIT007.....	75
Tabla 39.	Tabla de resultados del caso de prueba CPIT007	76
Tabla 40.	Caso de prueba CPIT008.....	76
Tabla 41.	Tabla de resultados del caso de prueba CPIT008	77
Tabla 42.	Caso de prueba CPIT009.....	77
Tabla 43.	Tabla de resultados del caso de prueba CPIT009	78
Tabla 44.	Tabla de resultados del cuestionario SUS.....	85

RESUMEN

En Ecuador una gran parte de la población es diagnosticada con enfermedades crónicas entre las principales están diabetes e hipertensión, para estos cuadros clínicos el Ministerio de Salud Pública del Ecuador posee guías de prácticas médicas que permiten a los especialistas emitir diagnósticos que ayudan a que los pacientes ingieran de forma diaria la medicación adecuada; con este antecedente se estructura para este proyecto de investigación un aplicativo llamado “It’s time”, el mismo que es desarrollado con un enfoque mediante la metodología iterativa e incremental y el uso de prototipos, la metodología utilizada genera iteraciones con el propósito de incrementar mejoras en características o funcionalidades, al igual que el uso de Firebase y Flutter tecnologías de vanguardia que permiten al proyecto ser adaptable en los diferentes tipos de plataformas; dando fluidez en un 97% de usabilidad determinada mediante la encuesta “Sistemas de escala de usabilidad” siendo amigable hacia los usuarios, los mismos que después de las pruebas de funcionalidad se determinó que un 70% de los encuestados sobrepasen los 40 años de edad, siendo visible el poco contacto de la tecnología en el uso diario, característica necesaria para el aplicativo. A pesar de dicha dificultad la aplicación evita caer en errores recurrentes que pueda cometer el usuario, facilitándole registros de alarmas y notificaciones cuando estas hayan sido programadas, dando un control adecuado de la medicación.

Palabras clave: It’s time, control, diagnósticos, usabilidad, amigable, aplicativo móvil.

ABSTRACT

In Ecuador, a large part of the population is diagnosed with chronic diseases, among the main ones are diabetes and hypertension, for these clinical pictures the Ministry of Public Health of Ecuador has medical practice guides that allow specialists to issue diagnoses that help patients take adequate medication on a daily basis; With this background, an application called "IT'S TIME" is structured for this research project, which is developed with an approach through the iterative and incremental methodology and the use of prototypes, the methodology used generates iterations with the purpose of increasing improvements in features or functionality, such as the use of cutting-edge Firebase and Flutter technologies that allow the project to be adaptable on different types of platforms; giving fluency in 97% of usability determined by means of the survey "Systems of usability scale" being friendly towards the users, the same ones that after the functionality tests it was determined that 70% of the respondents are over 40 years of age, being visible the little contact of the technology in daily use, a necessary characteristic for the application . Despite this difficulty, the application avoids falling into recurring errors that the user may commit, providing alarm records and notifications when they have been programmed, giving adequate control of the medication.

Keywords: It's time, control, diagnostics, usability, friendly, mobile application.

AGRADECIMIENTO

Agradezco a mi madre Katherine, mi padre Marco, a mis hermanas Mishelle y Raquel, a mi sobrino Diego, a mis abuelos y tíos maternos los cuales estuvieron en toda mi trayectoria hasta la actualidad, siendo guía e impartiendo consejos y enseñanzas para ser de mí la mejor versión.

Agradezco a las personas que me han acompañado en este trayecto como: amigos, compañeros y profesores ya que de todos ustedes aprendí y me llevo las mejores experiencias y conocimientos.

Agradezco a mi tutor el cuál fue una guía en el transcurso para el desarrollo de este proyecto de titulación.

Y finalmente agradezco de forma muy especial a mí, ya que sin mi esfuerzo, dedicación, conocimiento y empeño nada de esto hubiera sido posible.

DEDICATORÍA

Dedico el presente proyecto de titulación al esfuerzo de mi familia y al mío, especialmente a mis padres ya que sin ellos no hubiera existido la pauta para el tema.

1. INTRODUCCIÓN

1.1. Planteamiento del problema

El Ministerio de Salud Pública del Ecuador mediante una encuesta realizada al 64.9% de la población, obtuvo como resultado que más del 25.8% de la población nacional [1] padecen de enfermedades crónicas o hereditarias, las cuales hacen del paciente dependiente de medicamentos de forma diaria para poder mantener un control óptimo para una vida saludable, la fragilidad mental de las personas para recordar los eventos diarios inhibe la oportuna ingesta de los medicamentos los mismos que son consumidos a destiempo, incumpliendo con la dosificación recomendada por el médico, con este antecedente el presente proyecto de titulación propone un aplicativo móvil que permita a los pacientes obtener un monitoreo amigable y constante para el consumo adecuado de sus medicinas.

El presente proyecto se formula como incógnita principal la resolución del siguiente problema:

¿Cómo establecer un control adecuado para la administración de medicamentos vía oral a pacientes diagnosticados con enfermedades crónicas o hereditarias?

Para poder solventar este problema se propone el desarrollo de un aplicativo móvil que contendrá un horario interno con alarmas programadas para que el usuario cuente con un recordatorio que le ayude a mantener un control disciplinado del mismo, evitando problemas de salud a futuro, además se suman notificaciones push emitidas posterior al primer recordatorio; las cuales permitirán que usuarios de la plataforma que siguen al paciente puedan visualizar si su ingesta no se registró, o si fue interrumpida. En el caso de tener una notificación de las anteriormente descritas, permitirán comunicarse con el paciente para poder dar aviso del consumo de su medicamento.

1.2. Objetivos

1.2.1. Objetivo general

- Crear un aplicativo móvil que permita registrar y notificar la ingesta de medicamentos para pacientes diagnosticados de enfermedades.

1.2.2. Objetivos específicos

- Revisión de literatura y un benchmarking, para establecer los requerimientos del aplicativo.
- Diseñar una interfaz cómoda y amigable para el registro de pacientes y enfermedades mediante prototipos de baja fidelidad.
- Definir la arquitectura y el modelado de datos para el proyecto.
- Implementar prototipos de alta fidelidad en función a los requerimientos establecidos.
- Realizar pruebas de usabilidad y definir la satisfacción del usuario con el entorno de la aplicación.

1.3. Alcance

En el presente proyecto integrador se plantea dar una solución a la mayor parte de pacientes dependientes de medicamentos, solventando sus problemas con respecto a los horarios de ingesta mediante el desarrollo de un aplicativo móvil que sirva en sistemas operativos tipo Android.

El aplicativo móvil permitirá al usuario objetivo o paciente crear una alarma o aviso programado para la ingesta de su diferente medicación diaria, el aplicativo móvil también tendrá un enfoque similar a una red social, con el propósito de que el usuario objetivo o paciente pueda tener seguidores que también puedan ayudar al paciente a recordar la ingesta de sus medicamentos ya que al usuario seguidor se le presentará una notificación

en el caso de que la alarma principal haya sido pospuesta; permitiendo tener un segundo recordatorio para el paciente.

1.4. Marco teórico

1.4.1. Pacientes dependientes de medicamentos

El proyecto de titulación tiene como propósito fundamental crear una aplicación de fácil acceso y uso para todos los pacientes dependientes de medicamentos de forma diaria por cualquier tipo de enfermedades, para ello vamos a tomar como base a pacientes con enfermedades crónicas como hipertensión y diabetes, esta clase de pacientes tienen un tratamiento específico dependiendo del tipo de enfermedad en nuestro país, el Ministerio de Salud Pública del Ecuador posee una estandarización para el control de la mayor parte de enfermedades. Los documentos con los que se realizan estos procesos por parte del área de salud se llaman “Guías de prácticas clínicas”, estas guías tienen un acceso público en las páginas oficiales del Ministerio de Salud Pública, en las guías podemos encontrar los medicamentos que se usan para los diferentes tratamientos.

El tratamiento de la hipertensión mediante la Guía de Práctica Clínica del 2019 se basa en los siguientes medicamentos: [2]

- Clortalidona, en dosis diarias de 12.5 mg a 25 mg mediante una ingesta tipo vía oral.
- Enalapril en dosis diarias de 2.5 mg a 5 mg vía oral, o dosis de 1.25 mg tres veces al día mediante una vía parenteral.
- Amlodipina dosis diarias de 5 mg vía oral.
- Losartán dosis diarias de 25 mg a 100 mg vía oral.
- Atenolol dosis diarias de 25 mg a 50 mg vía oral.
- Carvedilol dos ingestas diarias de 6.25 mg vía oral.

El tratamiento de diabetes mediante la Guía de Práctica Clínica del 2017 se basa en los siguientes medicamentos: [3]

- Ácido acetil salicílico, dosis diarias de 75 mg a 162 mg vía oral.
- Clopidogrel, dosis diarias de 75 mg vía oral.
- Enalapril, dosis diarias de 5 mg vía oral.
- Glibenclamida, dosis diarias de 5 mg vía oral.
- Glicazida, dosis diarias de 40 mg a 80 mg vía oral.
- Glimepirida, dosis diarias de 1 mg vía oral, en intervalos de una a dos semanas.
- Glucagón, dosis diarias de 1 mg vía subcutánea o intramuscular.
- Insulina humana de acción intermedia, dosis diarias subcutánea de 10 UI.
- Insulina humana de acción rápida, vía subcutánea intramuscular o intravenosa.
- Losartan, dosis diarias de 50 mg vía oral.
- Metformina dosis diarias de 500 mg vía oral.
- Simvastatina dosis de 20 mg a 40 mg diarias vía oral y comúnmente en horarios nocturnos.

1.4.2. Notificaciones push

En aplicaciones móviles una notificación push es el medio de comunicación mediante mensajes emitidos desde los diferentes puntos del sistema en el cual interactúa la aplicación, es decir la interacción de dos usuarios de diferentes móviles dentro de la misma red. [4] Como por ejemplo los diferentes tipos de redes sociales como WhatsApp, Telegram, Signal o Messenger entre otros los cuales permiten a los usuarios generar notificaciones entre ellos mediante mensajes, los cuales pueden llegar al dispositivo estando en reposo o cuando la aplicación se ejecuta en segundo plano.

1.4.3. Benchmarking

El benchmarking es una metodología que permite realizar un análisis comparativo y evaluativo de los diferentes tipos de servicios, productos o procesos de diversas compañías, el motivo principal de la realización de un benchmarking para una organización es poder analizar la mayor parte de características de las organizaciones que tienen el mismo giro de negocio o las organizaciones que pueden ser competitivas dentro del mercado, obteniendo las características fuertes y débiles de la competencia y

dando de esta manera la posibilidad de que la empresa despunte del resto, ofreciendo mejores servicios o productos con procesos de mejor calidad y con mayor ventaja en el mercado. Permitiendo a la organización validar de cierta manera las desventajas de la organización con respecto a toda su competencia directa. [5]

¿Qué tipos de benchmarking existen?

Existen tres tipos de benchmarking: interno, de competencia o externo y funcional o estratégico.

- El benchmarking interno se lo realiza con el propósito de evaluar y analizar los departamentos dentro de la organización de este modo verificando los departamentos que necesitan mejorar sus procesos para dar mejores resultados para la empresa. [5]
- El benchmarking de competencia o externo compara nuestra organización con las organizaciones que operan dentro del mismo giro de negocio llegando a analizar las prácticas que colaboran a estas organizaciones a tener un mejor desempeño y ver nuestras debilidades ante ellos saber en qué procesos nuestra organización podría mejorar para ser más competitiva. [5]
- El benchmarking funcional o estratégico analiza las estrategias o procesos determinados específicamente que logran dar fortaleza a las organizaciones que forman parte de la competencia [5] el motivo de esta clase de benchmarking no es solo validar dichos procesos que dan beneficios a las organizaciones de la competencia, sino que también tiene como objetivo principal implementarlo dentro de la organización buscando fortalecer completamente todos los aspectos de la organización. [6]

1.4.4. Diseño de interfaces

Para el diseño de interfaces se realizó primero un análisis detallando como primera instancia los colores corporativos, estos colores se los debe seleccionar de una manera que permitan al usuario final tener una interacción agradable y sencilla al momento de utilizar la aplicación, para los colores corporativos del proyecto nos basamos en los

colores primarios y secundarios de Material Design, que son el color blanco y negro. [7] La guía de Material Design describe que es importante que al seleccionar los colores que permitan generar elementos interactivos dentro de las aplicaciones, que de la misma manera por los colores existan elementos que destaquen de otros teniendo así un nivel de importancia, que sean legibles y comprensibles para el usuario ya sea este mediante el uso de iconos o textos. También el material investigado explica que los colores utilizados deben ser expresivos con los colores corporativos [7] ya que la paleta que se va a usar dentro del proyecto es la más sencilla pero comúnmente usada y con buenos resultados, como lo podemos validar en aplicaciones como Uber, TikTok, BeReal entre muchas otras el uso de estos dos colores permite al usuario final poder tener una interacción satisfactoria con el sistema al igual que también evita la variación de cromatismos y exageración de colores si hubiese sido el caso.

¿Qué es una interfaz cómoda y amigable?

Se puede definir en primer lugar una interfaz es aquella página o aplicación que interactúa de forma directa con nuestro usuario final o cliente, es decir se compone de todo aquello que ve el cliente botones, imágenes, textos entre otros.

Para el desarrollo de las interfaces existen diferentes tipos de prácticas entre ellas podemos encontrar prácticas como la armonía entre elementos, claridad, síntesis, seguridad y prevención, resolver en las necesidades reales del problema y una de ellas es la práctica de menos es más [8], todas estas prácticas permiten a los desarrolladores poder generar interfaces amigables con el usuario y más que nada permiten al usuario tener un fácil uso con el producto desarrollado.

1.4.5. Metodología iterativa incremental

La metodología iterativa incremental nos permite desarrollar diferentes tipos de sistemas mediante el desarrollo de diferentes etapas consideradas en el desarrollo de la metodología en cascada éstas pueden ser análisis, diseño, implementación, pruebas y mantenimiento. Permitiendo de esta manera generar diferentes versiones del producto requerido al igual que en el desarrollo de aplicaciones con las diferentes metodologías de desarrollo ágil, esta metodología nos permite generar mejoras y cambios en cada

versión generada dando paso a un mejor y coherente desarrollo hasta tener completo el sistema. Dando cada vez mejores resultados en los entregables para los clientes en cuestión.

1.5. Herramientas de desarrollo

1.5.1. Flutter

Una herramienta para desarrollo móvil o un kit de desarrollo de software SDK (por sus siglas en inglés de software development kit) creado por Google [5] permitiendo aportes fáciles, expresivos y flexibles en la creación de las interfaces de usuario. Flutter es un SDK que permite al desarrollador manipular todo mediante widgets los mismos que permitirán tener un control de nuestra aplicación ya que permiten describir la visibilidad de la vista de la aplicación dependiendo configuraciones y estados.



Figura 1. Logotipo de Flutter.

1.5.2. Dart

Lenguaje de programación que permite optimizar aplicaciones en varias plataformas. [6]



Figura 2. Logotipo de Dart.

1.5.3. Firebase

Un gestor de base de datos en la nube ofertado por Google y muy factible para la gestión de información que podría generar la aplicación. [7] Brindando una gran gama de servicios muy utilizados en la actualidad para el desarrollo de diferentes aplicativos web o móviles.



Figura 3. Logotipo de Firebase.

1.5.4. Figma

Figma es un aplicativo web que permite tener una comunidad de colaboradores en el ámbito del diseño de interfaces para aplicaciones web, aplicaciones móviles entre otros productos digitales, con el objetivo de buscar accesibilidad para cualquier tipo de usuario. [8]



Figura 4. Logotipo de Figma.

1.5.5. Git

Git es un gestor que permite a los usuarios tener un control de revisión distribuido, rápido y escalable con diferentes tipos de archivos mediante el uso de diferentes tipos de comandos. [9]



Figura 5. Logotipo de Git.

1.5.6. GitHub

GitHub es una plataforma que ofrece una diversidad de varios productos que permiten a los usuarios clasificar y colaborar dentro de diferentes proyectos. El motor que permite la modificación de los diferentes archivos dentro de los proyectos es Git. [10]



Figura 6. Logotipo de GitHub.

1.5.7. Android Studio

Android Studio es una herramienta de Google que permite a los desarrolladores poder tener un IDE adecuado y 100% compatible para el desarrollo de aplicaciones en el ambiente de plataformas Android como celulares, wear OS, Android TV, Android Auto o Android Things. [11]



Figura 7. Logotipo de Android Studio.

1.5.8. Visual Studio Code

Visual Studio Code es un IDE multiplataforma ligera pero completamente potente como editor de código fuente o diferentes tipos de archivos teniendo también soporte y compatibilidad con diferentes tipos de lenguajes de programación. [12] Personalizable, modificable y amigable con los usuarios que se dispongan a modificar proyectos con dicha herramienta, permitiendo incluir también modificaciones de otros usuarios mediante la implementación de plugins dentro de la herramienta.



Figura 8. Logotipo de Visual Studio Code.

1.5.9. Postman

Postman es un aplicativo que permite a los desarrolladores poder generar peticiones de forma sencilla en las pruebas de un API de tipo REST. [13]



Figura 9. Logotipo de Postman.

2. METODOLOGÍA

La metodología implementada en el proyecto es el desarrollo iterativo e incremental [18], la cual permite la realización de la aplicación mediante diferentes periodos llamadas iteraciones, permitiendo desde el primer prototipo entregar pequeños avances funcionales y con contenido concreto al cliente. De esta manera se logra tener una retroalimentación necesaria para que el desarrollo sea completamente adecuado.

Esta metodología permite una reducción de modificaciones de parte del cliente ya que se tiene una constante comunicación con el mismo y de esta manera se puede tener un mejor progreso con el incremento de las diferentes necesidades o funcionalidades requeridas por el cliente, de esta manera el equipo obtiene también un incremento en la rapidez de la entrega del producto final.

2.1. Benchmarking

La realización de un benchmarking nos debe permitir ver que existe dentro del mercado que tipo de competencia se tiene y que podemos acoplar de dicha competencia para beneficio del desarrollo de nuestro producto. [19]

¿Explique benchmarking de qué va a realizar?

Para la realización del benchmarking vamos a realizar en primera instancia el benchmarking externo permitiéndonos visualizar cuáles son las características de aplicaciones similares, realizando una investigación sobre las diferentes aplicaciones dentro del mercado tanto en la App Store o en la Play Store ambas tiendas abarcan la mayor parte del mercado los cuales son iOS y Android.

Las aplicaciones que tiene que ver con el giro de negocio de nuestro proyecto de titulación son las siguientes:

- APR Alarma y aviso de remedio: en esta aplicación podemos recalcar los siguientes atributos, podemos asignar un nombre a la alarma pertinente, contiene un apartado sobre el estado de la alarma tal como activo o inactivo, se puede programar el horario necesario en el que necesitamos el aviso, se tiene un control de cada cuanto tiempo debe ser ingerido el medicamento, esto permite que solamente con la primer fecha

de ingreso sea la única para poder generar el resto de alarmas hasta que el estado de la alarma pase a un estado de inactividad, una característica interesante de la aplicación es que permite personalizar el sonido o canción para ser reproducido con la alarma.

- **Reminder: Recordatorio**, está aplicación cuenta con el acceso de widgets en el home de Android, asigna a los recordatorios diferentes tipos de prioridades valorados mediante estrellitas siendo cinco estrellas activas la más importante y siendo una estrella la menos importante, también se tiene un listado de recordatorios preprogramados por el aplicativo para el usuario.
- **Recordatorios de medicamentos**, está aplicación cuenta con notificaciones internas dentro del sistema, al igual que calcula estadísticas mediante las actividades registradas por el usuario dentro del sistema como el tipo de regularidad con la ingesta de los medicamentos de manera semanal, analiza la presión arterial, la cantidad de pasos al día o en un periodo de tiempo realizados por el paciente. Posee un conteo de medicamentos permitiendo al paciente tener un registro de cuando debería el generar una nueva cita médica para poder volver a pedir medicamentos en el subcentro de salud público más cercano o comprar los medicamentos en su defecto.
- **Med Control** es una aplicación que permite visualizar un conteo regresivo para la ingesta del próximo medicamento, permite también visualizar el tipo de ingesta o procedimiento, estos pueden ser vía oral o mediante inyección.
- **Recordatorio de medicamentos y pastillas: Medisafe**, es un aplicativo que permite acoplarse a horarios dependiendo de un régimen mediante las diferentes ingestas alimenticias como desayuno, almuerzo y merienda. Posee un apartado en el cual permite al usuario tener acceso a una búsqueda en la web a páginas enfocadas en un aspecto médico y permite también agendar citas con sus médicos de cabecera.
- **Mi recordatorio de pastillas**: es un aplicativo que cuenta con las características principales del resto de aplicaciones, pero destaca una funcionalidad sobre todas la cual permite al paciente tener un mapa y su ubicación en tiempo real de las farmacias más cercanas a su ubicación.

Según la descripción de las aplicaciones antes mencionadas se puede ver como las aplicaciones comparten ciertas características al igual que cada una de ellas posee una

característica que la diferencia del resto y las hace de cierta forma competitiva dentro del mercado de aplicaciones con la finalidad del registro de alarmas o recordatorios de ingesta medicinal, así como lo podemos ver en la tabla 1.

Tabla 1. Detalle de características de las diferentes aplicaciones usadas en el benchmarking

Característica / Aplicación	APR	Reminder: Recordatorio	Recordatorios de medicamentos	Med Control	Medisafe	Mi recordatorio de pastillas
Registro de alarma	X	X	X	X	X	X
Detallar el nombre de la alarma	X	X	X	X	X	X
Detallar la hora de ingesta	X	X	X	X	X	X
Visualización del recordatorio	X	X	X	X	X	X
Personalización del audio	X					
Prioridades en el recordatorio		X				
Estadísticas de ingesta			X			
Conteo Regresivo				X		
Búsqueda en la web					X	
Visualización de mapa con farmacias cercanas						X

2.1.1. Análisis FODA.

Como podemos observar dentro de la competencia, todas permiten poder generar un registro de la ingesta, permitiendo al cliente final que sería el usuario, crear un recordatorio personal de su registro e independientemente de cualquier otro tipo de registro que queramos ingresar. Todas las aplicaciones tienen como característica principal lo anteriormente descrito algunas sobresalen con opciones como poder generar estadísticas o validar ubicaciones de centros de salud o farmacéuticas entre otras, por ello mediante una de las reuniones con todo el equipo se determinó que el aplicativo debería tener algo que el resto de aplicativos no contenga y permita de esta forma despuntar del resto de aplicaciones. Esta característica es el poder tener un listado de seguidores con los cuales el usuario pueda compartir su registro para poder tener este control de la ingesta, otra de las ventajas del proyecto es que es un aplicativo delimitado esto quiere decir que es específico para personas con enfermedades como diabetes e hipertensión y de esta manera contine únicamente información de estas

enfermedades y cuáles son los medicamentos aprobados por el Ministerio de Salud Pública del Ecuador para poder ser recetados por los médicos pertinentes.

Tabla 2. Tabla de análisis FODA

<p>Fortalezas</p> <ul style="list-style-type: none"> - La necesidad de contar con un sistema de registro de alarmas. - Poder gestionar de forma adecuada la información dentro del sistema. 	<p>Debilidades</p> <ul style="list-style-type: none"> - No poder tener estadísticas de la información. - No tener una funcionalidad de activación y desactivación de alarmas.
<p>Oportunidades</p> <ul style="list-style-type: none"> - La implementación de poder compartir la información creada con diferentes personas para evitar dejar de ingerir los medicamentos. 	<p>Amenazas</p> <ul style="list-style-type: none"> - Caída de los servicios de terceros. - El desconocimiento tecnológico por parte del cliente final.

Definición de requerimientos funcionales y no funcionales de la aplicación.

2.1.2. Requerimientos funcionales.

Tabla 3. Requerimientos funcionales

Código	Descripción
RFIT001	Poder crear un usuario.
RFIT002	Poder modificar la información de un usuario.
RFIT003	Poder iniciar y cerrar sesión dentro del aplicativo.
RFIT004	Poder crear un registro nuevo denominado alarma.
RFIT005	Poder modificar una alarma.
RFIT006	Poder eliminar una alarma.
RFIT007	Poder visualizar las alarmas creadas.
RFIT008	Poder visualizar todos los usuarios dentro del sistema.
RFIT009	Poder seleccionar un usuario de la lista como seguidor.
RFIT010	Poder eliminar los usuarios seguidos.
RFIT011	Visualizar los usuarios seguidores.
RFIT012	Poder enviar un mensaje mediante la plataforma WhatsApp desde la pantalla de seguidores.
RFIT013	Poder recibir notificaciones push desde un servicio API de tipo REST.

2.1.3. Requerimientos no funcionales.

Tabla 4. Requerimientos no funcionales

Código	Descripción
RNFIT001	Desarrollo del aplicativo con el lenguaje de programación DART.
RNFIT002	Desarrollar el aplicativo con el SDK Flutter.
RNFIT003	Gestionar la información con Firebase.
RNFIT004	Gestionar el API de tipo REST con los servicios de Firebase.
RNFIT005	Tener un control de la longitud de la contraseña del usuario.
RNFIT006	Realizar una interfaz cómoda y amigable para el usuario final.

2.2. Prototipado

El prototipado permite generar una base tangible de cómo será el producto final más que nada en funcionalidad esto permite que tanto el desarrollador como el cliente van a poder tomar las decisiones necesarias antes de comenzar el proceso de desarrollo e implementación. De esta manera la interacción cliente – desarrollador es más proactiva con forme a las necesidades del cliente y permitiendo llegar a un producto objetivo conciso y definido.

2.2.1. Fases del prototipado

La implementación de prototipos en los proyectos por lo general permite al desarrollador tener solo una base o una guía terminando estos prototipos solamente como documentación del proceso que se tuvo en el desarrollo de este, pero también existen prototipos que como en nuestro caso son pequeños entregables los cuales permiten al desarrollador ir trabajando en marcha sobre el producto final.

Para lograr conseguir esta clase de prototipos se realizan las siguientes fases permitiendo tener una interacción pertinente del cliente con el equipo de desarrollo para llegar de forma adecuada al producto final. Entre ellos encontramos la comunicación, fase que permite entablar un diálogo entre las partes interesadas definiendo los requerimientos del cliente y las capacidades del desarrollador con respecto a estos requerimientos, permitiendo llegar a un acuerdo entre ambas partes para tener un entregable formal por iteración. Plan rápido,

fase que permite la planificación en alcance y tiempo de duración necesario para cada uno de los prototipos tratados con el cliente en la reunión o fase de comunicación. Modelado y diseño rápido, en este apartado tenemos la creación del mockup que permitirá detallar los aspectos de software visibles por los usuarios del producto final como por ejemplo las interfaces a ser utilizadas y la navegación entre pantallas. Construcción, el grupo de desarrollo se encarga de construir de forma sencilla un aplicativo basándose en la fase plan rápido y en la fase de modelado y diseño rápido, los cuales fueron definidos con antelación por las partes interesadas. Llegando finalmente a la fase de despliegue en la cual el equipo de desarrollo muestra el prototipo a los usuarios obteniendo de ellos una retroalimentación con nuevos requerimientos siendo estos mejoras o modificaciones en funcionalidad, también se puede obtener como retroalimentación cambios requeridos por el cliente.

Finalizando todos estos procesos y analizando la retroalimentación de los usuarios y del cliente se repite de forma iterativa nuevamente las fases anteriores llegando de esta manera a generar las iteraciones necesarias e incrementando las retroalimentaciones de cada una de las iteraciones, hasta llegar al producto final deseado por el cliente.

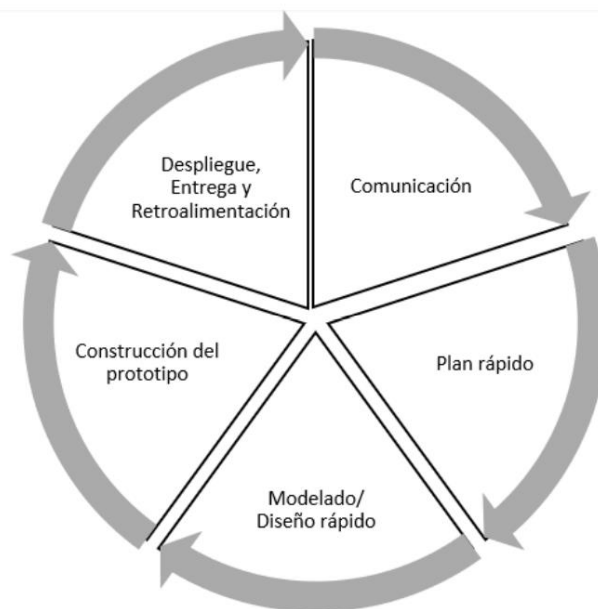


Figura 10. Fases del prototipado.

2.3. Prototipo móvil

Para el desarrollo del prototipo móvil se van a generar iteraciones que permitan al cliente poder tener un producto entregable basados en los requerimientos funcionales.

2.3.1. Iteración 0

En la iteración cero solventaremos todos los requerimientos necesarios hasta poder cumplir con los requerimientos funcionales: “RFIT001”, “RFIT002”, “RFIT003”. Y los requerimientos no funcionales: “RNFIT001”, “RNFIT002”, “RNFIT003”, “RNFIT004”, “RNFIT005”, “RNFIT006”.

2.3.1.1. Prototipo 0

2.3.1.1.1. Comunicación

En la primera reunión con el cliente se llegó a la definición de los siguientes requerimientos iniciales y al alcance de la aplicación, que tiene como objetivo permitir tener un control de la ingesta de medicamentos a pacientes con enfermedades como diabetes e hipertensión.

2.3.1.1.2. Plan rápido

2.3.1.1.2.1. Objetivos

- Diseñar e implementar la base de datos con las entidades correspondientes.
- Diseñar la arquitectura del aplicativo móvil.
- Configurar el ambiente de desarrollo necesario para la construcción del aplicativo móvil.
- Probar la conexión entre la base de datos con el aplicativo móvil.

2.3.1.1.2.2. Historias de usuario

Para la realización de las historias de usuario se determinará un código, la definición de la historia de usuario dependiendo de los objetivos planteados, se asignará un peso y una

prioridad por historia de usuario, los mismos que serán evaluados y estimados por el desarrollador. Para está estimación se utilizará la técnica de Planning Poker.

Tabla 5. Historias de usuario para el prototipo 0.

Código	Historia de usuario	Peso	Prioridad
HUIT001	Diseñar y construir la base de datos.	8	Alta
HUIT002	Diseñar la arquitectura del aplicativo móvil.	5	Alta
HUIT003	Configurar el ambiente de desarrollo.	8	Alta
HUIT004	Probar la conexión del aplicativo con la base de datos.	5	Alta
	Total	26	-

2.3.1.1.2.3. Tiempo estimado

Tabla 6. Tiempo estimado para el prototipo 0.

Prototipo	Tiempo estimado	Desde	Hasta
0	3 días	16-02-2022	18-02-2022

2.3.1.1.3. Diseño

Basado en la realización de los mockups se obtuvo el tipo de información necesaria para cada uno de los documentos necesarios a gestionar en la base de datos de Firestore.

Documentos

1. Gestión de usuarios
2. Gestión de alarmas

La arquitectura que se va a implementar es Cliente-Servidor está arquitectura nos permitirá consumir información necesaria para el cliente en caso de necesitarla, está basada mayormente en dos actores el proveedor que en nuestro caso es el servidor el mismo que nos va a permitir gestionar los diferentes tipos de peticiones, las mismas que permitirán guardar o enviar información entre el proveedor y el cliente de esta manera se generará la información contenida en las diferentes interfaces y el consumidor que serían todos los clientes o usuarios dentro de nuestra base de datos. [14] Los clientes serán los que provean

de información a nuestro servidor y serán aquellos que hagan peticiones para poder visualizar la información contenida. La *figura 11* detalla de forma gráfica la arquitectura cuando el backend se gestiona con los servicios de Firebase.



Figura 11. Arquitectura Cliente-Servidor con Firebase

La arquitectura de diseño implementada dentro del aplicativo es la arquitectura M.V.C. la cual significa Modelo-Vista-Controlador, como se implementa esta arquitectura dentro del proyecto.

Para poder entender cómo se utiliza esta arquitectura debemos primero entender cómo se encuentra estructurada nuestra base de datos, al ser una base no relacional está conformada por documentos los mismos son la forma en la que esta clase de bases de datos almacena la información mediante el uso del tipo de variable JSON usado por varios lenguajes, pero desarrollado por los creadores de JavaScript, ya que JSON significa JavaScript Object Notation en sus siglas en inglés o Notación de Objeto JavaScript en español.

De esta manera organizan los documentos las bases no relacionales, en nuestro proyecto las bases no relacionales tienen la siguiente estructura.

El aplicativo consta de cuatro documentos que contendrán la información requerida entre ellos tenemos “alarm”, “followers”, “pills”, “users”. Cada documento contiene una estructura definida de la siguiente manera clave - valor.

- El documento “alarm” contiene las siguientes claves: “disease”, “pill”, “time”, “uid”.
- El documento “followers” contiene las siguientes claves: “follower”, “user”.
- El documento “pills” contiene las siguientes claves: “diabetes”, “hypertension”. En este caso tenemos documentos dentro de documentos y la estructura de ambos es mediante las siguientes claves: “name”, “time”.

- El documento “users” contiene las siguientes claves: “email”, “id”, “lastname”, “name”, “phone”, “token”.

Una vez entendido la forma de contener la información dentro de nuestro backend podemos entender de mejor manera cómo funciona la arquitectura de diseño dentro de nuestro proyecto basándonos en la estructura de los documentos se parte para la creación de los modelos necesarios a ser realizados, posterior a ello se gestionan las vistas las cuales son las interfaces que verá el usuario donde se encontraran los componentes necesarios para la interacción con cliente-servidor en dichas interfaces se encontraran formularios donde el usuario ingresará la información requerida al igual que existirán interfaces donde el usuario podrá ver reflejada la información contenida en el servidor de una manera legible, y para finalizar los controladores serán los archivos contenidos en las carpetas “providers” y “services” archivos que permitirán tener el control de los datos a ser almacenados en el servidor.

El patrón para usarse en la construcción del aplicativo es el patrón Provider el mismo que nos permitirá poder tener un mayor control de los estados que se van a ver involucrados dentro del flujo del aplicativo. Este patrón es la lógica que nos permite tener el control de nuestros modelos anteriormente explicados.

La *figura 12* explica de forma gráfica el manejo de providers dentro del aplicativo encapsulando la información que será requerida en diferentes puntos de la aplicación dentro de un provider para que de esta manera dicha información pueda ser consumida, modificada y mostrada dependiendo del componente que necesite del provider.

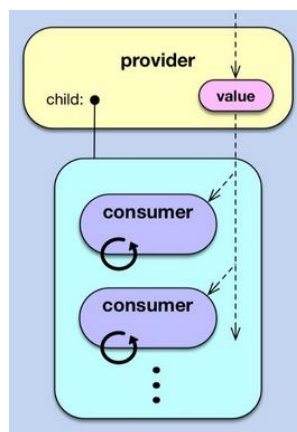


Figura 12. Implementación del patrón provider.

Flutter al manejar todos sus componentes como widgets los redibuja en tiempo de ejecución por lo que trabajar con el patrón provider podemos tener un control de los estados de los widgets y de la información que necesitemos en los diferentes widgets.

En la *figura 13* podremos visualizar el diagrama de navegación de la aplicación móvil, para dicha navegación se realizó mediante el análisis de los módulos existentes.

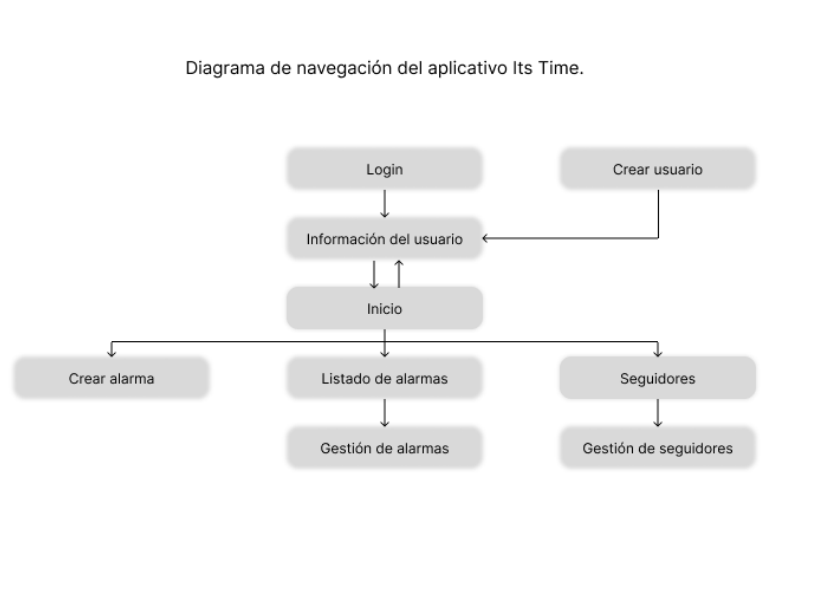


Figura 13. Diagrama de navegación.

2.3.1.1.4. Construcción

El primer proceso para realizarse fue el diseño de los prototipos de baja fidelidad los mismos que están representados en la *figura 14* y *anexo 1*, para ello utilizamos la aplicación web Figma la cual nos permite generar los diseños de la aplicación móvil para poder presentar una idea inicial al cliente de cómo podría ser el producto final y poder abstraer los posibles campos necesarios para la realización de los documentos gestionados en el gestor de base de datos de Firebase.

¿Cuáles fueron las prácticas que se utilizaron para el diseño de las interfaces?

Para el diseño de las interfaces enfocamos las siguientes prácticas como necesarias para el desarrollo de cada una de las pantallas del aplicativo. La primera práctica fue resolver de manera óptima las necesidades del aplicativo en nuestro caso los requerimientos levantados

en la primer reunión, posterior a ello y mediante el uso de las guías de material design validamos los colores corporativos o paleta de colores a usarse dentro del entorno visual del aplicativo, para ello tomamos una paleta que daba mayor facilidad y solvencia contenida por dos colores el blanco y negro, colores que nos servirán para el desarrollo de las interfaces. [19] Este procedimiento permite al equipo solventar prácticas como la armonía entre los elementos tratando de mitigar al máximo que el aplicativo tenga una gama extensa de colores, ayudando con claridad para los usuarios. En el apartado de armonía entre los elementos también se optó por tratar de estandarizar todos los botones a utilizarse exceptuando los botones que utilizan únicamente un icono como base, el resto de los botones con texto interno tienen una forma similar.

Una de las mejores prácticas que se utilizó dentro del desarrollo de las interfaces cómodas y amigables del proyecto fue la práctica denominada “menos es más”, esta práctica permite no aglomerar de mucha información o componentes visuales a las interfaces evitando al usuario una fatiga al momento de utilizar el aplicativo ya que mientras mayor contenido tenga su uso se verá reflejado con una mayor dificultad, mientras que si las interfaces que solamente contienen el contenido justo y necesario para su interactividad van a ser más amigables con el usuario ya que no tendrá que sacrificar mucho de su tiempo para su uso y esto tendrá como resultado un mayor y fácil entendimiento de la lógica de negocio del proyecto, del porque tomar esta práctica es sencillo ya que la mayor parte de clientes finales o potenciales clientes que son personas con enfermedades como diabetes e hipertensión suelen ser personas con una edad avanzada las cuáles tienen una interacción muy reducida con cualquier aspecto tecnológico.

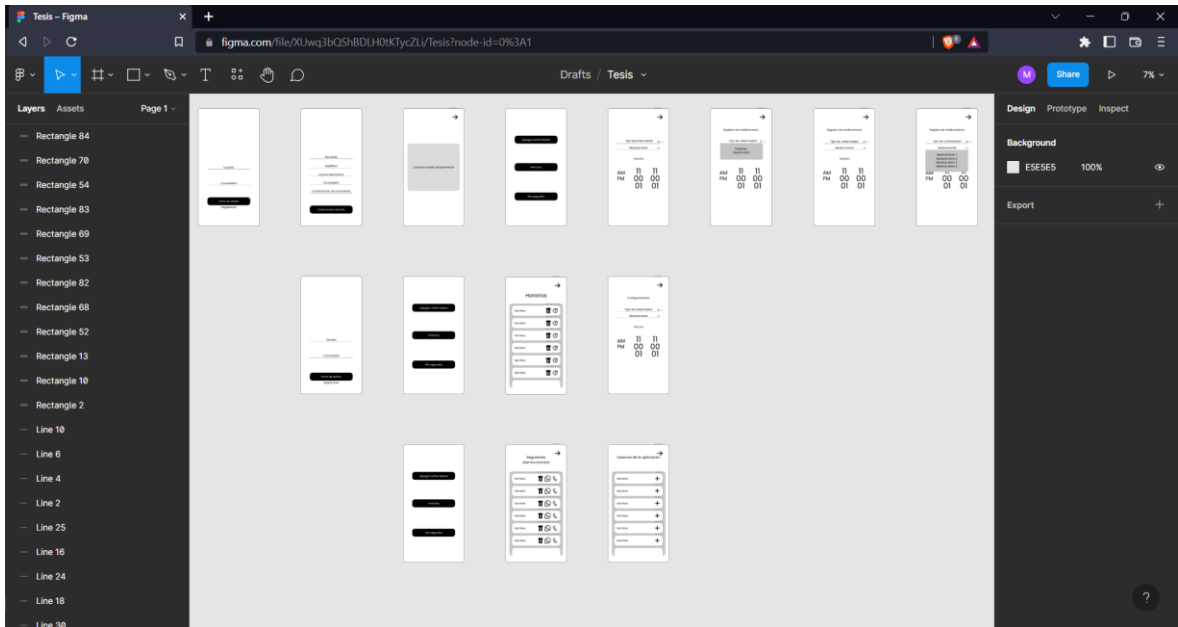


Figura 14. Creación de los mockups en Figma

En la *figura 15* podremos ver la creación de la base de datos en Firebase la cual nos permitirá tener el control de los datos de nuestra aplicación en donde gestionaremos los documentos.

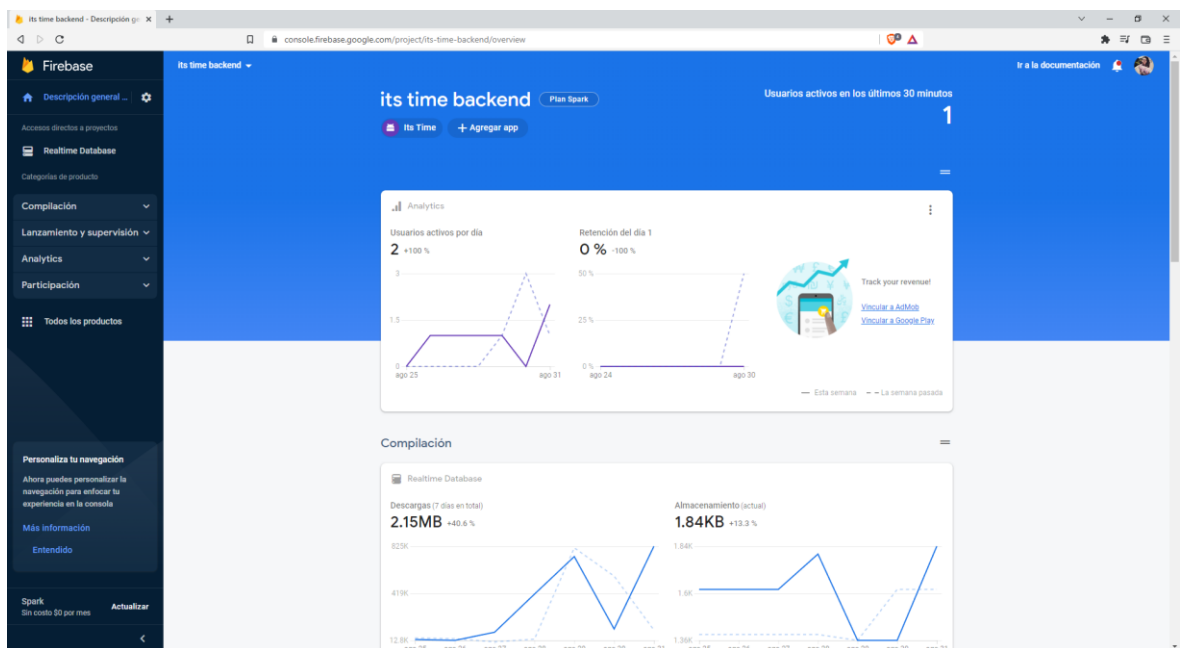


Figura 15. Base de datos Its time.

En la *figura 16* se puede visualizar la página donde se tiene el control de la base de datos no relacional generada para la posterior implementación con el aplicativo móvil.

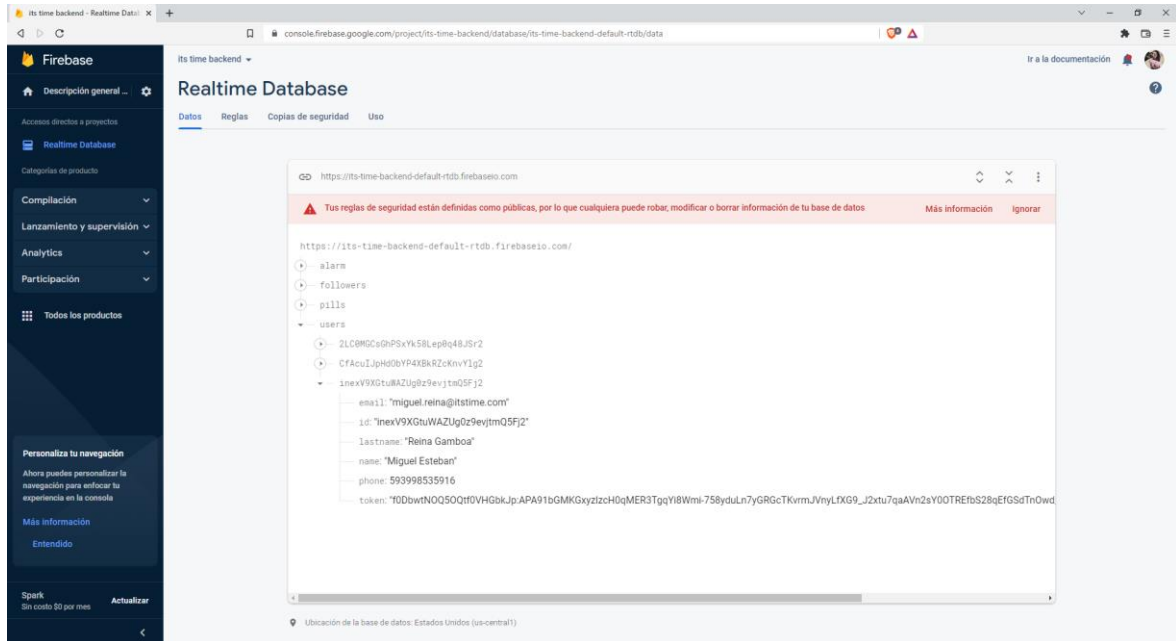


Figura 16. Visualización de los documentos de Firebase.

Para la culminación de la construcción del prototipo 0 procedemos a validar la información de la base de datos dentro del aplicativo móvil ya que de esta manera se puede ratificar la conexión del aplicativo con la base de datos. Dando por culminada la historia de usuario HUIT004.

Información del usuario

Nombre del usuario
Miguel Esteban

Apellidos del usuario
Reina Gamboa

Correo electrónico
miguel.reina@itstime.com

Número de teléfono
593998535916

Guardar

Figura 17. Pantalla de información del usuario, validación de conexión con la base de datos.

2.3.1.1.5. Despliegue

La culminación de la construcción del prototipo 0 se realizaron de forma satisfactoria, la estimación de realizada de peso y prioridad de las historias de usuario no tuvieron que ser modificadas.

Tabla 7. Modificación de análisis de historias de usuarios del prototipo 0.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT001	8	8	-	100%	Alta	Alta
HUIT002	5	5	-	100%	Alta	Alta
HUIT003	8	8	-	100%	Alta	Alta
HUIT004	5	5	-	100%	Alta	Alta
Total	26	26	0	100%	-	-

2.3.1.2. Prototipo 1

2.3.1.2.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 0 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la pantalla de “Inicio de sesión” de la aplicación.

2.3.1.2.2. Plan rápido

2.3.1.2.2.1. Objetivos

- Diseñar las pantallas del aplicativo móvil asignadas para el prototipo 1.
- Construir el prototipo 1 del aplicativo en base a los mockups presentados con anterioridad al cliente.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 1 según los criterios de aceptación definidos en las historias de usuario.

2.3.1.2.2.2. Historias de usuario

Tabla 8. Historias de usuario para el prototipo 1.

Código	Historia de usuario	Peso	Prioridad
HUIT005	Registro de un nuevo usuario en el aplicativo.	5	Alta
HUIT006	Ingreso del usuario en el aplicativo.	5	Alta
HUIT007	Visualización del menú principal del aplicativo.	2	Media
	Total	12	-

2.3.1.2.2.3. Tiempo estimado

Tabla 9. Tiempo estimado para el prototipo 1.

Prototipo	Tiempo estimado	Desde	Hasta
1	4 días	18-02-2022	21-02-2022

2.3.1.2.3. Diseño

Firestore tiene la disponibilidad de que los desarrolladores tengan una facilidad para la implementación de una autenticación de usuarios dentro de aplicaciones existentes o aplicaciones a ser desarrolladas, estos servicios permiten al desarrollador crear un servicio de autenticación mediante plataformas existentes como Google, Facebook entre otros el

seleccionado para el aplicativo fue el más sencillo sin la conexión a un tercero, el cual nos permite ingresar un correo electrónico y una contraseña las cuales se guardaran dentro de un apartado específico dentro de nuestra base de datos denominado Autenticación, visualizado en la *figura 18*.

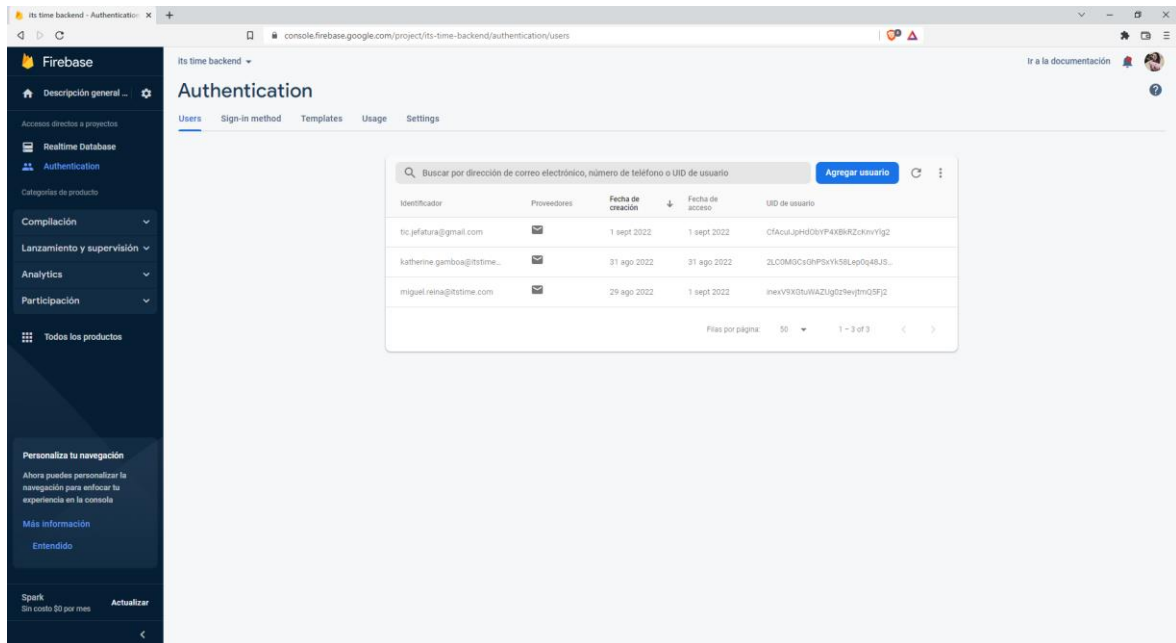


Figura 18. Servicio de autenticación provisto por Firebase.

2.3.1.2.4. Construcción

Para ello deberemos comenzar a consumir el API de Firebase, nos ayudamos de Postman para poder implementar y probar las peticiones que va a generar nuestra aplicación. Este servicio nos permite tener información como el correo ingresado, contraseña, fecha de ingreso, un token y más información dependiendo de los requerimientos del cliente, en el proyecto únicamente se necesitara correo con el que el usuario se registra, la contraseña creada y su token para poder diferenciar el usuario, ya que para poder tener un mejor control y al ser este un servicio distinto al de nuestra base de datos denominada en tiempo real, se procede a generar un documento de usuarios con la información necesaria.



Figura 19. Pantalla para la creación de un nuevo usuario.



Figura 20. Ingreso de información del nuevo usuario.

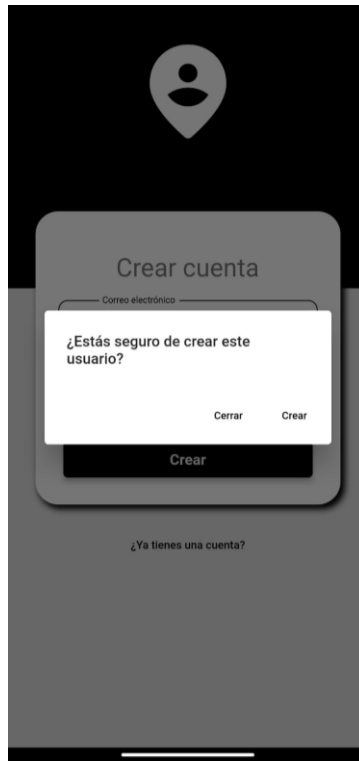


Figura 21. Alerta de creación de un nuevo usuario.

Posterior a la creación del usuario el sistema le permite ingresar dentro de una pantalla donde del cliente deberá llenar información requerida para la funcionalidad necesaria para satisfacer el objetivo del proyecto.

11:37 PM 61%

Información del usuario

Nombre del usuario

Apellidos del usuario

Correo electrónico
miguel.reina@epn.edu.ec

Número de teléfono
593999999999

Guardar

Figura 22. Formulario para la actualización de información del usuario.

11:37 PM 61%

Información del usuario

Nombre del usuario
Miguel Esteban

Apellidos del usuario
Reina Gamboa

Correo electrónico
miguel.reina@epn.edu.ec

Número de teléfono
593998535916

Guardar

Figura 23. Ingreso de la información requerida por usuario.

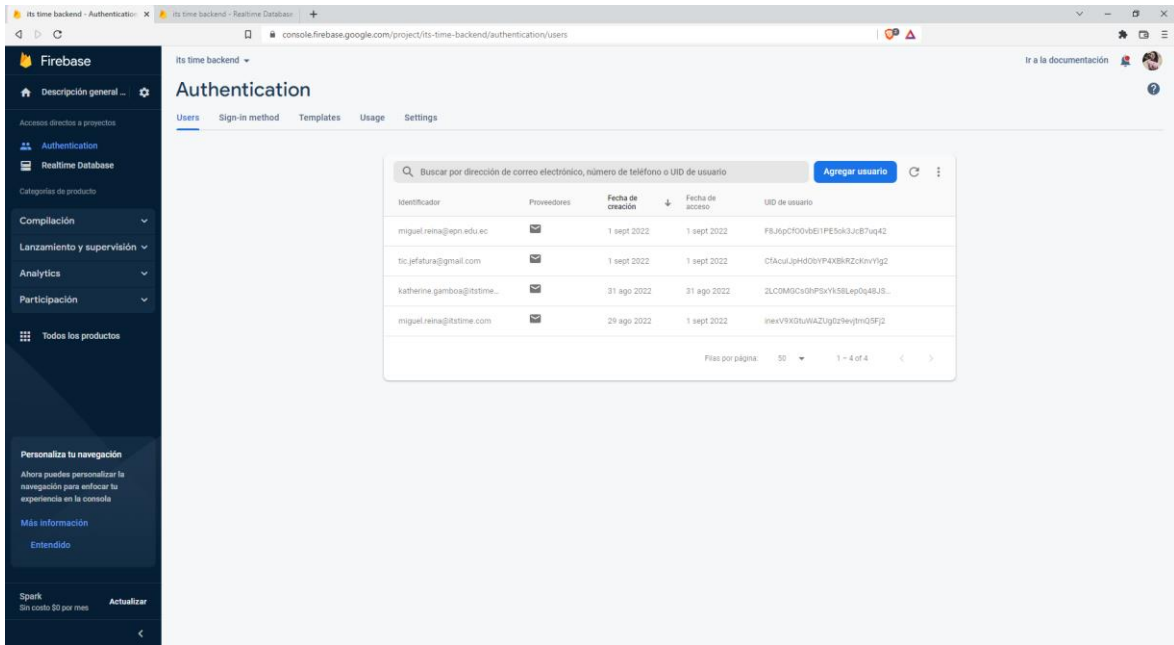


Figura 24. Validación del nuevo usuario dentro del servicio de autenticación.

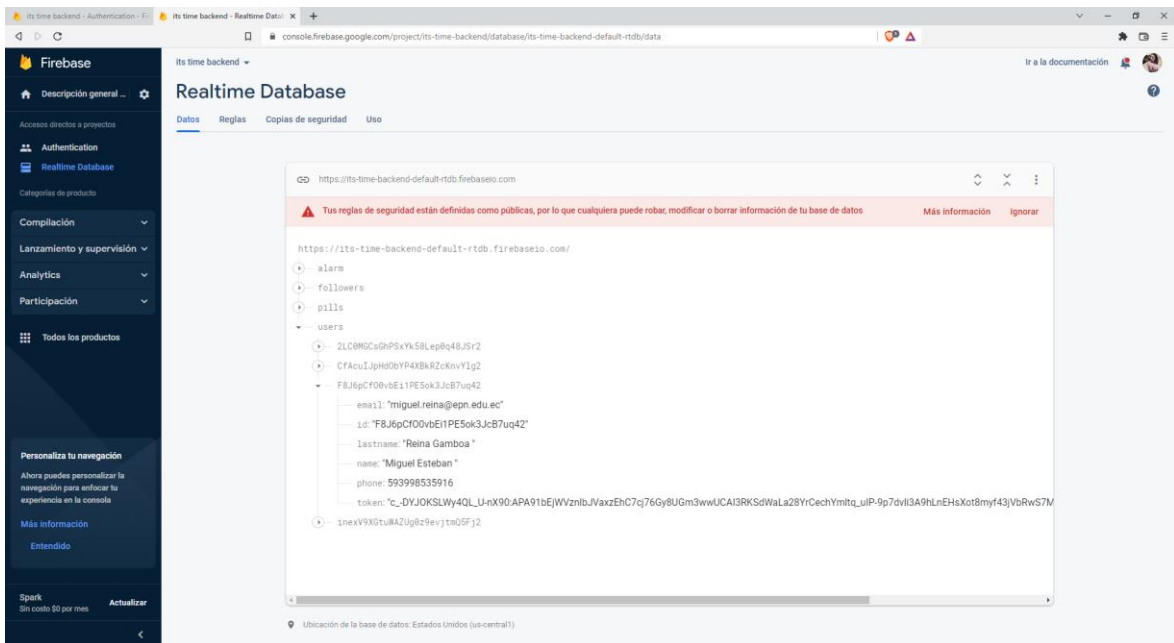


Figura 25. Validación del nuevo usuario dentro del documento usuarios en la base de datos.

Posterior a la validación de la información del usuario se navega a la pantalla de inicio.



Figura 26. Pantalla de inicio.

2.3.1.2.5. Despliegue

Para el despliegue del prototipo 1 no se modificaron las estimaciones en peso y prioridad.

Tabla 10. Modificación de análisis de historias de usuarios del prototipo 1.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT005	5	5	-	100%	Alta	Alta
HUIT006	5	5	-	100%	Alta	Alta
HUIT007	2	2	-	100%	Media	Media
Total	12	12	0	100%	-	-

2.3.2. Iteración 1

En la iteración uno solventaremos todos los requerimientos necesarios hasta poder cumplir con los requerimientos funcionales “RFIT004”, “RFIT005”, “RFIT006”, “RFIT007”. Y los requerimientos no funcionales: “RNFIT003”, “RNFIT004”, “RNFIT006”.

2.3.2.1. Prototipo 2

2.3.2.1.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 1 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la pantalla de “Crear alarma” de la aplicación.

2.3.2.1.2. Plan rápido

2.3.2.1.2.1. Objetivos

- Diseñar la pantalla del aplicativo móvil asignadas para el prototipo 2.
- Construir el prototipo 2 del aplicativo en base a los mockups.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 2 según los criterios de aceptación definidos en las historias de usuario.

2.3.2.1.2.2. Historias de usuario

Tabla 11. Historias de usuario para el prototipo 2.

Código	Historia de usuario	Peso	Prioridad
HUIT008	Creación del formulario para crear una nueva alarma.	8	Alta
HUIT009	Validación de la creación de la alarma en la base de datos.	5	Media
HUIT010	Visualización de la información mediante Postman.	2	Media
	Total	15	-

2.3.2.1.2.3. Tiempo estimado

Tabla 12. Tiempo estimado para el prototipo 2.

Prototipo	Tiempo estimado	Desde	Hasta
2	4 días	05-03-2022	08-03-2022

2.3.2.1.3. Diseño

Partiendo de las historias de usuario generadas para el prototipo 2 se prosigue a la creación del formulario para la creación de alarmas tomando como referencia los mockups, visualizados en la figura 27 los cuales sirvieron como base para la creación del formulario final.

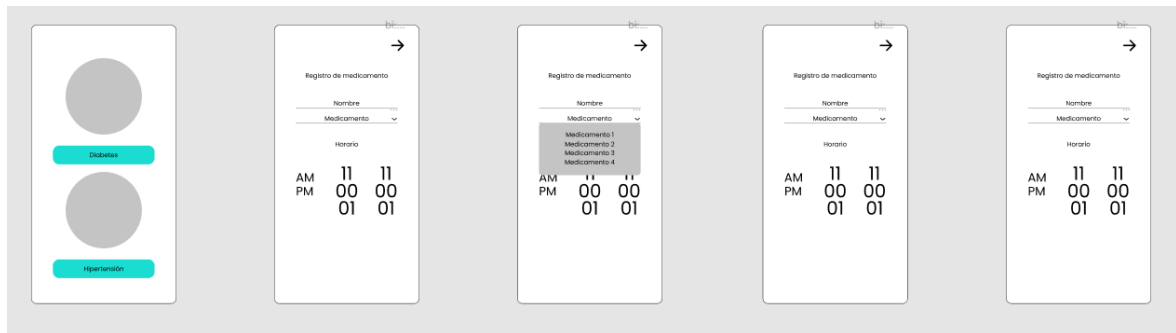


Figura 27. Prototipo de baja fidelidad del formulario de creación de alarmas.

2.3.2.1.4. Construcción

Para la realización del formulario en primera instancia generamos los puntos de consulta de nuestra base de datos mediante el API provista por Firebase posterior a ello generamos los servicios que nos permitirán el envío de la información desde nuestro formulario a la base de datos. Generada la lógica de consumo se construyó el formulario el cuál contendrá una validación donde dependiendo del tipo de enfermedad manejado con una variable guardada en un provider podremos desplegar un segundo dropdown que nos permita visualizar la información específica por enfermedad esta información hace alusión a los medicamentos aprobados por el ministerio de salud pública de la República del Ecuador.

En las siguientes figuras vamos a poder apreciar el funcionamiento del formulario creado para el ingreso de una nueva alarma dentro del sistema.

The screenshot shows a mobile application interface for creating a new alarm. At the top, there is a black header with the text "Nueva alarma". Below the header, the form is divided into three sections:

- Seleccione la enfermedad:** A dropdown menu with the text "Tipo de enfermedad" and a downward arrow.
- Seleccione el medicamento:** A dropdown menu that is currently empty.
- Seleccione la hora de consumo:** A time selection interface showing "0 hours" and "0 min." with a list of numbers (1, 2, 3) for each unit.

At the bottom of the form, there is a black button with the text "Guardar".

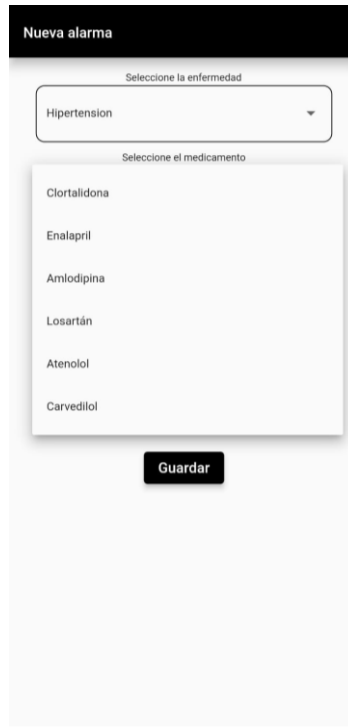
Figura 28. Formulario para la creación de una alarma.

This screenshot shows the same "Nueva alarma" form, but with the "Diabetes" option selected in the "Seleccione la enfermedad" dropdown. The "Selección el medicamento" dropdown is now open, displaying a list of medication names:

- Ácido acetil salicílico
- Clopidogrel
- Glibenclamida
- Glicazida
- Glucagón
- Insulina
- Insulina intramuscular
- Losartan
- Metformina
- Simvastatina

The "Guardar" button remains at the bottom of the form.

Figura 29. Formulario con las opciones para la enfermedad "diabetes".



Nueva alarma

Seleccione la enfermedad

Hipertension

Seleccione el medicamento

Clortalidona

Enalapril

Amlodipina

Losartán

Atenolol

Carvedilol

Guardar

Figura 30. Formulario con las opciones para la enfermedad “hipertensión”.



Nueva alarma

Seleccione la enfermedad

Hipertension

Seleccione el medicamento

Enalapril

Seleccione la hora de consumo

14	57
15	58
16	59
17 hours	0 min.
18	1
19	2
20	3

Guardar

Figura 31. Formulario con la información necesaria para crear una alarma.

Para poder ratificar el ingreso de la información dentro de nuestra base de datos hacemos uso del aplicativo Postman el cual nos permitirá mediante un end point realizar la consulta de las alarmas existentes dentro de nuestra base de datos.

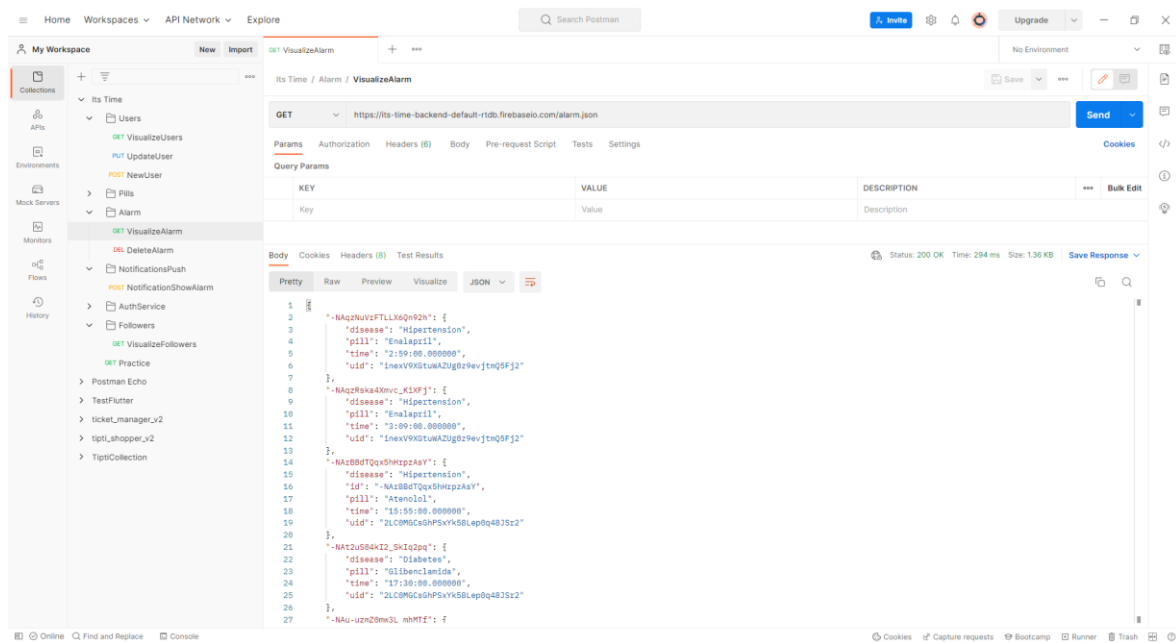


Figura 32. Consulta de alarmas creadas mediante Postman.

2.3.2.1.5. Despliegue

Para el despliegue del prototipo 2 no se modificaron las estimaciones en peso y prioridad.

Tabla 13. Modificación de análisis de historias de usuarios del prototipo 2.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT008	8	8	-	100%	Alta	Alta
HUIT009	5	5	-	100%	Media	Media
HUIT010	2	2	-	100%	Media	Media
Total	15	15	0	100%	-	-

2.3.2.2. Prototipo 3

2.3.2.2.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 2 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la pantalla de gestión de alarmas denominada “Listado de alarmas” dentro de la aplicación.

2.3.2.2.2. Plan rápido

2.3.2.2.2.1. Objetivos

- Diseñar las pantallas del aplicativo móvil asignadas para el prototipo 3.
- Construir el prototipo 3 del aplicativo en base a los prototipos de baja fidelidad.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 3 según los criterios de aceptación definidos en las historias de usuario.

2.3.2.2.2.2. Historias de usuario

Tabla 14. Historias de usuario para el prototipo 3.

Código	Historia de usuario	Peso	Prioridad
HUIT011	Creación de la navegabilidad de pantallas para la gestión de alarmas.	5	Media
HUIT012	Generación de widgets reutilizables.	2	Baja
HUIT013	Listar las alarmas posteriormente creadas.	5	Alta
HUIT014	Actualizar una alarma seleccionada.	8	Alta
HUIT015	Eliminar una alarma seleccionada.	5	Alta
	Total	25	-

2.3.2.2.2.3. Tiempo estimado

Tabla 15. Tiempo estimado para el prototipo 3.

Prototipo	Tiempo estimado	Desde	Hasta
3	10 días	20-03-2022	30-03-2022

2.3.2.2.3. Diseño

Mediante la guía de los prototipos de baja fidelidad generados al inicio del proyecto se fue generando cada una de las pantallas necesarias para la gestión de alarmas, satisfaciendo las historias generadas para el prototipo 3.

2.3.2.2.4. Construcción

Para la creación de la navegabilidad de la gestión de las alarmas dentro del sistema se generaron los widgets necesarios para la navegabilidad de la pantalla de inicio, a una pantalla que me permita visualizar las alarmas existentes mediante un widget personalizado con la información de la alarma, entre ellos el nombre de la enfermedad, el nombre del medicamento y la hora de la alarma, tal como se muestra en la *figura 33*, dentro de este widget también podremos visualizar las opciones para su posterior gestión.

Cada tarjeta contenida con la información de una alarma se podrá eliminar con el icono de color rojo representado con un basurero, el icono de color negro situado al lado derecho representado con una flecha circular, nos permitirá seleccionar la alarma navegando a un formulario en donde el usuario podrá tener la posibilidad de modificar la información de esta. Este formulario lo podemos visualizar en la *figura 34*, en la cual se selecciona la primera alarma del listado de alarmas de la *figura 33*.

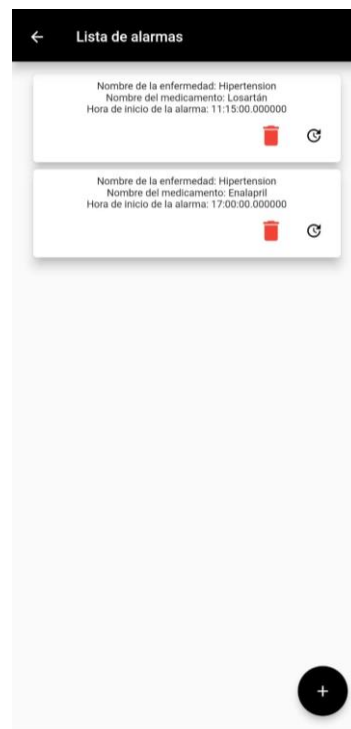


Figura 33. Pantalla con el listado de alarmas.

The screenshot shows a mobile application interface for updating an alarm. At the top, there is a black header with the text 'Actualizar alarma'. Below the header, there are three sections, each with a title and a dropdown menu:

- Seleccione la enfermedad:** A dropdown menu with 'Hipertension' selected.
- Seleccione el medicamento:** A dropdown menu with 'Losartán' selected.
- Seleccione la hora de consumo:** A time picker interface showing '11 hours 15 min.' selected. The time is displayed in a grid format with hours and minutes.

At the bottom of the form, there is a black button with the text 'Guardar'.

Figura 34. Pantalla con el formulario de actualización.

2.3.2.2.5. Despliegue

Para el despliegue del prototipo 3 no se modificaron las estimaciones en peso y prioridad.

Tabla 16. Modificación de análisis de historias de usuarios del prototipo 3.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT011	5	5	-	100%	Media	Media
HUIT012	2	2	-	100%	Baja	Baja
HUIT013	5	5	-	100%	Alta	Alta
HUIT014	8	8	-	100%	Alta	Alta
HUIT015	5	5	-	100%	Alta	Alta
Total	25	25	0	100%	-	-

2.3.3. Iteración 2

En la iteración uno solventaremos todos los requerimientos necesarios hasta poder cumplir con los requerimientos funcionales “RFIT004”, “RFIT005”, “RFIT006”, “RFIT007”. Y los requerimientos no funcionales: “RNFIT003”, “RNFIT004”, “RNFIT006”.

2.3.3.1. Prototipo 4

2.3.3.1.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 3 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la pantalla de “Seguidores” de la aplicación.

2.3.3.1.2. Plan rápido

2.3.3.1.2.1. Objetivos

- Diseñar las pantallas del aplicativo móvil asignadas para el prototipo 4.
- Construir el prototipo 4 del aplicativo en base a los mockups presentados con anterioridad al cliente.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 4 según los criterios de aceptación definidos en las historias de usuario.

2.3.3.1.2.2. Historias de usuario

Tabla 17. Historias de usuario para el prototipo 4.

Código	Historia de usuario	Peso	Prioridad
HUIT016	Creación de pantallas y navegación para el apartado de seguidores.	5	Alta
HUIT017	Visualización de todos los usuarios existentes dentro de la base de datos.	5	Alta
HUIT018	Implementación de la lógica y los servicios necesarios para la asignación de seguidores.	8	Alta
HUIT019	Visualización de los usuarios seguidores.	5	Alta
	Total	23	-

2.3.3.1.2.3. Tiempo estimado

Tabla 18. Tiempo estimado para el prototipo 4.

Prototipo	Tiempo estimado	Desde	Hasta
4	6 días	11-04-2022	16-04-2022

2.3.3.1.3. Diseño

Mediante la guía de los prototipos de baja fidelidad generados al inicio del proyecto se fue generando cada una de las pantallas necesarias para la gestión de alarmas, satisfaciendo las historias generadas para el prototipo 4.

2.3.3.1.4. Construcción

Para la construcción de este prototipo se realizó primero la navegación entre las diferentes pantallas a ser utilizadas, comenzando el flujo desde la pantalla de inicio siguiendo a una pantalla de seguidores la misma que en un principio se encontrara vacía como se podrá visualizar en la figura 35, en la parte inferior derecha encontraremos un botón que tendrá el icono más redirigiéndonos a la pantalla donde encontraremos todos los usuarios registrados dentro de nuestra base de datos evidenciado en la figura 36, la tarjeta de usuarios contendrá un espacio definido para una imagen la cual podrá ser gestionada en un prototipo posterior, el nombre del usuario y en la parte inferior derecha un icono en color negro definido con un signo de más el cual nos permitirá registrarlo como un seguidor.

De esta forma si regresamos a la pantalla de seguidores podremos visualizar los seguidores agregados a nuestra lista, tal y como se puede apreciar en la *figura 37*.

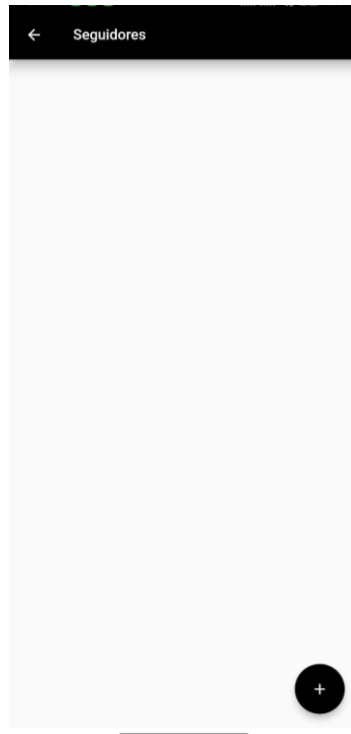


Figura 35. Pantalla de seguidores al iniciar la aplicación.

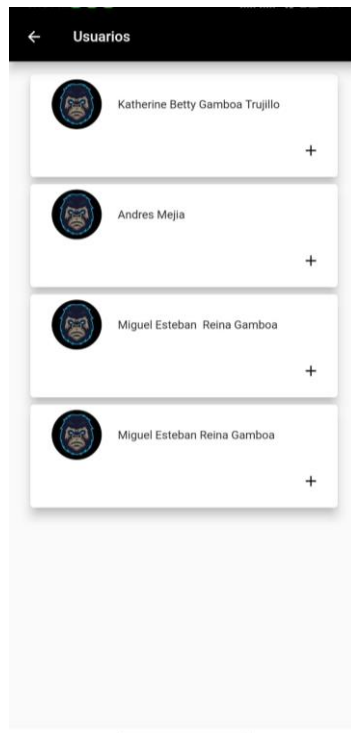


Figura 36. Pantalla que contiene todos los usuarios registrados en la aplicación.

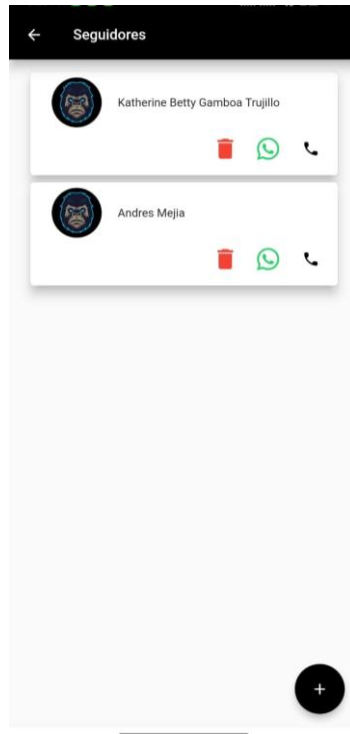


Figura 37. Pantalla donde se visualiza los usuarios seguidores.

En la *figura 37* podemos apreciar que en la parte inferior de cada una de las tarjetas de nuestros usuarios seguidores tenemos 3 iconos los cuales nos permitirán como clientes poder realizar las siguientes acciones, el icono de color rojo identificado con un basurero nos permitirá eliminar dicho seguidor de nuestra lista, el icono de color verde identificado con el logotipo de la red social WhatsApp permitirá al usuario poder generar un chat con dicha persona para de esta manera poder tener la posibilidad de avisarle en caso de que tenga una alarma pospuesta, al igual que el icono negro del final identificado con el símbolo de un teléfono convencional permitirá generar una llamada local al usuario en cuestión.

2.3.3.1.5. Despliegue

Para el despliegue del prototipo 4 no se modificaron las estimaciones en peso y prioridad.

Tabla 19. Modificación de análisis de historias de usuarios del prototipo 4.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT016	5	5	-	100%	Alta	Alta
HUIT017	5	5	-	100%	Alta	Alta

HUIT018	8	8	-	100%	Alta	Alta
HUIT019	5	5	-	100%	Alta	Alta
Total	23	23	0	100%	-	-

2.3.4. Iteración 3

En la iteración uno solventaremos todos los requerimientos necesarios hasta poder cumplir con los requerimientos funcionales “RFIT012”, “RFIT013”. Y los requerimientos no funcionales: “RNFIT003”, “RNFIT004”, “RNFIT006”.

2.3.4.1. Prototipo 5

2.3.4.1.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 4 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la lógica para el envío de notificaciones push emitidas desde el servidor al cliente.

2.3.4.1.2. Plan rápido

2.3.4.1.2.1. Objetivos

- Diseñar el formato de la información requerida para la notificación push requerido en el prototipo 5.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 5 según los criterios de aceptación definidos en las historias de usuario.

2.3.4.1.2.2. Historias de usuario

Tabla 20. Historias de usuario para el prototipo 5.

Código	Historia de usuario	Peso	Prioridad
HUIT020	Revisión de documentación para la implementación de notificaciones push provistas por Firebase.	2	Baja

HUIT021	Configuración del aplicativo para la recepción de notificaciones push desde el servidor.	8	Alta
HUIT022	Diseño del formato de información requerida para el envío de información de notificaciones push en el cliente.	5	Alta
HUIT023	Validación del envío y recepción de una notificación push en un dispositivo.	8	Alta
	Total	23	-

2.3.4.1.2.3. Tiempo estimado

Tabla 21. Tiempo estimado para el prototipo 5.

Prototipo	Tiempo estimado	Desde	Hasta
5	5 días	16-05-2022	20-05-2022

2.3.4.1.3. Diseño

Puesto que las notificaciones provistas por Firebase vienen con un formato predeterminado para las notificaciones push, el aplicativo web nos permite personalizar nuestra notificación mediante el servicio Cloud Messaging el cual nos permite redactar notificaciones desde el servidor en caso de requerirlo dando al desarrollador la posibilidad de ingresar parámetros como “Título de la notificación”, “Texto de la notificación”, “Imagen de notificación (opcional)”, “Nombre de la notificación” al igual que nos permite ver una simulación de cómo se vería las notificaciones en los dos sistemas operativos móviles más utilizados como lo son Android y iOS, la *figura 38* y *figura 39* nos permitirán un prototipo de baja fidelidad de cómo se vería la notificación push en el cliente.



Figura 38. Visualización de la notificación push en su estado inicial.



Figura 39. Visualización de la notificación push en la vista expandida.

Esta herramienta nos permite también generar las notificaciones dependiendo la orientación o enfoque de estas la cual nos permite validar la aplicación y seleccionar un segmento de usuarios, al igual que se puede programar la notificación push mediante el parámetro de hora, el cual viene por defecto con el valor ahora esto genera una notificación automática al ser enviada.

Se puede configurar la notificación mediante eventos de conversación la misma que permitirá al desarrollador poder generar eventos dentro del aplicativo, como por ejemplo redireccionar a una pantalla específica al recibir la notificación, cuando está es pulsada por el usuario.

Configuraciones adicionales para la misma como asignarle un canal datos personalizados que pueden ser utilizados dentro de las pantallas o widgets de nuestro aplicativo al ser utilizado, se puede configurar si la notificación va a emitir un sonido al igual que un tiempo de duración ya que está puede permanecer sin ser activada dentro del teléfono en el segmento de notificaciones.

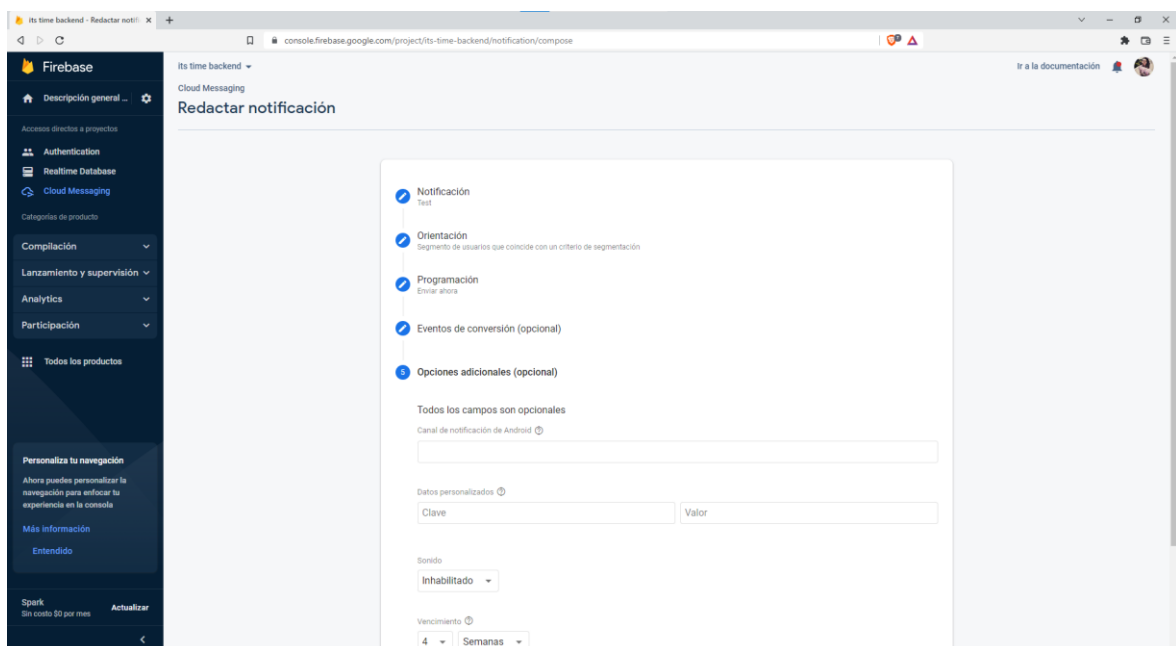


Figura 40. Servicio Cloud Messaging de Firebase.

El servicio cloud messaging también permite posterior a la configuración dentro de nuestro aplicativo, la posibilidad de utilizar un end point para poder utilizar servicios como Postman para el envío de notificaciones o la posibilidad de gestionar de forma adecuada en el backend consumiendo los datos necesarios para el envío de este.

2.3.4.1.4. Construcción

Para la construcción del prototipo 5 se prosigue a realizar la configuración del servicio de cloud messaging en el código fuente. Los pasos de configuración están detallados en la documentación provista por Firebase y Flutter al igual que una página que contiene la información compilada de ambas partes dando una documentación más entendible para los desarrolladores que están comenzando o desarrolladores expertos de ambas plataformas, esta documentación fue realizada para el uso del plugin “firebase_messaging”.

Plugins requeridos:

```
32 | firebase_auth: ^3.3.20
33 | firebase_core: ^1.18.0
```

Figura 41. Dependencias necesarias para el servicio Cloud Messaging.

Posterior a la instalación del plugin y basándonos en la documentación generamos los servicios de Google necesarios para la comunicación de Flutter con Firebase.

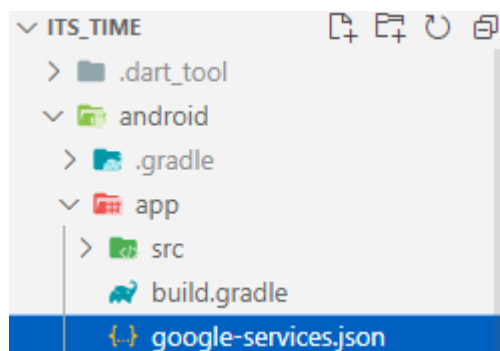


Figura 42. Ubicación del archivo de servicios de Google.

Y se modifican las dependencias de los archivos tipo build.gradle de nuestro proyecto de Android.

Una vez terminada las configuraciones de archivos, debemos tener en consideración la siguiente información al generar esta conexión con el servicio de notificaciones push de Firebase este servicio asigna un token a cada dispositivo, la duración del mismo es desde la instalación de la aplicación hasta su eliminación esto quiere decir que si posterior a hacer pruebas o algún problema de compilación, cuando se vuelve a instalar el aplicativo configurado dentro del cliente el token se genera nuevamente, esto es importante recalcar puesto que si el desarrollador omite esta información al generar un token y almacenarlo de forma indefinida en la base puede que las notificaciones no lleguen cuando sea apropiado.

Para solventar esta problemática es conveniente tratar el token dentro de un provider y configurarlo dentro de la información del usuario que inicie sesión ya que de esta manera se podrá identificar con facilidad a que dispositivo es al que le debe llegar la notificación push.

El servicio tiene la nos va a permitir obtener el token tiene el siguiente método.

```
27     static Future initializeApp() async {
28         await Firebase.initializeApp();
29         token = await FirebaseMessaging.instance.getToken();
30         FirebaseMessaging.onBackgroundMessage(_onBackgroundHandler);
31         FirebaseMessaging.onMessage.listen(_onMessageHandler);
32         FirebaseMessaging.onMessageOpenedApp.listen(_onMessageOpenHandler);
33     }
```

Figura 43. Función que genera el token por dispositivo.

Con el token de cada dispositivo generado y guardado dentro de nuestra base de datos podemos utilizarlo para generar las notificaciones push, para ello utilizaremos el aplicativo Postman el cual nos permitirá, mediante el envío de un JSON en un end point generado por el servicio de cloud messaging podremos enviar de forma satisfactoria la notificación desde el servidor y recibirla en el dispositivo del cliente.

Para poder enviar las notificaciones push desde Postman necesitamos estructurar el JSON con la información del dispositivo cliente, que en este caso será el token y la información de la notificación y la data necesaria que se requiera mostrar dentro de la pantalla al abrir la aplicación.


```

1  {
2    .... "to": "string",
3    .... "notification": {
4    .... | .... "title": "string",
5    .... | .... "body": "string"
6    .... },
7    .... "data": {
8    .... | .... "type": "string"
9    .... }
10 }

```

Figura 44. Estructura del JSON para la notificación push.

Para poder validar que funciona nuestro servicio generamos una petición mediante Postman y validamos que el dispositivo reciba la notificación push con la información del JSON.

```

1  {
2    .... "to": "c..._DY3OKSLWY4QL_U-nX90:APA91bEjWznlbJvaxzEhC7cj76Gy8UGm3wUcAI3RKSdWaLa28YrCechYmItq_uIP-9p7dvl13A9hLnEhsXot8myF43jVbRnS7Mbsio3Ik00HI4M0LQxCuM7AABS6IN_Tgx2LHDs..._2n",
3    .... "notification": {
4    .... | .... "title": "Alarma Hipertensión",
5    .... | .... "body": "Recuerda consumir tus medicamentos"
6    .... },
7    .... "data": {
8    .... | .... "type": "La pastilla que debes tomar es Enalapril."
9    .... }
10 }

```

Figura 45. Información que se envía desde Postman.

```

.... "notification": {
.... | .... "title": "Alarma Hipertensión",
.... | .... "body": "Recuerda consumir tus medicamentos"
.... },
.... "data": {
.... | .... "type": "La pastilla que debes tomar es Enalapril."
.... }

```

Figura 46. Información de la notificación y datos.

```

1  {
2    "multicast_id": 612541528998710399,
3    "success": 1,
4    "failure": 0,
5    "canonical_ids": 0,
6    "results": [
7    |   {
8    |   |   "message_id": "0:1662173769571228%62d7d9c762d7d9c7"
9    |   }
10  ]
11 }

```

Figura 47. Respuesta del end point.

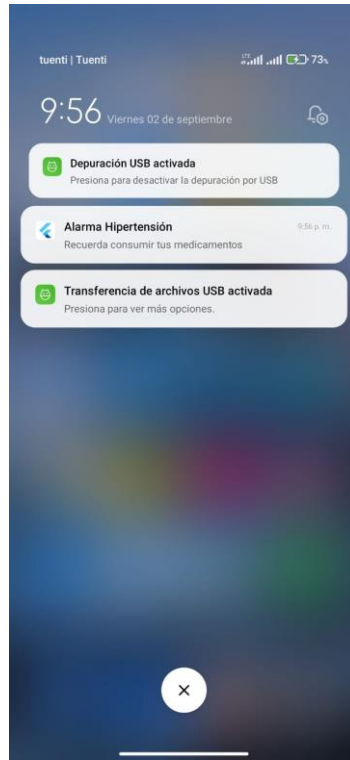


Figura 48. Visualización de la notificación en el dispositivo.

2.3.4.1.5. Despliegue

Para el despliegue del prototipo 5 no se modificaron las estimaciones en peso y prioridad.

Tabla 22. Modificación de análisis de historias de usuarios del prototipo 5.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT020	2	2	-	100%	Baja	Baja
HUIT021	8	8	-	100%	Alta	Alta
HUIT022	5	5	-	100%	Alta	Alta
HUIT023	8	8	-	100%	Alta	Alta
Total	23	23	0	100%	-	-

2.3.4.2. Prototipo 6

2.3.4.2.1. Comunicación

En la reunión posterior a la retroalimentación del prototipo 5 se llegó a la conclusión de asignar las siguientes historias de usuario al equipo de desarrollo, enfocándose en la implementación de la lógica para el envío masivo a los usuarios seguidores.

2.3.4.2.2. Plan rápido

2.3.4.2.2.1. Objetivos

- Diseñar las pantallas del aplicativo móvil asignadas para el prototipo 1.
- Construir el prototipo 1 del aplicativo en base a los mockups presentados con anterioridad al cliente.
- Evaluar la funcionalidad apta para los requerimientos del prototipo 1 según los criterios de aceptación definidos en las historias de usuario.

2.3.4.2.2.2. Historias de usuario

Tabla 23. Historias de usuario para el prototipo 6.

Código	Historia de usuario	Peso	Prioridad
HUIT024	Diseño del formato para el envío masivo de notificaciones push a usuarios seguidores.	5	Alta
HUIT025	Validación de funcionalidad.	5	Alta
	Total	10	-

2.3.4.2.2.3. Tiempo estimado

Tabla 24. Tiempo estimado para el prototipo 6.

Prototipo	Tiempo estimado	Desde	Hasta
2	2 días	10-06-2022	11-06-2022

2.3.4.2.3. Diseño

Basado en la información recopilada en el prototipo 5, se prosigue a diseñar el JSON que contendrá la lógica del envío masivo de notificaciones para los seguidores de un usuario en específico.

2.3.4.2.4. Construcción

Para el desarrollo del JSON necesario para poder generar las notificaciones masivas se procede a consultar dentro de la documentación y dentro de los parámetros se tiene el parámetro “registration_ids” la cual recibe una lista de strings la misma que nos permite ingresar los tokens de los usuarios requeridos, los mismos que esta almacenados en la base de datos.

Para ello se realizan las pruebas mediante el aplicativo Postman.

```
1  {
2    .... "registration_ids": [
3      .... "string",
4      .... "string" .....
5    .... ],
6    .... "notification": {
7      .... | ..... "title": "string",
8      .... | ..... "body": "string"
9      .... | ..... },
10   .... "data": {
11   .... | ..... "type": "string"
12   .... | ..... }
13 }
```

Figura 49. Estructura del JSON para notificaciones push masivas.

```
1  {
2    .... "registration_ids": [
3      .... "f0DbwtN0Q50QtF0VHGbkJp:APA91b6MKGxyzlzcH0qMER3TgqYi8Wmi-758yduLn7y6RGcTKvzmJVnyLFXG9_3Jxtu7qaAvn2eY0TREFbS28qEfgSdTn0wd_YhgN6M1PDbbeExfJE0bocJ0xoIfX0Feb1HH1cA40pb",
4      .... "c_-DY3OKSLWy4QL_U-nX98:APA91bEjWzn1bJVaxzEHC7cj766y8U6m3wwUCAI3RKSDwaLa28YrCechYmltq_uIP-9p7dv1i3A9hLnEHsXot8myf43jVbRwS7Mbsio3IK08HI4M6LQxUM7AABSbIn_Tgx21Hds_2n"
5    .... ],
6    .... "notification": {
7      .... | ..... "title": "Recuerda la ingesta a tu amigo",
8      .... | ..... "body": "El usuario Miguel Esteban Reina Gamboa pospuso su medicamento"
9      .... | ..... },
10   .... "data": {
11   .... | ..... "type": "Toma contacto con Miguel Esteban Reina Gamboa para que no se olvide de ingerir su medicación."
12   .... | ..... }
13 }
```

Figura 50. JSON con la información de prueba.

```
6    .... "notification": {
7    .... | ..... "title": "Recuerda la ingesta a tu amigo",
8    .... | ..... "body": "El usuario Miguel Esteban Reina Gamboa pospuso su medicamento"
9    .... | ..... },
10   .... "data": {
11   .... | ..... "type": "Toma contacto con Miguel Esteban Reina Gamboa para que no se olvide de ingerir su medicación."
12   .... | ..... }
```

Figura 51. Información de la notificación y datos de notificaciones masivas.

```

1  {
2    "multicast_id": 472668179303757361,
3    "success": 2,
4    "failure": 0,
5    "canonical_ids": 0,
6    "results": [
7      {
8        "message_id": "0:1662203513687315%62d7d9c762d7d9c7"
9      },
10     {
11       "message_id": "0:1662203513688618%62d7d9c762d7d9c7"
12     }
13   ]
14 }

```

Figura 52. Respuesta del end point.

Como podemos apreciar en la respuesta nosotros podemos en este apartado validar el envío satisfactorio de las notificaciones o el error de envío en caso de existir, esto mediante los parámetros “success” que nos permiten validar el envío correcto de las notificaciones de los dos usuarios ingresados en el body del end point, y cero errores en el parámetro “failure”.



Figura 53. Dispositivo emulado para las pruebas de envío.



Figura 54. Dispositivo físico con el aplicativo, para las pruebas de envío masivo.

2.3.4.2.5. Despliegue

Para el despliegue del prototipo 4 no se modificaron las estimaciones en peso y prioridad.

Tabla 25. Modificación de análisis de historias de usuarios del prototipo 6.

Código	Puntos estimados	Puntos ejecutados	Puntos modificados	Porcentaje ejecutado	Prioridad estimada	Prioridad modificada
HUIT024	5	5	-	100%	Alta	Alta
HUIT021	5	5	-	100%	Alta	Alta
Total	10	10	0	100%	-	-

3. RESULTADOS

3.1. Pruebas de funcionalidad

Para la realización de pruebas de funcionalidad, se procederá con la asignación de diferentes tareas al extracto de personas que permitirán evaluar la funcionalidad del aplicativo, tomando en consideración las historias de usuario de los prototipos realizados. La finalidad de estas pruebas es validar las especificaciones detalladas en el proyecto.

El aplicativo fue evaluado por un grupo de personas diagnosticadas con enfermedades como diabetes, hipertensión o ambas, los cuales realizarán los casos de uso estipulados en el presente proyecto.

Tabla 26. Caso de prueba CPIT001

Identificador	CPIT001
Funcionalidad	Registro de un nuevo usuario
Procedimiento	<ol style="list-style-type: none">1. Abrir la aplicación en el dispositivo.2. Dar clic en la opción “Crear nueva cuenta”.3. Ingresar los datos requeridos en el formulario.4. Dar clic en el botón “Crear”.5. Validar la creación del usuario, en la ventana emergente con el botón de “Crear”.6. Llenar el formulario de “Información del usuario”.7. Dar clic en el botón “Guardar”.8. Visualizar la pantalla de inicio.
Resultado esperado	Visualización de la pantalla de inicio

Tabla 27. Tabla de resultados del caso de prueba CPIT001

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se abrió el aplicativo en el dispositivo.	100%	-	-
Se logró navegar a la pantalla del formulario para la creación del usuario.	100%	-	-

Se visualizo la ventana emergente para la validación de la creación de un nuevo usuario.	100%	-	-
Se visualizo el formulario de información del usuario.	100%	-	-
Se visualizo la pantalla de inicio.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT001.

Tabla 28. Caso de prueba CPIT002

Identificador	CPIT002
Funcionalidad	Cierre de sesión e Inicio de sesión del usuario creado.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla de inicio. 2. Dar clic en el icono de cierre de sesión ubicado en la parte superior derecha. 3. Visualizar la pantalla de inicio de sesión. 4. Ingresar los datos del usuario creado con anterioridad. 5. Dar clic en el botón “Iniciar Sesión”. 6. Visualizar la pantalla de inicio.
Resultado esperado	Visualización de la pantalla de inicio

Tabla 29. Tabla de resultados del caso de prueba CPIT002

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla de inicio.	100%	-	-
Se logró cerrar sesión y visualizar la pantalla de inicio de sesión.	100%	-	-
Se inicio sesión con los datos del usuario creado con anterioridad.	100%	-	-
Se visualizo la pantalla de inicio.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT002.

Tabla 30. Caso de prueba CPIT003

Identificador	CPIT003
Funcionalidad	Creación de una nueva alarma.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla de inicio. 2. Dar clic en el botón “Crear alarma”. 3. Visualizar el formulario de creación de la alarma. 4. Llenar el formulario con la información requerida. 5. Guardar el formulario. 6. Visualizar el listado de alarmas creadas.
Resultado esperado	Visualización de la pantalla del listado de alarmas.

Tabla 31. Tabla de resultados del caso de prueba CPIT003

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla de inicio.	100%	-	-
Se logró navegar al formulario de creación de una nueva alarma.	100%	-	-
Se logró crear la nueva alarma	100%	-	-
Se visualizo la pantalla con el listado de alarmas.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT003.

Tabla 32. Caso de prueba CPIT004

Identificador	CPIT004
Funcionalidad	Visualización de las alarmas creadas.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla de inicio. 2. Dar clic en el botón “Listado de alarmas”. 3. Visualizar las alarmas creadas del usuario registrado.
Resultado esperado	Visualización de la pantalla del listado de alarmas.

Tabla 33. Tabla de resultados del caso de prueba CPIT004

Resultado esperado	Resultado obtenido (%)
---------------------------	-------------------------------

	Si	No	Parcialmente
Se visualizo la pantalla de inicio.	100%	-	-
Se logró navegar pantalla del listado de alarmas.	100%	-	-
Se visualizo la pantalla con el listado de alarmas.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT004.

Tabla 34. Caso de prueba CPIT005

Identificador	CPIT005
Funcionalidad	Actualización de una alarma creada.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla con el listado de alarmas. 2. Seleccionar la alarma a ser modificada, mediante el ícono de actualizar de color negro. 3. Visualizar el formulario de actualización de alarma. 4. Validar que la información de la alarma a modificar sea correcta. 5. Modificar los datos de la alarma, ya sea uno o todos los datos del formulario. 6. Dar clic en el botón “Guardar”. 7. Visualizar las modificaciones de la alarma en la pantalla del listado de alarmas.
Resultado esperado	Visualización de la pantalla del listado de alarmas.

Tabla 35. Tabla de resultados del caso de prueba CPIT005

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla del listado de alarmas.	100%	-	-
Se logró seleccionar una alarma.	100%	-	-
Se visualizo la información correcta de la alarma seleccionada.	100%	-	-
Se logro modificar los campos del formulario.	100%	-	-

Se logro guardar los cambios realizados de la alarma seleccionada.	100%	-	-
Se logro visualizar el listado de alarmas con los cambios realizados en la alarma seleccionada.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT005.

Tabla 36. Caso de prueba CPIT006

Identificador	CPIT006
Funcionalidad	Eliminación de una alarma creada.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla del listado de alarmas. 2. Eliminamos la alarma seleccionada mediante, el uso del botón de borrar definido por el icono de un basurero y de color rojo. 3. Visualizamos que la alarma desaparece del listado de alarmas.
Resultado esperado	Visualización de la pantalla del listado de alarmas.

Tabla 37. Tabla de resultados del caso de prueba CPIT006

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla del listado de alarmas.	100%	-	-
Se logró eliminar una alarma.	100%	-	-
Se visualizo la ausencia de la alarma seleccionada.	100%	-	-
Se logro visualizar el listado de alarmas.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT006.

Tabla 38. Caso de prueba CPIT007

Identificador	CPIT007
----------------------	---------

Funcionalidad	Agregar un seguir a la lista de seguidores.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar la pantalla de inicio. 2. Dar clic en el botón “Seguidores”. 3. Navegará a la pantalla de seguidores la cual estará vacía. 4. Dar clic en el botón agregar situado en la parte inferior derecha simbolizado con el icono de más. 5. Navegara a la pantalla de usuarios. 6. Se listarán todos los usuarios existentes en la base de datos. 7. Seleccionará el nuevo seguidor dando clic en el icono de más ubicado en la tarjeta del usuario seleccionado. 8. Visualizar el listado de seguidores.
Resultado esperado	Visualización de la pantalla del listado de seguidores.

Tabla 39. Tabla de resultados del caso de prueba CPIT007

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla de inicio.	100%	-	-
Se logró navegar a la pantalla de seguidores.	100%	-	-
Se logró navegar a la pantalla de usuarios del sistema.	100%	-	-
Se logró agregar un nuevo seguidor.	100%	-	-
Se logró visualizar el nuevo seguidor en el listado de seguidores.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT007.

Tabla 40. Caso de prueba CPIT008

Identificador	CPIT008
Funcionalidad	Eliminación de un seguidor registrado.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar el listado de seguidores. 2. Seleccionar el usuario a eliminar.

	3. Visualizar la ausencia del usuario en el listado de seguidores.
Resultado esperado	Visualización de la pantalla del listado de seguidores.

Tabla 41. Tabla de resultados del caso de prueba CPIT008

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se visualizo la pantalla del listado de seguidores.	100%	-	-
Se logró eliminar un seguidor.	100%	-	-
Se visualizo la ausencia del seguidor seleccionado.	100%	-	-
Se logro visualizar el listado de seguidores actualizado.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT008.

Tabla 42. Caso de prueba CPIT009

Identificador	CPIT009
Funcionalidad	Visualización de notificación push.
Procedimiento	<ol style="list-style-type: none"> 1. Tener encendido el dispositivo, estando fuera de la aplicación. 2. Visualizar la notificación push emitida por el servidor. 3. Interactuar con la notificación. 4. Visualizar las opciones de la pantalla de la alarma. 5. Dar clic en el botón “Aceptar” o “Posponer”. 6. Visualizar la ventana emergente. 7. Dar clic en el botón “Aceptar”. 8. Visualizar la pantalla de inicio de la aplicación.
Resultado esperado	Visualización de la pantalla de inicio.

Tabla 43. Tabla de resultados del caso de prueba CPIT009

Resultado esperado	Resultado obtenido (%)		
	Si	No	Parcialmente
Se tuvo el dispositivo encendido, no importa si está habilitado o bloqueado.	100%	-	-
Se visualizó la notificación.	100%	-	-
Se interactuó con la notificación.	100%	-	-
Se visualizó la pantalla de la alarma.	100%	-	-
Se visualizó la pantalla emergente.	100%	-	-
Se visualizo la pantalla de inicio.	100%	-	-

Los encuestados que realizaron el caso de prueba con éxito fueron un de 10 de 10 encuestados dando como resultado un 100% de funcionalidad y usabilidad del caso de prueba CPIT009.

3.2. Pruebas de compatibilidad

Al ser una aplicación móvil, usar alguna herramienta para obtener un reporte de las versiones Android compatibles con la app.

3.3. Pruebas de usabilidad

Para la realización de pruebas de usabilidad del aplicativo móvil se utilizó el cuestionario SUS (System Usability Scale o Sistema de Escalas de Usabilidad) la cual permite evaluar la aplicación mediante un formulario que contiene un total de 10 preguntas, teniendo 5 posibles respuestas las cuales evalúan tomando a 1 como “Totalmente en desacuerdo” y a 5 como “Totalmente de acuerdo”.

Las preguntas que utiliza el cuestionario SUS son las siguientes. [15]

- 1) Creó que usaría este sistema frecuentemente.
- 2) Encuentro este sistema innecesariamente complejo.
- 3) Creo que el sistema fue fácil de usar.

- 4) Creó que necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema.
- 5) Las funciones de este sistema están bien integradas.
- 6) Creó que el sistema es muy inconsistente.
- 7) Imagino que la mayoría de la gente aprendería a usar este sistema en forma muy rápida.
- 8) Encuentro que el sistema es muy difícil de usar.
- 9) Me siento confiado al usar este sistema.
- 10) Necesité aprender muchas cosas antes de ser capaz de usar este sistema.

Posterior a la realización de los casos de prueba a cada evaluador se le envió el cuestionario SUS para poder recopilar la información de usabilidad, de esta manera se obtuvo la siguiente información.

Resultados de la encuesta.

Como podemos observar mediante la muestra del cien por ciento de usuarios encuestados en la pregunta número uno de la encuesta SUS afirman que si usarían la aplicación de manera frecuente para tener el control de la ingesta de sus medicamentos, ya que de está forma ellos podrían generar un nuevo hábito permitiéndose mejorar la ingesta de su medicación buscando controlar la enfermedad y en el mejor de los casos hacer que está enfermedad desaparezca, la mayor parte de los encuestados dieron como favorable la iniciativa del aplicativo ya que puede ser un aplicativo excelente en sus rutinas.

¿Usaría el aplicativo de manera frecuente?

10 respuestas

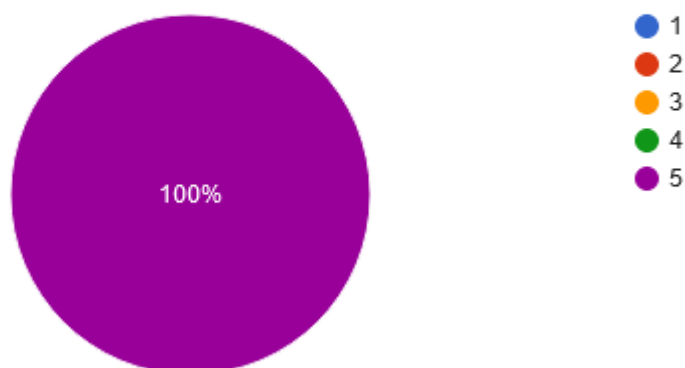


Figura 55. Pregunta 1. Encuesta SUS

El resultado de la pregunta número dos permite validar que los usuarios que interactuaron con el sistema no tuvieron dificultad en utilizarlo dando de esta manera una referencia a que las prácticas implementadas en el diseño de interfaces dieron un resultado optimo, ya que las buenas prácticas de desarrollo fueron indispensables en el diseño de las interfaces, el uso de la práctica “menos es más”, permitió no saturar la aplicación de botones, imágenes o textos sino solamente contar con los elementos indispensables ya que uno de los motivos principales era tener la consideración de que el mayor porcentaje de usuarios finales son personas que no tienen mucho contacto con la tecnología o un contacto casi escaso ya que en la actualidad la mayor parte de personas se han visto obligadas a utilizar medios tecnológicos de manera forzosa, pero esto no quiere decir que todos se hayan acoplado a su uso constante por lo que al tener un aplicativo conciso y preciso en su enfoque dio como resultado una aprobación por parte de los usuarios finales y encuestados.

¿Encontró el sistema innecesariamente complejo?

10 respuestas

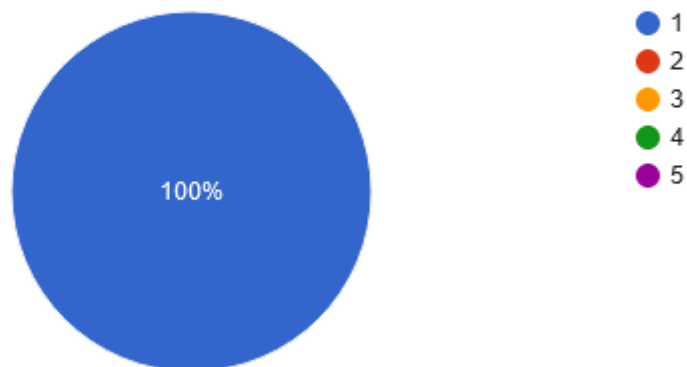


Figura 56. Pregunta 2. Encuesta SUS

En la encuesta podemos visualizar que de los diez encuestados uno de ellos asignó una calificación de cuatro el motivo de este fue que mientras el usuario estaba realizando las pruebas de usabilidad del sistema pudo visualizar e interactuar de formas con las que usuarios podrían tener posibles errores. Uno de estos era el asumir que ya tenían un usuario creado e ingresar directamente correos creados con anterioridad por los usuarios, pero de

diferentes plataformas asumiendo que funcionaban. Este más que un fallo dentro del desarrollo es un fallo de desconocimiento de las diferentes tecnologías algunos encuestados asumían que al utilizar el aplicativo ellos podían ingresar con correos o usuarios creados con anterioridad en las redes sociales comunes o servicios de correos electrónicos comúnmente usados por la mayor parte de cibernautas, ya que nuestro proyecto no tiene un dominio especificado el mismo puede ingresar con un correo cualquiera no obstante se explico a los encuestados que para poder tener acceso a las características del aplicativo era necesario crear un usuario dentro del aplicativo ya que no se cuenta con una base de datos alimentada con información sino que la misma debe ser generada por aquellos usuarios interesados en el uso de las diferentes funcionalidades del aplicativo.

¿Creé que el sistema fue fácil de usar?

10 respuestas

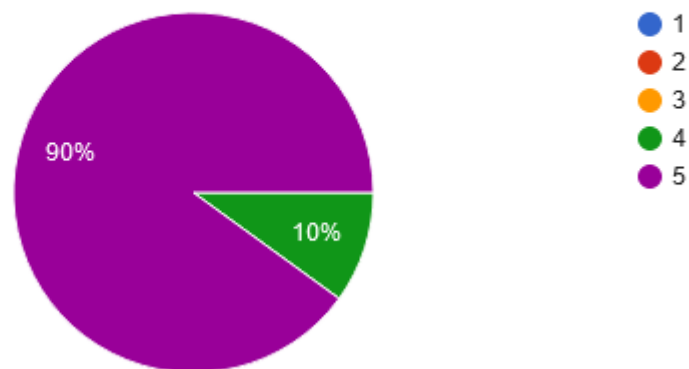


Figura 57. Pregunta 3. Encuesta SUS

Está pregunta dio como resultado que un 20% de la muestra necesite de cierta forma la guía de una persona con conocimientos previos ya que como se explicó en la pregunta tres se podía visualizar caminos en los que los usuarios encuentren fallas dando como resultado un mal uso del aplicativo.

¿Creé que necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema?

10 respuestas

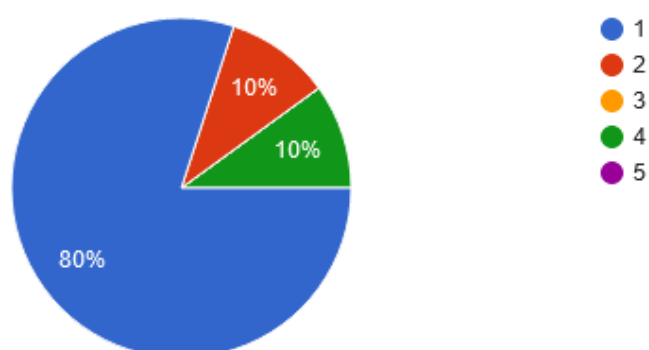


Figura 58. Pregunta 4. Encuesta SUS

Este resultado muestra una inconformidad mínima de parte de los encuestados ya que dentro de las pruebas no se obtuvo resultados como el desconocimiento del significado de iconos como el de actualizar o eliminar, que aunque sean intuitivos no son conocidos por todos los usuarios encuestados.

¿Las funciones de este sistema están bien implementadas?

10 respuestas

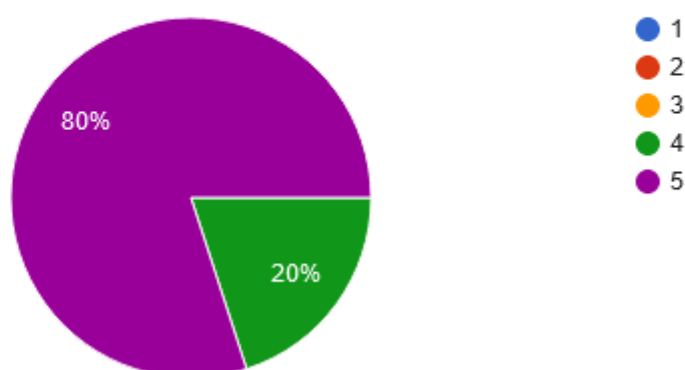


Figura 59. Pregunta 5. Encuesta SUS

La perspectiva de la mayor parte de usuarios es que la consistencia del aplicativo es óptimo ya que realiza lo requerido y lo mencionado en las pruebas de uso, el cuál es poder gestionar sus alarmas usadas para la ingesta.

¿Creé que el sistema es muy inconsistente?

10 respuestas

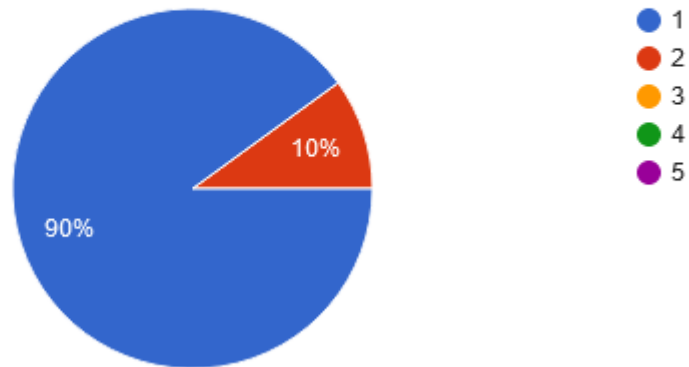


Figura 60. Pregunta 6. Encuesta SUS

A pesar de los resultados anteriores el aplicativo fue sencillo de comprender, usar e interactuar para todos los encuestados, así como se ve en la imagen siguiente.

¿Imagina que la mayoría de la gente aprendería a usar este sistema en forma muy rápida?

10 respuestas

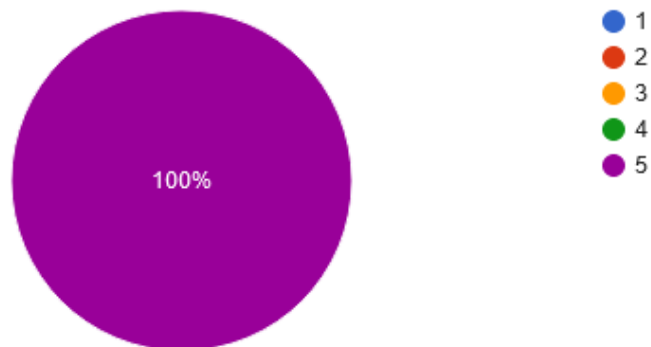


Figura 61. Pregunta 7. Encuesta SUS

Los encuestados concuerdan con la practicidad y la facilidad de uso del aplicativo en sus funciones.

¿Encuentra que el sistema es muy difícil de usar?

10 respuestas

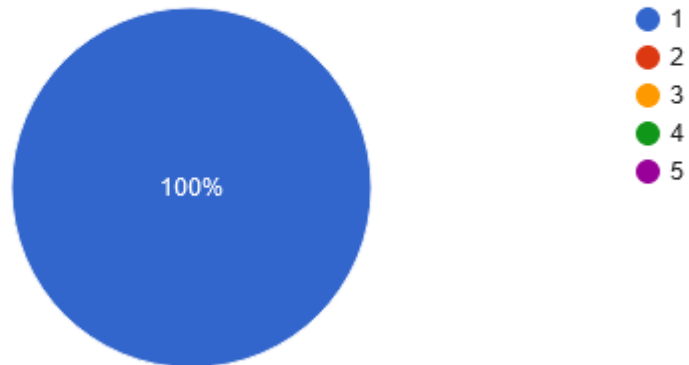


Figura 62. Pregunta 8. Encuesta SUS

Los encuestados se sienten seguros de utilizar el aplicativo logrando tener la mejor interacción con el sistema generando los diferentes tipos de uso del aplicativo.

¿Se siente confiado al usar este sistema?

10 respuestas

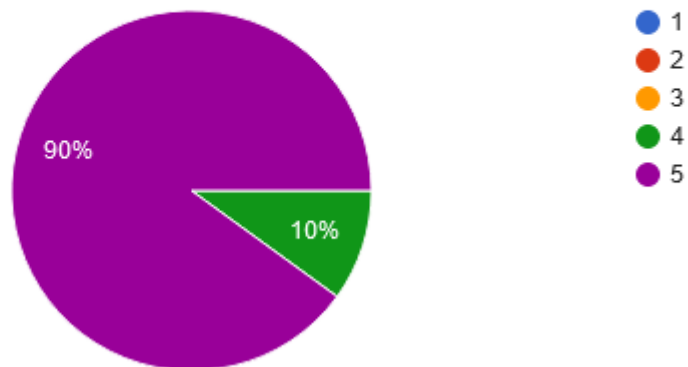


Figura 63. Pregunta 9. Encuesta SUS

El resultado de la última pregunta es satisfactorio ya que el 90% de los encuestados sintió ser capaz de utilizar el aplicativo con el conocimiento actual sobre el uso de la tecnología. Esto nos da como resultado el poder apreciar que aunque la mayor parte de posibles usuarios pueden ser personas que no utilicen en su diario vivir la tecnología o aplicaciones de diferentes tipos de índole ellos pudieron familiarizarse con todo el proceso de uso de la aplicación Its time.

¿Necesité aprender muchas cosas antes de ser capaz de usar este sistema?

10 respuestas

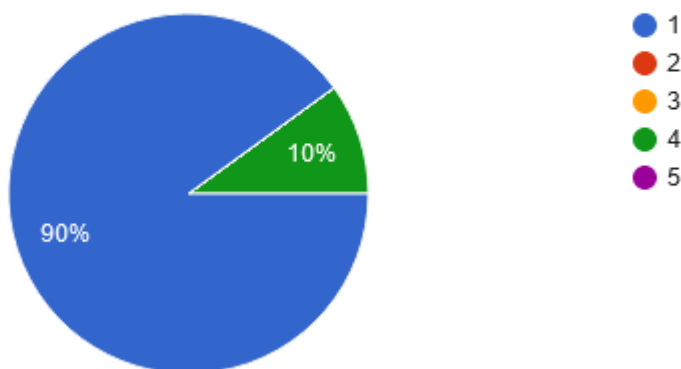


Figura 64. Pregunta 10. Encuesta SUS

El resultado de esta encuesta es valorado sobre 100 para ello se utiliza los siguientes pasos:

- 1) Una vez obtenidos los resultados por evaluador, generamos un promedio por pregunta.
- 2) El resultado de las preguntas impares, $x-1$ donde x es el promedio de las preguntas.
- 3) El resultado de las preguntas pares, $5-x$ donde x es el promedio de las preguntas.
- 4) Sumamos todos los resultados obtenidos.
- 5) El resultado de la sumatoria lo multiplicamos por 2,5.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
E1	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
E2	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	4,00
E3	5,00	1,00	5,00	1,00	4,00	2,00	5,00	1,00	4,00	1,00
E4	5,00	1,00	5,00	4,00	5,00	1,00	5,00	1,00	5,00	1,00
E5	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
E6	5,00	1,00	4,00	2,00	4,00	1,00	5,00	1,00	5,00	1,00
E7	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
E8	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
E9	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
E10	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00	5,00	1,00
PROMEDIO	5,00	1,00	4,90	1,40	4,80	1,10	5,00	1,00	4,90	1,30
PROMEDIO SUS	4,00	4,00	3,90	3,60	3,80	3,90	4,00	4,00	3,90	3,70

Tabla 44. Tabla de resultados del cuestionario SUS

Simbología:

E1 = Evaluador 1;

P1 = Pregunta 1;

Dando como resultado un total de 97 sobre 100.

El cuestionario SUS sugiere que un aplicativo con una muy buena usabilidad es aquel que como resultado es mayor a 68 por lo que se concluye que la aplicación hasta el prototipo actual es un aplicativo con una muy buena usabilidad, también dando como resultado una muy factible adaptabilidad a la hora de ser usada por los clientes finales que en nuestro caso serían los evaluadores. Este resultado también da como resultado no solamente que el aplicativo es eficiente y fácil de usar también nos da como muestra que las personas con esta clase de enfermedades sienten la necesidad de utilizar un elemento nuevo dentro de su rutina para mejorar los hábitos con los que cuentan actualmente ya que como se detalla en la parte inicial del proyecto las personas tendemos a olvidarnos algunos eventos o actividades requeridas en nuestro día a día y el aplicativo al ser aceptado por esta muestra de encuestados nos permite validar que aplicaciones como la que se desarrollo dentro de este proyecto pueden tener un impacto favorable en los pacientes. Al igual que con el tiempo este tiene como propósito mejorar los hábitos para poder tratar de prescindir del aplicativo y poder tener el control de su ingesta de forma rutinaria.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- El objetivo general del presente proyecto se logró con satisfacción ya que el aplicativo móvil logra gestionar alarmas, creadas por un usuario al igual que recibir notificaciones para hacerle acuerdo del medicamento, dando como resultado un mejor control de su ingesta, el cuál satisface la necesidad de la creación del aplicativo.
- Se reviso la información necesaria para entender, ¿qué es un benchmarking? y ¿cómo implementarlo para recabar los requerimientos del sistema? Comprendiendo los diferentes tipos de análisis que se pueden realizar para poder generar una necesidad y llegar a ser competitivos con los diferentes aplicativos dentro del mercado. Dándonos como resultado que sin tener en consideración el control de usuarios y alarmas, el poder dar al usuario la facilidad de comunicación con su red de seguidores nos daba un plus ante el resto de las aplicaciones.
- Se diseñaron interfaces cómodas y amigables mediante el uso de buenas prácticas a tomarse en consideración en el diseño de interfaces como cumplir con las necesidades del aplicativo, como buscar la armonía entre los diferentes componentes dentro de las interfaces al igual que el uso de componentes realmente necesarios al momento de diseñar las interfaces evitando saturar de componentes o información irrelevante para el usuario final.
- Se diseño interfaces cómodas y amigables teniendo en consideración aspectos como la importancia de los componentes y los diferentes tipos de paletas de colores para implementarlos como colores corporativos. Permitiendo obtener un aplicativo diseñado con dos colores básicos pero que permiten flexibilidad para su abstracción ya que mediante su uso se llegó a un aplicativo de fácil uso, comprensión y adaptabilidad de los usuarios finales.
- En la “iteración 0” una problemática a ser solucionada era la forma de implementación de desarrollo. ¿Cuál era el camino más conveniente? Dicha pregunta fue solventada en la reunión realizada en el “prototipo 0” donde se estableció la historia de usuario “HUIT002”, en la que el equipo opto por desarrollar el aplicativo mediante el uso de la arquitectura Cliente-Servidor. Como conclusión podemos

determinar que el uso de la metodología Iterativa-Incremental permite mediante las reuniones que cada prototipo se pueda solventar diferentes problemáticas encontradas en el proceso de desarrollo.

- Dentro de la “iteración 0” encontramos un problema. ¿Cómo manipular la información contenida dentro del código? Al igual que en el anterior punto la reunión fue clave para la solución de dicha cuestión ya que pudimos determinar varios caminos para solucionarlo, como por ejemplo mediante el uso de patrón Bloc el cual parte después de tener una excelente concepción de la lógica de negocio, algo que para dicha reunión no estaba bien definido. Por lo que en la reunión se llegó a la conclusión de que el uso del patrón Provider era el óptimo para su uso ya que no dependía de la lógica de negocio sino de saber cómo abstraer la información concreta que va a ser utilizada en diferentes secciones del aplicativo.
- Se definió la arquitectura necesaria para el desarrollo de la aplicación, dando como resultado una arquitectura con un grado alto de comprensión y aplicabilidad para proyectos con el mismo tipo de índole o propósito.
- Se definieron los diferentes modelos para la obtención de información dando como resultado una forma adecuada de abstraer la información contenida en nuestra base de datos.
- Se implementaron prototipos de baja fidelidad o mockups con el propósito de tener aterrizada la idea principal y la meta a llegar en cada uno de los prototipos o validar lo que en realidad satisfacía los diferentes requerimientos del proyecto.
- Se implementaron los prototipos de alta fidelidad de forma satisfactoria teniendo en consideración los prototipos, los requerimientos, las buenas prácticas dentro del diseño de interfaces y también procurando que el usuario final se sienta a gusto y confiado al utilizar el aplicativo.
- Flutter es un SDK que permite a los equipos de desarrollo tener una gama de ventajas al momento de desarrollar aplicaciones ya que su curva de aprendizaje es muy alta permitiéndonos generar aplicativos intuitivos y sencillos de desarrollar y mantener, como nuestro aplicativo It's time.
- La mayor parte de personas evaluadas que tienen diabetes o hipertensión, o a la vez que tienen familiares con estas enfermedades emitieron comentarios positivos con respecto al desarrollo de esta clase de aplicaciones ya que es un problema muy común

en la mayor parte de pacientes, más aún cuando estos son diagnosticados por primera vez los primeros meses suelen tener bastante irregularidad en la ingesta.

- Se realizaron las pruebas de usabilidad para poder tener un análisis métrico con respecto a si el aplicativo cumplía con los requerimientos establecidos.
- Mediante el uso de las pruebas de usabilidad y del cuestionario SUS se pudo también abstraer la satisfacción del usuario con respecto al uso del sistema, en diferentes ámbitos como, por ejemplo: si el usuario considerase utilizar el aplicativo como aplicación recurrente para el uso diario en el control de su ingesta, el cual fue satisfactorio ya que el cien por ciento de los encuestados dijo que si lo usaría.
- Los resultados del cuestionario SUS sugieren que el aplicativo tiene una usabilidad del 97%, esto quiere decir que los usuarios no tuvieron problemas con el uso de esta que se hizo sencillo su interactividad con los componentes de este.
- En la “iteración 0” en el “prototipo 1” el inconveniente con mayor relevancia fue la selección del servicio adecuado de Firebase para el almacenamiento de información ya que contamos con dos servicios muy robustos “Firestore Database” y “Realtime Database” ambos servicios usan documentos ya que son bases de datos no SQL. No obstante, la mejor opción para ser implementado en el proyecto fue el servicio “Realtime Database” ya que este nos permite testear mediante una API sencilla, el cual nos permitió tener un control mediante la herramienta Postman y de mayor accesibilidad dentro del uso en el código ya que solamente se realizaban los controladores con los URL’s necesarios.
- Para solventar la realización del aplicativo y la historia de usuario “HUIT020” se investigó la información necesaria que nos permita emitir notificaciones clientes finales y a sus usuarios seguidores solventando al igual la historia de usuario “HUIT024” para que tomen contacto con el usuario. Dando como culminación esa difusión del mensaje requerido para los involucrados con nuestro cliente final o paciente.

4.2. Recomendaciones

- Se recomienda dar una pequeña inducción del uso del aplicativo a los usuarios cuando estos pasan de cierta edad o no tienen tanto contacto o uso con la tecnología, ya que, aunque es muy intuitiva y de fácil uso debemos recordar que no todas las personas tienen los conocimientos necesarios para el uso de esta clase de tecnologías.
- Es recomendable indagar muy bien sobre las diferentes formas de benchmarking con el propósito de validar que características son indispensables en el desarrollo de un aplicativo y que características carecen la competencia para poder tomar esta como fortaleza para poder despuntar en el mercado, al igual que tener en consideración diferentes tipos de características para ser implementadas a futuro dentro de nuestro aplicativo para que este tenga mayor impacto y acogida por parte de los usuarios finales.
- Es necesario que los usuarios del aplicativo tengan una conexión estable con internet, ya que de esta forma se puede recibir las notificaciones.
- Se recomienda realizar un backend que permita gestionar algunas funcionalidades de forma automatizada.
- Es recomendable orientar a los usuarios o clientes finales con el aplicativo es decir dar como primera instancia un tutorial guiado ya sea este por alguien que conozca del aplicativo o mediante un desarrollo futuro dando como resultado una inducción para los nuevos usuarios.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. d. s. pública, «Ministerio de salud pública,» 22 10 2020. [En línea]. Available: <https://www.salud.gob.ec/msp-promueve-acciones-para-prevenir-enfermedades-cronicas-en-la-poblacion/>.
- [2] M. d. S. P. d. Ecuador, Guía de Práctica Clínica - Hipertensión arterial, Quito: Dirección Nacional de Normalización-MSP, 2019.
- [3] M. d. S. Pública, Guía de Práctica Clínica - Diabetes mellitus tipo 2, Quito: Dirección Nacional de Normalización MSP, 2017.
- [4] B. L. Gutierrez Bautista, «Notificaciones push,» de *Sistema móvil para difusión de notificaciones push y asistencia de servicios “caso: dtic-u.m.s.a. proyecto hageo”*, La Paz - Bolivia, 2015, pp. 30 - 31.
- [5] G. Author, «Rockcontent blog,» 25 Agosto 2017. [En línea]. Available: <https://rockcontent.com/es/blog/que-es-benchmarking/>. [Último acceso: 1 Septiembre 2022].
- [6] «Tableau,» Tableau Software, [En línea]. Available: <https://www.tableau.com/es-mx/learn/articles/benchmarking-types-and-examples>. [Último acceso: 4 Septiembre 2022].
- [7] Google, «Material Design,» Material Design, [En línea]. Available: <https://material.io/design/color/the-color-system.html#color-theme-creation>. [Último acceso: 10 Septiembre 2022].
- [8] F. G. d. Zuñiga, «Arsys,» Arsys, 08 Septiembre 2015. [En línea]. Available: <https://www.arsys.es/blog/buenas-practicas-interfaces-usuario>. [Último acceso: 10 Octubre 2022].
- [9] Google, «Flutter,» Google, [En línea]. Available: <https://docs.flutter.dev/>. [Último acceso: 17 01 2022].
- [10] Google, «Dart,» Google, [En línea]. Available: <https://dart.dev/>. [Último acceso: 17 01 2022].
- [11] Google, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs>. [Último acceso: 17 01 2022].
- [12] Figma, «Figma,» Figma, [En línea]. Available: <https://www.figma.com/developers/api>. [Último acceso: 17 01 2022].
- [13] Git, «Git,» [En línea]. Available: <https://git-scm.com/docs/git>. [Último acceso: 2022 02 05].
- [14] Github, «Github,» [En línea]. Available: <https://docs.github.com/es/get-started/learning-about-github/githubs-products>. [Último acceso: 2022 02 05].
- [15] Android, «Developers Android,» [En línea]. Available: <https://developer.android.com/docs>. [Último acceso: 2022 02 05].

- [16 V. S. Code, «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/Docs>.
] [Último acceso: 2022 02 05].
- [17 I. Postman, «Postman,» [En línea]. Available: <https://www.postman.com/company/about-postman/>. [Último acceso: 1 septiembre 2022].
- [18 «proyectos ágiles,» [En línea]. Available: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>. [Último acceso: 10 12 2022].
- [19 G. Author, «Rockcontent,» 25 Agosto 2017. [En línea]. Available:
] <https://rockcontent.com/es/blog/que-es-benchmarking/>. [Último acceso: 10 Diciembre 2012].
- [20 O. Blancarte, «Reactive Programming,» Reactive Programming, [En línea]. Available:
] <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>. [Último acceso: 21 06 2022].
- [21 M. Design, «Material,» Google, [En línea]. Available: <https://material.io/design/color/the-color-system.html#tools-for-picking-colors>. [Último acceso: 12 Octubre 2022].
- [22 usability.gov, «usability.gov,» [En línea]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Último acceso: 03 09 2022].
- [23 J. Sanchez, «XurxoDev,» XurxoDev, 26 03 2020. [En línea]. Available:
] <http://xurxodev.com/introduccion-al-patron-bloc/#:~:text=Un%20BloC%20es%20un%20componente,y%20arquitectura%20de%20tu%20aplicaci%C3%B3n..> [Último acceso: 21 06 2022].

6. ANEXOS

6.1. Anexo 1

Los prototipos de baja fidelidad se encuentran almacenados en el siguiente link.

<https://www.figma.com/file/XUwq3bQShBDLH0tKTycZLi/Tesis?node-id=0%3A1>

6.2. Anexo 2

El código fuente del aplicativo móvil desarrollado en Flutter se encuentra almacenado en el siguiente link.

https://github.com/MiguelReina/its_time.git

6.3. Anexo 3

Las capturas de pantalla de la aplicación están almacenadas en el siguiente link.

<https://drive.google.com/drive/folders/16NFUudSMLNZMGYbaOgfvqXwdN8esUjUd?usp=sharing>

6.4. Anexo 4

Casos de prueba, los casos de prueba están almacenados en el siguiente link.

https://epnecuador-my.sharepoint.com/:f:/g/personal/miguel_reina_epn_edu_ec/Ev4zJulKrGVJho6-czAxSvEBldKSQsmXL0iq_u8wrPklwQ?e=bneQoD

6.5. Anexo 5

Formulario para realizar la encuesta SUS.

<https://forms.gle/yMWjwGJdqW4dftYj6>

6.6. Anexo 6

Los resultados de la encuesta SUS están almacenados en el siguiente link.

<https://drive.google.com/drive/folders/1ZXJta6PXDCyVJ8VT-KMUNqQRYKqBZcvJ?usp=sharing>

6.7. Anexo 7

El APK del aplicativo se encuentra almacenado en el siguiente link.

https://drive.google.com/drive/folders/1qeZqa_nGqy7wzsvrLqlhH_15v550WsK8?usp=sharing

6.8. Anexo 8

Las peticiones realizadas en Postman están almacenadas en el archivo del siguiente link.

<https://drive.google.com/drive/folders/1mtDFyFC7dG016Xi8CRtuYYXXWijQAra7?usp=sharing>

6.9. Anexo 9

Los resultados de la encuesta SUS se encuentran almacenados en el siguiente link.

https://epnecuador-my.sharepoint.com/:f:/g/personal/miguel_reina_epn_edu_ec/Ev7eXCfY4oNKIDaOuJrc4JcBqGnOzlpFC07T_avVNrLCIQ?e=EkWMwd