



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

**APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL Y  
REGISTRO DE TRATAMIENTOS MÉDICOS.**

---

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a  
la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Irving Mauricio Taipe Sailema

**TUTOR:** Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

**Ambato - Ecuador**

**septiembre – 2021**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL Y REGISTRO DE TRATAMIENTOS MÉDICOS, desarrollado bajo la modalidad de Proyecto de Investigación por el señor Irving Mauricio Taipe Sailema, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2021.

-----  
Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

TUTOR

## AUTORÍA

El presente Proyecto de Investigación titulado: APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL Y REGISTRO DE TRATAMIENTOS MÉDICOS es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2021.



-----  
Irving Mauricio Taipe Sailema

C.C. 180430746-8

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Irving Mauricio Taipe Sailema, estudiante de la Carrera de Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad de Proyecto de Investigación, titulado APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL Y REGISTRO DE TRATAMIENTOS MÉDICOS nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, septiembre 2021.

-----  
Ing. Pilar Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Hernando Buenaño, Mg.  
PROFESOR CALIFICADOR

-----  
Ing. Julio Balarezo, PhD.  
PROFESOR CALIFICADOR

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2021.



-----  
Irving Mauricio Taípe Sailema

C.C. 180430746-8

AUTOR

## DEDICATORIA

*Dedico mi trabajo de titulación a Dios todo poderoso que ha guiado mi vida y me ha bendecido siempre hasta alcanzar esta gratificante meta.*

*A mis padres Segundo y María por enseñarme que el mejor regalo es la educación y que debo encaminarme bajo la humildad a lo largo de mi vida, gracias por todo.*

*A mis hermanos Gabriel y Pamela por confiar en mis capacidades y motivarme a tomar riesgos más allá de lo que yo creía posible.*

*A mis amigos por estar presentes en los buenos y malos momentos, su apoyo ha sido un gran aliciente para mi formación personal.*

## AGRADECIMIENTO

*Agradezco a la “Universidad Técnica de Ambato”, especialmente a la “Facultad de Ingeniería en Sistemas, Electrónica e Industrial” por convertirse en mi alma mater y permitirme cumplir mi sueño de ser un profesional.*

*Al Ingeniero Franklin Mayorga, tutor del presente trabajo, quien con su experiencia, enseñanza y paciencia supo guiarme en el desarrollo del mismo.*

*A los mejores padres Segundo y María, quienes con amor, alegría, paciencia y disciplina han sido los principales partícipes de mi formación personal y profesional.*

*A mis hermanos Gabriel y Pamela, por sus consejos y energía positiva brindada a través de los años.*

*A todos mis familiares, abuelos, tíos, tías, primos, y primas porque han sabido demostrarme su apoyo a lo largo de toda la carrera.*

*A aquellos compañeros de la universidad, que se han convertido en amigos para toda la vida, porque cada uno de ellos ha sido parte fundamental para culminar esta etapa.*

*A mis amigos y amigas que a pesar de no vernos tan seguido aún seguimos conservando una bonita amistad.*

## ÍNDICE

APROBACIÓN DEL TUTOR .....	i
AUTORÍA .....	ii
APROBACIÓN TRIBUNAL DE GRADO .....	iii
DERECHOS DE AUTOR .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
RESUMEN EJECUTIVO .....	xv
ABSTRACT .....	xvi
CAPITULO I .....	1
MARCO TEÓRICO .....	1
1.1. Tema de Investigación .....	1
1.2. Antecedentes Investigativos .....	1
1.2.1 Contextualización del problema .....	2
1.2.2 Fundamentación teórica .....	4
1.2.2.1 Desarrollo Web .....	4
1.2.2.2 Aplicaciones híbridas .....	4
1.2.2.3 SQL .....	4
1.2.2.4 MySQL .....	5
1.2.2.5 PHPMyAdmin .....	5
1.2.3 Servicios Web .....	5
1.2.3.1 Servicio SOAP .....	6
1.2.3.2 Servicio REST .....	6
1.2.4 Laravel .....	6
1.2.4.1 MVC: Modelo – Vista – Controlador .....	7
1.2.5 Tecnologías de desarrollo móvil multiplataforma .....	8
1.2.5.1 Ionic .....	8
1.2.5.1.1 Características .....	9
1.2.5.2 Flutter .....	10
1.2.5.3 React Native .....	11
1.2.5.4 Xamarin .....	11
1.2.6 Android .....	12



1.2.6.1	Ventajas .....	12
1.2.6.2	Desventajas.....	12
1.2.7	IOS.....	13
1.2.7.1	Ventajas .....	13
1.2.7.2	Desventajas.....	13
1.2.8	Metodologías Ágiles.....	14
1.2.9	Manifiesto Ágil.....	14
1.2.10	Scrum.....	15
1.2.10.1	Roles de Scrum .....	15
1.2.10.1.1	Scrum Master.....	15
1.2.10.1.2	Product Owner.....	16
1.2.10.1.3	Equipo de desarrollo .....	17
1.2.10.1.4	Eventos .....	17
1.2.10.1.5	Artefactos .....	18
1.2.11	Xtreme Programming (XP) .....	19
1.2.11.1	Roles .....	19
1.2.11.1.1	Fases .....	20
1.2.12	Mobile-D .....	26
1.2.12.1	Fases de Metodología Mobile-D .....	27
1.3.	Objetivos.....	30
1.3.1.	Objetivo General.....	30
1.3.2.	Objetivos Específicos .....	30
CAPITULO II.....		31
METODOLOGÍA.....		31
2.1	Materiales .....	31
2.2	Métodos .....	34
2.2.1	Modalidad de la Investigación.....	34
2.3	Población y Muestra .....	35
2.4	Procesamiento y Análisis de Datos .....	45
CAPITULO III .....		46
RESULTADOS Y DISCUSIÓN .....		46
3.1	Análisis y discusión de los resultados .....	46
3.1.1	Análisis de medicamentos más comunes, prescritos por el médico .....	46

3.1.2	Comparación de los tipos de frameworks en el desarrollo móvil multiplataforma 47	
3.1.3	Análisis comparativo de las metodologías Scrum, Extreme Programming (XP) y Mobile-D49	
3.2	Desarrollo de la propuesta .....	50
3.2.1	Fase de Exploración.....	50
3.2.1.1	Levantamiento de Requerimientos .....	50
3.2.1.2	Establecimiento de grupo de interés .....	51
3.2.1.3	Planificación del proyecto .....	51
3.2.2	Fase de Inicialización .....	52
3.2.2.1	Herramientas para iniciar la configuración del entorno.....	52
3.2.3	Fase de Producción.....	56
3.2.3.2	Creación de modelos en Laravel.....	58
3.2.3.3	Creación de los Controladores.....	61
3.2.3.4	Creación de las Rutas .....	69
3.2.3.5	Arquitectura del Proyecto .....	69
3.2.3.6	Dominio y hosting .....	71
3.2.3.7	Diseño lógico de base de datos .....	71
3.2.3.8	Diagrama de Actividades.....	72
3.2.3.9	Diagrama de secuencia .....	78
3.2.4	Fase de Producción.....	84
3.2.5	Fase de estabilización .....	104
3.2.5.1	Publicación de la API Rest .....	104
3.2.5.2	Generación de APK, del aplicativo móvil .....	104
3.2.6	Fase de Pruebas .....	105
3.2.6.1	Muestra del funcionamiento de la aplicación .....	106
CAPÍTULO IV .....		113
CONCLUSIONES Y RECOMENDACIONES .....		113
4.1	Conclusiones.....	113
4.2	Recomendaciones.....	113
BIBLIOGRAFÍA.....		114
ANEXOS .....		116

## ÍNDICE DE TABLAS

Tabla 1: Formato de entrevista .....	34
Tabla 2. Población y Muestra .....	35
Tabla 3: Resultados de la Entrevista.....	44
Tabla 4: Análisis de medicamentos de Pacientes con más de 65 años .....	46
Tabla 5: Análisis de Medicamentos de Pacientes de 56 a 65 años .....	46
Tabla 6: Análisis de Medicamentos de Pacientes de 46 a 55 años .....	46
Tabla 7: Análisis de Medicamentos de Pacientes de 36 a 45 años .....	47
Tabla 8: Análisis de Medicamentos de Pacientes de 26 a 35 años .....	47
Tabla 9: Cuadro comparativo entre Flutter, Ionic y Xamarin.....	48
Tabla 10 :Análisis Comparativo de las metodologías.....	49
Tabla 11: Definición de Usuarios .....	51
Tabla 12: Rutas utilizadas en la Api Rest .....	69
Tabla 13: Iteración 1.Registro de datos personales del médico.....	84
Tabla 14: Iteración 2: Editar datos personales del doctor .....	85
Tabla 15: Iteración 3. Agregar medicamentos.....	87
Tabla 16: Iteración 4. Eliminar medicamentos.....	89
Tabla 17: Iteración 5. Visualizar historial de pacientes .....	91
Tabla 18: Iteración 6. Buscador de pacientes .....	93
Tabla 19: Iteración 7. Agregar recetas a pacientes .....	95
Tabla 20: Iteración 8. Registro de datos personales del paciente .....	97
Tabla 21: Iteración 9. Editar datos personales del paciente.....	99
Tabla 22: Iteración 10. Visualización de la receta por parte del paciente.....	100
Tabla 23: Iteración 11. Alarma personalizada .....	102

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Modelo Vista Controlador.....	8
Ilustración 2: Scrum Master .....	16
Ilustración 3: Product owner .....	16
Ilustración 4: Equipo de desarrollo.....	17
Ilustración 5: Ciclo de desarrollo de Mobile-D.....	27
Ilustración 6. Pregunta 1.....	35
Ilustración 7. Pregunta 2.....	36
Ilustración 8. Pregunta 3.....	37
Ilustración 9. Pregunta 4.....	38
Ilustración 10. Pregunta 5.....	38
Ilustración 11. Pregunta 6.....	39
Ilustración 12. Pregunta 7.....	40
Ilustración 13. Pregunta 8.....	41
Ilustración 14. Pregunta 9.....	41
Ilustración 15. Pregunta 10.....	42
Ilustración 16: Descarga de Visual Studio Code .....	53
Ilustración 17: Descarga de node.js.....	53
Ilustración 18: Ejecución Postman .....	55
Ilustración 19: Instalación de composer .....	55
Ilustración 20: Comprobación de la instalación de composer .....	56
Ilustración 21: Instalación laravel.....	56
Ilustración 22: Proyecto en Laravel.....	57
Ilustración 23: Estructura general de laravel .....	58
Ilustración 24: Creación del modelo Usuarios.....	59
Ilustración 25: Creación del modelo Medicamentos .....	59
Ilustración 26: Creación del modelo Alarmas .....	59
Ilustración 27: Creación del modelo Recetas .....	60
Ilustración 28: Comando para insertar los registros en las tablas.....	60
Ilustración 29: Comando para crear el Controlador Administrador .....	61
Ilustración 30: Métodos para el Administrador .....	61

Ilustración 31:Método para la creación de Doctores .....	62
Ilustración 32:Métodos para mostrar y eliminar usuarios .....	62
Ilustración 33: Comando para la creación del controlador Alarmas.....	63
Ilustración 34: Método para eliminar alarmas .....	63
Ilustración 35:Método para crear la receta .....	63
Ilustración 36:Método para mostrar Usuarios .....	64
Ilustración 37: Comando para la creación del controlador Doctores.....	64
Ilustración 38:Método para mostrar alarmas por usuarios .....	64
Ilustración 39: Método para crear la receta .....	65
Ilustración 40: Comando para la creación del controlador Medicamentos.....	65
Ilustración 41: Método para mostrar medicamentos .....	65
Ilustración 42:Métodos para agregar y eliminar los medicamentos .....	66
Ilustración 43:Creación del Controlador de recetas.....	66
Ilustración 44: Método para mostrar recetas por usuarios.....	67
Ilustración 45: Métodos para agregar y eliminar recetas.....	67
Ilustración 46: Creación del controlador Usuarios .....	67
Ilustración 47: Método para actualizar datos del usuario .....	68
Ilustración 48: Método para el inicio de sesión.....	68
Ilustración 49: Método para el registro de usuarios.....	68
Ilustración 50: Hosting Cpanel .....	71
Ilustración 51: Diagrama de Base de datos.....	71
Ilustración 52: Diagrama de actividad. Registro de datos del Médico .....	72
Ilustración 53: Diagrama de actividades. Editar datos del Médico .....	72
Ilustración 54: Diagrama de actividades. Agregar medicamento .....	73
Ilustración 55: Diagrama de actividades. Eliminar medicamento .....	73
Ilustración 56: Diagrama de actividades. Registro del paciente.....	74
Ilustración 57: Diagrama de actividades. Editar datos del paciente .....	75
Ilustración 58: Diagrama de actividades. Alarma personalizada.....	75
Ilustración 59: Diagrama de actividades. Buscador de pacientes .....	76
Ilustración 59.1: Diagrama de actividades. Buscador de pacientes.....	77
Ilustración 60: Diagrama de actividades. Agregar receta .....	78
Ilustración 61: Diagrama de actividades. visualizar receta.....	78

Ilustración 62: Diagrama de actividades. Historial de pacientes .....	78
Ilustración 63: Diagrama de secuencia. Registro de datos del médico.....	78
Ilustración 64: Diagrama de secuencia. Editar datos personales del médico .....	79
Ilustración 65: Diagrama de secuencia. Agregar medicamento .....	79
Ilustración 66: Diagrama de secuencia. Eliminar medicamento.....	80
Ilustración 67: Diagrama de secuencia. Registro de datos del paciente .....	80
Ilustración 68: Diagrama de secuencia. Editar datos personales del paciente .....	81
Ilustración 69: Diagrama de secuencia. Agregar alarma personal.....	81
Ilustración 70: Diagrama de secuencia. Búsqueda de paciente .....	82
Ilustración 71:Diagrama de secuencia. Agregar receta a paciente .....	82
Ilustración 72: Diagrama de secuencia. Abrir receta.....	83
Ilustración 73: Diagrama de secuencia. Historial del paciente .....	83
Ilustración 74: Iteración 1. Desarrollo home_page.....	84
Ilustración 75: Iteración 1. Diseño (creación de perfil doctor).....	85
Ilustración 76: Iteración 2. Desarrollo de la página editar datos .....	86
Ilustración 77: Iteración 2. Diseño Editar datos personales del médico .....	86
Ilustración 78: Iteración 2. Diseño Guardar datos personales .....	87
Ilustración 79: Iteración 3. Desarrollo de la página para agregar medicamentos .....	88
Ilustración 80: Iteración 3. Medicamentos agregados .....	88
Ilustración 81: Iteración 3. Diseño de la página agregar medicamentos .....	89
Ilustración 82: Iteración 4. Desarrollo del método para eliminar medicamentos .....	90
Ilustración 83: Iteración 4. Diseño de eliminar medicamentos .....	90
Ilustración 84: Iteración 5. Desarrollo para visualizar el historial de pacientes .....	91
Ilustración 85: Iteración 5. Diseño visualización de pacientes.....	92
Ilustración 86: Iteración 5. Visualización de historial de pacientes.....	92
Ilustración 87: Iteración 5. Visualización del detalle de recetas.....	93
Ilustración 88: Iteración 6. Desarrollo de buscador de pacientes .....	94
Ilustración 89: Iteración 6. Diseño del buscador de pacientes.....	94
Ilustración 90: Iteración 6. Diseño de paciente no registrado.....	95
Ilustración 91: Iteración 7. Desarrollo de la página agregar recetas.....	96
Ilustración 92: Iteración 7. Diseño para recetar medicamentos.....	96

Ilustración 93: Iteración 7. Diseño para agregar horario de medicación .....	97
Ilustración 94: Iteración 8. Desarrollo para el registro de pacientes.....	98
Ilustración 95: Iteración 8. Diseño del registro de pacientes .....	98
Ilustración 96: Iteración 9. Desarrollo de la página para editar los datos del paciente.....	99
Ilustración 97: Iteración 9. Diseño de la página para editar los datos personales.....	100
Ilustración 98: Iteración 10. Desarrollo de la visualización de la receta .....	101
Ilustración 99: Iteración 10. Diseño para visualizar las recetas.....	101
Ilustración 100: Iteración 10. Diseño del detalle de la receta .....	102
Ilustración 101: Iteración 10. Desarrollo de la página para agregar una alarma personalizada .	103
Ilustración 102: Iteración 10. Diseño para agregar una alarma personalizada .....	103
Ilustración 103: Ambiente de Producción .....	104
Ilustración 104: Comando para generar el releas de producción.....	104
Ilustración 105: Generar APK firmada.....	104
Ilustración 106: Pruebas de ejecución .....	105
Ilustración 107: Solución del problema de idioma .....	106
Ilustración 108: Funcionamiento de la aplicación. Registro de usuario .....	106
Ilustración 109: Funcionamiento de la aplicación. Registro de usuario .....	107
Ilustración 110: Funcionamiento de la aplicación. Editar datos personales .....	108
Ilustración 111: Funcionamiento de la aplicación. Editar datos .....	109
Ilustración 112: Funcionamiento de la aplicación. Editar datos.....	109
Ilustración 113: Funcionamiento de la aplicación. Agregar medicamentos .....	110
Ilustración 114: Funcionamiento de la aplicación. Agregar medicamentos .....	110
Ilustración 115: Funcionamiento del aplicativo móvil. Buscar medicamentos .....	111
Ilustración 116: Funcionamiento del aplicativo móvil. Buscar medicamentos .....	111
Ilustración 117: Funcionamiento del aplicativo móvil. Agregar receta médica .....	112
Ilustración 118: Funcionamiento del aplicativo. Agregar medicamento que se recetará .....	112
Ilustración 119: Funcionamiento del aplicativo. Selección del medicamento.....	113
Ilustración 120: Funcionamiento del aplicativo. Agregar información para la receta.....	113
Ilustración 121: Funcionamiento del aplicativo. Recetar a paciente .....	114
Ilustración 122: Funcionamiento del aplicativo. Visualización de la receta .....	114
Ilustración 123: Funcionamiento del aplicativo. Visualización del detalle de la receta. ....	115
Ilustración 124: Funcionalidad del aplicativo. Aplicación en el dispositivo móvil.....	115
Ilustración 125: Funcionalidad del aplicativo. Aplicación en el dispositivo móvil.....	116

## RESUMEN EJECUTIVO

El desarrollo de aplicaciones móviles es un tema muy común hoy en día, ya que, este cubre un sinnúmero de funcionalidades, que van desde pequeñas aplicaciones cotidianas en el hogar, hasta aplicativos de gran relevancia en varias áreas profesionales. Actualmente, la mayoría de personas disponen de dispositivos móviles inteligentes, facilitando su acceso y participación a distintas apps.

El campo de la medicina a pesar de los siglos no ha perdido su importancia o relevancia en nada, donde la relación médico- paciente tan importante que ha trascendido el tiempo, sin tomar en cuenta etnias, géneros u otras características; esta relación se enfoca primordialmente en el médico que da su diagnóstico y tratamiento a una persona, y el paciente que sigue las indicaciones al pie de la letra para su posterior recuperación.

El presente proyecto tiene como finalidad la creación de un aplicativo móvil orientado a la interacción del médico con su paciente, intentando ayudar a las personas con sus inconvenientes sobre la toma de medicamentos, para así reducir y mitigar en parte los problemas consecuentes por una irresponsabilidad latente del ser humano en el orden de sus prescripciones y tratamientos.

El aplicativo móvil está diseñado en el framework Ionic por la gran facilidad que este brinda al crear aplicaciones multiplataforma, sin la necesidad de codificar por separado; adicionalmente se utiliza el framework Laravel para la creación del API Rest, que abarca todos los métodos que se utilizará en el desarrollo de la app.

**Palabras clave:** Aplicación móvil, framework, medicina, API Rest, médico, paciente.



## **ABSTRACT**

The development of mobile applications is a very common topic nowadays, since it covers a myriad of functionalities, ranging from small everyday applications at home, to highly relevant applications in various professional areas. Currently, most people have smart mobile devices, facilitating their access and participation in different apps.

The field of medicine, despite the centuries, has not lost its importance or relevance in anything, the doctor-patient relationship being so important that it has transcended time, without taking ethnic groups, genders or other characteristics into account; This relationship focuses primarily on the treating physician who gives a diagnosis and corresponding treatment to a person, and this patient follows the instructions to the letter for their subsequent recovery.

The purpose of this project is to create a mobile application aimed at the interaction of the doctor with his patient, trying to help people with their problems with taking medications, in order to reduce and partially mitigate the problems resulting from latent irresponsibility. Of the human being in the order of their prescriptions and treatments.

The mobile application is designed in the Ionic framework due to the great ease it offers to create multiplatform applications, without the need to code separately; additionally, the Laravel framework is used to create the Rest API that covers all the methods that will be used in the development of the app.

**Keywords:** Mobile application, framework, medicine, API Rest, doctor, patient.

# CAPITULO I

## MARCO TEÓRICO

### 1.1. Tema de Investigación

“APLICACIÓN MÓVIL MULTIPLATAFORMA PARA EL CONTROL Y REGISTRO DE TRATAMIENTOS MÉDICOS.”

### 1.2. Antecedentes Investigativos

De acuerdo con las investigaciones que se han encontrado anteriormente, se puede resaltar los resultados obtenidos por David Castro Villena en su tesis "Sistema de Gestión y Control de Medicación", trabajo realizado en la Universidad Politécnica de Cataluña, desarrolló una aplicación donde el paciente puede recibir notificaciones de los diferentes medicamentos que incluya su plan de medicación en las horas en las que sea necesario hacer uso de alguno de ellos. Esta aplicación incluye una serie de información relevante como la imagen del medicamento, el nombre y la dosis que debe utilizar, también puede informar cuando se detecte que escasea la cantidad disponible de algún fármaco e incluso consultar toda la información introducida por el personal médico sobre el plan de medicación, de una forma fácil y clara [1].

En el proyecto de tesis llevada a cabo por Carlos Aragonés Lacarta con el nombre "Aplicaciones para dispositivos móviles basadas en NFC (Near Field Communication) comunicación de campo cercano para el control, gestión y monitorización de la medicación en pacientes de larga duración", trabajo realizado en la Universidad Politécnica de Cataluña, desarrolló una aplicación móvil que contaba con un sistema de estadísticas a mostrar en forma gráfica y un diagrama de navegabilidad que muestra al paciente las farmacias cercanas a su posición actual además de trazar una ruta hasta la farmacia seleccionada. Este trabajo fue de gran utilidad ya que ayudó a los pacientes tener un mejor control al momento de tomar una de las medicaciones que requería [2].

De la misma manera, Franklin Jhino Arias Moreno y Harold Ayrton Ruiz Rojas con el tema de tesis llamado "Aplicación web y móvil de monitoreo y control del tratamiento de los pacientes del hospital Nacional Arzobispo Loayza", trabajo realizado en la

Universidad de San Martín de Porres, desarrollaron una aplicación web y móvil donde el usuario puede visualizar sus datos médicos, lista de tratamiento a las cuales se esté sometiendo, además que cuenta con el acceso a la información del doctor que está a cargo del tratamiento. El proyecto realizado tuvo gran impacto ya que ayudó al hospital realizar un mejor seguimiento de los tratamientos de farmacología para el beneficio del hospital y sobre todo de los pacientes, ya que les permitió a estos tener la información y los tiempos en que tenían que administrarse un medicamento [3].

### **1.2.1 Contextualización del problema**

En los últimos años el aumento del uso de dispositivos móviles se ha convertido en una parte integral en la vida cotidiana de las personas, lo que conduce a la expansión del desarrollo de aplicaciones y el desafío de utilizarlas para las diferentes plataformas. Hoy en día, las aplicaciones están dirigidas a dispositivos móviles específicos, la expectativa radica en que sea ejecutable en la mayor cantidad de dispositivos; es decir, que sean multiplataforma, combinando lo mejor de la programación nativa y web, esto proporciona un entorno inmejorable para el desarrollo de aplicaciones capaces de usar de forma eficiente los recursos del dispositivo y esté sea utilizado en el mayor número de plataformas, reduciendo el tiempo y costo por mantenimiento o crecimiento del software desarrollado [4].

El uso de tecnología móvil en el ámbito médico es un elemento de gran auge en la actualidad; su empleo es efectuado con el objetivo de alcanzar una mejora en la atención médica para el paciente contribuyendo a potenciar la asistencia médica. La tecnología móvil facilita una mejor prevención en lo concerniente a la salud, ya que ofrece al usuario la posibilidad de llevar un mejor control al momento de administrar la propia medicación para así evitar consecuencias a largo plazo por la mala administración de fármacos causando a futuro un deterioro en la enfermedad del paciente [4].

En relación con el mundo actual, se han encontrado datos que demuestran que las personas se administran medicamentos de manera diaria o en frecuencias altas, ya sea por una enfermedad crónica, o por afecciones medicas temporales; la problemática de los pacientes radica al momento que presentan olvidos en el seguimiento del tratamiento

farmacológico e inclusive el uso inadecuado de los medicamentos provocando graves consecuencias y un significativo deterioro en el estado de salud del paciente [5].

“La Organización Mundial de Salud (OMS) ha definido que alrededor del 50% de los pacientes no toman sus medicamentos como se recomienda y puede genera un alargamiento en el tratamiento o provocar una progresión innecesaria de la enfermedad lo cual reduciría las capacidades funcionales, una menor calidad de vida e incluso la muerte” [5].

En el Ecuador por ejemplo los gobiernos nacionales realizan grandes esfuerzos e inversiones públicas para mejorar las condiciones de vida de sus habitantes, viendo como el producto de estas inversiones llegan a ser menos representativas y efectivas, entre los sectores que mayor incidencia tiene sobre la calidad de vida de sus habitantes se halla el sector de la salud [6]. En este sentido se puede observar que la tecnología informática en el Ecuador no es tratada de una forma óptima la cual ayude de manera sistemática con datos o información del avance del tratamiento de los pacientes por medio de la red.

En la ciudad de Ambato, lo concerniente al área de salud, no posee una buena difusión de aplicativos para la administración de medicamentos; si bien, existen aplicaciones móviles similares o relacionadas al seguimiento de tratamientos médicos, estas se centran en enfermedades específicas dejando de lado otras, o bien estas presentan desafíos en su uso, sin pensar en las limitaciones o dificultades que los pacientes posean, es por ello que se busca ayudar, tanto a los pacientes en la administración de sus medicamentos, como a los médicos en el seguimiento de sus pacientes; por este motivo se desarrollará dicha aplicación móvil con el fin de facilitar y mejorar la asistencia médica, mediante un recordatorio adecuado de la fecha y hora en la que deben ingerirse los medicamentos.

## **1.2.2 Fundamentación teórica**

### **1.2.2.1 Desarrollo Web**

El desarrollo web es la programación necesaria para la construcción del sitio web. Se divide en dos, la parte del cliente y la parte del servidor. En la parte del cliente estaríamos hablando de HTML (HyperText Markup Language) Lenguaje de Marcas de Hipertexto y CSS (Cascading StyleSheets) Hojas de estilo en cascada, código básico para creación de páginas web, y JavaScript y DOM (Document Object Model) Modelo de Objeto de Documento, para la interacción con el usuario. En la parte del servidor se trabaja con PHP (Hypertext Pre-Processor) Preprocesador de Hipertexto, ASP.NET (Active Server Pages) Páginas de Servidor Activo, JSP (JavaServer Pages) Páginas de Servidor Java, etc. Ya que con ello se contruiría el back-end, la parte de la web que el usuario no ve. Su objetivo es el diseño de bases de datos y asegurar la seguridad de la web [7].

### **1.2.2.2 Aplicaciones híbridas**

Una aplicación híbrida se basa en el desarrollo de una página móvil con capacidad para manejar los elementos nativos del dispositivo (cámara y GPS (Global Positioning System) Sistema de posicionamiento global entre otros). Para esto se utiliza la nueva versión del lenguaje HTML conocida como HTML5, que está siendo utilizada de manera creciente por los equipos de desarrollo. Estas tecnologías son complementarias, pues permiten prestar servicios a la mayor cantidad de dispositivos móviles que existen en el mercado [8].

### **1.2.2.3 SQL**

SQL (Structured Query Language) es un lenguaje estándar de las bases de datos relacionales ya que permite especificar diversos tipos de operaciones y es capaz de conjugar las operaciones del Algebra y el cálculo relacional con operadores adicionales, además permite realizar consultas para recuperar o modificar información de bases de datos, así como hacer cambios en ellas [9].

SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado, y dispone de instrucciones para crear, insertar, actualizar, eliminar y consultar las bases de datos relacionales [9].

#### **1.2.2.4 MySQL**

“MySQL es el sistema de administración de bases de datos (Database Management System, DBMS), desarrollado y proporcionado por MySQL AB”. Es un sistema de gestión de base de datos relacional, que fue escrito en C y C++ y destaca por su gran adaptación en diferentes entornos de desarrollo, permitiendo interactuar con los lenguajes de programación como PHP, Perl y Java [10].

#### **1.2.2.5 PHPMyAdmin**

Es una herramienta web que permite controlar y manejar bases de datos MySQL donde permite crear, eliminar, alterar tablas, borrar, editar y añadir campos ejecutando cualquier sentencia SQL [10].

### **1.2.3 Servicios Web**

Los servicios Web son aplicaciones de negocio autodescriptivas y modulares que exponen la lógica empresarial como servicios a través de Internet a través de la interfaz programable y donde el protocolo de Internet (IP) puede ser utilizado para encontrar e invocar esos servicios. “Un servicio web es el elemento clave en la integración de diferentes sistemas de información”, ya que los sistemas de información pueden basarse en diferentes plataformas, lenguajes de programación y tecnologías [11].

### **1.2.3.1 Servicio SOAP**

La implementación de servicios web por protocolo simple de acceso a objetos (SOAP) se desarrolló como una alternativa al estándar CORBA (Common Object Request Broker Architecture). Para garantizar el transporte de datos en SOAP, se utilizan protocolos como HTTP, SMTP, entre otros, en formato XML. En este enfoque, un proveedor de servicios publica una descripción del servicio o una interfaz para el registro de servicios, por lo que el solicitante del servicio puede encontrar una instancia de servicio correcta y utilizarla. Algunos problemas de rendimiento en SOAP se producen al formar el mensaje SOAP ya que agrega un encabezado adicional y partes al cuerpo al mensaje. “Los servicios Web basados en SOAP incluyen una variedad de estándares, tales como WSDL, WSBPEL, WS-Security, WS- Addressing. Estas normas fueron desarrolladas por organizaciones de normalización, como W3C y OASIS” [11].

### **1.2.3.2 Servicio REST**

Un servicio REST se define como una agregación de diferentes recursos que pueden ser alcanzados desde un identificador universal de recurso (URI) base. Un recurso representa a una entidad del mundo real cuyo estado está expuesto y puede cambiarse accediendo a un URI [11].

### **1.2.4 Laravel**

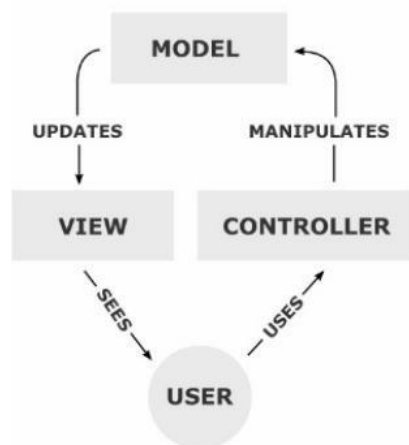
Laravel es un framework de código abierto para el desarrollo de aplicaciones web en PHP 5 que posee una sintaxis simple, expresiva y elegante. “Fue creado en 2011 por Taylor Otwell, inspirándose en Ruby on Rails y Symfony”, Laravel facilita el desarrollo simplificando el trabajo con tareas comunes como la autenticación, el enrutamiento, gestión sesiones, el almacenamiento en caché entre otros. Las principales características de Laravel son:

- Está diseñado para desarrollar bajo el patrón MVC (modelo - vista - controlador), para facilitar el trabajo en equipo, el mantenimiento y la reutilización del código.
- Integra un sistema ORM de mapeado de datos relacional llamado Eloquent.
- Permite la gestión de bases de datos y la manipulación de tablas desde código, manteniendo un control de versiones de estas mediante su sistema de Migraciones.
- Utiliza un sistema de plantillas para las vistas llamado Blade y así facilitar la creación de vistas mediante el uso de layouts, herencia y secciones.
- Facilita la extensión de funcionalidad mediante paquetes o librerías externas.
- Incorpora un intérprete de línea de comandos llamado Artisan que ayuda con las tareas rutinarias como la creación de distintos componentes de código, base de datos, migraciones, gestión de rutas, cachés, colas, tareas programadas entre otros [12].

#### 1.2.4.1 MVC: Modelo – Vista – Controlador

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario [12].

Ilustración 1: Modelo Vista Controlador



Fuente: (Laravel 5, 2015)



## 1.2.5 Tecnologías de desarrollo móvil multiplataforma

### 1.2.5.1 Ionic

Ionic es un framework de código abierto basado en la tecnología de Apache Cordova que permite crear aplicaciones móviles híbridas. A diferencia de las aplicaciones nativas, las aplicaciones híbridas son implementadas usando tecnologías web (HTML, CSS y JavaScript) [13].

Además, son aquellas que se desarrollan en un lenguaje de programación de propósito general y que luego se puede compilar o exporta a cualquier plataforma o dispositivo con cambios mínimos o nulos [4].

#### 1.2.5.1.1 Características

Ionic Framework presenta las siguientes características:

- **Alto rendimiento:** Ionic está construido para ser rápido, gracias a la mínima manipulación del DOM y con aceleraciones de transiciones por hardware.
- **AngularJS & Ionic:** Utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones robustas el cual facilitaría el desarrollo de las aplicaciones.
- **Diseño:** Ionic ha sido diseñado para trabajar con todos los dispositivos móviles ya que cuenta con muchos componentes usados en móviles, tipografía, elementos interactivos entre otros.
- **CLI:** Mediante la utilización de la consola permite crear, construir y compilar las aplicaciones móviles.

- **Cordova**, Ionic cuenta con un marco de desarrollo conocido como Cordova el cual utiliza nos permite utilizar los plugins que estén desarrollados en el framework en el cual se puede instalar mediante líneas de comandos [4].

#### **1.2.5.2 Flutter**

Flutter es un framework de desarrollo de aplicaciones móviles multiplataforma creado por Google. Es de código abierto y permite construir aplicaciones tanto para Android como para iOS [13].

También usa Skia para el renderizado 2D. El objetivo de esta herramienta es permitir a los desarrolladores construir aplicaciones multiplataforma a partir de una única base de código, la cual es compilada a código nativo para cada una de las plataformas objetivo. Además, se aprovecha de la flexibilidad de Dart en cuanto a su compilación y ejecución para obtener ciclos de desarrollo más rápidos y tiempos de ejecución más bajos [13].

Flutter construye los widgets a nivel de aplicación, lo que hace que el desarrollador pueda personalizarlos y extenderlos de manera sencilla, consiguiendo una mayor libertad en el diseño de las aplicaciones [13].

#### **1.2.5.3 React Native**

React Native es un framework utilizado principalmente para el desarrollo de aplicaciones móviles, además permite trabajar para la plataforma universal de Windows. Permite utilizar las mismas librerías usadas para el desarrollo de aplicaciones web, pero para plataformas móviles. Esto hace que esta herramienta resulte especialmente atractiva para desarrolladores u organizaciones que ya tengan aplicaciones web implementadas con estas tecnologías [13].

#### **1.2.5.4 Xamarin**

Xamarin utiliza lenguaje C# no sólo para la implementación de la lógica de negocio, sino que también provee de controles de interfaz personalizados para cada plataforma. Sin embargo, esto hace que la implementación de estas interfaces con C# no se pueda compartir entre plataformas, haciendo que el porcentaje de código compartido sea mucho menor que en otras soluciones, como Flutter [13].

## **1.2.6 Android**

Es un sistema operativo basado en Linux creado por Android Inc. Está diseñado para Smartphone, pero también se puede encontrar en otros dispositivos como tablets, televisores y relojes inteligentes [14].

### **1.2.6.1 Ventajas**

- Permite una gran personalización.
- Es un sistema muy abierto, permitiendo que haya más aplicaciones desarrolladas para esta plataforma.
- Su IDE oficial se puede encontrar para cualquier plataforma de escritorio [14].

### **1.2.6.2 Desventajas**

- Al existir tantos fabricantes distintos, con un hardware muy heterogéneo, no hay tanta integración software-hardware como en los dispositivos de Apple.
- Google desarrolla el sistema operativo, pero depende de cada uno de los fabricantes el hacer llegar las últimas actualizaciones hasta los usuarios.
- Al ser la plataforma más utilizada se puede encontrar mayor cantidad de malware[14].

## **1.2.7 IOS**

IOS fue desarrollado para dispositivos iPhone, iPad, en el cual es una Tablet que fue basada en el iPhone el cual es un intermediario entre el iPhone y el Mac. “Actualmente se encuentra en segundo lugar, por detrás de Android, con respecto a la cuota de mercado mundial” [14].

### **1.2.7.1 Ventajas**

- Las nuevas actualizaciones del sistema operativo se filtran rápidamente hasta los usuarios.
- Al ser un sistema más cerrado, se ejerce un mayor control sobre las aplicaciones que pueden instalar los usuarios, mejorando la seguridad [14].

### **1.2.7.2 Desventajas**

- Su interfaz es sencilla y fácil de utilizar.

- El IDE oficial para el desarrollo de aplicaciones iOS sólo está disponible para Mac.
- Apple sólo diseña dispositivos de gama alta, en consecuencia, hay menos usuarios que puedan adquirirlos [14].

### **1.2.8 Metodologías Ágiles**

El término ágil surge como iniciativa de un conjunto de expertos en el área de desarrollo de software con el fin de optimizar el proceso de creación de este, el cual era caracterizado por ser rígido y con mucha documentación [15].

### **1.2.9 Manifiesto Ágil**

El manifiesto detalla cuatro valores clave para habilitar equipos de alto rendimiento, los cuales se menciona a continuación:

- Valorar más a los individuos y su iteración que a los procesos y las herramientas.
- Valorar más a el software que funciona que a la documentación exhaustiva.
- Valorar más la colaboración con el cliente que la negociación contractual.
- Valorar más la respuesta al cambio que el seguimiento de un plan [16].

### **1.2.10 Scrum**

“Desarrollada por Ken Schwaber, Jeff Sutherland y Mike”. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Sus principales características son:

- El desarrollo de software se realiza mediante iteraciones denominadas sprints, con una duración de 30 días, el resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- Reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [16].

#### **1.2.10.1 Roles de Scrum**

El marco de trabajo Scrum tiene tres roles los cuales son:

##### **1.2.10.1.1 Scrum Master**

“Al respecto Paula canal (2015) afirma que el Scrum master (SM) es la persona especializada en la metodología Scrum”, por ende, es el líder del equipo que ayuda a gestionar de manera ágil y a romper barreras u obstáculos que entorpezcan el trabajo del equipo [17].

Ilustración 2: Scrum Master



Fuente: (Aplicación scrum, 2019)

### 1.2.10.1.2 Product Owner

La Escuela europea de dirección y empresa (2018) expresa que el product owner es la persona que se apropia del producto y para ello debe conocer el negocio y producto final deseado. El product owner es el encargado de generar el listado de tareas a realizar durante la elaboración del proyecto, brindando entregas parciales y de valor [17].

Según Cordoves Franco y Felipe Germán (2018) el product owner tiene entre sus principales funciones generar el historial de usuario del producto y a su vez es la persona encargada de la relación con terceros, cabe aclarar que el product owner puede ser una persona interna del equipo de trabajo o puede ser una persona externa brindada por el usuario o negocio que solicita el desarrollo [17].

Ilustración 3: Product owner



Fuente: (Aplicación scrum, 2019)

### 1.2.10.1.3 Equipo de desarrollo

El equipo de desarrollo es el encargado de la entrega del producto, José Rubén Laínez Fuentes (2014) afirma que este equipo está conformado por un grupo de personas con conocimientos y habilidades predestinadas al análisis, diseño, desarrollo de software. Una cualidad importante de los equipos de desarrollo es que están conformados con un mínimo de tres personas y un máximo de nueve, este límite es definido debido a que si el equipo es muy grande los canales de comunicación serían difíciles de controlar generando muchos inconvenientes en la ejecución del proyecto, disminuyendo así la agilidad del mismo [17].

Ilustración 4: Equipo de desarrollo



Fuente: (Aplicación de Scrum, 2019)

### 1.2.10.1.4 Eventos

Marc Bara (2019) afirma que el marco de trabajo Scrum está compuesto por seis eventos los cuales son:

- **Sprint:** El sprint es la base de toda la metodología, es el tiempo asignado para la ejecución de un grupo de tareas que tiene como objetivo una entrega parcial del producto o la creación de un incremento potencial de este. Es recomendable que el sprint sea máximo de un mes y mínimo de dos semanas.
- **Planeación del sprint (Sprint Planning):** Es el evento donde el equipo de desarrollo, el scrum master y el product owner se reúnen para definir las tareas a realizar en el sprint y plantear la duración del mismo. “Cuando el sprint tiene una duración de un mes

se tiene calculado que su planeación sea de ocho horas, en caso de que sea menor a un mes disminuirá el tiempo de este evento”.

- **Objetivo del sprint (Sprint Goal):**. Este objetivo permite que el equipo de desarrollo tenga una dirección y estén al tanto del porque se está construyendo el incremento.
- **Scrum diario (Daily scrum):** Este evento es una reunión diaria en donde el equipo informa sus actividades presentes, futuras y qué inconvenientes tienen, esto con el fin de que el equipo se apoye entre sí. Esta reunión tiene una duración máxima de quince minutos.
- **Revisión del sprint (Sprint Review):** Tiene como objetivo evaluar el incremento en la ejecución del proyecto y adaptar la lista de requisitos del producto si es necesario. El sprint tiene una duración aproximada de cuatro horas.
- **Retrospectiva de Sprint (Sprint Retrospective):** Este evento sucede después de la revisión del sprint y antes del siguiente sprint Planning. La reunión tiene una duración aproximada de tres horas.

#### 1.2.10.1.5 Artefactos

Los artefactos son el resultado de la implementación del marco de trabajo scrum o en pocas palabras se puede definir como la representación de trabajo o valor en diversas formas. Los artefactos de scrum son los siguientes:

- **Lista de producto (Product backlog):** La lista de producto es en sí las descripciones generales de los requisitos y funciones esperadas para el proyecto.
- **Lista de sprint (Sprint backlog):** Es una parte del product backlog seleccionada para el sprint y es agregado un plan de entrega.

- **Incremento:** Es el grupo de todos los ítems completados de la lista del sprint y el valor de todos los incrementos de los Sprint anteriores [17].

### **1.2.11 Xtreme Programming (XP)**

La metodología XP trabaja con el cliente ya que se debe realizar iteraciones cada dos semanas, donde no existe más documentación que el código en sí; la retroalimentación del cliente al sistema permite la modificación necesaria en cada versión [18].

XP utiliza historias de usuarios la cual no puede demorar en desarrollarse más de una semana. Se debe definir un estándar en el tipo de codificación, lo cual le permite a los programadores tener definido un solo estilo al momento de programar.

La metodología XP busca agilidad, disciplina y un aporte de soluciones fáciles en el desarrollo de un proyecto de software. Se fundamenta en los siguientes principios:

- Minimizar los periodos de desarrollo.
- Implicar al cliente desde el inicio del proyecto [18].

#### **1.2.11.1 Roles**

- **Cliente**

El cliente es el responsable de conducir el proyecto. Define el proyecto y sus objetivos. Cuanto más preciso es su trabajo y cuanto mayor sea su involucración, mayores serán las oportunidades de éxito.

- **Programador**

Una vez que se han comprendido las historias de usuario, el XP adjudica a los programadores la responsabilidad de tomar decisiones técnicas. Los desarrolladores estiman el tiempo que les va a tomar cada historia.



- **Encargado de Pruebas (Tester)**

El encargado de pruebas ayuda al cliente a definir y escribir las pruebas de aceptación de las historias de usuario. Este rol en un equipo XP también es responsable realizar test periódicamente e informar de los resultados al equipo.

- **Encargado de Seguimiento (Tracker)**

Hace el seguimiento de acuerdo con la planificación. La métrica más importante para XP es la velocidad del equipo, que se define como el tiempo ideal estimado para las tareas frente al tiempo real dedicado.

- **Entrenador (Coach)**

No es un rol cubierto en todos los equipos de XP. Su papel es guiar y orientar al equipo, especialmente cuando un equipo comienza a trabajar siguiendo la metodología XP.

- **Gestor (Big Boss)**

Es el gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado. El cliente puede asumir este papel.

#### **1.2.11.1.1 Fases**

##### **Fase de exploración**

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuario”. Los programadores estiman los tiempos de desarrollo en base a esta información.

##### **Fase de planificación**

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias usuario, y, asociadas a éstas, las entregas.

##### **Fase de iteraciones**

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que

implementa las historias de usuario asignadas a la iteración.

### **Fase de puesta en producción**

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

### **Planificación**

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes.

### **Programación en parejas**

La programación en parejas incrementa la calidad del código sin impactar en la fecha de entrega.

### **Pruebas**

#### **Pruebas Unitarias**

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-driven programming”) [19].

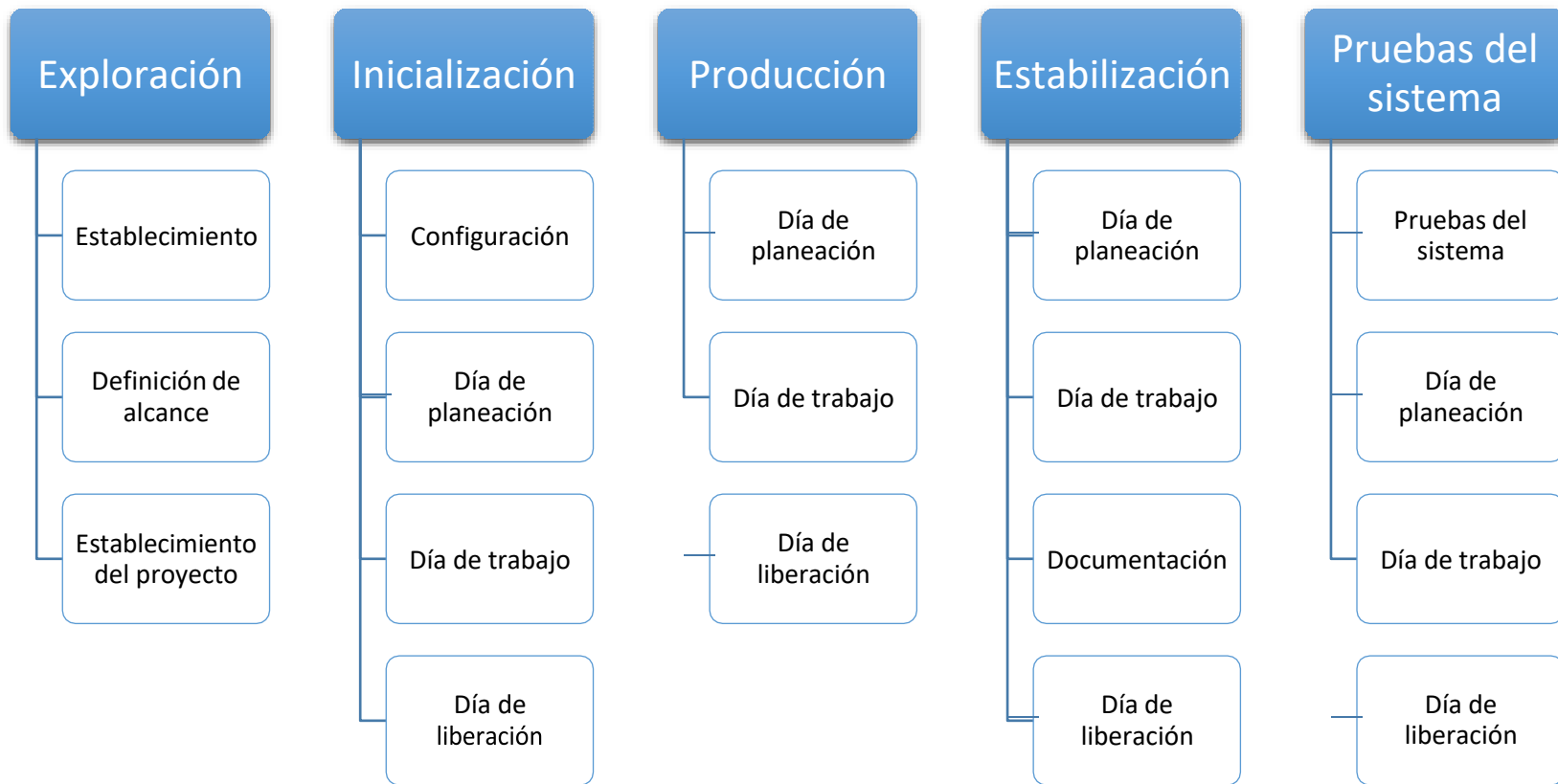
#### **1.2.12 Mobile-D**

Es una metodología ágil para el desarrollo de aplicaciones móviles, que proporciona un ciclo de desarrollo corto, puede detectar errores y realizar una verificación funcional con los usuarios [20].

Esta metodología se utiliza para equipos que tienen menos de diez desarrolladores trabajando y en poco tiempo menos de 10 semanas. Mobile-D combina extrema programación (XP) para el proceso de desarrollo [20].

### 1.2.12.1 Fases de Metodología Mobile-D

Ilustración 5: Ciclo de desarrollo de Mobile-D



Fuente: (Elaborado por el autor en base a [20])

## **Exploración**

Los objetivos de la primera etapa son establecer los grupos de partes interesadas, que son uno de los principales indicadores de lo que se espera de la aplicación móvil, definir los objetivos de la aplicación móvil y elegir qué miembros van a participar en el desarrollo de la aplicación móvil [20].

Las salidas obtenidas en esta fase son las siguientes:

- Los requisitos iniciales
- Plan del proyecto.
- Descripción de los procesos

## **Inicialización**

En esta se prepara el diseño arquitectónico, diagramas de casos de uso, diseño de interfaz de usuario (UI) y sus diferentes funcionalidades.

La documentación conseguida es la siguiente:

- El plan actual del proyecto.
- La versión de la arquitectura del software y la descripción del diseño.
- Requisitos iniciales modificados.
- Interfaces de Usuario.
- Diagramas de Casos de Uso [20].

## **Producción**

La fase de producción incluye la implementación real. Se divide en:

- Los días de planificación: tienen como objetivo analizar, mejorar y priorizar los requisitos, planificar los contenidos de la iteración actual y preparar los casos de prueba de aceptación que se usará el día del lanzamiento.

- Los días laborables: implementan funcionalidades en el desarrollo guiado por pruebas de software (TDD).
- Los días de lanzamiento: se lanza una versión funcional para pruebas de aceptación de cliente utilizando los casos de prueba desarrollados durante los días de planificación. Además del modelo informativo y de vista de usuario, durante la producción de la fase, el equipo de desarrollo utiliza el caso de uso, el componente, el diagrama de clase de la actividad y secuencia, así como el patrón MVC para llevar la implementación y alinear la comprensión de los miembros del equipo [20].

### **Estabilización**

En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema.

### **Pruebas del Sistema**

Su propósito es que la aplicación sea estable y funcional para que los clientes la utilicen. La aplicación una vez terminada se la integra y la prueba en base a los requisitos del cliente y suprimen todos los errores descubiertos [20].

## **1.3. Objetivos**

### **1.3.1. Objetivo General**

Desarrollar una aplicación móvil multiplataforma para el control y registro de tratamientos médicos.

### **1.3.2. Objetivos Específicos**

- Establecer un análisis sobre la toma de medicamentos en aspectos como dosis y horarios bajo un asesoramiento médico.
- Investigar diferentes frameworks que permita desarrollar aplicaciones multiplataforma.
- Implementar una aplicación móvil multiplataforma para el control y registro de tratamientos médicos en una empresa médica.

## CAPITULO II

### METODOLOGÍA

#### 2.1 Materiales

La presente investigación se sirvió de la utilización de dos instrumentos para la recopilación de información, en primer lugar, se utilizó una encuesta cuyo objetivo fue verificar la aceptabilidad que tendría el desarrollo de una aplicación móvil para el control y registro de los tratamientos médicos.

En segunda instancia se empleó una entrevista realizada al Dr. Manuel Cruz en la que se obtuvieron pautas técnicas para el desarrollo del aplicativo móvil

<b>ENTREVISTA</b>	
<b>N°</b>	<b>Preguntas</b>
<b>1</b>	¿Cuántas consultas promedio realiza usted al día?
<b>2</b>	¿Cómo lleva usted el control y monitoreo de sus pacientes?
<b>3</b>	¿Le interesaría usar una aplicación móvil que lleve el registro de medicación y horarios recetados a sus pacientes?
<b>4</b>	¿Alguno de sus pacientes ha tenido problemas de salud causado por un suministro erróneo de medicamentos?
<b>5</b>	¿Cree usted que traería beneficios la implementación de una aplicación móvil para el control y registro de tratamientos médicos en la pandemia actual o similar?
<b>6</b>	¿Bajo qué criterios usted suministra los medicamentos a sus pacientes, incluidos la cantidad y el tiempo en que estos deben ser ingeridos?
<b>7</b>	¿Cuál grupo de edad considera que descuida más las indicaciones prescritas en la receta?
<b>8</b>	En su experiencia, ¿Cuáles serían las ventajas que esta aplicación traería para los profesionales de la salud y sus pacientes?
<b>9</b>	¿Existen pacientes que vuelven a pedir la receta por perdida?

*Tabla 1: Formato de entrevista*

*Elaborado por: El investigador*

## ENCUESTA

### Pregunta 1

¿En qué rango de edad se encuentra usted?

- De 18 a 25 años
- De 26 a 35 años
- De 36 a 45 años
- De 46 a 55 años
- De 56 a 65 años
- Más de 65 años

### Pregunta 2

¿Dispone usted de un teléfono inteligente?

- Si
- No

### Pregunta 3

Si su respuesta anterior fue afirmativa. ¿Cuál es el sistema operativo de su dispositivo móvil?

- Android
- IOS
- Otro (especifique)

### Pregunta 4

¿De dónde acostumbra a descargar las aplicaciones?

- Google Play
- App Store (Apple)
- Windows Phone Store
- Amazon Appstore
- Otro (especifique)

### **Pregunta 5**

¿Qué tan probable es que usted aprenda a manejar una aplicación móvil que le ayude a controlar y cuidar su salud?

- Muy probable
- Poco probable
- Nada probable

### **Pregunta 6**

¿Qué grado de dificultad tiene usted para recordar el nombre de los medicamentos prescritos por el médico?

- Alto
- Medio
- Bajo
- Ninguno

### **Pregunta 7**

¿Ha utilizado alguna aplicación que le permita llevar registro de sus horarios en la que debe ingerir sus medicamentos?

- Si
- No

### **Pregunta 8**

¿Con que facilidad usted recuerda la dosificación de los medicamentos prescritos por el médico?

- Alta
- Media
- Baja

### **Pregunta 9**

¿De qué forma preferiría recibir los avisos y/o recordatorios de su aplicativo móvil?

- Mensajes de texto (SMS)



- Correo electrónico
- Notificación del dispositivo móvil
- Otro (especifique)

### **Pregunta 10**

¿Le gustaría que su receta se encuentre en su dispositivo móvil?

- Si
- No

## **2.2 Métodos**

### **2.2.1 Modalidad de la Investigación**

La presente investigación se realizó mediante las siguientes modalidades:

- **Investigación bibliográfica:** La investigación será bibliográfica debido a que se tomará como apoyo a nuestra investigación el uso de libros, revistas, revistas digitales, papers, tesis del área informática entre otros, para la elaboración del marco teórico sobre el control y registro de tratamientos médicos basado en aplicaciones móviles multiplataforma.
- **Investigación de campo:** La investigación será de campo debido a que se interactuará con las personas que necesiten ayuda para recordar la hora de ingerir sus medicamentos mediante una app que ayude a este propósito.
- **Investigación experimental:** La investigación será experimental debido a que se llevará pruebas de funcionamiento del aplicativo móvil.

### **2.3 Población y Muestra**

Para la selección de la población y muestra del presente proyecto se tomarán en cuenta a los pacientes y al doctor del Centro Médico Familiar; por medio de instrumentos para recabar datos como la entrevista y la encuesta.

**Población:** Para la recolección de datos mediante la encuesta la población está representada por los pacientes que se atendieron durante la semana del 12 al 16 de octubre en el Centro Médico Familiar del Dr. Manuel Cruz, obteniendo una población total de 57 pacientes en dicha semana. Mientras que la recolección de información mediante la entrevista será realizada al Dr. Manuel Cruz, quien está a cargo del Centro Médico

Familiar, teniendo una población de 1 para el instrumento actual.

**Muestra:** Al realizar encuestas sobre la atención médica para conocer el grado de satisfacción de los pacientes o realizar preguntas que no estén relacionadas con problemas de salud, un tamaño de muestra estadísticamente significativo no es tan importante; pudiendo obtener información valiosa sobre las necesidades y experiencias de los pacientes con una muestra moderada [21].

Al desarrollar el presente proyecto, se recopiló la información a través de encuestas con la ayuda de Google Form, para llevar a cabo un análisis de la información dada por los pacientes y sus expectativas para la aplicación móvil.

La encuesta se realizó a una muestra de 15 pacientes del Centro Médico Familiar.

*Tabla 2. Muestra*

<b>N°</b>	<b>Descripción de la muestra</b>	<b>Cantidad</b>	<b>Porcentaje</b>
1	Doctor	1	6,25 %
2	Pacientes	15	93,75 %
Total		16	100 %

*Elaborado por: El investigador*

## Resultado de la Entrevista

Al realizar la entrevista al r. Manuel Cruz se obtuvo los siguientes resultados:

ENTREVISTA		
Pregunta	Respuesta	Conclusión
1. ¿Cuántas consultas promedio realiza usted al día?	El promedio de consultas diarias en el consultorio es de 6 pacientes y a domicilio es de 4 a 5 pacientes.	La mayoría de pacientes atendidos por el Dr. Manuel Cruz, pertenecen al grupo de “adultos mayores”, los cuales son polifármacos debido a la gran cantidad de medicamentos que ingieren a causa del deterioro cognitivo provocado por su edad avanzada.  Los médicos son profesionales que trabajan con sus pacientes conjunto de normas y reglamentos, tanto para los controles como para recetar un medicamento; estas normas
2. ¿Cómo lleva usted el control y monitoreo de sus pacientes?	El control se lleva a través de un historial clínico como el del Ministerio de Salud Pública.	
3. ¿Le interesaría usar una aplicación móvil que lleve el registro de medicación y horarios recetados a sus pacientes?	Si, para poder realizar un historial de la medicación recetada a los pacientes en cualquier momento.	
4. ¿Alguno de sus pacientes ha tenido problemas de salud causado por un suministro erróneo de medicamentos?	Por ejemplo, los pacientes geriátricos son polifármacos, por lo que suelen confundir la medicación.	
5. ¿Cree usted que traería beneficios la implementación de una aplicación móvil	Esta aplicación no solo sería útil en la pandemia; cualquier ayuda tecnológica que recuerde horarios y fechas de medicación es muy beneficiosa.	

para el control y registro de tratamientos médicos en la pandemia actual o similar?		pueden ser establecidas bajo parámetros nacionales y/o internacionales dependiendo el caso.
6. ¿Bajo qué criterios usted suministra los medicamentos a sus pacientes, incluidos la cantidad y el tiempo en que estos deben ser ingeridos?	Esta aplicación no solo sería útil en la pandemia; cualquier ayuda tecnológica que recuerde horarios y fechas de medicación es muy beneficiosa.	
7. ¿Cuál grupo de edad considera que descuida más las indicaciones prescritas en la receta?	Los adultos, porque ellos al poco tiempo de sentirse bien abandonan el tratamiento, muchas veces este queda a la mitad.	
8. En su experiencia, ¿cuáles serían las ventajas que esta aplicación traería para los profesionales de la salud y sus pacientes?	Con el consumo correcto de medicamentos se evitaría la resistencia a fármacos, en los adultos; y complicaciones propias de sus patologías en el caso de adultos mayores.	
9. ¿Existen pacientes que vuelven a pedir la receta por perdida?	Principalmente esto se da en adultos y en adultos mayores, razón que exige recurrir al expediente clínico.	

*Tabla 3: Resultados de la Entrevista*

*Elaborado por: El investigador*

### Pregunta 1:

¿En qué rango de edad se encuentra usted?

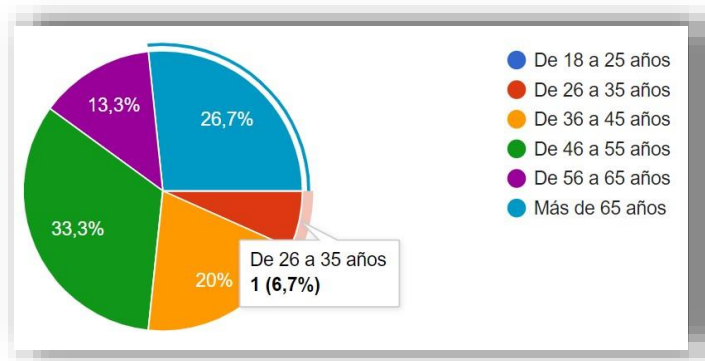


Ilustración 6. Pregunta 1.

Elaborado por: El investigador

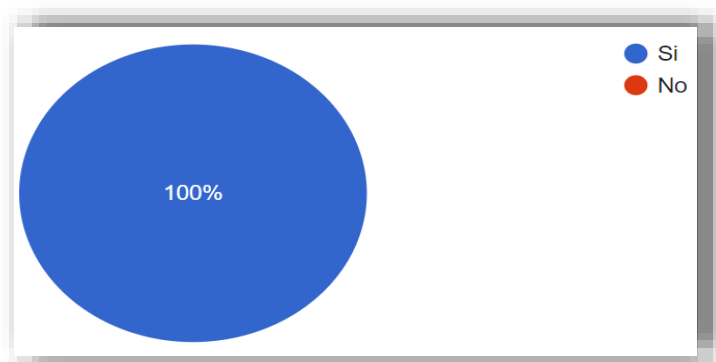
### Análisis e Interpretación:

De una muestra de 15 personas, se obtuvo que el 33.3% de los pacientes que asistieron al centro médico familiar donde labora el doctor Manuel Cruz están dentro del rango de edad comprendido entre los 46 a 55 años, siendo este el grupo mayoritario; seguidamente con un 26.7% están los pacientes cuyas edades superan los 65 años; finalmente el otro grupo de edades más representativo es el de 36 a 45 años con un 20%. Los otros 3 grupos minoritarios en años son los de 56 a 65, de 26 a 35 y 18 a 25 con el 13.3%, 6.7% y 0% respectivamente.

Los datos indican que las personas que más requieren de una atención o servicio médico son aquellas con edades superiores a los 35 años.

### Pregunta 2:

¿Dispone usted de un teléfono inteligente?



*Ilustración 7. Pregunta 2*

*Elaborado por: El investigador*

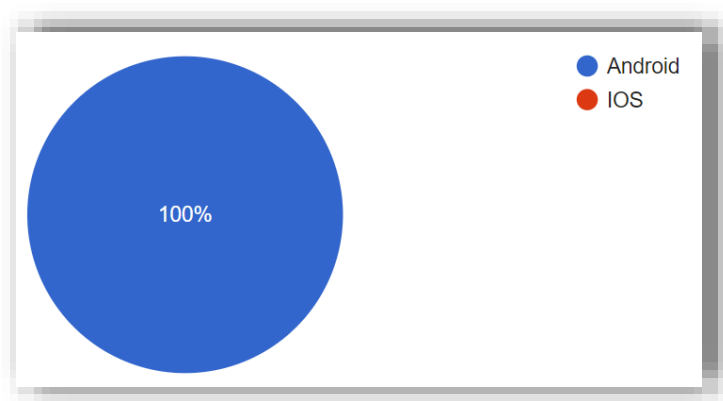
### **Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 100% de los pacientes posee un teléfono inteligente.

Los porcentajes obtenidos indican que en la actualidad la gran mayoría de personas disponen y usan dispositivos inteligentes, lo cual también es un indicativo del conocimiento de utilización de al menos 1 aplicación móvil, lo cual facilita el manejo de la aplicación móvil propuesta en el presente trabajo.

### **Pregunta 3:**

Si su respuesta anterior fue afirmativa ¿Cuál es el sistema operativo de su dispositivo móvil?



*Ilustración 8. Pregunta 3*

*Elaborado por: El investigador*

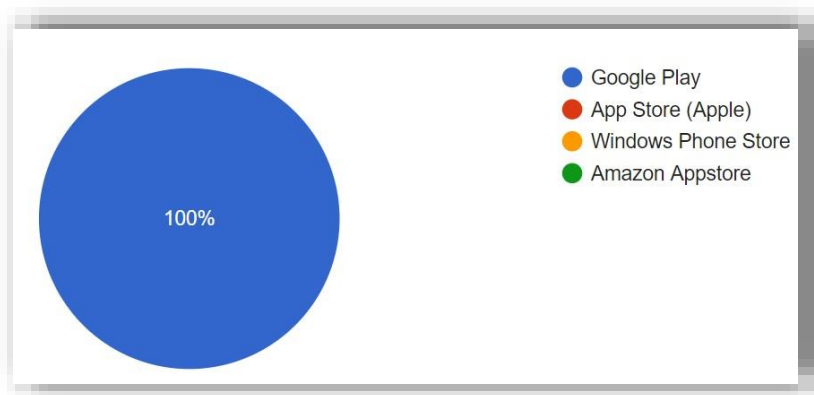
### **Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 100% de pacientes utilizan dispositivos inteligentes con sistema operativo Android.

Los porcentajes obtenidos indican que la mayoría de personas sin importar su rango de edad prefiere utilizar dispositivos inteligentes con el sistema operativo Android posiblemente por el tema de costos y su interfaz amigable.

### **Pregunta 4:**

¿De dónde acostumbra descargar las aplicaciones?



*Ilustración 9. Pregunta 4*

*Elaborado por: El investigador*

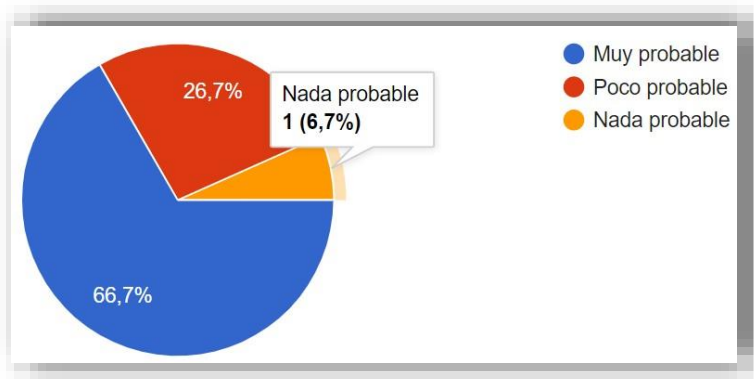
### **Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 100% de pacientes descarga sus aplicaciones en la tienda de Google Play Store.

Los porcentajes obtenidos en la presente pregunta son un complemento a los porcentajes obtenidos a la anterior pregunta, ya que los dispositivos móviles con sistema operativo Android usan la app Google Play Store para la descarga de sus aplicaciones.

### **Pregunta 5:**

¿Qué tan probable es que usted aprenda a manejar una aplicación móvil que le ayude a controlar y cuidar su salud?



*Ilustración 10. Pregunta 5  
Elaborado por: El investigador*

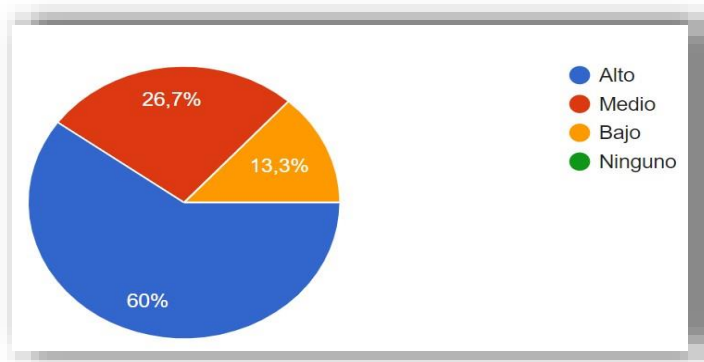
**Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, el 66.7% de los pacientes es muy probable que aprenda a manejar la aplicación móvil, mientras que el 26.7% afirma que es poco probable y un 6.7% indica que es nada probable.

Los porcentajes obtenidos muestran que la mayoría de personas encuestadas están interesadas en utilizar una app en beneficio de su salud.

**Pregunta 6:**

¿Qué grado de dificultad tiene usted para recordar el nombre de los medicamentos prescritos por el médico?



*Ilustración 11. Pregunta 6  
Elaborado por: El investigador*

**Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, el 60% de los pacientes tiene un grado de dificultad alto para recordar el nombre de sus medicamentos prescritos

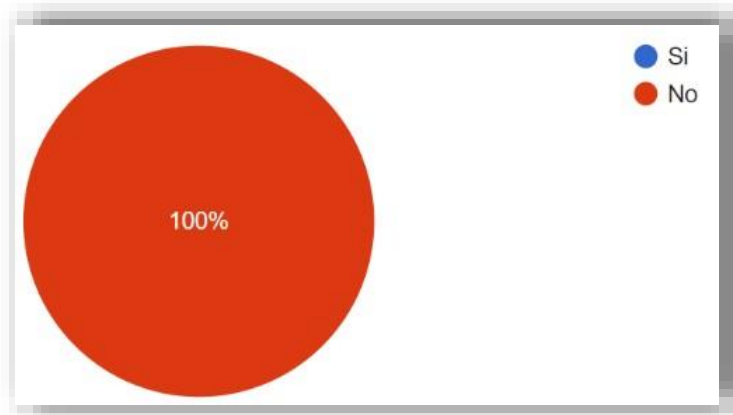


por el médico, mientras que el 26.7% tiene un grado de dificultad medio para recordar los nombres y el 13.3% indica que tiene una baja dificultad.

Los porcentajes obtenidos muestran que más del 50% de las personas encuestadas tienen problemas para recordar los nombres de los medicamentos prescritos por su médico.

**Pregunta 7:**

¿Ha utilizado alguna aplicación que le permita llevar registro de sus horarios en la que debe ingerir sus medicamentos?



*Ilustración 12. Pregunta 7*

*Elaborado por: El investigador*

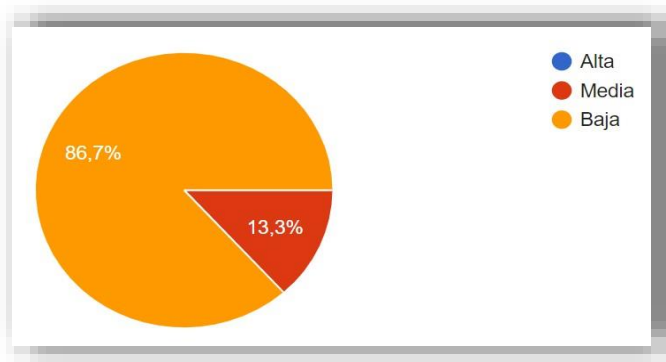
**Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 100% de pacientes no han utilizado alguna aplicación que le permita llevar registros de los horarios en los que debe ingerir sus medicamentos.

Los porcentajes obtenidos demuestran que las personas encuestadas no tienen experiencia utilizando el tipo de aplicación propuesta en el presente proyecto, lo cual da la oportunidad de enseñar los usos de esta app.

**Pregunta 8:**

¿Con que facilidad usted recuerda la dosificación de los medicamentos prescritos por el médico?



*Ilustración 13. Pregunta 8*

*Elaborado por: El investigador*

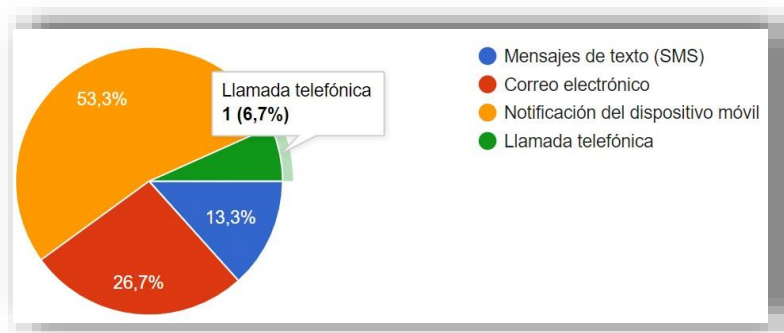
### **Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 86.7% de los pacientes tienen un bajo grado de facilidad para recordar la dosificación de sus medicamentos, mientras que el 13.3% indica que tiene un grado de facilidad medio.

Los porcentajes obtenidos muestran que la mayoría de personas poseen dificultad para recordar las dosis de sus medicinas; por lo que en complemento con la pregunta 7 se puede observar que el problema en recordar las indicaciones médicas recae más en la dosificación.

### **Pregunta 9:**

¿De qué forma preferiría recibir los avisos y/o recordatorios de su aplicativo móvil?



*Ilustración 14. Pregunta 9*

*Elaborado por: El investigador*

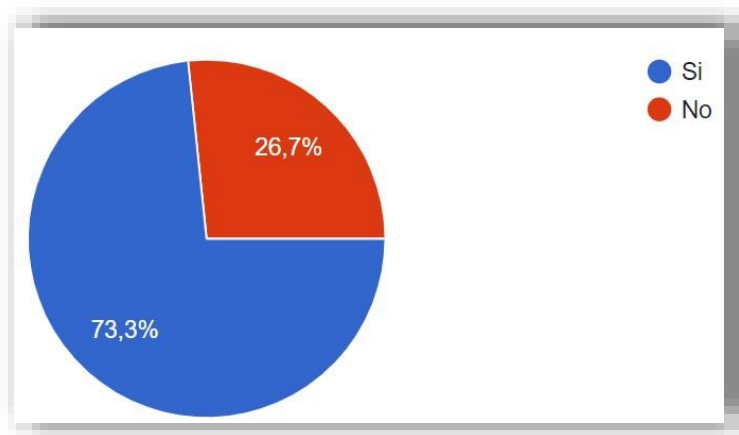
### **Análisis e Interpretación:**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 53.3% de los pacientes prefieren recibir los avisos y/o recordatorios a través de notificaciones; seguidamente del 26.7% que señala su preferencia por correo electrónico; mientras que el 13.3% optó por mensajes de texto y el 6.7% prefiere llamada telefónica.

Los porcentajes obtenidos muestran que más de la mitad de personas encuestadas se siente cómoda con recibir las alertas a través de notificaciones, lo cual es un claro indicio de como las personas usan dispositivos inteligentes en su vida diaria.

### **Pregunta 10:**

¿Le gustaría que su receta se encuentre en una aplicación móvil?



*Ilustración 15. Pregunta 10*

*Elaborado por: El investigador*

### **Análisis e Interpretación**

De una muestra de 15 personas con diferentes rangos de edad, se obtuvo que el 73.3% prefiere que su receta se encuentre en una app, mientras que el 26.7% está en desacuerdo.

Los porcentajes obtenidos indican claramente la preferencia de las personas por tener respaldada su prescripción médica en un dispositivo móvil, específicamente en una aplicación destinada a esta tarea.

## **2.4 Procesamiento y Análisis de Datos**

Una vez procesada la información obtenida en las encuestas y la entrevista, se evidencian las dificultades que presentan tanto el médico como los pacientes, a la hora de prescribir y tomar los medicamentos, respectivamente; razón por la cual el análisis correspondiente constata la necesidad de que se diseñe un aplicativo móvil que ayude en la medicación. Los datos de la entrevista junto con los resultados obtenidos en las encuestas permitirán la realización del aplicativo, enfocado principalmente a los requerimientos más importantes obtenidos de los análisis.

## CAPITULO III

### RESULTADOS Y DISCUSIÓN

#### 3.1 Análisis y discusión de los resultados

##### 3.1.1 Análisis de medicamentos más comunes, prescritos por el médico

Los datos que se presentan son el resultado de una investigación en colaboración con el Dr. Manuel Cruz, llevado a cabo en pacientes durante el mes de junio del 2021 y divididos en grupos de edad como los utilizados en las encuestas; los medicamentos indicados en las siguientes tablas, corresponden a las recetas o tratamientos de las enfermedades más comunes que acontecieron los pacientes en dicho mes.

Tabla 4: Análisis de medicamentos de Pacientes con más de 65 años

<b>Pacientes con más de 65 años</b>			
<b>Medicamento</b>	<b>Dosis por día</b>	<b>Horario</b>	<b>Tratamiento</b>
Sertralina	1 vez por día	Antes de la merienda	Demencia senil y depresión
Tamsulosina	1 vez por día	Antes de la merienda	Próstata
Ibuprofeno	3 veces al día	Después de cada comida	Dolor causado por la artritis

Elaborado por: El investigador

Tabla 5: Análisis de Medicamentos de Pacientes de 56 a 65 años

<b>Pacientes de 56 a 65 años</b>			
<b>Medicamento</b>	<b>Dosis por día</b>	<b>Horario</b>	<b>Tratamiento</b>
Tamsulosina	1 vez por día	Antes de la merienda	Próstata
Levofloxacina	1 vez por día	Antes de la merienda	Vías urinarias
Diclofenaco	3 veces al día	Antes de cada comida	
Metformina	2 veces al día	Antes del desayuno y merienda	Diabetes

Elaborado por: El investigador

Tabla 6: Análisis de Medicamentos de Pacientes de 46 a 55 años

<b>Pacientes de 46 a 55 años</b>			
<b>Medicamento</b>	<b>Dosis por día</b>	<b>Horario</b>	<b>Tratamiento</b>
Losartán	1 vez por día	Antes del desayuno	Hipertensión arterial
Amlodipino	1 vez por día	Antes de la merienda	
Lansoprazol	1 vez por día	Antes de la merienda	Gastritis crónica
Metoclopramida	3 veces al día	Antes de cada comida	Vaciado gástrico

Elaborado por: El investigador

Tabla 7: Análisis de Medicamentos de Pacientes de 36 a 45 años

<b>Pacientes de 36 a 45 años</b>			
<b>Medicamento</b>	<b>Dosis por día</b>	<b>Horario</b>	<b>Tratamiento</b>
Prednisolona	1 vez por día	Antes de la merienda	Rinitis crónica
Nitrofurantoína	2 veces al día	Antes del desayuno Antes de la merienda	Cistitis
Enalapril	1 vez por día	Antes del desayuno	Presión arterial alta

Elaborado por: El investigador

Tabla 8: Análisis de Medicamentos de Pacientes de 26 a 35 años

<b>Pacientes de 26 a 35 años</b>			
<b>Medicamento</b>	<b>Dosis por día</b>	<b>Horario</b>	<b>Tratamiento</b>
Simvastatina	1 vez por día	Antes de la merienda	Colesterol alto
Aciclovir	3 veces al día	Antes de cada comida	Herpes
Paroxetina	1 vez por día	Después de la merienda	Ansiedad Social

Elaborado por: El investigador

Analizando las tablas 4, 5, 6, 7, 8; se obtiene que la mayoría de los medicamentos se consumen 1 vez por día, habiendo casos en que se puede tomar hasta 3 veces en un mismo día; también se puede constatar que la mayor parte de los medicamentos se consumen previo a la comida, más comúnmente antes de la merienda; además, cada tratamiento será diferente uno del otro, dependiendo del diagnóstico médico, donde algunas enfermedades presentan más de un medicamento por recomendación médica.

Los resultados en esta investigación son importantes, ya que servirán para establecer los lineamientos y parámetros necesarios para el diseño del aplicativo móvil, tales como los medicamentos para determinados tratamientos, incluido la dosificación y horarios de estos.

### **3.1.2 Comparación de los tipos de frameworks en el desarrollo móvil multiplataforma**

Para determinar la mejor tecnología en el desarrollo d una aplicación multiplataforma se debe tomar en cuenta aspectos como rendimiento, lenguaje, compatibilidad, entre otros. Por consiguiente, en la tabla 5 se expone un cuadro comparativo de las características entre las tecnologías más utilizadas para el desarrollo multiplataforma.

Tabla 9: Cuadro comparativo entre Flutter, Ionic y Xamarin

Características	Flutter	Ionic	Xamarin
<b>Lenguaje</b>	Dart	HTML, CSS, JavaScript, Angular, React Native, Vue.js	C#
<b>Rendimiento</b>	Alto	Medio	Alto
<b>Reusabilidad</b>	Medio	Alto	Alto
<b>Manejo de gráficas pesadas</b>	Bajo	Alto	Alto
<b>Ejecución</b>	Motor de gráfico personalizado	Navegador web y Emulador de Android SDK	Emulador d Android SDK
<b>Comunidad</b>	En constante crecimiento	Popular en los desarrolladores	Popular en los desarrolladores
<b>Interfaz</b>	Componentes propios	HTML, CSS	Componentes nativos
<b>Plataformas Compatibles</b>	Android, iOS, Windows, macOS y Linux	Android, iOS, Windows, web y PWA	Android, IOS, Windows
<b>Precio</b>	Open Source	Open Source y para uso comercial	Open Source
<b>Facilidad de aprendizaje</b>	Bajo	Alto	Medio
<b>Compatibilidad de las librerías para notificaciones</b>	Flutter_local_notificatio n, firebase_Messaging	Local_notificatio n, OneSignal	INotificationManag er

Elaborado por: El investigador

De acuerdo con los datos de la tabla 10, se decide utilizar la tecnología Ionic, gracias a las características y componentes que ofrece el mismo para desarrollar una aplicación

multiplataforma de manera eficiente y rápida; ya que, en el apartado de implementación de notificaciones, Ionic posee librerías o plugins que contiene un servicio para enviar notificaciones Push fuera de Google y Firebase; además, al tener una alta facilidad de aprendizaje, brinda un manejo más sencillo del framework volviéndolo popular entre los desarrolladores.

### 3.1.3 Análisis comparativo de las metodologías Scrum, Extreme Programming (XP) y Mobile-D

Para determinar la metodología de trabajo, se procede a realizar un cuadro comparativo entre las metodologías ágiles más comunes en el desarrollo de aplicaciones híbridas.

Tabla 10 :Análisis Comparativo de las metodologías

	Scrum	Programación Extrema (XP)	Mobile-D
Tamaño de Equipo	El equipo debe estar conformado de 5 a 9 personas	El equipo debe estar conformado de 2 a 10 personas	El equipo debe estar conformado de 10 personas o menos
Fases	Planificación de sprint Reunión del equipo Scrum Refinamiento del backlog Revisión del sprint Retrospectiva del sprint	Planificación Diseño Codificación Pruebas Lanzamiento	Exploración Inicialización Producción Estabilización Pruebas
Comunicación	El Product Owner es el responsable de gestionar la comunicación entre el cliente y el equipo de trabajo con el fin de minimizar el riesgo en caso de malentendidos durante la fase de desarrollo	La comunicación debe ser constante entre el cliente y el equipo de desarrollo	El jefe del proyecto es el encargado de gestionar la comunicación entre el cliente y el equipo de trabajo para lograr un mejor desarrollo de la aplicación
Correcciones del aplicativo	Al carecer de tiempo límite, concede solicitudes de cambio al usuario de forma continua.	El grupo de trabajo mantiene el orden de prioridad solicitado por el cliente, sin hacer	Impulsa el descubrimiento de errores y correcciones en cortos períodos.



		cambios en la iteración.	
Entregas de prototipos	Requiere de grandes y disciplinados equipos de desarrolladores.	Utiliza cronogramas para la entrega de prototipos.	Favorece la entrega de prototipos en cortos tiempos.

Elaborado por: El investigador

De acuerdo con los resultados en la tabla 11, en el presente proyecto se utilizará la metodología Mobile-D, ya que, no requiere equipos de muchas personas para el desarrollo del proyecto; además, cuenta con un orden marcado que facilitará la detección de errores en cortos periodos de tiempo, y a su vez permitirá la entrega de prototipos de calidad; sin descuidar la interacción entre el equipo de trabajo con el cliente.

### 3.2 Desarrollo de la propuesta

Las fases que se aplicarán son:

- Exploración
- Inicialización
- Producción
- Estabilización
- Pruebas

#### 3.2.1 Fase de Exploración

##### 3.2.1.1 Levantamiento de Requerimientos

Se realizó una reunión con el médico, el cual explicó cómo lleva su registro de pacientes, en función del diagnóstico y el medicamento correspondiente, con el fin de obtener los principales parámetros para el desarrollo del aplicativo móvil, considerando las necesidades tanto del doctor como del paciente.

A continuación, se presentan enlistados los requerimientos principales para el desarrollo del aplicativo móvil.

- Desarrollo de una aplicación móvil para el control y registro de tratamientos médicos
- **Registro de Usuarios**

Al realizar el levantamiento de requerimientos los campos a crear son:

- Cédula
- Nombre
- Apellido

- Email
- Edad
- Género
- Rol
- Fotografía
- Portada
- Contraseña
- Confirmar Contraseña
- Horario de alimentación
- **Registro de medicamentos**

Al realizar el levantamiento de requerimientos los campos a crear son:

  - Nombre del medicamento
  - Descripción
  - estado
- **Visualización de recetas por parte de médico y paciente**

Al realizar el levantamiento de requerimientos los campos a crear son:

  - Nombre del médico de quien receta
  - Fecha
  - Nombre del medicamento
  - Días de tratamiento
  - Tiempo
  - Indicaciones
- **Visualización del historial de pacientes**
  - Buscador de paciente por apellido o cédula
  - Recetas creadas por pacientes
- **Creación de una alarma personalizada**

Al realizar el levantamiento de requerimientos los campos a crear son:

  - Agregar Medicamento
  - Días de tratamiento
  - Tiempo
  - Indicaciones
  - Estado

- **Generar notificaciones de los medicamentos prescritos por el médico**

Al realizar el levantamiento de requerimientos la notificación mostrara el medicamento que debe ingerir el paciente.

- La aplicación móvil debe ser de fácil uso para personas familiarizadas con la tecnología
- Instalación del backend
- Instalación de librerías para el desarrollo del proyecto.
- Desarrollo del proyecto en Ionic
- Desarrollo de una Api Rest en Laravel
- Autenticación utilizando Tokens
- Utilización del plugin Local-notification
- Contratación de un hosting

### 3.2.1.2 Establecimiento de grupo de interés

Para esta investigación se tomó como grupo de interés a las personas residentes del Cantón Ambato que asisten en calidad de pacientes al centro médico familiar quienes tienen un rango de edad entre 18 a mayores de 65 años. Estas personas son chequeadas por el Doctor Manuel Cruz, quien evalúa la condición médica de los pacientes, llevando un registro de las medicinas prescritas a cada una de ellas y facilitando el desarrollo práctico de este estudio.

### 3.2.1.3 Planificación del proyecto

#### Definición de usuarios

Tabla 11: Definición de Usuarios

Tipo de Usuario	Tareas	Módulos
Administrador	Creación de Doctores	Aplicativo móvil
Doctor	Creación de Medicamentos Recetar medicamentos a pacientes. Revisar historial de Pacientes. Eliminar Receta Médica.	Aplicativo móvil

	Editar datos personales Búsqueda de Pacientes	
Usuario	Creación de cuenta Búsqueda de recetas Creación de alarma personalizada	Aplicativo móvil

Elaborado por: El investigador

### Selección de entorno

- Instalación del editor de código
- Instalación del entorno de desarrollo
- Postman para verificar si se está obteniendo la información de la API.
- Instalación de Laravel para el desarrollo de la API.
- Contratación del dominio y hosting
- Creación de Base de Datos
- Uso de Angular para el consumo del API.
- Uso de Ionic para el desarrollo del aplicativo móvil multiplataforma.
- Laptop Dell, procesador Intel Core i7 con 8GB de RAM.
- Teléfono Samsung con Android.
- Android Studio para la creación de la Apk.

### 3.2.2 Fase de Inicialización

#### 3.2.2.1 Herramientas para iniciar la configuración del entorno

Para dar inicio al desarrollo del proyecto, se llevaron a cabo un conjunto de procesos enfocados en la configuración del entorno, donde se realizará la instalación de una serie de herramientas, las cuales se detallan a continuación.

## Instalación de Angular

Primeramente, se instalará el editor de código, como requisito para la instalación de Angular; en el presente proyecto se utilizará Visual Studio Code.

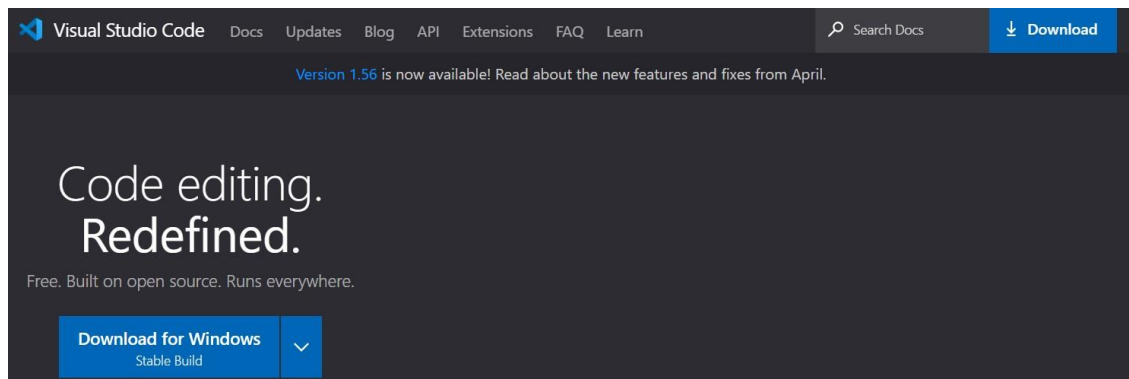


Ilustración 16: Descarga de Visual Studio Code

Elaborado por: El investigador

A continuación, se accederá al siguiente enlace <https://nodejs.org/es/>; para descargar el entorno de ejecución de JavaScript.



Ilustración 17: Descarga de node.js

Elaborado por: El investigador

Para iniciar la instalación de Angular, en la terminal se ejecutará el siguiente comando:

```
npm install -g @angular/cli
```

### **Instalación de Ionic y Cordova**

En la instalación de Ionic y Cordova, se deberá ingresar las siguientes líneas de comando en la terminal:

```
npm install .g cordova
```

```
npm install -g @ionic/cli
```

Realizado todo esto, se tendrán instaladas las herramientas necesarias para la creación de proyectos en Ionic, y con ello la creación de aplicaciones móviles y/o de escritorio, de alta calidad y rendimiento, mediante la utilización de tecnologías web con integraciones para marcos populares como: Angular, React y Vue.

### **Postman**

Esta herramienta es importante, ya que realiza las pruebas del API Rest, a la vez que obtiene todos los tipos de peticiones, como:

- http
- get
- post
- put
- delete

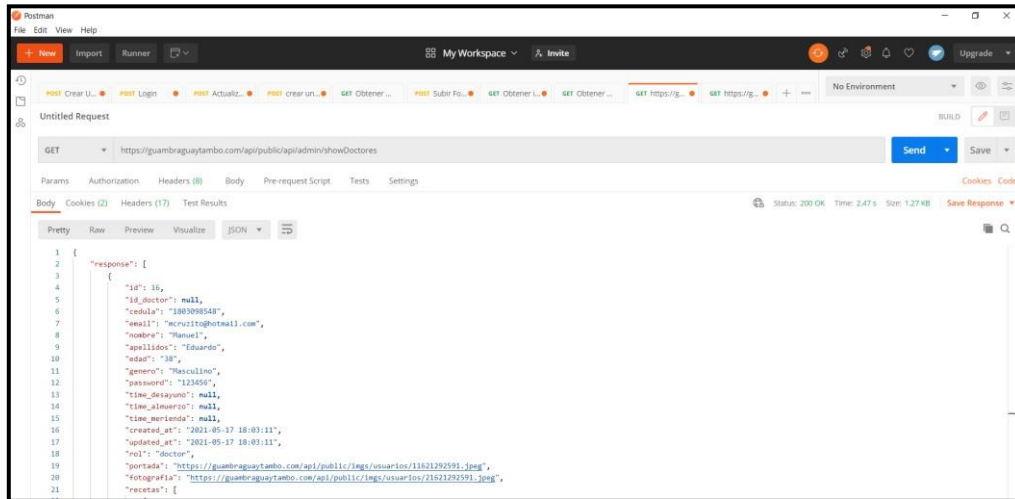


Ilustración 18: Ejecución Postman

Elaborado por: El investigador

## Laravel

Previo a la creación del API Rest se debe instalar Laravel con la versión 7, para lo cual es necesario tener el composer, ya que esta herramienta de gestión de dependencias del php, y ayudará en la instalación de las librerías indispensables para este proyecto.

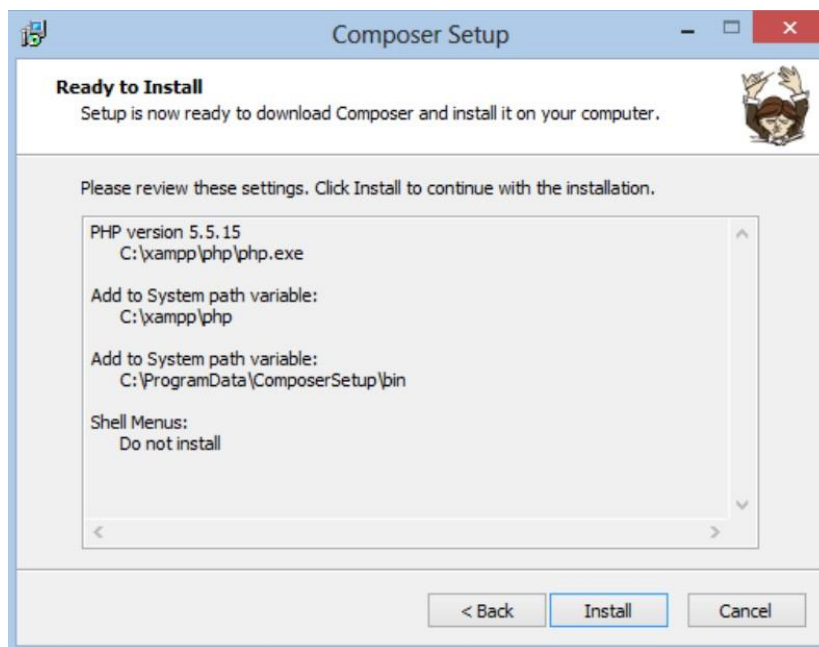
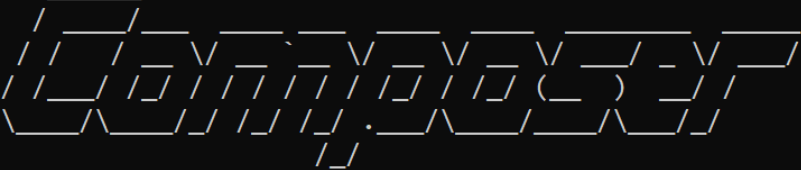


Ilustración 19: Instalación de composer

Elaborado por: El investigador

```
C:\Users\DELL 2018>composer -v  
  
Composer version 1.10.10 2020-08-03 11:35:19
```

*Ilustración 20: Comprobación de la instalación de composer*

*Elaborado por: El investigador*

### 3.2.3 Fase de Producción

A continuación, se creará un nuevo proyecto con el comando:

```
composer global require laravel/installer
```

---



*Ilustración 21: Instalación laravel*

*Elaborado por: El investigador*



## laravel new api

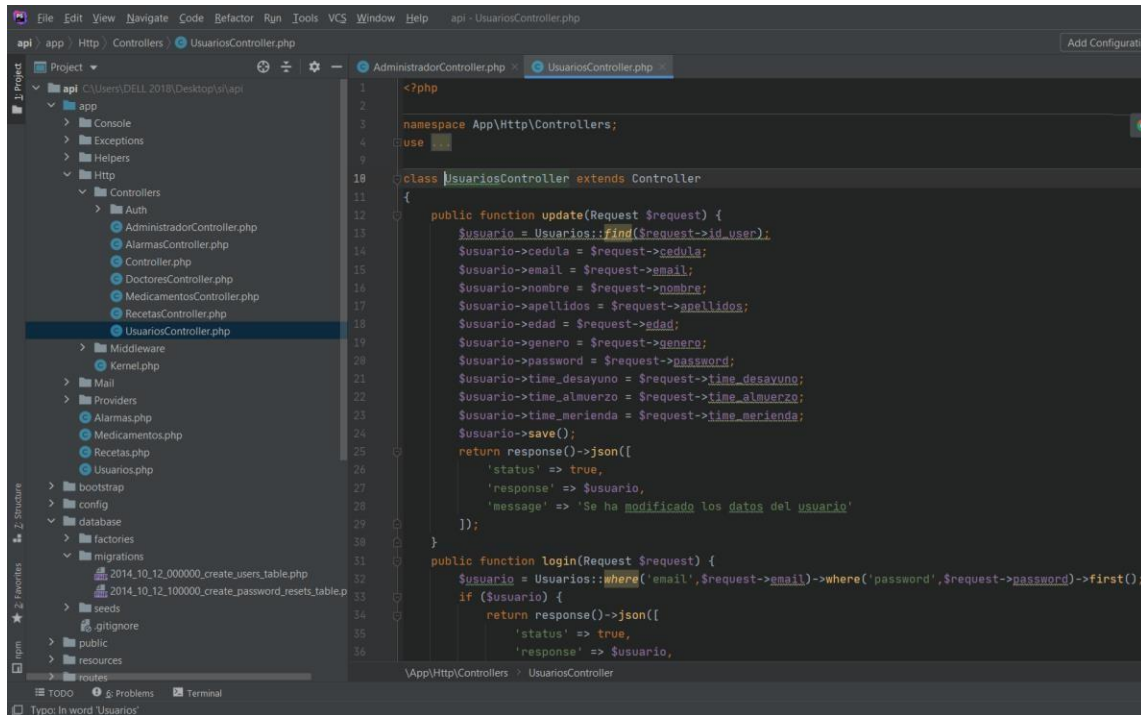


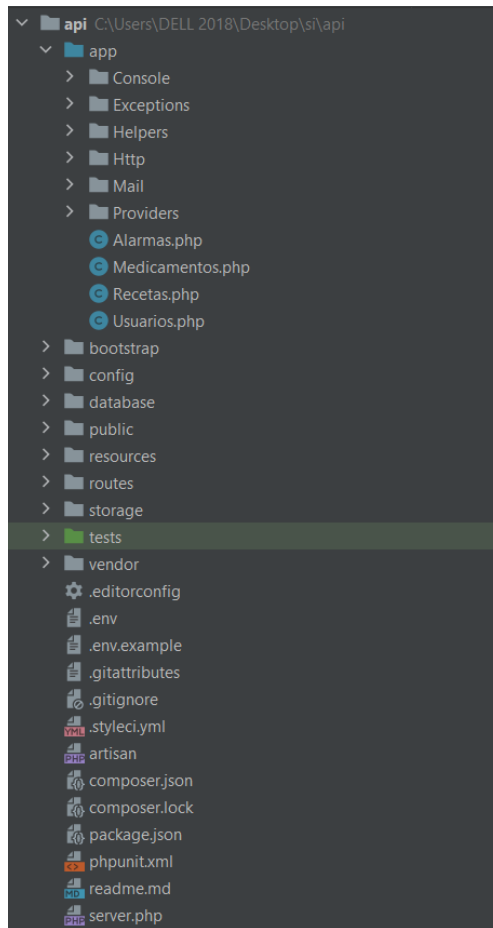
Ilustración 22: Proyecto en Laravel

Elaborado por: El investigador

### 3.2.3.1 Creación del API Rest en Laravel

Al crear el API Rest en Laravel, se observa que la ilustración 22, en el inicio del proyecto, se obtiene una estructura general, donde contiene:

- app: Contiene los Modelos
- resources/views: Contiene las vistas
- app/Http/Controllers: Contiene los controladores
- routes: Donde se definirá las rutas



*Ilustración 23: Estructura general de laravel*

*Elaborado por el investigador*

### **3.2.3.2 Creación de modelos en Laravel**

Laravel utiliza el modelo Eloquent Object Relational Mapping (ORM), que sirve para transformar los datos de un sistema orientado a objetos, a una base de datos; además, contiene un patrón de diseño llamado Active Record para el almacenamiento de información en la base de datos.

Para crear una plantilla para cada modelo es utilizando PHP artisan donde nos proporcionará la mayor parte de comandos que nos ayudará desarrollar aplicaciones como por ejemplo generará tablas a nuestra base de datos MySQL además de rellenar datos en las tablas automáticamente.

Para ello se deberá crear el modelo desde la consola con el comando

- *Php artisan make\_model Usuarios*

```
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;

class Usuarios extends Authenticatable
{
    use Notifiable;
    protected $table = "usuarios";
    protected $fillable = ['cedula','fotografia','portada','email','nombre','apellidos','rol','edad','genero','password','time_desayuno',
        'time_almuerzo','time_merienda'];
}
```

*Ilustración 24: Creación del modelo Usuarios*

*Elaborado por el investigador*

- *Php artisan make\_model Medicamentos*

```
<?php

namespace App;
use Illuminate\Database\Eloquent\Model;

class Medicamentos extends Model
{
    protected $table = "medicamentos";
    protected $fillable = ['nombre','descripcion','estado'];
}
```

*Ilustración 25: Creación del modelo Medicamentos*

*Elaborado por el investigador*

- *Php artisan make\_model Alarmas*

```
<?php

namespace App;
use Illuminate\Database\Eloquent\Model;

class Alarmas extends Model
{
    protected $table = "alarmas";
    protected $fillable = ['id_receta','medicamento','dias','tiempo','indicaciones','caduca_dia','cada_comida','status','created_at'];
}
```

*Ilustración 26: Creación del modelo Alarmas*

*Elaborado por el investigador*

- *Php artisan make\_model Recetas*

```
class Recetas extends Model
{
    protected $table = "recetas";
    protected $fillable = ['id_doctor', 'id_usuario', 'tipo', 'observaciones', 'creado', 'status'];
    protected $appends = ['doctor', 'usuario', 'medicamentos'];

    public function medicamentos()
    {
        return $this->hasOne( related: 'App\Alarmas', foreignKey: 'id_receta', localKey: 'id');
    }

    public function getMedicamentosAttribute()
    {
        return $this->medicamentos()->get();
    }

    public function doctor()
    {
        return $this->hasOne( related: 'App\Usuarios', foreignKey: 'id', localKey: 'id_doctor');
    }

    public function getDoctorAttribute()
    {
        return $this->doctor()->first();
    }

    public function usuario()
    {
        return $this->hasOne( related: 'App\Usuarios', foreignKey: 'id', localKey: 'id_usuario');
    }

    public function getUsuarioAttribute()
    {
        return $this->usuario()->first();
    }
}
```

*Ilustración 27: Creación del modelo Recetas*

*Elaborado por el investigador*

Una vez creado los modelos que se guardarán en la base de datos también se deberá relacionar las tablas para no tener algún inconveniente al momento de guardar los datos.

A continuación, se ejecutará el comando que insertará registros en las tablas

```
C:\Users\DELL 2018\Desktop\si\api>php artisan db:seed
```

*Ilustración 28: Comando para insertar los registros en las tablas*

*Elaborado por el investigador*

### 3.2.3.3 Creación de los Controladores

Laravel nos ayuda a crear los controladores de una forma organizada sin necesidad de definir toda la lógica de las rutas, además, permite agrupar toda la lógica de las peticiones o solicitudes dentro de una misma clase.

Para crear dichos controladores se ejecutará el comando

- `php artisan make:Controller AdministradorController`

```
C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller AdministradorController
```

*Ilustración 29: Comando para crear el Controlador Administrador*

*Elaborado por el investigador*

```
<?php
namespace App\Http\Controllers;
use ...

class AdministradorController extends Controller
{
    public function update(Request $request) {
        $usuario = Usuarios::find($request->id_user);
        $usuario->cedula = $request->cedula;
        $usuario->email = $request->email;
        $usuario->nombre = $request->nombre;
        $usuario->apellidos = $request->apellidos;
        $usuario->edad = $request->edad;
        $usuario->genero = $request->genero;
        $usuario->password = $request->password;
        $usuario->time_desayuno = $request->time_desayuno;
        $usuario->time_almuerzo = $request->time_almuerzo;
        $usuario->time_merienda = $request->time_merienda;
        $usuario->save();
        return response()->json([
            'status' => true,
            'response' => $usuario,
            'message' => 'Se ha modificado los datos del usuario'
        ]);
    }
}
```

*Ilustración 30: Métodos para el Administrador*

*Elaborado por el investigador*

```

public function addDoctor(Request $request) {
    $exist = Usuarios::where('email',$request->email)->orWhere('cedula',$request->cedula)->first();
    if ($exist) {
        return response()->json([
            'status' => false,
            'message' => 'Los datos ingresados ya estan en uso'
        ]);
    }

    $portada = base64_decode($request->portada);
    $file = 'imgs/usuarios/1'.time().'.jpeg';
    $success = file_put_contents($file, $portada);
    $portada = 'https://guambraguaytambo.com/api/public/'.$file;

    $fotografia = base64_decode($request->fotografia);
    $file1 = 'imgs/usuarios/2'.time().'.jpeg';
    $success1 = file_put_contents($file1, $fotografia);
    $fotografia = 'https://guambraguaytambo.com/api/public/'.$file1;

    $usuario = Usuarios::create([
        'cedula' => $request->cedula,
        'email' => $request->email,
        'nombre' => $request->nombre,
        'apellidos' => $request->apellidos,
        'edad' => $request->edad,
        'genero' => $request->genero,
        'password' => $request->contrasena,
        'rol' => 'doctor',
        'portada' => $portada,
        'fotografia' => $fotografia
    ]);
}

```

*Ilustración 31:Método para la creación de Doctores*

*Elaborado por el investigador*

```

public function showDoctores() {
    $doctores = Usuarios::where('rol','doctor')->get();
    foreach($doctores as $doctor){
        $doctor->recetas = Recetas::where('id_doctor',$doctor->id)->get();
    }
    return response()->json([
        'response' => $doctores
    ]);
}

public function delete($id) {
    $usuario = Usuarios::find($id);
    $usuarios = Usuarios::all();
    foreach($usuarios as $user){
        if ($user->id_doctor === $id) {
            $user->id_doctor = 0;
            $user->save();
        }
    }
    $usuario->delete();
    return response()->json([
        'status' => true
    ]);
}
}

```

*Ilustración 32:Métodos para mostrar y eliminar usuarios*

*Elaborado por el investigador*

- *php artisan make:Controller AlarmasController*

C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller AlarmasController

*Ilustración 33: Comando para la creación del controlador Alarmas*

*Elaborado por el investigador*

```
<?php
namespace App\Http\Controllers;
use ...

class AlarmasController extends Controller
{
    public function delete($id) {
        $alarma = Recetas::find($id);
        $alarma->status = false;
        $alarma->save();
        if ($alarma) {
            return response()->json([
                'status' => true,
                'message' => 'Alarma eliminada'
            ]);
        }
        return response()->json([
            'status' => false,
            'message' => 'La alarma no pudo ser eliminada'
        ]);
    }
}
```

*Ilustración 34: Método para eliminar alarmas*

*Elaborado por el investigador*

```
public function add(Request $request) {
    $receta = Recetas::create([
        'id_usuario' => $request->id_user,
        'tipo' => 'Personal',
        'creado' => $request->creado
    ]);
    $create = Alarmas::create([
        'id_receta' => $receta->id,
        'medicamento' => $request->medicamento,
        'días' => $request->días,
        'tiempo' => $request->tiempo,
        'indicaciones' => $request->indicaciones,
        'caduca_dia' => $request->caduca_dia,
        'every' => $request->every
    ]);
    if ($create) {
        return response()->json([
            'status' => true,
            'response' => $create,
            'message' => 'Alarma creada'
        ]);
    }
    return response()->json([
        'status' => false,
        'message' => 'La alarma no pudo ser creada'
    ]);
}
```

*Ilustración 35: Método para crear la receta*

*Elaborado por el investigador*

```

public function showByUser($id) {
    return response()->json([
        'response' => Alarmas::where('id_user',$id)->where('status',true)->orderBy('id', 'DESC')->get()
    ]);
}
}

```

*Ilustración 36: Método para mostrar Usuarios*

*Elaborado por el investigador*

- *php artisan make:Controller DoctoresController*

```

C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller DoctoresController

```

*Ilustración 37: Comando para la creación del controlador Doctores*

*Elaborado por el investigador*

```

<?php
namespace App\Http\Controllers;
use ..

class DoctoresController extends Controller
{
    public function getAllUsers($id) {
        return response()->json([
            'response' => Usuarios::where('rol','usuario')->where('id','<>',$id)->get(),
        ]);
    }

    public function usuariosQuery($query) {
        return response()->json([
            'response' => Usuarios::where('rol','usuario')->where('apellidos','like','%'.$query.'%')->orWhere('cedula',
                'like','%'.$query.'%')->get()
        ]);
    }

    public function alarmasByUser($id) {
        return response()->json([
            'response' => Alarmas::where('id_user',$id)->where('pon','Medico')->where('status', true)->get(),
        ]);
    }
}

```

*Ilustración 38: Método para mostrar alarmas por usuarios*

*Elaborado por el investigador*



```

public function add(Request $request) {
    $create = Alarmas::create([
        'id_user' => $request->id_user,
        'por' => 'Medico',
        'medicamento' => $request->medicamento,
        'dias' => $request->dias,
        'tiempo' => $request->tiempo,
        'indicaciones' => $request->indicaciones,
        'caduca_dia' => $request->caduca_dia,
        'every' => $request->every,
    ]);
    if ($create) {
        return response()->json([
            'status' => true,
            'response' => $create,
            'message' => 'Alarma creada'
        ]);
    }
    return response()->json([
        'status' => false,
        'message' => 'La alarma no pudo ser creada'
    ]);
}
}

```

Ilustración 39: Método para crear la receta

Elaborado por el investigador

- `php artisan make:Controller MedicamentosController`

```
C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller MedicamentosController
```

Ilustración 40: Comando para la creación del controlador Medicamentos

Elaborado por el investigador

```

<?php

namespace App\Http\Controllers;
use ...

class MedicamentosController extends Controller
{
    public function medicamentosQuery($query) {
        return response()->json([
            'response' => Medicamentos::where('nombre', 'like', '%' . $query . '%')->where('status', 1)->get()
        ]);
    }

    public function showMedicamentos() {
        return response()->json([
            'response' => Medicamentos::where('status', true)->get(),
        ]);
    }
}

```

Ilustración 41: Método para mostrar medicamentos

Elaborado por el investigador

```

public function add(Request $request) {
    $create = Medicamentos::create([
        'nombre' => $request->nombre,
        'descripcion' => $request->descripcion,
    ]);
    if ($create) {
        return response()->json([
            'status' => true,
            'response' => $create,
            'message' => 'Medicamento creado'
        ]);
    }
    return response()->json([
        'status' => false,
        'message' => 'El medicamento no pudo ser creado'
    ]);
}

public function delete($id) {
    $medicamento = Medicamentos::find($id);
    $medicamento->status = false;
    $medicamento->save();
    return response()->json([
        'status' => true,
        'message' => 'Medicamento eliminado'
    ]);
}
}
}

```

*Ilustración 42: Métodos para agregar y eliminar los medicamentos*

*Elaborado por el investigador*

- *php artisan make:Controller RecetasController*

```
C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller RecetasController
```

*Ilustración 43: Creación del Controlador de recetas*

*Elaborado por el investigador*

```

<?php

namespace App\Http\Controllers;
use ...

class RecetasController extends Controller
{
    public function showByUser($id) {
        return response()->json([
            'response' => Recetas::where('id_usuario',$id)->where('status',true)->get(),
        ]);
    }
}

```

*Ilustración 44: Método para mostrar recetas por usuarios*

*Elaborado por el investigador*

```

public function add(Request $request) {
    $usuario = $request->usuario;
    $receta = Recetas::create([
        'id_usuario' => $usuario['id'],
        'id_doctor' => $request->id_doctor,
        'tipo' => 'Medico',
        'creado' => $request->creado
    ]);

    $medicamentos = $request->medicamentos;
    foreach($medicamentos as $medicamento){
        Alarmas::create([
            'id_receta' => $receta->id,
            'medicamento' => $medicamento['medicamento'],
            'dias' => $medicamento['dias'],
            'tiempo' => $medicamento['tiempo'],
            'indicaciones' => $medicamento['indicaciones'],
            'caduca_dia' => $medicamento['caduca_dia'],
            'every' => $medicamento['every']
        ]);
    }
    return response()->json([
        'status' => true
    ]);
}

public function delete($id) {
    $receta = Recetas::where('id',$id)->first();
    $receta->save();
    return response()->json([
        'status' => true,
    ]);
}
}

```

*Ilustración 45: Métodos para agregar y eliminar recetas*

*Elaborado por el investigador*

- *php artisan make:Controller UsuariosController*

```
C:\Users\DELL 2018\Desktop\si\api>php artisan make:Controller UsuariosController
```

*Ilustración 46: Creación del controlador Usuarios*

*Elaborado por el investigador*

```

<?php
namespace App\Http\Controllers;
use ...

class UsuariosController extends Controller
{
    public function update(Request $request) {
        $usuario = Usuarios::find($request->id_user);
        $usuario->cedula = $request->cedula;
        $usuario->email = $request->email;
        $usuario->nombre = $request->nombre;
        $usuario->apellidos = $request->apellidos;
        $usuario->edad = $request->edad;
        $usuario->genero = $request->genero;
        $usuario->password = $request->password;
        $usuario->time_desayuno = $request->time_desayuno;
        $usuario->time_almuerzo = $request->time_almuerzo;
        $usuario->time_merienda = $request->time_merienda;
        $usuario->save();
        return response()->json([
            'status' => true,
            'response' => $usuario,
            'message' => 'Se ha modificado los datos del usuario'
        ]);
    }
}

```

*Ilustración 47: Método para actualizar datos del usuario*

*Elaborado por el investigador*

```

public function login(Request $request) {
    $usuario = Usuarios::where('email',$request->email)->where('password',$request->password)->first();
    if ($usuario) {
        return response()->json([
            'status' => true,
            'response' => $usuario,
            'message' => 'Has iniciado sesión como ' . $usuario->nombre . ' ' . $usuario->apellidos
        ]);
    }
    return response()->json([
        'status' => false,
        'message' => 'Email o contraseña no existen'
    ]);
}

```

*Ilustración 48: Método para el inicio de sesión*

*Elaborado por el investigador*

```

public function register(Request $request) {
    $exist = Usuarios::where('email',$request->email)->orWhere('cedula',$request->cedula)->first();
    if ($exist) {
        return response()->json([
            'status' => false,
            'message' => 'Los datos ingresados ya estan en uso'
        ]);
    }
    $usuario = Usuarios::create([
        'cedula' => $request->cedula,
        'email' => $request->email,
        'nombre' => $request->nombre,
        'apellidos' => $request->apellidos,
        'edad' => $request->edad,
        'genero' => $request->genero,
        'password' => $request->password,
        'time_desayuno' => $request->time_desayuno,
        'time_almuerzo' => $request->time_almuerzo,
        'time_merienda' => $request->time_merienda,
    ]);
    if ($usuario){
        return response()->json([
            'status' => true,
            'message' => 'Registro con éxito'
        ]);
    }
    return response()->json([
        'status' => false,
        'message' => 'No es posible realizar el registro, vuelve a intentarlo mas tarde'
    ]);
}

```

*Ilustración 49: Método para el registro de usuarios*

*Elaborado por el investigador*

Para cada controlador se ha desarrollado tareas a realizar, donde cada una se ejecutará al consumir la API REST.

### 3.2.3.4 Creación de las Rutas

Las rutas son capas muy importantes en el framework, ya que, se encarga de manejar las peticiones y respuestas del usuario, ya sea Get Post, Put, entre otros. Laravel cuenta con una carpeta raíz llamada routes, donde se encontrará el archivo api.php, que servirá para la creación de rutas en nuestra aplicación.

En la tabla, se documentan las rutas utilizadas del API REST implementada.

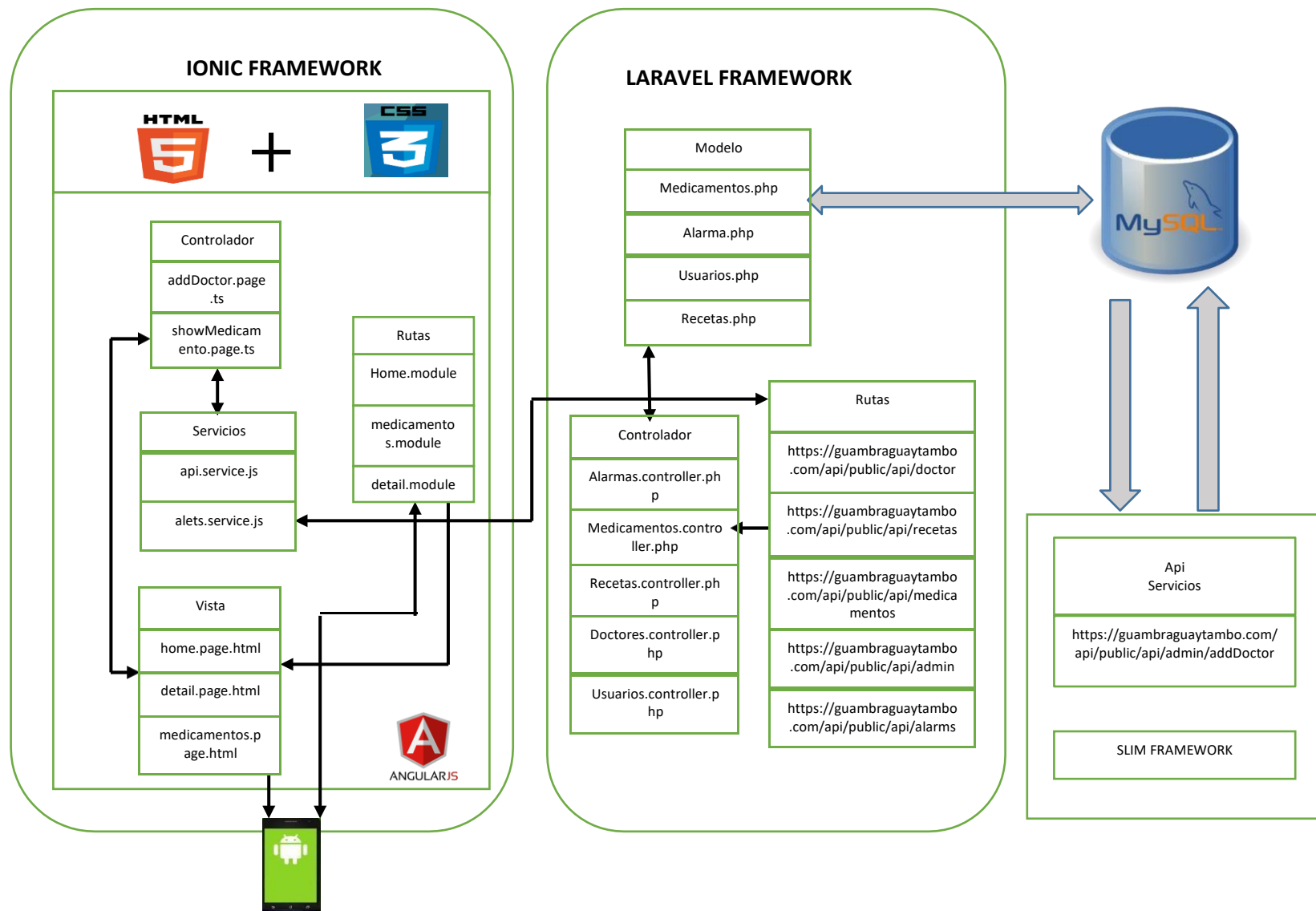
Método	Ruta	Descripción
post	api/admin/adddoctor	Agregar doctores
delete	api/admin/delete/{id}	Eliminar usuarios por id
get	api/admin/showdoctores	Mostrar Doctores
post	api/alarms/add	Agregar alarma
delete	api/alarms/delete/{id}	Eliminar alarma
get	api/alarms/showbyuser/{id}	Mostrar alarma por id
post	api/doctor/add	Agregar doctores
get	api/doctor/alarmasbyuser/{id}	Mostrar alarma del usuario por id
get	api/doctor/getallusers/{id}	Mostrar doctores por id
post	api/medicamentos/ad	Agregar medicamentos
get	api/medicamentos/delete/{id}	Eliminar medicamentos por id
get	api/medicamentos/showmedicamentos	Mostrar medicamentos
post	api/recetas/add	Agregar receta
delete	api/recetas/delete/{id}	Eliminar receta
get	api/recetas/showbyuser/{id}	Mostrar receta por usuario
post	api/user_init/login	Iniciar sesión
put	api/user_init/logout/{id}	Cerrar sesión
post	api/user_init/register	Registrar usuario
put	api/user_init/update	Actualizar usuario

Tabla 12: Rutas utilizadas en la Api Rest

Elaborado por el investigador

### 3.2.3.5 Arquitectura del Proyecto

A continuación, se detalla la arquitectura del proyecto donde se explicará el funcionamiento y el consumo de la API REST.



### 3.2.3.6 Dominio y hosting

Es importante contar con un hosting ya que nos permitirá almacenar la base de datos y la API con el fin de transferir nuestro contenido a la aplicación y así visualizar la información.

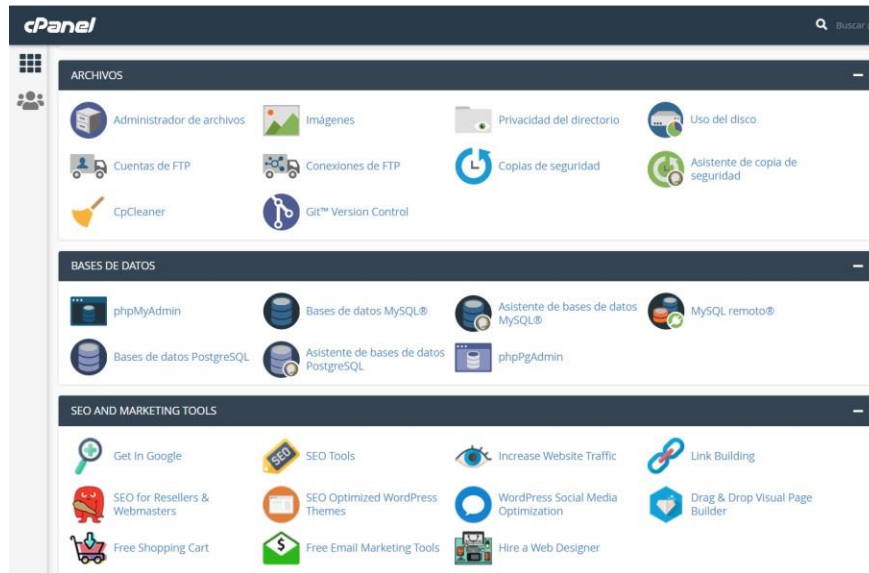


Ilustración 50: Hosting Cpanel

Elaborado por: El investigador

### 3.2.3.7 Diseño lógico de base de datos

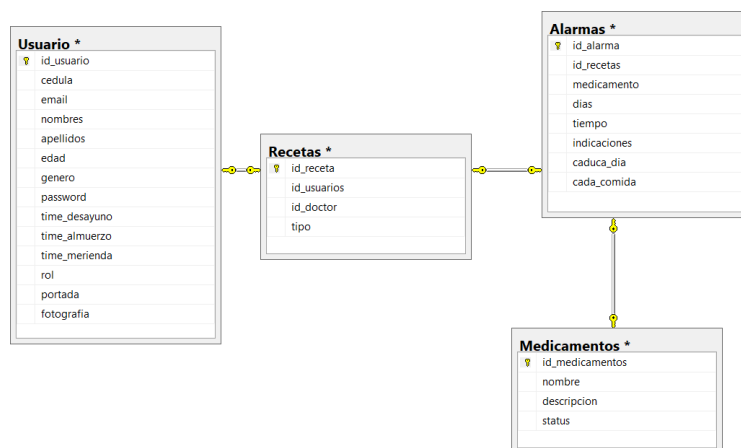


Ilustración 51: Diagrama de Base de datos

Elaborado por: El investigador

### 3.2.3.8 Diagrama de Actividades

- **Registro de datos del Médico**

El administrador es la persona encargada de crear los perfiles para los doctores mediante el uso de datos personales y guardar estos en la base de datos para el posterior inicio de sesión.

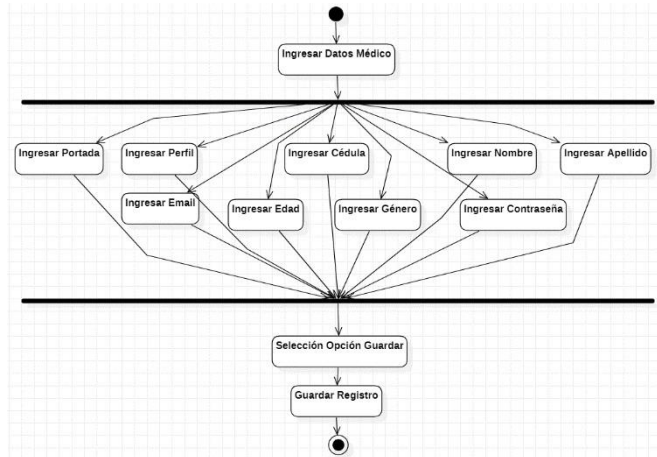


Ilustración 52: Diagrama de actividad. Registro de datos del Médico

Elaborado por el investigador

- **Editar datos personales del Médico**

Cuando el médico ingrese con su usuario al aplicativo móvil, podrá editar sus datos personales mediante la opción editar, esto en caso de ingreso erróneo de información o actualización de datos.

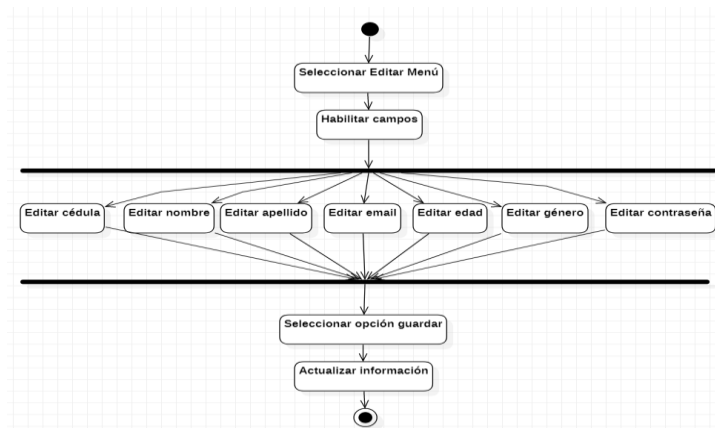


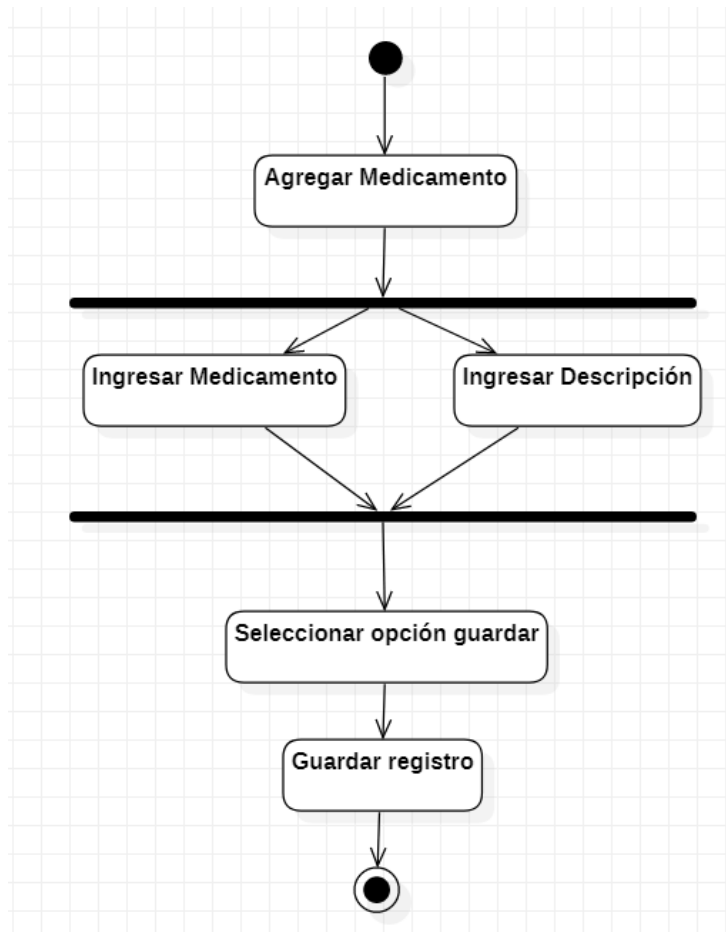
Ilustración 53: Diagrama de actividades. Editar datos del Médico

Elaborado por el investigador



- **Agregar medicamento**

El médico desde su perfil, podrá agregar y guardar los medicamentos que el considere pertinentes para las prescripciones médicas, los cuales se almacenarán en la base de datos.



*Ilustración 54: Diagrama de actividades. Agregar medicamento*

*Elaborado por el investigador*

- **Eliminar Medicamento**

El médico desde su perfil, podrá eliminar los medicamentos que él considere no necesarios para las prescripciones médicas, actualizando la base de datos considerando que dispone de un estado activo e inactivo.

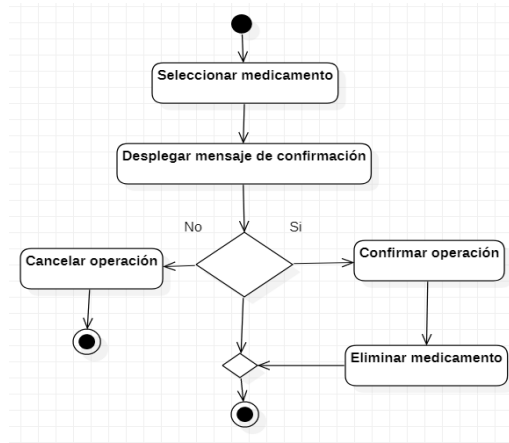


Ilustración 55: Diagrama de actividades. Eliminar medicamento

Elaborado por el investigador

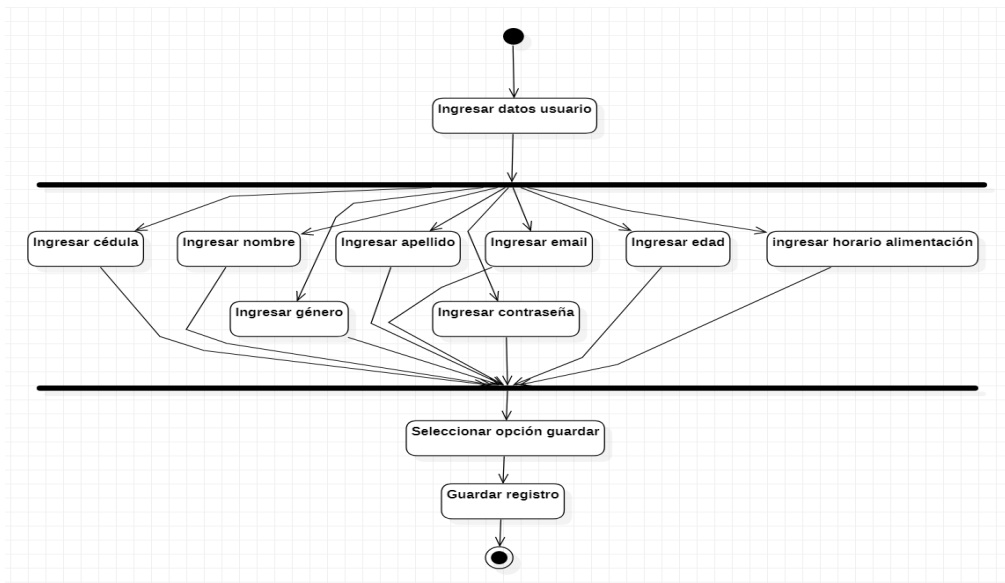
id	nombre	descripcion	status
11	Diclofenaco 50 mg	Detiene la producción del cuerpo de una sustancia ...	1
12	Aciclovir 400 mg	Sirve para reducir el dolor y acelerar la curación...	1
13	Metformina 500 mg	Ayuda a controlar la cantidad de glucosa en su san...	1
14	Metformina 750 mg	Ayuda a controlar la cantidad de glucosa en su san...	1
15	Metformina 850 mg	Ayuda a controlar la cantidad de glucosa en su san...	1
16	Cetirizina 10 mg	Se utiliza en los tratamientos de la alergia	1
17	Enalapril 10 mg	Combinación para tratar la presión arterial alta.	1
18	Atenolol 50 mg	Combinación con otros medicamentos para tratar la ...	1
19	Paroxetina 20 mg	Utilizado para tratar depresión, el trastorno de p...	1
20	Amlodipino 5 mg	Reduce la presión arterial.	1
21	Paracetamol 500 mg	Tratamiento de los síntomas del dolor leve y fiebr...	1
22	Levotiroxina 50 ug	Sirve para tratar el hipotiroidismo.	1
23	Omeprazol 20 mg	Medicamento para tratar el acidez estomacal.	1
24	Amoxicilina 500 mg	Medicamento para tratar la neumonía.	1
25	Loratadina 10 mg	Se usa para tratar alergias.	1
26	Clarithromicina 500 mg	Medicamento para tratar infecciones bacterianas.	1
27	Azitromicina 500 mg	Medicamento para tratar la bronquitis.	1
28	Lansoprazol 30 mg	Se usa para tratar los síntomas de enfermedad por ...	1

Ilustración 56: Base de Datos. Estado del Medicamento

Elaborado por el investigador

- **Registro de paciente**

Cada paciente se encargará de crear su propia cuenta, registrándose al ingresar al sistema, donde será necesario proporcionar sus datos personales y posteriormente guardarlos, para realizar futuros inicios de sesión.



*Ilustración 57: Diagrama de actividades. Registro del paciente*

*Elaborado por el investigador*

- **Editar datos personales del paciente**

Cuando el paciente ingrese con su usuario al aplicativo móvil, podrá editar sus datos personales mediante la opción editar, esto en caso de ingreso erróneo de información o actualización de datos.

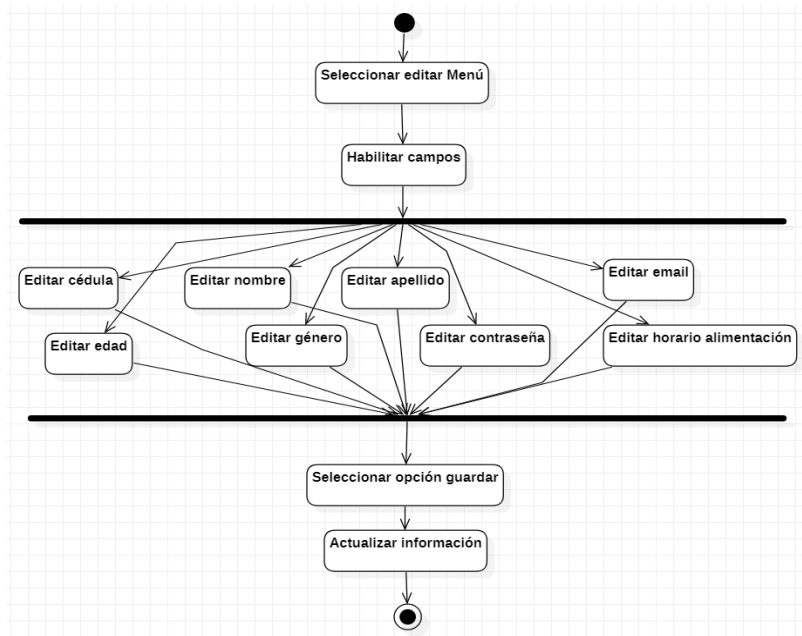


Ilustración 58: Diagrama de actividades. Editar datos del paciente

Elaborado por el investigador

- **Agregar alarma personal**

El paciente desde su perfil, podrá agregar y guardar sus propias alarmas personalizadas de las prescripciones médicas recetadas por su médico tratante, las cuales se almacenarán en una base de datos.

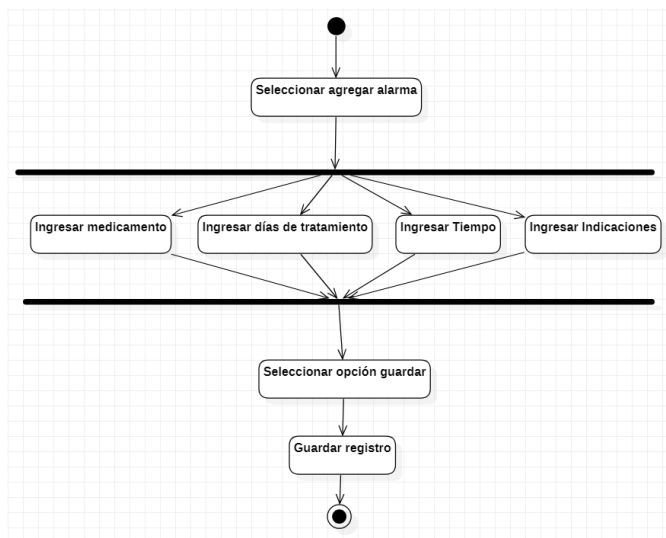
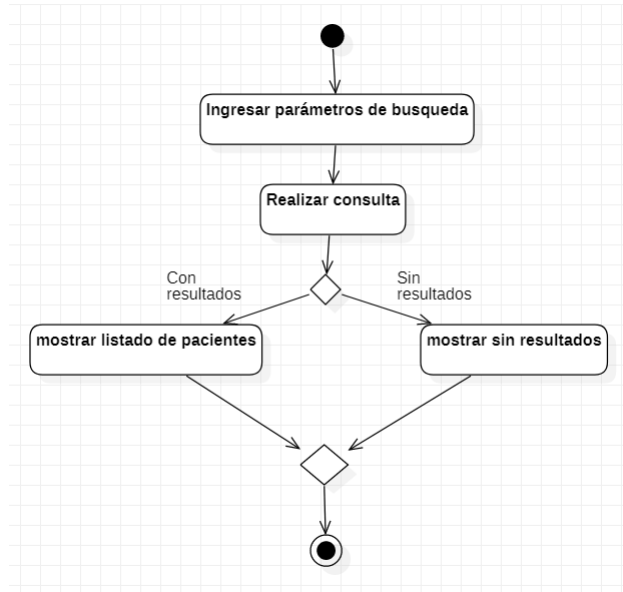


Ilustración 59: Diagrama de actividades. Alarma personalizada

Elaborado por el investigador

- **Búsqueda de paciente**

El doctor desde su perfil, podrá realizar fácilmente la búsqueda de sus pacientes registrados en el aplicativo, mediante el apellido o cédula del paciente.



*Ilustración 59.1: Diagrama de actividades. Buscador de pacientes*

*Elaborado por el investigador*

- **Agregar receta a paciente**

El doctor desde su perfil y posterior búsqueda del paciente podrá agregar el respectivo medicamento y dosificación, según el diagnóstico realizado, ingresando el número de días, horarios e indicaciones del tratamiento.

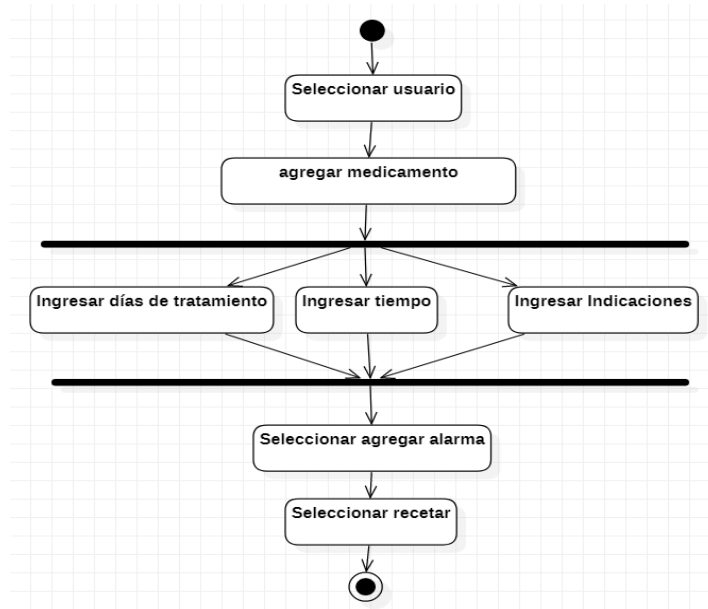


Ilustración 60: Diagrama de actividades. Agregar receta

Elaborado por el investigador

- **Abrir receta**

Todo paciente registrado, podrá visualizar su receta, donde encontrará el medicamento, el número de días, horarios e indicaciones del tratamiento prescritos por su médico.

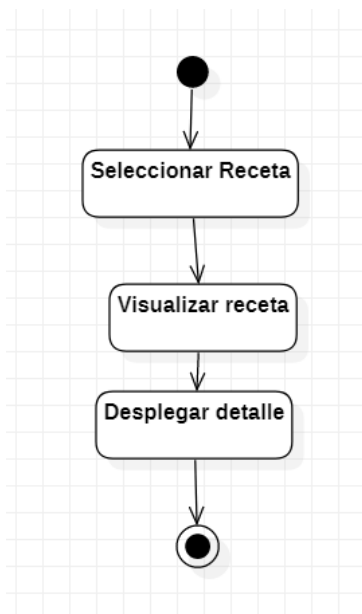


Ilustración 61: Diagrama de actividades. visualizar receta

Elaborado por el investigador

- **Historial del paciente**

El médico desde su perfil, podrá llevar un historial de todos los pacientes registrados en el aplicativo, donde podrá visualizar las recetas prescritas con anterioridad a sus pacientes.

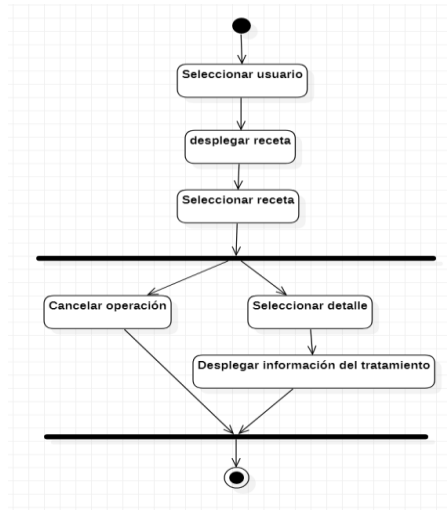


Ilustración 62: Diagrama de actividades. Historial de pacientes

Elaborado por el investigador

### 3.2.3.9 Diagrama de secuencia

- **Registro de datos del médico**

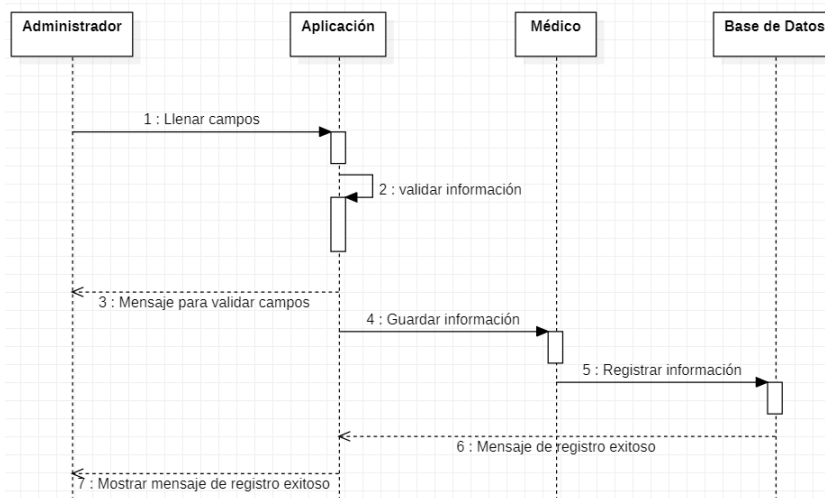
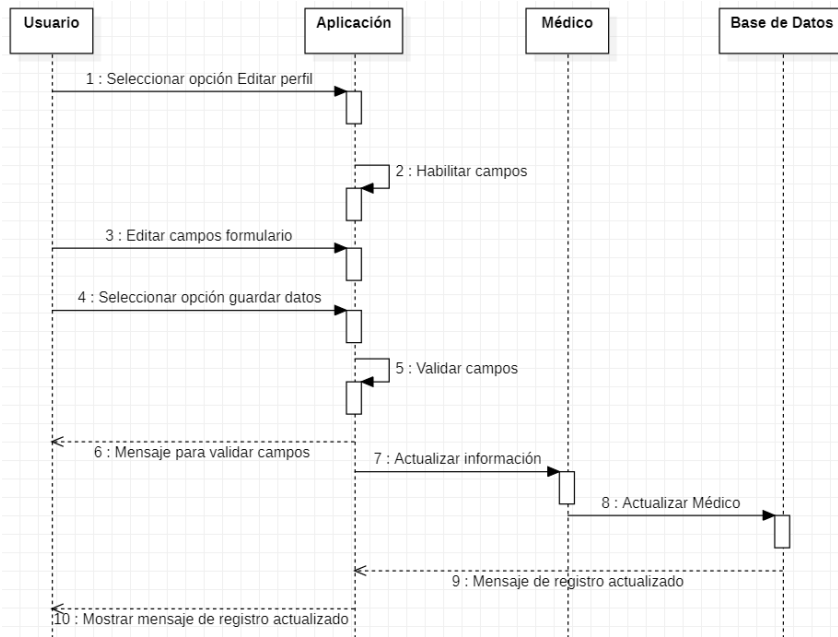


Ilustración 63: Diagrama de secuencia. Registro de datos del médico

Elaborado por el investigador

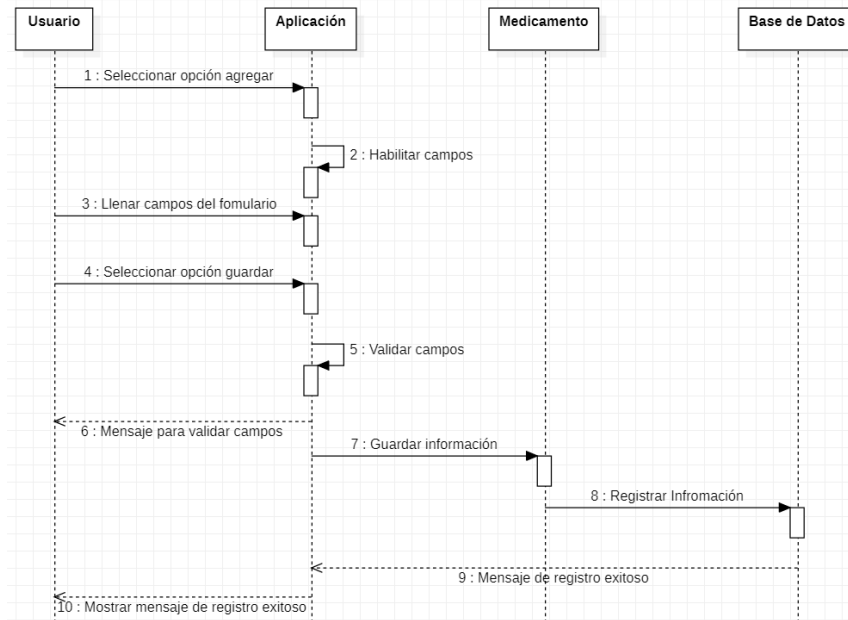
- **Editar datos personales del médico**



*Ilustración 64: Diagrama de secuencia. Editar datos personales del médico*

*Elaborado por el investigador*

- **Agregar medicamento**

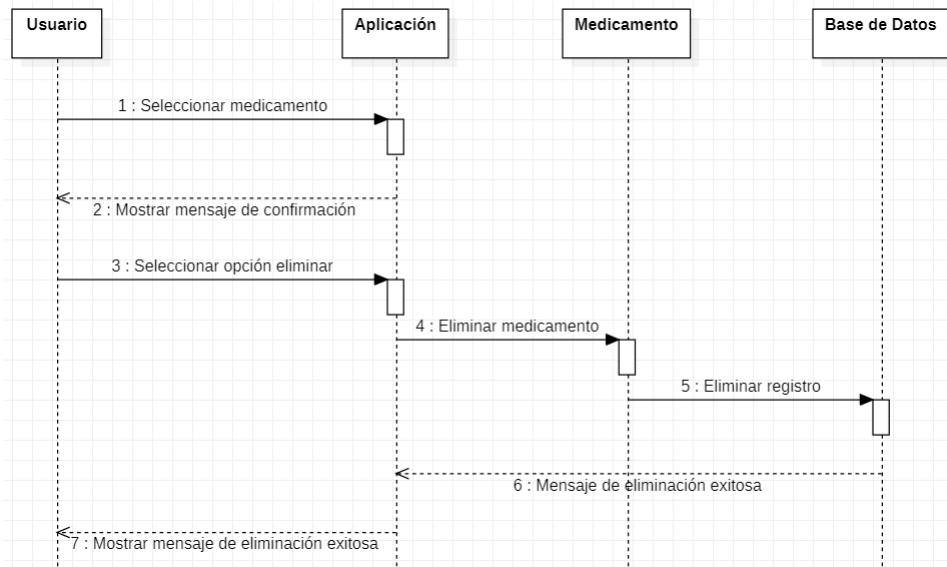


*Ilustración 65: Diagrama de secuencia. Agregar medicamento*

*Elaborado por el investigador*



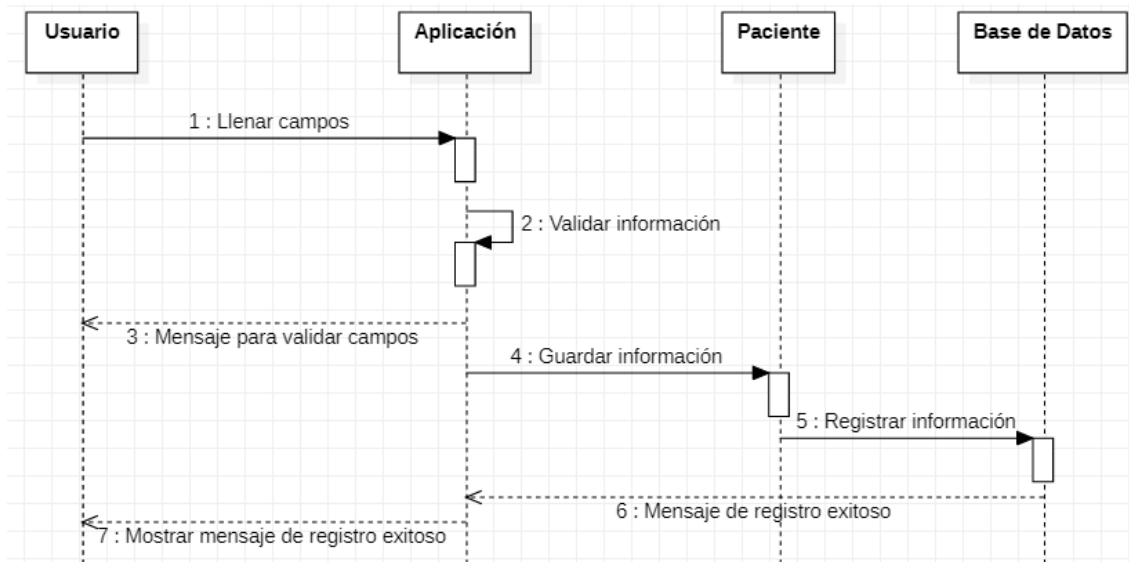
- **Eliminar medicamento**



*Ilustración 66: Diagrama de secuencia. Eliminar medicamento*

*Elaborado por el investigador*

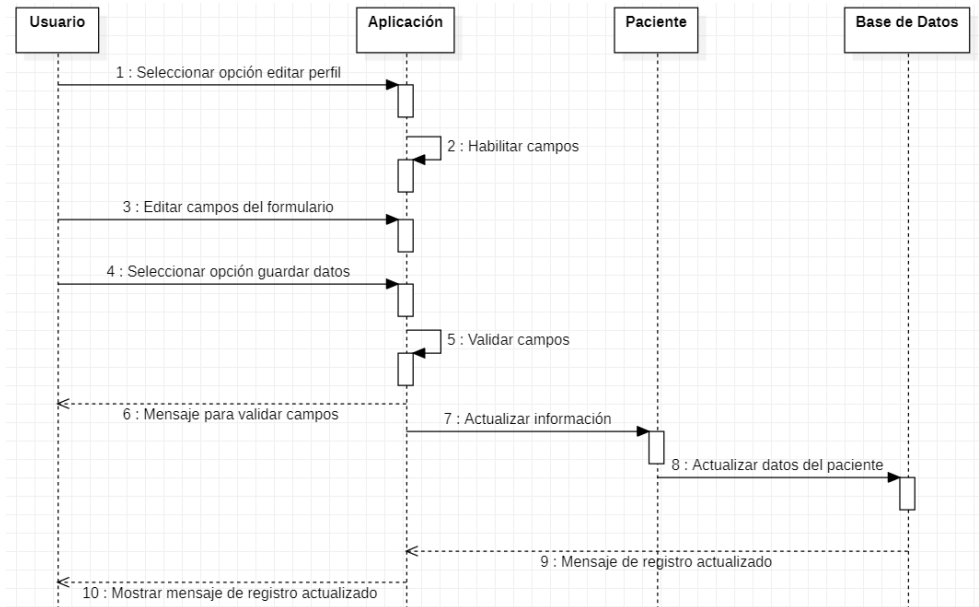
- **Registro del paciente**



*Ilustración 67: Diagrama de secuencia. Registro de datos del paciente*

*Elaborado por el investigador*

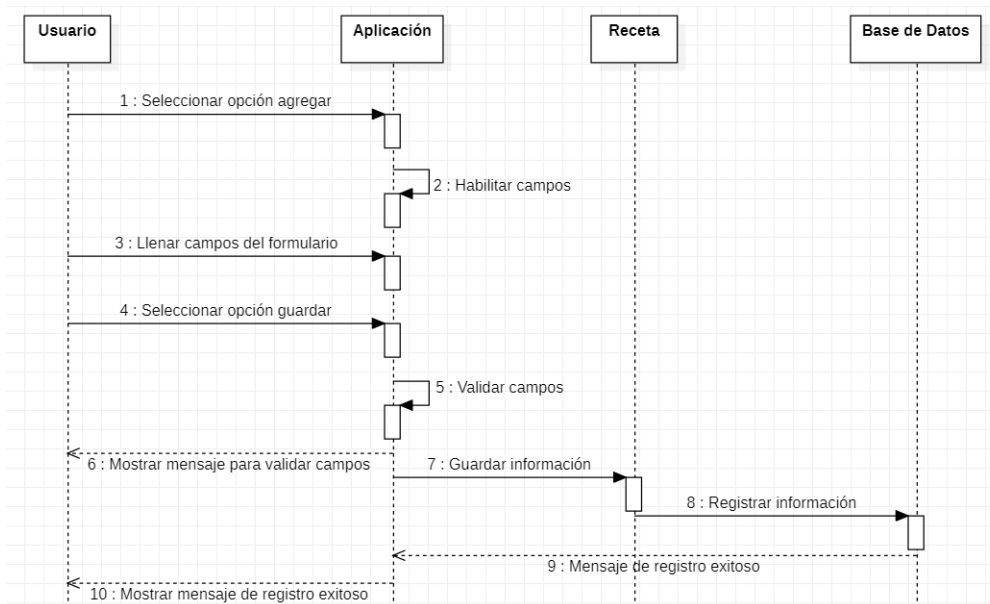
- **Editar datos personales del paciente**



*Ilustración 68: Diagrama de secuencia. Editar datos personales del paciente*

*Elaborado por el investigador*

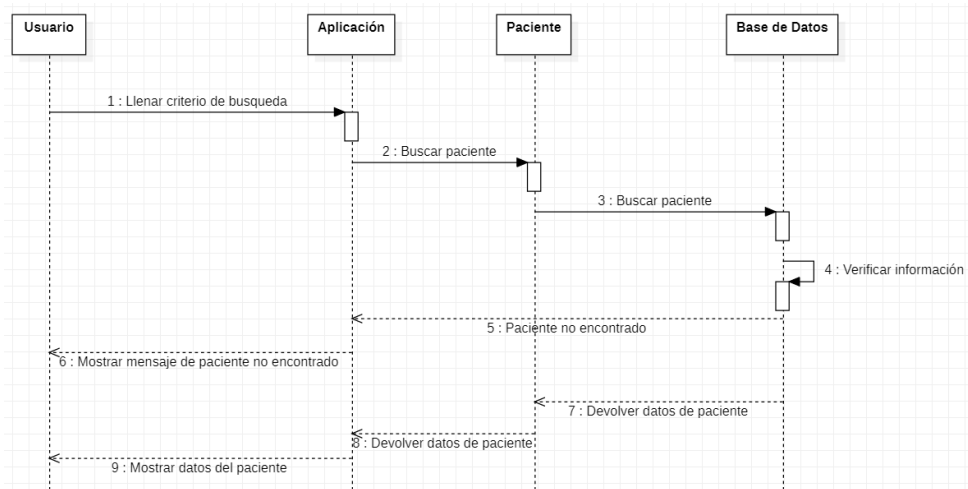
- **Agregar alarma personal**



*Ilustración 69: Diagrama de secuencia. Agregar alarma personal*

*Elaborado por el investigador*

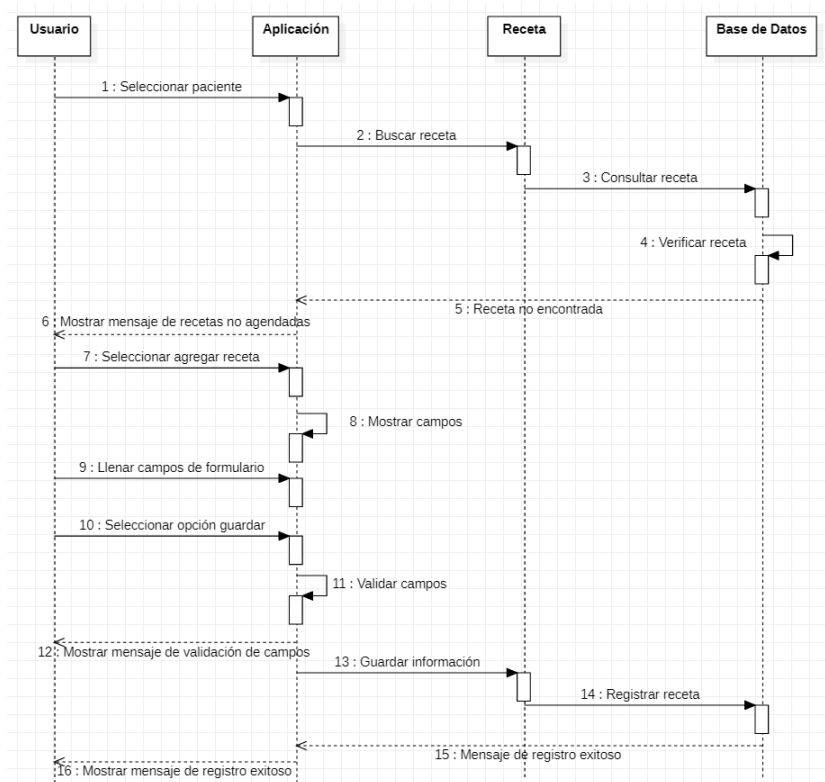
- **Búsqueda de paciente**



*Ilustración 70: Diagrama de secuencia. Búsqueda de paciente*

*Elaborado por el investigador*

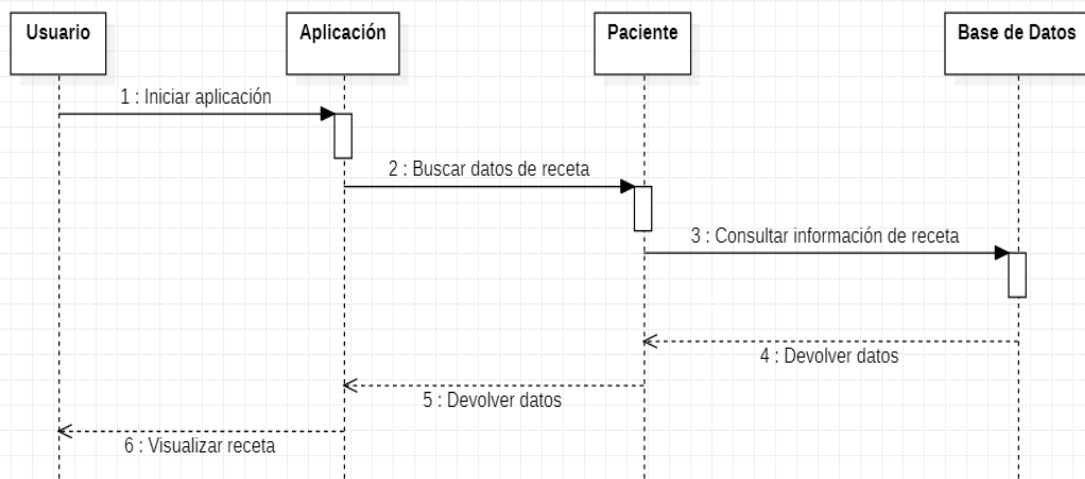
- **Agregar receta a paciente**



*Ilustración 71: Diagrama de secuencia. Agregar receta a paciente*

*Elaborado por el investigador*

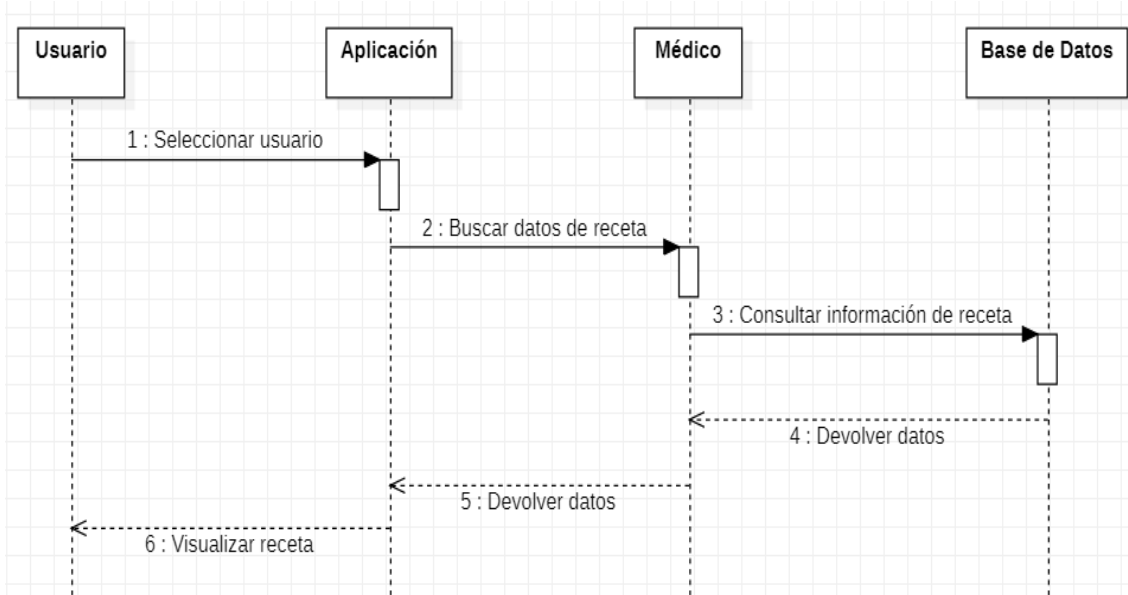
- **Visualizar receta**



*Ilustración 72: Diagrama de secuencia. Abrir receta*

*Elaborado por el investigador*

- **Historial del paciente**



*Ilustración 73: Diagrama de secuencia. Historial del paciente*

*Elaborado por el investigador*

### 3.2.4 Fase de Producción

#### 3.2.4.1 Iteración 1.

##### Día de planificación

- Registro de usuario en su rol de médico que van acceder al sistema

Descripción	
Como	Administrador
Necesito	Registrar la información de médicos
Para	Que tengan acceso al sistema
Criterio de Aceptación	
Dado	Que el usuario, en su rol de administrador, ingrese a la pantalla “Registro de Médicos”
Cuando	Llene los campos del formulario
Entonces	La aplicación registrará los datos del médico en la base de datos

Tabla 13: Iteración 1.Registro de datos personales del médico

Elaborado por el investigador

El código establecido para realizar la Iteración 1 del proyecto es el siguiente:

##### Home\_page

```
<ion-content>
  <div [formGroup]="addForm">
    <ion-item color="light">
      <ion-label color="dark">
        Portada
      </ion-label>
    </ion-item>
    <div class="ion-text-center" (click)="presentActionSheet(0)">
      <img style="width: 100%; height: 120px;" [src]="portada" alt="">
    </div>
    <hr>
    <ion-item color="light">
      <ion-label color="dark">
        Perfil
      </ion-label>
    </ion-item>
    <div class="ion-text-center" (click)="presentActionSheet(1)">
      <img style="width: 100px; border-radius: 100%;" [src]="perfil" alt="">
    </div>
    <hr>
    <div class="ion-padding-horizontal">
      <ion-input type="text" formControlName="cedula" placeholder="Cedula"></ion-input>
      <ion-input type="text" autocapitalize="on" formControlName="nombre" placeholder="Nombre"></ion-input>
      <ion-input type="text" autocapitalize="on" formControlName="apellidos" placeholder="Apellidos"></ion-input>
      <ion-input type="email" formControlName="email" placeholder="Email"></ion-input>
      <ion-input type="number" formControlName="edad" placeholder="Edad"></ion-input>
      <ion-input readonly formControlName="genero" placeholder="Genero" (click)="presentGeneroSheet()"></ion-input>
      <ion-input type="password" formControlName="contrasena" placeholder="Contraseña"></ion-input>
      <ion-input type="password" formControlName="confirmar_contrasena" placeholder="Confirmar contraseña"></ion-input>
      <ion-button (click)="checkForm()" [disabled]="!addForm.valid" expand="block">
        Guardar
      </ion-button>
    </div>
  </div>
</ion-content>
```

Ilustración 74: Iteración 1. Desarrollo home\_page

Elaborado por el investigador

## Presentación del Diseño

Se evidencia la ejecución de la iteración, donde el usuario podrá crear su perfil.

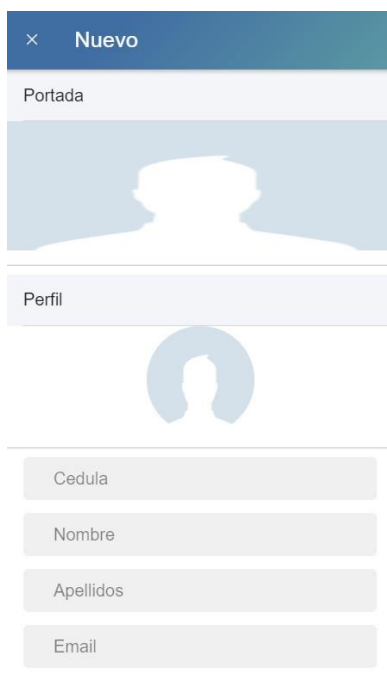


Ilustración 75: Iteración 1. Diseño (creación de perfil doctor)

Elaborado por el investigador

### 3.2.4.2 Iteración 2.

#### Día de planificación

- **Editar datos personales del doctor**

Descripción	
Como	Médico
Necesito	Visualizar la información ingresada
Para	Corregir los datos y actualizar la información en la base de datos
Criterio de Aceptación	
Dado	Que el usuario, en su rol de médico, ingrese a la pantalla “Menú”
Cuando	Seleccione la opción modificar
Entonces	El sistema actualizará la información del médico

Tabla 14: Iteración 2: Editar datos personales del doctor

Elaborado por el investigador

El código establecido para realizar la Iteración 2 del proyecto es el siguiente:

## perfil\_page

```
<ion-content>
  <div class="ion-padding">
    <div class="ion-text-center ion-padding">
      <img style="width: 100%; height: 120px;" [src]="auth.currentUser.portada" alt="">
      <img style="width: 60px; height: 60px; border-radius: 100%" [src]="auth.currentUser.fotografia" alt="">
    </div>
  </div>
  <div class="ion-margin" [formGroup]="userForm">
    <div class="title">Datos personales</div>
    <br>
    <ion-input [disabled]="!editForm" formControlName="cedula" type="text" placeholder="Cedula"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="nombre" type="text" placeholder="Nombre"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="apellidos" type="text" placeholder="Apellidos"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="email" type="email" placeholder="Email"></ion-input>
    <ion-row>
      <ion-col size="6" class="no-padding-left">
        <ion-input [disabled]="!editForm" formControlName="edad" type="number" placeholder="Edad"></ion-input>
      </ion-col>
      <ion-col size="6" class="no-padding-right">
        <div (click)="presentGeneroSheet()" class="input">{{userForm.controls.genero.value}}</div>
      </ion-col>
    </ion-row>
    <ion-input [disabled]="!editForm" formControlName="password" type="password" placeholder="Contraseña"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="confirma_password" type="password" placeholder="Confirmar contraseña"></ion-input>
  </div>
</ion-content>
<ion-footer>
  <ion-toolbar>
    <ion-item detail (click)="logout()" *ngIf="!editForm">
      <ion-icon name="close" slot="start" color="danger"></ion-icon>
      <ion-label style="color: var(--ion-color-danger);">Cerrar sesión</ion-label>
    </ion-item>
    <ion-item detail (click)="editData()" *ngIf="editForm">
      <ion-icon name="save-outline" slot="start" color="primary"></ion-icon>
      <ion-label style="color: var(--ion-color-primary);">Guardar Datos</ion-label>
    </ion-item>
  </ion-toolbar>
</ion-footer>
```

Ilustración 76: Iteración 2. Desarrollo de la página editar datos

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración 2, donde el usuario podrá editar su perfil.



Ilustración 77: Iteración 2. Diseño Editar datos personales del médico

Elaborado por el investigador

Datos Personales

FAMILIAR

Datos personales

1803098548

Manuel

Eduardo

mcruzito@hotmail.com

38 Masculino

Guardar Datos

Ilustración 78: Iteración 2. Diseño Guardar datos personales

Elaborado por el investigador

### 3.2.4.3 Iteración 3.

#### Día de planificación

- **Agregar Medicamento**

<b>Descripción</b>	
Como	Médico
Necesito	Ingresar la información de los medicamentos
Para	Recetar a pacientes
<b>Criterio de Aceptación</b>	
Dado	Que el medico se encuentre en la pantalla “Agregar medicamento”
Cuando	Seleccione Agregar medicamento
Entonces	La aplicación registrará el medicamento y la descripción.

Tabla 15: Iteración 3. Agregar medicamentos

Elaborado por el investigador



El código establecido para realizar la Iteración 3 del proyecto es el siguiente:

### add-medicamento\_page

```
<ion-header class="bg_gradient">
  <ion-toolbar color="white">
    <ion-buttons slot="start">
      <ion-button (click)="modalcontroller.dismiss({status:false})">
        <ion-icon slot="start" name="close"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>Agregar medicamento</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div class="ion-margin">
    <div class="ion-text-center">
      <ion-icon color="medium" style="font-size: 10rem;" name="bandage-outline"></ion-icon>
      <br><br>
    </div>
    <ion-input type="text" (ionChange)="ionInputChanged($event)" placeholder="Medicamento"></ion-input>
    <ion-textarea rows="3" (ionChange)="ionAreaChanged($event)" placeholder="Descripcion (minimo 10 caracteres)"></ion-textarea>
    <br>
    <!-- <div class="ion-text-center">
      <ion-note>Centro Médico Familiar</ion-note>
    </div -->
    <br>
    <ion-button (click)="add()" expand="full">
      <ion-icon slot="start" name="bandage-outline"></ion-icon>
      Agregar medicamentos
    </ion-button>
  </div>
</ion-content>
```

Ilustración 79: Iteración 3. Desarrollo de la página para agregar medicamentos

Elaborado por el investigador

### Día de liberación

Se evidencia la ejecución de la iteración 3, donde el usuario con perfil de doctor podrá agregar medicamentos.

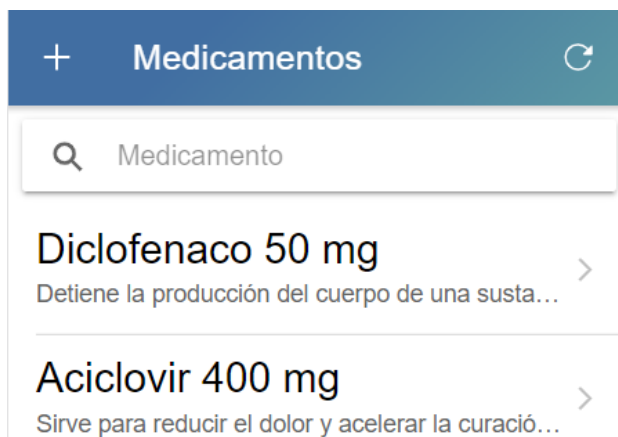


Ilustración 80: Iteración 3. Medicamentos agregados

Elaborado por el investigador

×
Agregar medicamento



➕
AGREGAR MEDICAMENTOS

*Ilustración 81: Iteración 3. Diseño de la página agregar medicamentos*

*Elaborado por el investigador*

### 3.2.4.4 Iteración 4.

#### Día de planificación

- **Eliminar Medicamentos**

<b>Descripción</b>	
Como	Médico
Necesito	Visualizar los medicamentos
Para	Eliminar los registros no deseados de la base de datos.
<b>Criterio de Aceptación</b>	
Dado	Que el usuario ingrese a la pantalla de medicamentos
Cuando	Seleccione el medicamento
Entonces	El sistema emitirá un mensaje de confirmación para eliminar el medicamento seleccionado

*Tabla 16: Iteración 4. Eliminar medicamentos*

*Elaborado por el investigador*

El código establecido para realizar la Iteración 4 del proyecto es el siguiente:

### Eliminar\_medramento\_page

```
<ion-header class="bg_gradient">
  <ion-toolbar color="white">
    <ion-buttons slot="start">
      <ion-button (click)="add()" *ngIf="auth.isAuthenticated()">
        <ion-icon slot="icon-only" name="add"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>
      Medicamentos
    </ion-title>
    <ion-buttons slot="end">
      <ion-button (click)="init()">
        <ion-icon slot="start" name="reload"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div *ngIf="auth.isAuthenticated()">
    <div *ngIf="estado === 1">
      <ion-searchbar placeholder="Medicamento" (ionChange)="onSearchChange($event)" [debounce]="1000"></ion-searchbar>
      <!-- <ion-item detail *ngFor="let medicamento of medicamentos" (click)="choose(medicamento)" -->
      <ion-item detail *ngFor="let medicamento of medicamentos" (click)="presentAlertConfirm(medicamento)" -->
    </div>
  </div>
</ion-content>
```

Ilustración 82: Iteración 4. Desarrollo del método para eliminar medicamentos

Elaborado por el investigador

### Día de liberación

Se evidencia la ejecución de la iteración, donde el usuario con perfil de doctor eliminar medicamentos.

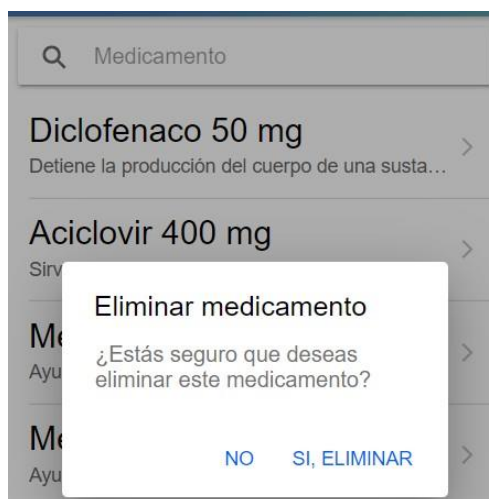


Ilustración 83: Iteración 4. Diseño de eliminar medicamentos

Elaborado por el investigador

### 3.2.4.5 Iteración 5.

#### Día de planificación

- **Historial de Pacientes**

<b>Descripción</b>	
Como	Médico
Necesito	Llevar el historial de los pacientes
Para	Visualizar sus recetas
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, en su rol de médico, ingrese a la pantalla “Usuarios”
Cuando	Seleccione el nombre de usuario
Entonces	Se visualizará la fecha que fue recetada, los medicamentos recetados y los días en los que deberá consumir los medicamentos.

Tabla 17: Iteración 5. Visualizar historial de pacientes

Elaborado por el investigador

El código establecido para realizar la Iteración 5 del proyecto es el siguiente:

#### historial\_page

```
<ion-content>
  <div *ngIf="estado === 1">
    <ion-card *ngFor="let receta of notifications" (click)="presentActionSheet(receta)">
      <ion-card-header>
        <ion-card-subtitle>{{receta.tipo | translate}}</ion-card-subtitle>
      </ion-card-header>
      <ion-card-content>
        <div *ngIf="receta.tipo === 'Medico'">
          <strong>Recetado por: </strong> {{receta.doctor.nombre}} {{receta.doctor.apellidos}}<br>
        </div>
        <strong>Fecha: </strong> {{receta.creado | date:'fullDate'}}<br>
        <strong>Medicamentos: </strong> {{receta.medicamentos.length}} <br>
      </ion-card-content>
    </ion-card>
  </div>

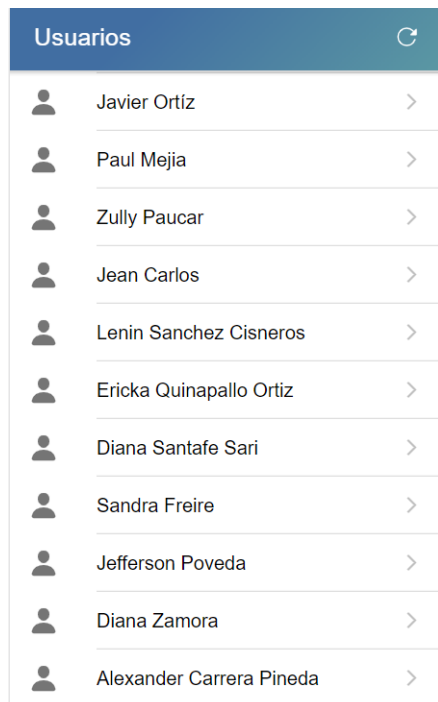
  <div *ngIf="estado === 2" class="ion-padding ion-text-center">
    <ion-text color="medium">Esta usuario no tiene recetas agendadas</ion-text>
  </div>
</ion-content>
```

Ilustración 84: Iteración 5. Desarrollo para visualizar el historial de pacientes

Elaborado por el investigador

## Día de liberación

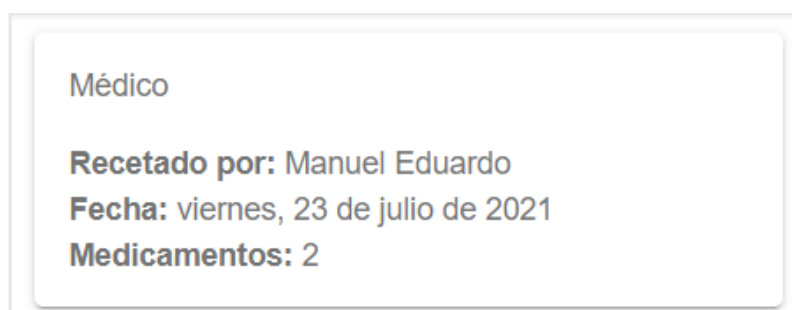
Se evidencia la ejecución de la iteración 5, donde el usuario con perfil de médico podrá visualizar el historial del paciente.



Usuarios	
Javier Ortiz	>
Paul Mejia	>
Zully Paucar	>
Jean Carlos	>
Lenin Sanchez Cisneros	>
Ericka Quinapallo Ortiz	>
Diana Santafe Sari	>
Sandra Freire	>
Jefferson Poveda	>
Diana Zamora	>
Alexander Carrera Pineda	>

*Ilustración 85: Iteración 5. Diseño visualización de pacientes*

*Elaborado por el investigador*



*Ilustración 86: Iteración 5. Visualización de historial de pacientes*

*Elaborado por el investigador*

<p>Medico</p> <p><b>Omeprazol 20 mg</b></p> <p><b>Tratamiento:</b> 3 días</p> <p><b>Tiempo:</b> Merienda</p> <p><b>Indicaciones:</b> Tomar una tableta antes de la merienda.</p>
<p>Medico</p> <p><b>Metoclopramida 10 mg</b></p> <p><b>Tratamiento:</b> 8 días</p> <p><b>Tiempo:</b> Antes de cada comida</p> <p><b>Indicaciones:</b> Tomar una tableta antes de cada comida.</p>

*Ilustración 87: Iteración 5. Visualización del detalle de recetas*

*Elaborado por el investigador*

### 3.2.4.6 Iteración 6

#### Día de planificación

- **Buscador de Pacientes**

<b>Descripción</b>	
Como	Médico
Necesito	Buscar a los pacientes por cédula o apellido
Para	Encontrar a los pacientes de manera ágil
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, en su rol de médico, ingrese a la pantalla “Usuarios”
Cuando	Ingrese la cédula o apellido y seleccione la opción buscar
Entonces	La aplicación desplegará el historial de usuario en base al criterio de búsqueda

*Tabla 18: Iteración 6. Buscador de pacientes*

*Elaborado por el investigador*

El código establecido para realizar la Iteración 6 del proyecto es el siguiente:

## Usuario\_page

```
<ion-content>
  <div *ngIf="auth.isAuthenticated()">
    <div *ngIf="estado === 1">
      <ion-searchbar placeholder="Apellido o cedula" (ionChange)="onSearchChange($event)" [debounce]="1000"></ion-searchbar>
      <ion-item detail *ngFor="let usuario of usuarios" (click)="historial(usuario)">
        <ion-icon slot="start" name="person"></ion-icon>
        <ion-label>{{usuario.nombre}} {{usuario.apellidos}}</ion-label>
      </ion-item>
    </div>
  </div>
</ion-content>
```

Ilustración 88: Iteración 6. Desarrollo de buscador de pacientes

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración 6, donde el usuario con perfil de doctor podrá buscar por apellido o cédula del paciente.

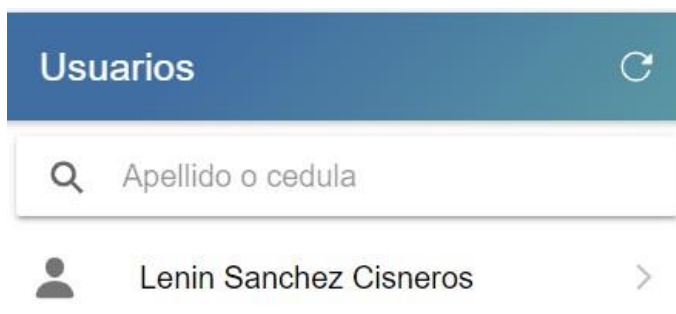


Ilustración 89: Iteración 6. Diseño del buscador de pacientes

Elaborado por el investigador

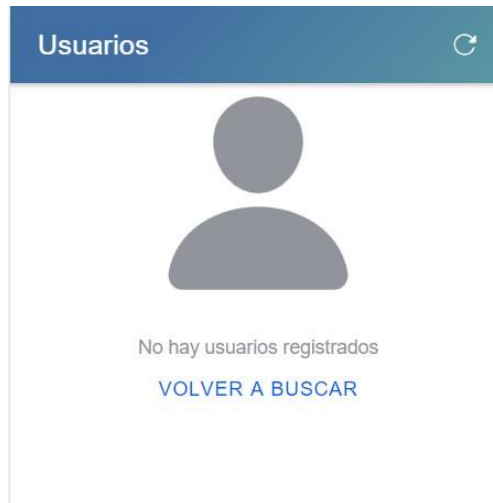


Ilustración 90: Iteración 6. Diseño de paciente no registrado

Elaborado por el investigador

### 3.2.4.7 Iteración 7

#### Día de planificación

- **Agregar Receta**

<b>Descripción</b>	
Como	Médico
Necesito	Agregar una alarma de la receta medica
Para	Notificar a sus pacientes en horarios de medicación
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, en su rol de médico, ingrese a la pantalla “Usuarios”
Cuando	Seleccione el usuario
Entonces	La aplicación habilitará los campos para recetar los medicamentos

Tabla 19: Iteración 7. Agregar recetas a pacientes

Elaborado por el investigador



El código establecido para realizar la Iteración 7 del proyecto es el siguiente:

```
<ion-title>Agregar Recetas</ion-title>
</ion-toolbar>
</ion-header>
<ion-content>
  <div class="ion-padding">
    <h6>Nombre: {{usuario.nombre}} {{usuario.apellidos}}</h6>
    <h6>Fecha: {{date | date:'fullDate'}}</h6>
    <hr>
  </div>
  <ion-card *ngFor="let receta of medicamentos" (click)="presentActionSheet(receta)">
    <ion-card-header>
      <ion-card-subtitle>Medico</ion-card-subtitle>
      <ion-card-title>{{receta.medicamento}}</ion-card-title>
    </ion-card-header>
    <ion-card-content>
      <strong>Tratamiento: </strong> {{receta.dias}} dias<br>
      <strong>Tiempo: </strong> {{receta.tiempo}} <br>
      <strong>Indicaciones: </strong>{{receta.indicaciones}}
    </ion-card-content>
  </ion-card>
  <ion-button (click)="presentMedicamentosModal()" expand="block">
    Agregar medicamento
  </ion-button>
</ion-content>
<ion-footer>
  <ion-toolbar>
    <ion-button (click)="recetar()" expand="full">
      Recetar
    </ion-button>
  </ion-toolbar>
</ion-footer>
```

Ilustración 91: Iteración 7. Desarrollo de la página agregar recetas

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración 7, donde el usuario con perfil de doctor podrá recetar a los pacientes.

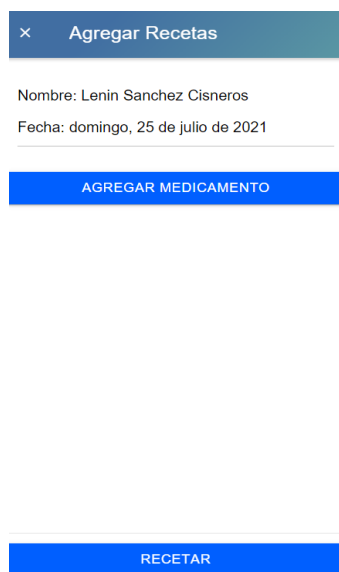


Ilustración 92: Iteración 7. Diseño para recetar medicamentos

Elaborado por el investigador

*Ilustración 93: Iteración 7. Diseño para agregar horario de medicación*

*Elaborado por el investigador*

### 3.2.4.8 Iteración 8

#### Día de planificación

- **Registro de Usuario**

<b>Descripción</b>	
Como	Usuario
Necesito	Registrar la información
Para	Tener acceso al sistema
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, ingrese a la pantalla “Registro de usuario”
Cuando	Llene los campos del formulario
Entonces	La aplicación registrará los datos del usuario en la base de datos

*Tabla 20: Iteración 8. Registro de datos personales del paciente*

*Elaborado por el investigador*

El código establecido para realizar la Iteración 8 del proyecto es el siguiente:

```
<ion-header class="bg_gradient">
  <ion-toolbar color="white">
    <ion-buttons slot="start">
      <ion-back-button [text]="Atras"></ion-back-button>
    </ion-buttons>
    <ion-title>Registro</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div class="ion-margin" [formGroup]="registerForm">
    <div class="ion-text-center">
      <ion-icon color="medium" style="font-size: 10rem;" name="person"></ion-icon>
      <br><br>
    </div>
    <div class="title">Datos personales</div>
    <ion-input formControlName="cedula" type="text" placeholder="Cedula"></ion-input>
    <ion-input formControlName="nombre" type="text" placeholder="Nombre"></ion-input>
    <ion-input formControlName="apellidos" type="text" placeholder="Apellidos"></ion-input>
    <ion-input formControlName="email" type="email" placeholder="Email"></ion-input>
    <ion-row>
      <ion-col size="6" class="no-padding-left">
        <ion-input formControlName="edad" type="number" placeholder="Edad"></ion-input>
      </ion-col>
      <ion-col size="6" class="no-padding-right">
        <div (click)="presentGeneroSheet()" class="input">{{registerForm.controls.genero.value}}</div>
      </ion-col>
    </ion-row>
    <ion-input formControlName="password" type="password" placeholder="Contraseña"></ion-input>
    <ion-input formControlName="confirm_password" type="password" placeholder="Confirmar contraseña"></ion-input>
  </div>
</ion-content>
```

Ilustración 94: Iteración 8. Desarrollo para el registro de pacientes

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración, donde el usuario podrá crear su perfil.



Ilustración 95: Iteración 8. Diseño del registro de pacientes

Elaborado por el investigador

### 3.2.4.9 Iteración 9

#### Día de planificación

- Editar datos personales de Usuario

<b>Descripción</b>	
Como	Usuario
Necesito	Visualizar la información ingresada
Para	Corregir los datos y actualizar la información en la base de datos
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, ingrese a la pantalla “Menú”
Cuando	Seleccione la opción modificar
Entonces	El sistema actualizará la información del usuario

Tabla 21: Iteración 9. Editar datos personales del paciente

Elaborado por el investigador

El código establecido para realizar la Iteración 9 del proyecto es el siguiente:

```
<ion-title>Menú</ion-title>
<ion-buttons slot="end" *ngIf="!editForm">
  <ion-button (click)="edit()">
    <ion-icon slot="start" name="pencil"></ion-icon>
  </ion-button>
</ion-buttons>
</ion-toolbar>
</ion-header>

<ion-content>
  <div class="ion-padding">
    <div class="ion-text-center ion-padding">
      
    </div>
  </div>
  <div class="ion-margin" [formGroup]="userForm">
    <div class="title">Datos personales</div>
    <ion-input [disabled]="!editForm" formControlName="cedula" type="text" placeholder="Cedula"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="nombre" type="text" placeholder="Nombre"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="apellidos" type="text" placeholder="Apellidos"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="email" type="email" placeholder="Email"></ion-input>
    <ion-row>
      <ion-col size="6" class="no-padding-left">
        <ion-input [disabled]="!editForm" formControlName="edad" type="number" placeholder="Edad"></ion-input>
      </ion-col>
      <ion-col size="6" class="no-padding-right">
        <div (click)="presentGeneroSheet()" class="input">{{userForm.controls.genero.value}}</div>
      </ion-col>
    </ion-row>
    <ion-input [disabled]="!editForm" formControlName="password" type="password" placeholder="Contraseña"></ion-input>
    <ion-input [disabled]="!editForm" formControlName="confirm_password" type="password" placeholder="Confirmar contraseña"></ion-input>
  </div>
</ion-content>
```

Ilustración 96: Iteración 9. Desarrollo de la página para editar los datos personales del paciente

Elaborado por el investigador

#### Día de liberación

Se evidencia la ejecución de la iteración 9, donde el usuario podrá actualizar los datos personales del perfil.

← ATRAS Menú

**Mediapp**

**Datos personales**

1804456737

Lenin

Sanchez Cisneros

ninel12008@hotmail.com

70 Masculino

.....

Guardar Datos >

*Ilustración 97: Iteración 9. Diseño de la página para editar los datos personales*

*Elaborado por el investigador*

### 3.2.4.10 Iteración 10

#### Día de planificación

- **Visualizar receta generada por Médico**

<b>Descripción</b>	
Como	Usuario
Necesito	Visualizar la receta generada por el médico
Para	Revisar los medicamentos que se debe ingerir
<b>Criterio de Aceptación</b>	
Dado	Que el usuario, ingrese a la pantalla “Alarmas”
Cuando	Seleccione la receta
Entonces	Se visualizará el medicamento que debe ingerir además del día de tratamiento, el tiempo y las indicaciones necesarias para ingerir.

*Tabla 22: Iteración 10. Visualización de la receta por parte del paciente*

*Elaborado por el investigador*

El código establecido para realizar la Iteración 10 del proyecto es el siguiente:

```
<ion-header class="bg_gradient">
  <ion-toolbar color="white">
    <ion-buttons slot="start">
      <ion-button (click)="menu()">
        <ion-icon slot="icon-only" name="person"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>
      Alarmas
    </ion-title>
    <ion-buttons slot="end">
      <ion-button (click)="init()">
        <ion-icon slot="start" name="reload"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div *ngIf="auth.isAuthenticated()">
    <div *ngIf="estado === 1">
      <ion-card *ngFor="let receta of localNotifications.notifications" (click)="detail(receta)">
        <ion-card-header>
          <ion-card-subtitle>{{receta.tipo | translate}}</ion-card-subtitle>
        </ion-card-header>
        <ion-card-content>
          <div *ngIf="receta.tipo === 'Medico'">
            <strong>Recetado por: </strong> {{receta.doctor.nombre}} {{receta.doctor.apellidos}}<br>
          </div>
            <strong>Fecha: </strong> {{receta.creado | date:'fullDate'}}<br>
            <strong>Medicamentos: </strong> {{receta.medicamentos.length}} <br>
          </ion-card-content>
        </ion-card>
      <br><br><br>
    </div>
  </div>
</ion-content>
```

Ilustración 98: Iteración 10. Desarrollo de la visualización de la receta

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración 10, donde el usuario podrá visualizar su receta.



Ilustración 99: Iteración 10. Diseño para visualizar las recetas

Elaborado por el investigador



**Lorazepam 1mg**

**Tratamiento:** 4 días

**Tiempo:** Merienda

**Indicaciones:** Tomar una tableta al día después de la merienda.

*Ilustración 100: Iteración 10. Diseño del detalle de la receta*

*Elaborado por el investigador*

### 3.2.4.11 Iteración 11

#### Día de Planificación

- **Ingresar alarma personalizada**

<b>Descripción</b>	
Como	Usuario
Necesito	Ingresar una alarma personalizada
Para	Guardar mis medicamentos recetados por otro médico
<b>Criterio de Aceptación</b>	
Dado	Que el usuario ingrese a la pantalla “Alarmas”
Cuando	Seleccione agregar alarma
Entonces	La aplicación Ingresará la información recetada por el medico aparte.

*Tabla 23: Iteración 11. Alarma personalizada*

*Elaborado por el investigador*

El código establecido para realizar la Iteración 11 del proyecto es el siguiente:

```
<ion-header class="bg_gradient">
  <ion-toolbar color="white">
    <ion-buttons slot="start">
      <ion-button (click)="modalController.dismiss({status:false})">
        <ion-icon slot="start" name="close"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>Agregar alarma</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div class="ion-margin" [formGroup]="recetaForm">
    <div class="ion-text-center">
      <ion-icon color="medium" style="font-size: 10rem; name="alarm"></ion-icon>
      <br><br>
    </div>
    <!-- <ion-text>Medicamento:</ion-text> -->
    <ion-input type="text" FormControlName="medicamento" placeholder="Medicamento"></ion-input>
    <!-- <ion-text>Días:</ion-text> -->
    <ion-input type="number" FormControlName="dias" placeholder="Días del tratamiento"></ion-input>
    <!-- <ion-text>Tiempo:</ion-text> -->
    <div class="input vertical-align-left" (click)="presentTimePicker()">{{recetaForm.controls.tiempo.value}}</div>
    <!-- <ion-text>Indicaciones:</ion-text> -->
    <ion-textarea rows="3" FormControlName="indicaciones" placeholder="Indicaciones (mínimo 10 caracteres)"></ion-textarea>
    <br>
    <!-- <div class="ion-text-center">
      <ion-note>Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus culpa, officii sint exercitationem iure.
    </div>
    <br> -->
    <ion-button (click)="validForm()" [disabled]="!recetaForm.valid" expand="full">
      <ion-icon slot="start" name="alarm"></ion-icon>
      Agregar alarma
    </ion-button>
  </div>
</ion-content>
```

Ilustración 101: Iteración 10. Desarrollo de la página para agregar una alarma personalizada

Elaborado por el investigador

## Día de liberación

Se evidencia la ejecución de la iteración 10, donde el usuario podrá agregar una alarma personalizada.



Ilustración 102: Iteración 10. Diseño para agregar una alarma personalizada

Elaborado por el investigador



### 3.2.5 Fase de estabilización

#### 3.2.5.1 Publicación de la API Rest

Se subió la API Rest desarrollada en Laravel al hosting mencionado en la ilustración 49, para que pueda conectarse con el ambiente de producción, permitiendo así el funcionamiento de la aplicación móvil.

```
export const environment = {  
  production: true,  
  API_URL: 'https://guambraguaytambo.com/api/public/api',  
};
```

*Ilustración 103: Ambiente de Producción*

*Elaborado por el investigador*

#### 3.2.5.2 Generación de APK, del aplicativo móvil

Para generar la APK se deberá generar el release de producción en la terminal de Ionic

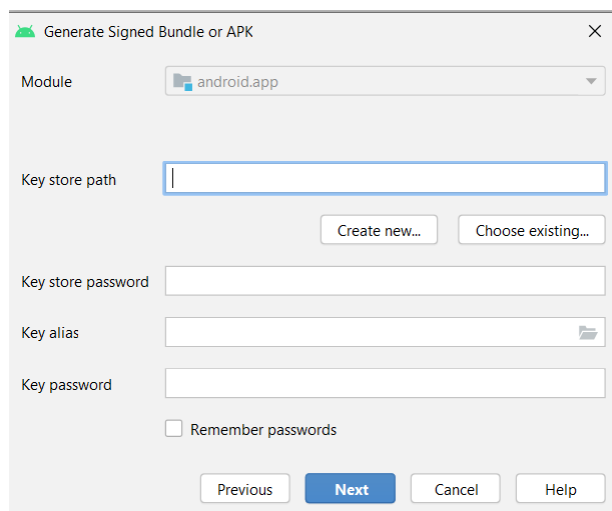
- `ionic cordova build android --prod --release`

```
PS C:\Users\DELL 2018\Desktop\si\proyecto_final\Nuevo\app-medi> ionic cordova build android --prod --release
```

*Ilustración 104: Comando para generar el releas de producción*

*Elaborado por el investigador*

Abrir Proyecto en android Studio para generar el APK firmada



*Ilustración 105: Generar APK firmada*

*Elaborado por el investigador*

### 3.2.6 Fase de Pruebas

Para su descarga, instalación y utilización la aplicación puede encontrarse en el siguiente enlace:

<https://drive.google.com/file/d/11CzeETWiLYUAn5BsggRwOdIlhShkbgBf/view?usp=sharing>

Como se puede observar en la ilustración 105, en el aplicativo móvil, la fecha se ejecuta en inglés, donde la mayoría de usuarios al ser adultos mayores tendrán dificultad de entender la fecha en la que fue recetada por lo cual se procederá cambiar al idioma español.



*Ilustración 106: Pruebas de ejecución*

*Elaborado por el investigador*

Como se muestra en la ilustración 106 se ha solucionado el problema de idioma, por ende, los usuarios tendrán un entorno de trabajo amigable.



*Ilustración 107: Solución del problema de idioma*

*Elaborado por el investigador*

### **3.2.6.1 Muestra del funcionamiento de la aplicación**

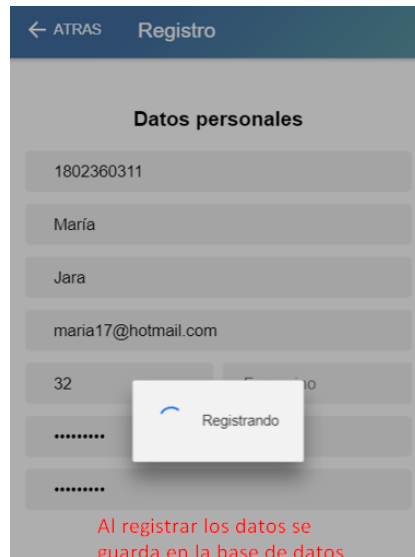
A continuación, se presentará el funcionamiento del aplicativo móvil cumpliendo todas las iteraciones establecidas en la metodología.

#### **Registro de Usuario**

El paciente podrá crear una cuenta de usuario donde quedará registrado en la base de datos.

*Ilustración 108: Funcionamiento de la aplicación. Registro de usuario*

*Elaborado por el investigador*



*Ilustración 109: Funcionamiento de la aplicación. Registrando Usuario*

*Elaborado por el investigador*

## **Editar datos personales**

El usuario tendrá la facilidad de editar sus datos personales en caso de que algún campo sea erróneo.



*Ilustración 110: Funcionamiento de la aplicación. Editar datos personales*

*Elaborado por el investigador*



*Ilustración 111: Funcionamiento de la aplicación. Editar Datos*

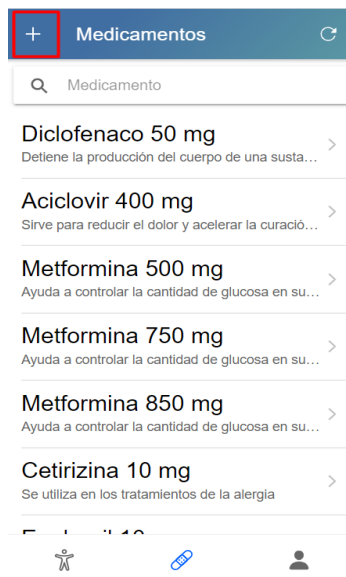
*Elaborado por el investigador*



*Ilustración 112: Funcionamiento de la aplicación. Editar Datos*

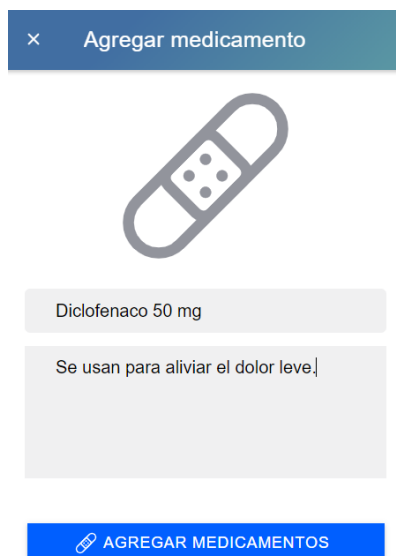
*Elaborado por el investigador*

## Agregar medicamento



*Ilustración 113: Funcionamiento de la aplicación. Agregar medicamentos*

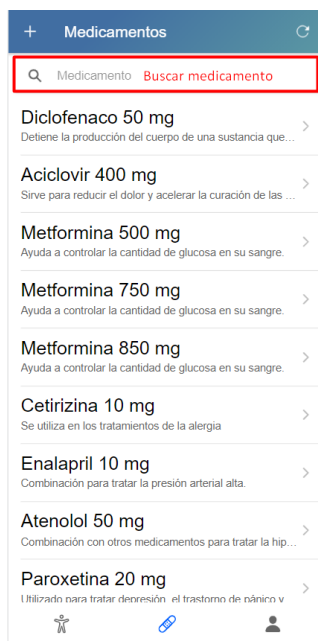
*Elaborado por el investigador*



*Ilustración 114: Funcionamiento de la aplicación. Agregar medicamentos*

*Elaborado por el investigador*

## Búsqueda del medicamento



*Ilustración 115: Funcionamiento del aplicativo móvil. Buscar medicamentos*

*Elaborado por el investigador*



*Ilustración 116: Funcionamiento del aplicativo móvil. Buscar medicamentos*

*Elaborado por el investigador*

## Agregar recetas médicas

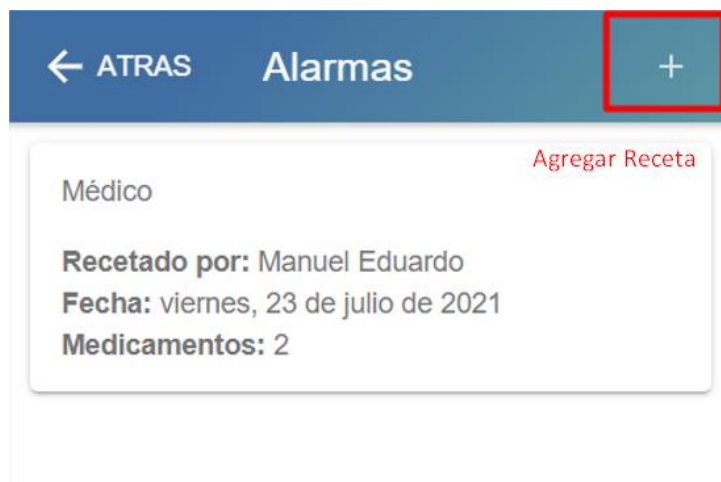


Ilustración 117: Funcionamiento del aplicativo móvil. Agregar receta médica

Elaborado por el investigador

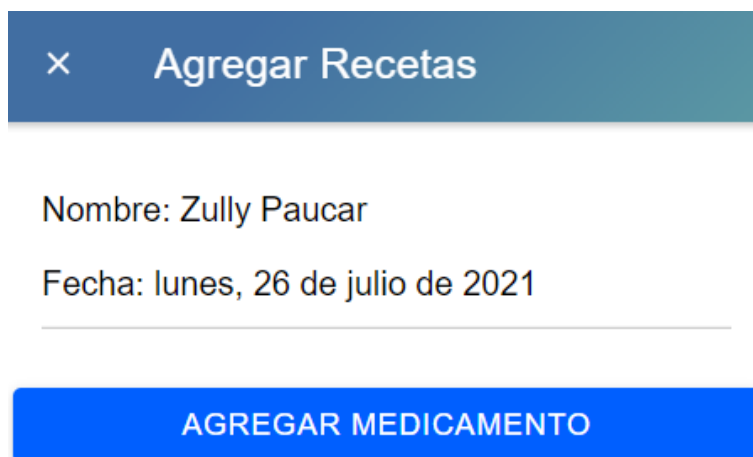
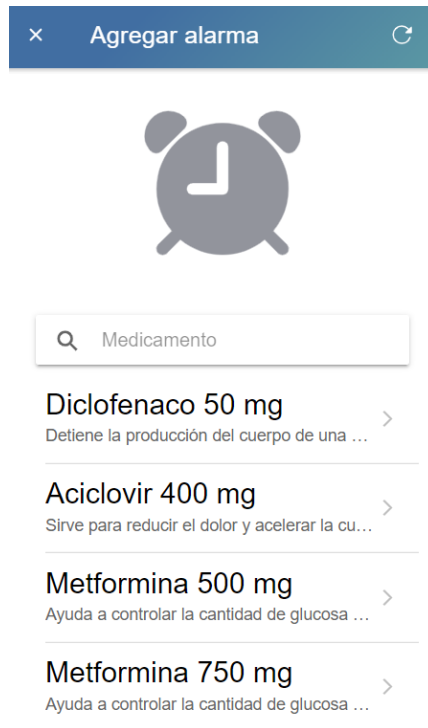


Ilustración 118: Funcionamiento del aplicativo. Agregar medicamento que se recetará

Elaborado por el investigador





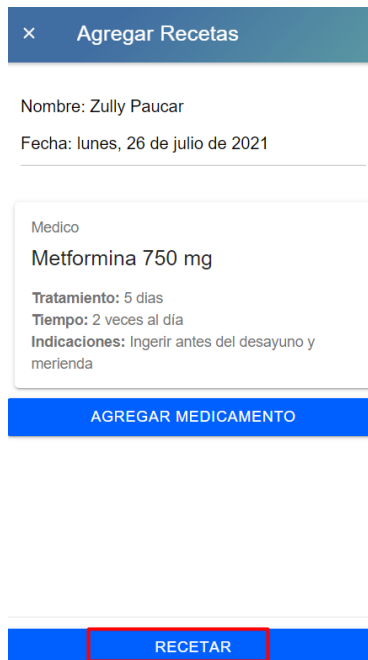
*Ilustración 119: Funcionamiento del aplicativo. Selección del medicamento*

*Elaborado por el investigador*



*Ilustración 120: Funcionamiento del aplicativo. Agregar información para la receta*

*Elaborado por el investigador*



*Ilustración 121: Funcionamiento del aplicativo. Recetar a paciente*

*Elaborado por el investigador*

## Visualización de la receta por parte del paciente



*Ilustración 122: Funcionamiento del aplicativo. Visualización de la receta*

*Elaborado por el investigador*



*Ilustración 123: Funcionamiento del aplicativo. Visualización del detalle de la receta.*

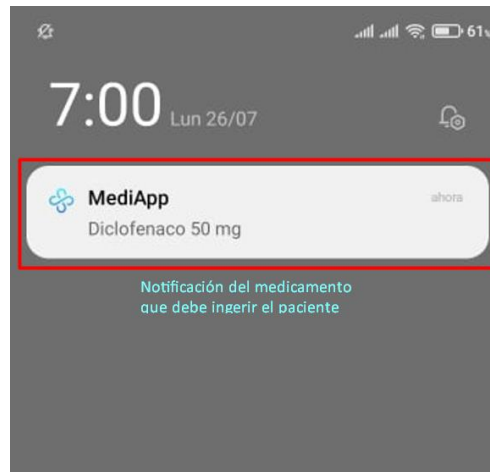
*Elaborado por el investigador*

## **Aplicación en dispositivo android**



*Ilustración 124: Funcionalidad del aplicativo. Aplicación en el dispositivo móvil*

*Elaborado por el investigador*



*Ilustración 125: Funcionalidad del aplicativo. Notificación en el dispositivo móvil*

*Elaborado por el investigador*

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- Se desarrolló el aplicativo móvil “Mediapp”, con la finalidad de realizar el control y registro de tratamientos médicos en varias plataformas móviles, mediante la metodología Mobile D, reduciendo los tiempos empleados en el desarrollo de la APP y siendo interactiva con el usuario
- Se llevó a cabo el análisis sobre la toma de medicamentos en pacientes del Centro Médico Familiar del Dr. Manuel Cruz, tomando en cuenta factores como dosificaciones y horarios, donde se pudo concluir que el tratamiento y la prescripción médica en cada paciente será diferente, dependiendo de su estilo de vida y la edad en que se encuentre en la actualidad.
- Se realizó una investigación sobre los diferentes frameworks disponibles para la elaboración del aplicativo Mediapp, siendo el framework Ionic el más factible para la creación de interfaces, ya que, al poseer una gran variedad de lenguajes y compatibilidad con varias plataformas, lo hace óptimo para el desarrollo de aplicaciones móviles multiplataforma; además, su alta reusabilidad redujo el tiempo empleado en la codificación.
- La aplicación Mediapp, fue diseñada en base a la recolección de datos tomados en el Centro Médico Familiar del Dr. Manuel Cruz; siendo factor determinante la constante comunicación que se mantuvo con el Doctor, para facilitar la elaboración del aplicativo obtenido.
- Se pudo establecer que los pacientes del Centro Médico Familiar, en distintos rangos de edades, muestran dificultades para recordar el nombre de algunos medicamentos o dosificaciones. Además, las personas encuestadas disponen de dispositivos Smart, donde pueden recibir los recordatorios programados en el aplicativo móvil.

## 4.2 Recomendaciones

- Se recomienda realizar una base de datos más extensa de los medicamentos, para futuros proyectos, ya que, permitirá realizar más combinaciones de medicamentos y brindar un mejor servicio a pacientes con casos especiales o críticos.
- Se recomienda usar otra alternativa de librería a más de Local notification, debido a que limita la compatibilidad de la aplicación realizada con algunas versiones de Android.
- Para mantener la aplicación actualizada, es recomendable investigar periódicamente los medicamentos que salen o entran al mercado farmacéutico, y considerar no solamente sus beneficios sino también los efectos que puede causar a largo plazo el consumo excesivo de determinado medicamento.

## BIBLIOGRAFÍA

- [1] D. Castro Villena, “Sistema de Gestión y Control de Medicación,” tesis, Universidad Técnica de Catalunya”, 2017.
- [2] C. Aragonés Lacarta, M. J. Blesa Aguilera, “Aplicaciones para dispositivos móviles basadas en NFC para el control, gestión y monitorización de la medicación en pacientes de larga duración,” tesis, “Universidad Politécnica de Catalunya”, 2016.
- [3] H. Ayrton Ruiz, F. Jhino Arias, “Aplicación Web Y Móvil De Monitoreo Y Control,” tesis, “Escuela Profesional de Ingeniería de computación y Sistemas”, 2017.
- [4] K. Atiencia Pogo, “Aplicación móvil multiplataforma que brinda información de las unidades de salud públicas y privadas de la ciudad de Loja”, tesis, Universidad Nacional de Loja, 2017.
- [5] L. L. Zullig *et al.*, “The new landscape of medication adherence improvement: Where population health science meets precision medicine,” *Patient Prefer. Adherence*, vol. 12, pp. 1225–1230, 2018.
- [6] C. Toapanta Edison, “Aplicación basada en arquitectura orientada a servicios para la gestión de la información de pacientes hipertensos”, tesis, Universidad Técnica de Ambato, 2018.
- [7] J. Pedro Barba and F. Berenguer, “Diseño y desarrollo web. Análisis de casos,” ES, *Patient*, 39, 2019.
- [8] R. Angulo, “Criterios para seleccionar la mejor opción para el desarrollo de aplicaciones,” *Patient*. 80–82, 2018.
- [9] L. Valencia Cabrera, «Metodologías ágiles de desarrollo de Software aplicadas a la gestión de proyectos empresariales,» *Natural Computing Site*, 11 11 2019. [En línea]. Available: <https://www.cs.us.es/cursos/bd/temas/BD-Tema-5.pdf>.
- [10] A. A. Pérez García and E. Egea López, “Desarrollo de herramientas web de gestión docente,” *Esc. Técnica Super. Ing. Telecomunicación*, vol. 0, no. 0, *Patient* 1–87, 2017.
- [11] D. Maya, E., y López, “Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web,” *Septima Conf. Dir. Tecnol. Inf. 2017*, no. July, *Patient* 12, 2017.

- [12] A. J. G. Sánchez, «Laravel 5,» 21 03 2018. [En línea]. Available: <https://www.lawebdelprogramador.com/pdf/9052-Laravel-5.html>.
- [13] V. Vázquez Rodríguez, “Desarrollo de aplicaciones móviles multiplataforma con Flutter,” *Univ. Almer.*, Patient 72, 2019.
- [14] E. Lerida Martinez, “Escuela Politécnica Superior de Linares,” Patient 1–89, 2019.
- [15] R. López Menéndez de Jiménez, “Metodologías ágiles de desarrollo de Software aplicadas a la gestión de proyectos empresariales,” Patient. 6–11, 2015.
- [16] M. Antonio Fonseca Vargas, Engels Ismael Obregón Hernandez, “Manifiesto Ágil,” Patient 68–70, 1377.
- [17] J. S. M. Cristian Camilo Narvaez Useche, Santiago Acosta Maldonado, “Aplicación de la metodología scrum e implementación de la estrategia de servicio con base al marco de referencia itil v 3.0 en el desarrollo de una aplicación web para una empresa de transporte público,” *Ayaañ*, vol. 8, no. 5, p. 55, 2019.
- [18] P. Esteban Medina, “Análisis, diseño y desarrollo de un sistema informático para la obtención de indicadores de sostenibilidad: Ambientales y sociales para la producción de leche en fincas del cantón Cayambe”, tesis, “Universidad Politécnica Salesiana Sede Quito”, pp. 1–63, 2018.
- [19] A. Collazo Garcia, M. Diaz Labrador, «La Programación Extrema,» 27 11 2018. [En línea]. Available: [https://www.researchgate.net/publication/318211906\\_La\\_programacion\\_extrema](https://www.researchgate.net/publication/318211906_La_programacion_extrema).
- [20] C. Antonio Muñoz, “Aplicación de la metodología mobile-d en el desarrollo de una app móvil para gestionar citas médicas del centro JEL Riobamba”, tesis, “Universidad Nacional De Chimborazo Carrera De Ingeniería En Sistemas Y,” 2020.
- [21] «Calculadora del tamaño de muestra,» 2021. [En línea]. Available: <https://es.surveymonkey.com/mp/sample-size-calculator/>



## Aplicación móvil multiplataforma para el control y registro de tratamientos médicos

\*Obligatorio

---

¿En qué rango de edad se encuentra usted? \*

- De 18 a 25 años
- De 26 a 35 años
- De 36 a 45 años
- De 46 a 55 años
- De 56 a 65 años
- Más de 65 años

¿Dispone usted de un teléfono inteligente? \*

- Si
- No

---

Si su respuesta anterior fue afirmativa. ¿Cuál es el sistema operativo de su dispositivo móvil? \*

- Android
- IOS
- Otro: \_\_\_\_\_

¿De dónde acostumbra descargar las aplicaciones? \*

Google Play

App Store (Apple)

Windows Phone Store

Amazon Appstore

Otro: \_\_\_\_\_

---

¿Qué tan probable es que usted aprenda a manejar una aplicación móvil que le ayude a controlar y cuidar su salud? \*

Muy probable

Poco probable

Nada probable

¿Qué grado de dificultad tiene usted para recordar el nombre de los medicamentos prescritos por el médico? \*

- Alto
  - Medio
  - Bajo
  - Ninguno
- 

¿Ha utilizado alguna aplicación que le permita llevar registro de sus horarios en la que debe ingerir sus medicamentos? \*

- Si
- No

¿Con que facilidad usted recuerda la dosificación de los medicamentos prescritos por el médico? \*

- Alta
- Media
- Baja

---

¿De qué forma preferiría recibir los avisos y/o recordatorios de su aplicativo móvil? \*

- Mensajes de texto (SMS)
- Correo electrónico
- Notificación del dispositivo móvil
- Otro: \_\_\_\_\_

¿Le gustaría que su receta se encuentre en su dispositivo móvil? \*

- Si
- No