



**UNIVERSIDAD
NACIONAL DE LOJA**



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables.

CARRERA DE INGENIERIA EN SISTEMAS

**“ARQUITECTURA INFORMÁTICA PARA GEO
LOCALIZACIÓN DE MEDICAMENTOS EN
FARMACIAS”**

“Trabajo de Titulación previo a la obtención
del título de Ingeniero en Sistemas”

Autor:

- Leonardo Favio Caraguay Caraguay

Director:

- José Oswaldo Guamán Quinche, Mg. Sc.

Tutor Académico:

- Edison Leonardo Coronel Romero, Mg. Sc

Loja-Ecuador

2019

CERTIFICACIÓN

Loja, 18 de marzo de 2019

Ing. José Oswaldo Guamán Quinche, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS, DE LA FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Que el Sr. Leonardo Favio Caraguay, egresado de la carrera de Ingeniería en Sistemas ha trabajado bajo mi tutoría el presente trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas, cuyo tema versa sobre **“ARQUITECTURA INFORMÁTICA PARA GEO LOCALIZACIÓN DE MEDICAMENTOS EN FARMACIAS”**, el mismo que ha sido dirigido, orientado y discutido bajo mi asesoramiento y cumple con la reglamentación pertinente, así como lo programado en el plan del proyecto, razones por las cuales reúne la suficiente validez técnica y práctica, por consiguiente autorizo su presentación y sustentación.

A handwritten signature in blue ink, appearing to be 'José Oswaldo Guamán Quinche', written over a horizontal dotted line.

Ing. José Oswaldo Guamán Quinche, Mg. Sc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

AUTORÍA

Yo, **LEONARDO FAVIO CARAGUAY CARAGUAY**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

Firma:

A handwritten signature in blue ink, appearing to be 'L. Caraguay', enclosed within a blue circular scribble.

Cédula: 1105903437

Fecha: Loja, 26 de febrero del 2020.

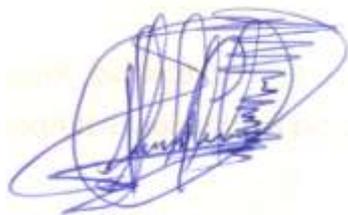
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo **LEONARDO FAVIO CARAGUAY CARAGUAY**, declaro ser el autor de la tesis titulada: “**ARQUITECTURA INFORMÁTICA PARA GEO LOCALIZACIÓN DE MEDICAMENTOS EN FARMACIAS**”, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización en la ciudad de Loja, a los veintiséis días del mes de febrero del dos mil veinte.



Firma:

Autor: Leonardo Favio Caraguay Caraguay.

Cedula: 1105903437.

Dirección: Loja, Ciudadela Samana.

Correo electrónico: lfcaraguayc@unl.edu.ec.

Teléfono: Celular: 0983886370.

DATOS COMPLEMENTARIOS:

Director de Tesis: Ing. José Oswaldo Guamán Quinche, Mg. Sc.

Tribunal de Grado: Ing. Ordoñez Ordoñez Pablo Fernando, Mg. Sc.

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.

Ing. Wilman Patricio Chamba Zaragocin, Mg. Sc.

Agradecimiento

Agradezco a Dios y la Virgen del Cisne por permitirme culminar con éxito esta etapa de mi formación académica y por el presente trabajo de titulación. Y de manera muy especial a mis padres y familiares que con su apoyo constante e incondicional hicieron de esta formación un éxito.

Además expreso mis sinceros agradecimientos a todos los docentes que formaron parte de mi instrucción académica que con sus conocimientos y experiencias se convirtieron en un pilar fundamental para la culminación exitosa de mi carrera profesional.

Mi agradecimiento al Mg. Sc. Patricio Valarezo gerente de PupilaBox por permitirme realizar mis practicas pre profesionales en su empresa donde pude adquirir conocimientos importantes para el desarrollo del presente proyecto.

Quisiera terminar mostrando mis sentimientos de gratitud hacia el director del presente trabajo de titulación el Mg. Sc. José Oswaldo Guamán Quinche, por el tiempo brindado y por su constante apoyo para la culminación exitosa del mismo.

Dedicatoria

El presente trabajo de titulación va dedicado a Dios y a la Virgen del Cisne por darme la salud, la sabiduría y las fuerzas para seguir adelante en los momentos difíciles.

A mis padres que son la pieza fundamental en todo lo que realizo, por apoyarme constantemente de todas las formas posibles, por sus consejos que fueron siempre la guía para terminar exitosamente mis carrera profesional.

A mis hermanos y hermanas que son siempre mi apoyo y fuerza en todas mis actividades y por ser los cómplices en todo lo que realizo.

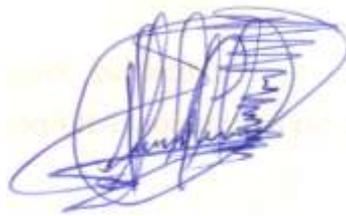
A mis abuelitos y demás familiares que siempre fueron un apoyo en mi formación académica.

A mis compañeros y amigos que formaron parte de este proceso de aprendizaje y compartieron varios momentos que el paso por esta etapa de formación nos deja.

Finalmente quiero dedicar este trabajo a todos los que formaron parte de mi instrucción tanto profesional como personal como los docentes de la carrera de ingeniería en sistemas, tutores de pasantías pre profesional y demás.

CESIÓN DE DERECHOS

Leonardo Favio Caraguay Caraguay, autor intelectual del presente Proyecto de Fin de Carrera, autorizo a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, y específicamente a la Carrera de Ingeniería en Sistemas, el total acceso a su contenido en lo que consideren necesario.



Leonardo Favio Caraguay Caraguay

C.I. 1105903437

Índice de Contenidos

CERTIFICACIÓN	2
AUTORÍA.....	III
Agradecimiento	V
Dedicatoria.....	6
CESIÓN DE DERECHOS.....	7
A. Título.....	VIII
B. Resumen.....	2
Summary.....	3
Índice de Contenidos	VIII
Índice de figuras	XI
Índice de tablas	14
C. Introducción	4
D. Revisión de literatura	6
1. Situación del mercado farmacéutico en Ecuador.	6
1.1 Mercado farmacéutico y la tecnología.	6
2. Arquitectura informática	6
2.1 Definición	6
2.2 Elementos	6
2.3 Tipos de Software	7
3. Geo localización	7
4. Framework para el desarrollo de aplicaciones móviles	10
5. Herramientas y tecnologías utilizadas para el desarrollo.	14
5.1 Web Service	14
6.1 Metodología de desarrollo robusta o tradicional.	23
6.2 Metodologías de desarrollo ágil.	23
E. Materiales y métodos	28
1. Métodos de investigación	28
1.1 Estudio de casos.	28
1.2 Observación activa:	28

2.	Técnicas de recolección de información.	28
2.1	Encuesta	28
2.2	Entrevista	29
2.3	Investigación Bibliográfica	29
3.	Metodología de desarrollo de software	29
4.	Metodología para la revisión de literatura	30
4.1.	Paso Uno: Definición de fuentes bibliográficas	30
4.2.	Paso Dos: Definición de palabras clave	30
4.3.	Paso Tres: Búsqueda de artículos	30
4.4.	Paso Cuatro: Criterios de exclusión	31
4.5	Paso Cinco: Análisis de Resultados	31
F.	Resultados	32
1.	Definición de una arquitectura informática.	32
1.1	Análisis de enfoques de desarrollo de aplicaciones.	32
1.2	Análisis comparativo de Web Services.	33
1.3	Análisis comparativo de Frameworks.	34
1.4	Análisis comparativo sobre los mapas Google Maps y Mapbox.	36
1.5	Análisis comparativo de metodologías.	37
1.6	Arquitectura de Software	39
2.	Desarrollo de la arquitectura definida	54
2.1	Exploración	54
2.2	Iniciación.	59
2.3	Producción	70
2.4	Estabilización	86
3.	Evaluar que la arquitectura permita la geo localización de los medicamentos en las diferentes farmacias.	89
3.1	Pruebas	89
3.2	Reporte de pruebas.	99
G.	Discusión	115
1.	Desarrollo de la propuesta alternativa.	115
2.	Valoración técnica económica ambiental	117
H.	Conclusiones	120
I.	Recomendaciones	121
J.	Referencias	122
K.	Anexos	125
1.	Anexo 1: Especificación de requisitos de software ERS.	125
1.1	INTRODUCCIÓN	125
1.2	Propósito	125

1.3	Alcance.....	126
1.4	Personal involucrado.....	126
1.5	Definiciones, Acrónimos y Abreviaturas.	127
1.6	Visión General del Documento.	127
1.7	Descripción General.....	127
1.8	Requisitos específicos.....	128
1.9	Requisitos funcionales.....	129
1.10	Requisitos no funcionales.....	135
1.11	Apéndices.....	136
	Apéndice 4: Resultados de la Encuesta.....	139
2.	Anexo 2: Pruebas de Aceptación.	144
2.1	PRUEBAS DE ACEPTACIÓN SEO.....	144
2.2	Resultados de la encuesta de pruebas de aceptación Seo.....	148
1)	Registro.....	148
	148	
2)	Inicio de Sesión.....	148
	148	
3)	Mi ubicación.....	149
	149	
4)	Farmacias en el mapa.....	149
	149	
2.3	Pruebas de Aceptación Pharma.....	152
2.4	Resultados de pruebas de aceptación Pharma.....	155
2.5	Interpretación de los resultados.	156
3.	Anexo 3: Certificado de traducción.....	157
4.	Anexo 4: Licencia.	158
	¡Error! Marcador no definido.	

Índice de figuras

Figura 1: Mapa Con Marcador [7].	8
Figura 2: Google Maps [9].	9
Figura 3: Open Streetmap [11].	10
Figura 4: Formato Soap [22].	15
Figura 5: Web Service Soap	16
Figura 6: Web Service Api Rest	18
Figura 7: Service GraphQL [25]	19
Figura 8: Fases De La Metodología Mobile-D [31].	26
Figura 9: Diseño De La Arquitectura	61
Figura 10: Diagrama De Clases	63
Figura 11: Diagrama De Casos De Uso	64
Figura 12: Prototipo Pantalla Inicio De Sesión	65
Figura 13: Prototipo Pantalla Inicio	65
Figura 14: Prototipo Pantalla Nueva Consulta	66
Figura 15: Prototipo Menú Lateral	66
Figura 16: Prototipo Mi Perfil	67
Figura 17: Prototipo Pantalla Registro	67
Figura 18: Prototipo Información Farmacia	68
Figura 19: Prototipo Inicio Usuario	68
Figura 20: Prototipo Nueva Consulta	69
Figura 21: Prototipo Resultados	69
Figura 22: Directorio De Proyecto Ionic En Visual Studio Code.	74
Figura 23: Proyecto En Repositorio Git	74
Figura 24: Código Vista De Registro De Usuario	76
Figura 25: Código De La Vista De Inicio De Sesión.	77
Figura 26: Código Vista De Selección De Estado	77
Figura 27: Código De Vista De Modal De Nuevas Consultas	78
Figura 28: Código Para Crear Un Mapa Con Mapbox.	79
Figura 29: Código Para Obtener Ubicación Actual	80
Figura 30: Código Nueva Consulta	81

Figura 31: Código Para Mostrar Los Resultados De Las Consultas	81
Figura 32: Código Inicio De Sesión (Servidor)	83
Figura 33: Código Para Registrar Usuarios (Servidor)	84
Figura 34: Configuración De Suscripción Para Mensajes En Tiempo Real.....	85
Figura 35: Código De Mensaje En Tiempo Real	85
Figura 36: Código Archivo Config	87
Figura 37: Código De Métodos Disponibles En El Servidor.	88
Figura 38: Prueba De Soporte De Datos De Entrada	101
Figura 39: Prueba De Soporte Ubicación	101
Figura 40: Prueba De Soporte No Conexión	102
Figura 41: Caso De Prueba Mi Ubicación	103
Figura 42: Caso De Prueba Inicio De Sesión	103
Figura 43: Caso De Prueba De Registro.	104
Figura 44: Caso De Prueba Mostrar Farmacias	104
Figura 45: Caso De Prueba Búsqueda De Farmacia	105
Figura 46: Caso De Prueba Detalles De Farmacia.....	105
Figura 47: Caso De Prueba Ruta Hacia La Farmacia	106
Figura 48: Caso De Prueba Búsqueda De Medicamento.....	106
Figura 49: Caso De Prueba Cambio De Estado	107
Figura 50: Caso De Prueba Nueva Consulta	107
Figura 51: Resultado De Caso De Prueba Registro	148
Figura 52: Resultado Caso De Prueba Inicio De Sesión.....	148
Figura 53: Resultado Caso De Prueba Mi Ubicación	149
Figura 54: Resultado Caso De Prueba Farmacias En El Mapa	149
Figura 55: Resultado Caso De Prueba Información De Las Farmacias.....	150
Figura 56: Resultado Caso De Prueba Buscar Farmacias	150
Figura 57: Resultado Caso De Prueba Buscar Medicamentos	151
Figura 58: Resultado Caso De Prueba Mostrar Ruta	151
Figura 59: Resultado Caso De Prueba Registro Pharma	155

Figura 60: Resultado Caso De Prueba Inicio De Sesión Pharma	155
Figura 61: Resultado Caso De Prueba Seleccionar Estado Pharma	155
Figura 62: Resultado Caso De Prueba Nuevas Consultas Pharma	155
Figura 63: Resultado Caso De Prueba Contestar Nuevas Consultas Pharma	156
Figura 64: Resultado Caso De Prueba Notificación Nueva Consulta Pharma	156
Figura 65: Resultado Caso De Prueba Visualización De Información Pharma	156
Figura 66: Diagrama De Casos De Uso	42
Figura 67: Modelo Conceptual	46
Figura 68: Diagrama De Clases Definición De Arquitectura	47
Figura 69: Diagrama De Actividad Generar Consulta	48
Figura 70: Diagrama De Actividad Elegir Estado	49
Figura 71: Diagrama De Actividad Autenticación Y Registro	50
Figura 72: Diagrama De Actividad Generar Respuesta	51
Figura 73: Diagrama De Despliegue	52
Figura 74: Arquitectura De Software	53

Índice de tablas

TABLA I: Cuadro Comparativo De Aplicaciones	32
TABLA II: Cuadro Comparativo De Web Services	33
TABLA III: Análisis De Frameworks De Desarrollo	35
TABLA IV: Cuadro Comparativo De Mapas	36
TABLA V: Análisis Comparativo De Metodologías[23].....	37
TABLA VI: Requerimientos Funcionales	55
TABLA VII: Requerimientos No Funcionales	56
TABLA VIII: Módulos Iniciales.....	57
TABLA IX: Casos De Pruebas	90
TABLA X: Caso De Prueba Mi Ubicación	92
TABLA XI: Caso De Prueba Inicio De Sesión.....	93
TABLA XII: Caso De Prueba Registro.	94
TABLA XIII: Caso De Prueba Visualizar Las Farmacias.	94
TABLA XIV: Caso De Prueba Información De Farmacia.....	95
TABLA XV: Caso De Prueba Búsqueda De Farmacia.....	95
TABLA XVI: Caso De Prueba Ruta.....	96
TABLA XVII: Caso De Prueba Buscar Medicamento.....	96
TABLA XVIII: Caso De Prueba Cambio De Estado De Farmacia	97
TABLA XIX: Caso De Prueba Nuevas Consultas.....	97
TABLA XX: Pruebas De Interfaz.....	100
TABLA XXI: Resultados Casos De Pruebas.....	108
TABLA XXII: Recursos Humanos	117
TABLA XXIII: Recursos Materiales	118
TABLA XXIV: Recursos Técnicos Y Tecnológicos.....	118
TABLA XXV: Total De Recursos Usados	119
TABLA XXVI: Definición, Siglas Y Abreviaturas.....	40
TABLA XXVII: Referencias Definición De Arquitectura	40
TABLA XXVIII: Arquitectura De Vistas 4+1	41
TABLA XXIX: Caso De Uso Crud	43
TABLA XXX: Caso De Uso Generar Consulta	43

TABLA XXXI: Caso De Uso Respuesta De Consultas.....	44
TABLA XXXII: Caso De Uso Visualizar Respuesta	44
TABLA XXXIII: Caso De Uso Modificar Estado	45
TABLA XXXIV: Caso De Uso Visualizar Estado	45

A. Título

**“ARQUITECTURA INFORMÁTICA PARA GEO
LOCALIZACIÓN DE MEDICAMENTOS EN FARMACIAS”**

B. Resumen

En el presente trabajo de titulación se redacta el desarrollo de una arquitectura informática que tiene como objetivo principal optimizar la búsqueda y ubicación de medicamentos en las diferentes farmacias además de conocer si la farmacia se encuentra abierta a través de dos aplicaciones móviles y un web service o servicio web. La primera aplicación móvil dirigida a los clientes de las farmacias donde les permite buscar medicamentos y conocer información necesaria de los mismos como su ubicación y disponibilidad entre otros, la segunda aplicación móvil dirigida a las farmacias que permite la interacción con el cliente a través de consultas en tiempo real conociendo los detalles del medicamento buscado por el cliente y permitiendo emitir una respuesta ante estas consultas y finalmente el servicio web encargado del manejo de los datos y de toda la interactividad entre las aplicaciones móviles.

El desarrollo del presente trabajo de titulación se dividió en tres metas como son la definición de la arquitectura, el desarrollo y las pruebas de la misma. En primera instancia fue necesario una revisión de la literatura con temas relacionados sobre arquitecturas informáticas, tipos de software, aplicaciones móviles, frameworks de desarrollo, mapas, bases de datos, servicio web y metodologías de desarrollo, luego en la sección de desarrollo se realizó cuadros comparativos donde se definió la arquitectura adecuada al proyecto dando como resultado dos aplicaciones móviles en el framework ionic conectados a través de un servidor web desarrollado con la tecnología GraphQL todo esto desarrollado según la metodología Mobile-D.

Luego basados en la metodología seleccionada se realizó el diseño de la arquitectura donde consta el levantamiento de requerimientos basados en las técnicas definidas en la sección de materiales y métodos se usa las encuestas y entrevistas para obtener la información necesaria además se utilizó el estándar IEEE Std. 830-1998 para manejar mejor esta información. Se realizó el desarrollo de las aplicaciones móviles y el servicio web apoyados en varias herramientas de desarrollo.

Finalmente se realizó las pruebas respectivas para esto se definió un ambiente simulado donde se utilizó estudiantes de la carrera de ingeniería en sistemas de la Universidad Nacional de Loja como la población seleccionada, todas las pruebas se basaron en el estándar IEEE 829 que está dirigido para las pruebas de software, dando como resultado un balance positivo de éxito en las pruebas realizadas.

Summary

In the present work of qualification, the development of a computer architecture is written, whose main objective is to optimize the search and location of medicines in the different pharmacies, in addition to knowing if the pharmacy is open through two mobile applications and a web service or web service. The first mobile application aimed at customers of pharmacies where it allows them to search for medications and know necessary information about them such as their location and availability among others, the second mobile application aimed at pharmacies that allows interaction with the client through consultations in real time knowing the details of the drug sought by the client and allowing to issue a response to these queries and finally the web service responsible for managing the data and all the interactivity between mobile applications.

The development of this degree work was divided into three goals such as the definition of architecture, development and testing of it. In the first instance, a review of the literature was necessary with topics related to computer architectures, types of software, mobile applications, development frameworks, maps, databases, web service and development methodologies, then in the development section tables were made comparisons where the appropriate architecture for the project was defined, resulting in two mobile applications in the ionic framework connected through a web server developed with GraphQL technology, all this developed according to the Mobile-D methodology.

Then, based on the selected methodology, the architecture design was carried out, which includes the lifting of requirements based on the techniques defined in the materials and methods section. Surveys and interviews are used to obtain the necessary information and the IEEE Std standard was used. 830-1998 to better handle this information. The development of mobile applications and the web service based on various development tools was carried out.

Finally, the respective tests were carried out for this, a simulated environment was defined where students of the systems engineering career of the National University of Loja were used as the selected population, all the tests were based on the IEEE 829 standard that is aimed at software tests, resulting in a positive balance of success in the tests performed.

C. Introducción

Con la integración de los teléfonos celulares en los hogares y más aún en cada individuo se abrieron nuevas líneas de uso para estos, no solo para la comunicación sino para acceder a varios servicios que va más allá del acceso a la información como mensajería, compras en línea, ofimática, videoconferencias, geo localización, entre otros.

Esta introducción de la tecnología en la vida cotidiana de las personas, trajo consigo una necesidad inevitable a las empresas de ser parte de este cambio incluyendo a la tecnología como una de sus herramientas fundamentales para madurar o crecer como empresa.

La industria farmacéutica es hoy en día una de las mayores fuerzas de comercio a nivel nacional y mundial aportando al crecimiento de la economía desde varios aspectos como las importaciones, ventas, movilización entre otros [1].

Luego de describir esta necesidad de la inmersión de la tecnología en las empresas, la industria farmacéutica inminentemente debe formar parte de esta tendencia tecnológica.

Hoy por hoy los clientes de farmacias cuentan con varias opciones tecnológicas para interactuar con estas pero siempre surge la necesidad de una nueva tecnología que unifique todos estos servicios en una sola plataforma tecnológica como su ubicación, disponibilidad, y su stock de medicamentos. O existe una herramienta dirigida solo a un servicio o no existe, entonces el cliente debe satisfacer sus necesidades de manera rustica como cuando decimos que un cliente quiere conocer si un medicamento existe en alguna farmacia o si ésta se encuentra disponible o muchas veces conocer si hay alguna cercana, el cliente debe acercarse de manera personal para contestar esta interrogante perdiendo tiempo y dinero en este proceso (Ver Anexo 3).

A partir de esta problemática encontrada se planteó la realización del presente trabajo que tiene como objetivo principal desarrollar una arquitectura informática que permita la geo localización de los medicamentos en las diferentes farmacias.

Para el cumplimiento de este objetivo principal se desprenden tres objetivos específicos que son: definir una arquitectura informática que permita la geo localización de los medicamentos en las diferentes farmacias; desarrollar una arquitectura informática que permita la geo

localización de los medicamentos en las diferentes farmacias; evaluar que la arquitectura permita la geo localización de los medicamentos en las diferentes farmacias.

Para el desarrollo del presente documento se basó en la estructura definida en los lineamientos que tiene la Universidad Nacional de Loja y el Área de la Energía, Las Industrias y los Recursos No Renovables el cual nos dicta un orden y esquema específico.

El resumen donde se describe una visión general del desarrollo del presente trabajo de titulación, la introducción que muestra a breves rasgos de donde nace el problema y también una descripción breve de la estructura del documento.

La revisión de la literatura donde se adquirió conocimientos importantes para el desarrollo de la arquitectura informática y se fundamentó el trabajo realizado, luego están los materiales y métodos necesarios para el correcto desarrollo de la arquitectura lo cual se puede evidenciar en la sección de resultados para luego en la sección de discusión hacer un análisis de los resultados obtenidos.

Finalmente en sus últimas secciones se encuentran las conclusiones y recomendaciones que surgen al término del desarrollo del presente trabajo, también hay la sección de bibliografía y anexos que fundamentan el trabajo realizado.

D. Revisión de literatura

1. Situación del mercado farmacéutico en Ecuador.

En Ecuador el Estado es el principal socio comercial de la industria farmacéutica pero sin embargo su demanda es cubierta en su mayor parte por la empresa privada.

La capacidad que tiene la oferta para incidir en el mercado está establecida por las estrategias de mercadeo de los productores y distribuidores de medicamentos, que incluye, entre otros, la entrega de los estímulos a instituciones, prescriptores, distribuidoras y farmacias. La estrategia utilizada por la industria farmacéutica y su relación con el prescriptor contribuyen a distorsionar la elección de los fármacos según su costo-efectividad [2].

1.1 Mercado farmacéutico y la tecnología.

En los tiempos actuales donde la transformación digital es inminente el uso de las tecnologías en las empresas ha dejado de ser una opción para convertirse en una estrategia de competitividad.

La industria de la salud también se ha visto obligada a sumarse a esta transformación con el afán de conseguir beneficios como la reducción de costes y mejora de rentabilidad. Pero ¿cómo está el mercado farmacéutico local y nacional? la evidencia muestra que no existe una estrecha relación de la tecnología con el mercado farmacéutico en relación a otros países etiquetados como más desarrollados.

2. Arquitectura informática

2.1 Definición

El término arquitectura se refiere a las decisiones sobre diseño de una estructura en informática, este término en general que se aplica a la estructura de un sistema informático o de una parte del mismo. El término se aplica asimismo al diseño del software de sistema por ejemplo el sistema operativo (el programa que controla la computadora) y también se refiere a la Estructura de sistemas o sistema de estructuras que consisten en elementos, sus propiedades externamente visibles y la relación entre ellos [3].

2.2 Elementos

Como se mencionó en la definición puede estar conformado por programas de software, hardware, programas de aplicación, servicios en la nube, etc.

2.3 Tipos de Software

Son tres los grandes grupos en los que se divide este concepto informático:

Software de sistema. Elementos que permiten el mantenimiento del sistema en global: sistemas operativos, controladores de dispositivos, servidores, utilidades, herramientas de diagnóstico, de corrección y optimización.

Software de programación. Diferentes alternativas y lenguajes para desarrollar programas de informática: editores de texto, compiladores, intérpretes, enlazadores, depuradores, entornos de desarrollo integrados (IDE).

Software de aplicación. Permite a los usuarios llevar a cabo una o varias tareas específicas en cualquier campo de actividad: aplicaciones ofimáticas, para control de sistemas y automatización industrial, software educativo [4].

3. Geo localización

3.1 Definición.

Entendemos por geo localización al conjunto de técnicas que permiten determinar la posición geográfica de un elemento (un ordenador, un teléfono móvil o cualquier dispositivo capaz de ser detectado) en el mundo real y hacer uso de esa información. Esta tecnología requiere de la perfecta sincronización entre hardware y software, es necesario un dispositivo con GPS o conexión a internet y un software que permita hacer uso de ellos en esta dirección.

- **Georreferenciación:** Es el proceso mediante el cual se localiza un objeto, lugar o persona en el espacio físico para posteriormente representarlo en un sistema de coordenadas o mapa. Un ejemplo habitual es la representación de tu posición en el mapa de tu ciudad y actualizarla a medida que te desplazas.
- **Geocodificación:** Es el proceso de obtención de coordenadas geográficas a partir de otro tipo de datos geográficos, como la dirección o el código postal. Al proceso contrario, la obtención de direcciones postales a partir de coordenadas se le denomina georreferenciación [5].

3.2 Mapas.

Un Mapa es un despliegue gráfico sobre información geográfica que permite entender las distancias, referencias y puntos importantes en relación a un lugar y permite mostrar a un usuario sus puntos de interés [6], como se puede observar en la Figura 1.

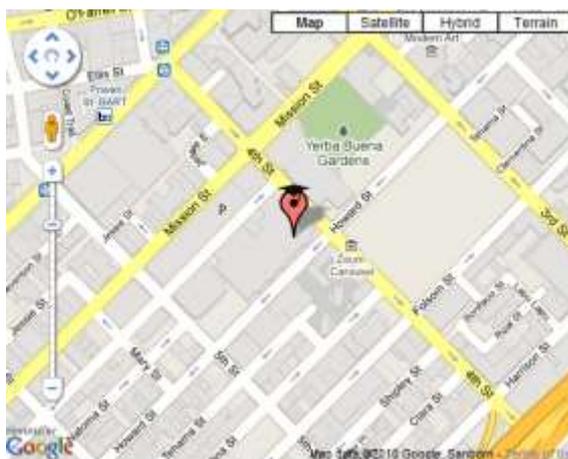


Figura 1: Mapa con marcador [7].

3.2.1 Google Maps.

Google Maps, permite la creación de pasos para llegar a alguna dirección creando una lista paso a paso para saber el cómo llegar a su destino, calculando el tiempo necesario y la distancia recorrida entre las ubicaciones. Google Maps puso a disposición de los desarrolladores sus códigos fuentes llamados APIS, que son un conjunto de comandos, funciones y protocolos informáticos que permiten integrar los mapas de Google Maps en cualquier aplicación con el uso de su codificación y con ello se pueden aplicar nuevas formas de ver el mundo como la vista satélite y relieve [8].



Figura 2: Google Maps [9].

3.2.1.1 Google Maps API

Google Maps API se trata de una tecnología que permite la visualización de Google Maps en tus propias páginas web o dispositivos móviles con JavaScript. El API proporciona unas determinadas herramientas para interactuar con los mapas y añadir contenido a los mismos a través de una serie de servicios, permitiendo llegar a crear aplicaciones con mapas de gran complejidad y robustez [9].

3.2.2 Open Street Maps

Este es un proyecto libre y gratuito. La idea en Open Street Maps es que los usuarios vayan colocando datos como calles, puntos de interés, y demás detalles en el mapa. Gracias a esta técnica, su velocidad de renovación es mucho más rápida que la de otros mapas. Además, se puede encontrar información de pueblos turísticos y algunas comunidades que por ser muy pequeñas no aparecen en los mapas convencionales. No cuenta con imágenes satelitales, visualmente es poco atractivo comparado con otros sistemas de mapas y para descargar los mapas para usarlos sin conexión hay que pagar. Puede usarse como aplicación nativa en varios dispositivos móviles. Es un mapa ideal para los que gustan de contribuir con información para lograr un gran detalle, aunque no cuente con la mejor presentación [10].

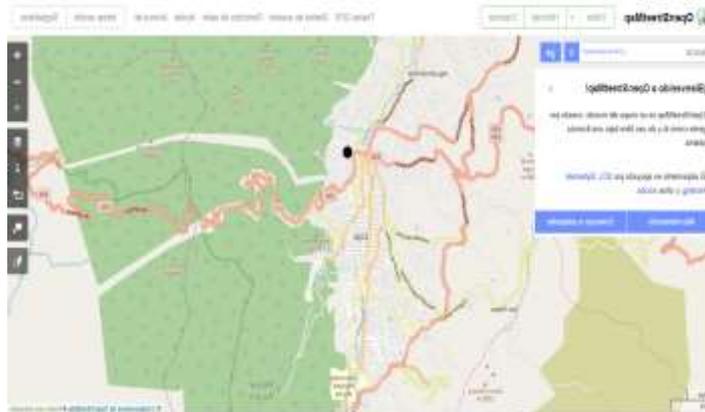


Figura 3: Open StreetMap [11].

3.2.3 MapBox

Mapbox es la plataforma de datos de ubicación para aplicaciones móviles y web. Proporciona bloques de construcción para agregar funciones de ubicación como mapas, búsqueda y navegación en cualquier experiencia que cree.

Mapbox se enorgullece de los principios de código abierto por lo que trabaja al aire libre permitiendo integrar tanto código como sea posible.

Contiene una gran diversidad de atractivos mapas amigables con el desarrollador para poder concentrar los esfuerzos en cosas diferentes como el desarrollo e integración de la herramienta.

Por ser una herramienta de código abierto permite a las compañías de análisis entender los grandes datos geográficos, las compañías de aviones no tripulados publican los sobrevuelos, los sitios de bienes raíces visualizan propiedades, las compañías de satélites procesan imágenes sin nubes y las compañías de seguros rastrean los activos, etc. [12]

4. Framework para el desarrollo de aplicaciones móviles

El desarrollo multiplataforma procura optimizar la relación costo/beneficio compartiendo la misma codificación entre las versiones para las distintas plataformas. Entre otras ventajas sobresalen: menor tiempo y costo de desarrollo; prestaciones cercanas a las nativas con acceso al hardware del dispositivo y disponibilidad de entornos potentes de desarrollo (Delphi, Visual Studio) o, en su lugar, utilización de

tecnologías bien conocidas por los desarrolladores web (HTML5, Javascript y CSS) que pueden trasladar sus conocimientos y experiencias al paradigma móvil [13].

4.1 Aplicaciones Web Móviles

Estas aplicaciones, diseñadas para correr dentro de un navegador, se desarrollan con tecnología web (HTML, CSS y JavaScript) y cuentan con una serie de características favorables: no necesitan adecuarse a ningún entorno operativo, son independientes de la plataforma y su puesta en marcha es rápida y sencilla. Por contrapartida, sus tiempos de respuesta decaen afectados por la interacción cliente-servidor, al mismo tiempo que resultan ser menos atractivas que las aplicaciones nativas ya que no se encuentran instaladas en el dispositivo [14].

4.1.1 Aplicaciones Híbridas

Las aplicaciones híbridas utilizan tecnologías web (HTML, Javascript y CSS) pero no son ejecutadas por un navegador. En su lugar, corren en un contenedor web del dispositivo con mayor acceso a sus capacidades específicas a través de una API. Las aplicaciones híbridas ofrecen grandes ventajas permitiendo la reutilización de código en las distintas plataformas, el acceso al hardware del dispositivo, y la distribución a través de las tiendas de aplicaciones [15].

4.1.2 Aplicaciones Interpretadas

Las aplicaciones interpretadas consisten en un proyecto base que se traduce en su mayor parte a código nativo mientras el resto es interpretado en ejecución. Se implementan de forma independiente de las plataformas utilizando diversas tecnologías y lenguajes, tales como Java, Ruby y XML, entre otros.

Como conclusión de las experiencias realizadas se señala que la mejor opción para aplicaciones multiplataforma son las aplicaciones híbridas [14]

4.2 Framework para el desarrollo de aplicaciones híbridas.

El uso de frameworks en el desarrollo de aplicaciones es el auge de estos días, existen un gran abanico de frameworks de donde se puede elegir dependiendo las necesidades y capacidades de los desarrolladores [16].

4.2.1 Ionic

Lo primero que se debe saber es que ionic se basa en la estructura MVC es decir modelo vista, controlador.

El MVC es un patrón de diseño que separa los datos, la lógica y las interfaces de usuario. Como su nombre indica, está separado en tres componentes: Modelo, Vista y Controlador. Está basado en la ideología de separación de conceptos y cumple perfectamente con los objetivos de los patrones de diseño [17].

1.-Modelo

Es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a las bases de datos para enviar o recibir información. Estas bases de datos pueden estar alojadas de forma local en nuestra app o de forma remota en un servidor externo.

2.-Vista

Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, como ejemplo podríamos decir que en una aplicación es el código HTML que nos permite mostrar la salida de los datos procesados.

3.-Controlador

Es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación.

En el caso MVVM (Modelo Vista VistaModelo) la iteración entre la vista y el controlador será en los dos sentidos, el controlador muestra los datos en la vista y si en la vista hay un cambio de datos, se actualiza el modelo automáticamente.

Entonces con respecto a Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS [18].

Características:

- **Alto rendimiento.**- la velocidad es importante. Tan importante que sólo se nota cuando no está en la app. Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery.
- **Centro nativo.**- ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, es que se desarrolla una vez, y compila para varias plataformas.
- **Diseño.**- limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

- **Un potente CLI.**- con un sólo comando se puede crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.
- **Comunidad.**- Una gran comunidad donde existe ayuda y soporte a cualquier duda o inconveniente [19].

4.2.2 PhoneGap

PhoneGap es un framework gratuito y de código abierto que permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y JavaScript. PhoneGap está realizado sobre Apache Cordova, fue creado por Nitobi, actualmente es propiedad de Adobe y crea de las llamadas aplicaciones híbridas, a mitad de camino entre las nativas y las puramente web, esto lo consigue a partir de una serie de APIs que permiten controlar características de los dispositivos como el acelerómetro, la cámara, las notificaciones o la geo localización [15].

Características:

- **Multiplataforma.**- PhoneGap permite crear actualmente aplicaciones móviles para: iPhone, Android, Windows Phone, Blackerry, Blackberry 10, webOS, Symbian y Bada
- **Rápida.**- En poco tiempo, si no se necesita muchos recursos (la aplicación es sencilla), PhoneGap permite desarrollar rápidamente con el plus que da el hecho de que automáticamente el desarrollo es multiplataforma. Llegar a más dispositivos, más rápido.
- **PhoneGap Build.**- Es el complemento perfecto para PhoneGap. Es un poderoso compilador en la nube que permite después de unos sencillos pasos (registro) acceder a todas las herramientas y todo el soporte necesario para compilar nuestras aplicaciones web en las plataformas móviles con las que trabaja PhoneGap.
- **Diseño.**- Este framework permite a los desarrolladores web de enfocarse en el desarrollo para los teléfonos inteligentes teniendo como base un código genérico con herramientas tales como JavaScript, HTML, CSS, y creando una interfaz de funciones foráneas para embeber una vista Web en el dispositivo móvil [15].

4.2.3 React

React es una biblioteca de JavaScript desarrollada en Facebook e Instagram utilizada para crear aplicaciones móviles nativas. Este es el marco de código abierto

recientemente desarrollado por Facebook, que permite a alguien escribir aplicaciones iOS (esto ahora también incluye la plataforma Android) usando JSX, un conocido código HTML parecido a un aspecto y JavaScript [20].

Es una innovación creada sobre la base del marco JavaScript de React.js; sin embargo, React.js no es compatible de forma nativa en aplicaciones IOS (o Android) a menos que se cuente la ejecución dentro de un navegador web como Safari o Chrome.

Características:

- **Híbrida.-** Inicialmente, Facebook solo desarrolló React Native para admitir IOS. Sin embargo, con su reciente compatibilidad con el sistema operativo Android, la biblioteca ahora puede representar UI móviles para ambas plataformas. Lo que significa que solo tiene que obtener el código escrito por una vez y que irá con múltiples plataformas para permitirle más de una aplicación nativa con un solo esfuerzo.
- **Ágil.-** En React Native se trata de brindar la velocidad y agilidad del desarrollo de aplicaciones web al espacio híbrido, con resultados nativos. Bajo el capó React Native se encuentra la popular biblioteca ReactJS UI de Facebook para aplicaciones web. React Native ofrece todo el mejor rendimiento de la aplicación de ReactJS, la abstracción DOM y los métodos de programación simplificados para el desarrollo móvil híbrido.
- **Reusabilidad.-** La reutilización de componentes se presenta como una gran ventaja para los desarrolladores. ReactJS le permite reutilizar los componentes desarrollados para otra aplicación que utiliza la misma funcionalidad. Esto ayudará a disminuir los esfuerzos de desarrollo mientras se asegura de que los componentes funcionen de manera suave y sin fallas [21].

5. Herramientas y tecnologías utilizadas para el desarrollo.

5.1 Web Service

Es un servicio o funcionalidad que se encuentra disponible a través de Internet o una Intranet , usa una forma estandarizada de mensajería (generalmente XML); no se encuentra atado a ningún sistema operativo ni ningún lenguaje de programación; también puede ser descubierto a través de un mecanismo de búsqueda. La gran ventaja que trae el protocolo HTTP es su esquema de mensajes especialmente diseñado y optimizado para ser utilizado en redes como Internet, por ello es que el

HTTP es el protocolo preferido para el transporte de mensajes de los Web Services [13].

Actualmente hay tres tipos de Web Service los que estandarizan su información a través de mensajes XML, en formato SOAP, llamados servicios SOAP; y los que no estandarizan su información, denominados servicios REST y el nuevo denominado GraphQL

5.1.1 Servicios SOAP

Los servicios SOAP (Simple Object Acces Protocol) implican el intercambio de mensajes XML, codificados según el protocolo SOAP.

Estos mensajes en formato SOAP (Ver figura 4) son movidos de un sistema a otro, utilizando HTTP; el sistema recibe el mensaje, hace lo que tiene que hacer, y devuelve una respuesta también en formato SOAP. Es un sistema simple, que no tiene en cuenta aspectos importantes del desarrollo de soluciones empresariales, pero que son tenidas en cuenta a través de extensiones a los estándares [22].

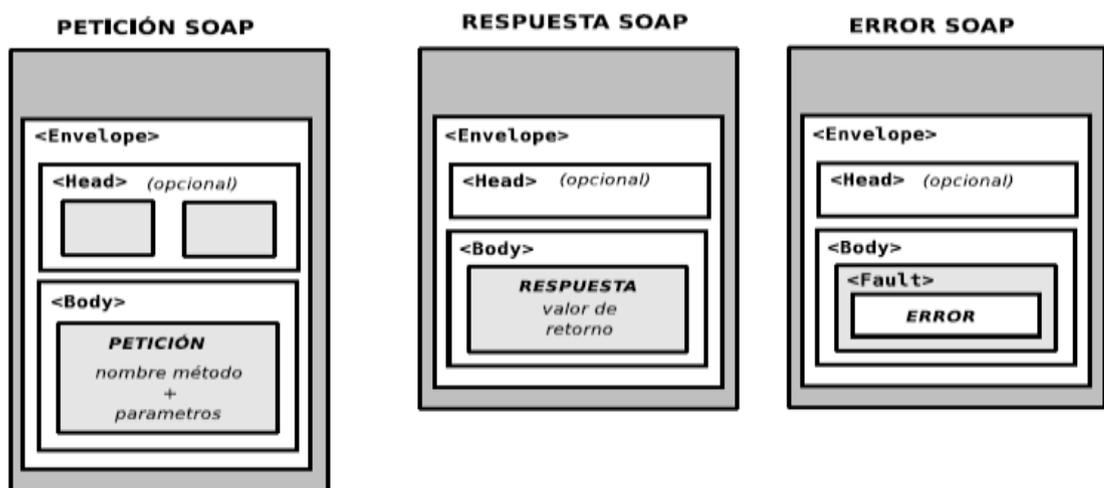


Figura 4: Formato Soap [22].

Los Web Services SOAP necesitan que se cumplan ciertas características:

- Excepto para datos binarios anexos, los mensajes deben ser transportados sobre SOAP.
- La descripción de un servicio debe ser hecha en WSDL.

- Uso de UDDI (Universal Description, Discovery and Integration).

Este tipo de web Service cuyo funcionamiento consta en la figura 4 necesita de ciertos elementos que interactúan entre sí:

Lógica. Se trata del componente que procesa la petición para generar la información solicitada por el cliente. Para realizar su función puede comunicarse con otros Servicios Web, acceder a bases de datos o bien invocar API de otras aplicaciones solicitando la información (o parte de ella) que ha de generar para enviar en formato XML.

SOAP (Simple Object Access Protocol). Protocolo de comunicación, basado en XML, que sirve para la invocación de los Servicios Web a través de un protocolo de transporte, como HTTP, SMTP, entre otros. Consta de tres partes: una descripción del contenido del mensaje, unas reglas para la codificación de los tipos de datos en XML y una representación de las llamadas RPC para la invocación y respuestas generadas por el Servicio Web.

UDDI (Universal Description, Discovery and Integration): Directorio donde es posible publicar los Servicios Web, permitiendo con ello que los posibles usuarios de ese servicio puedan obtener toda la información necesaria para la invocación y ejecución del Servicio Web [22].

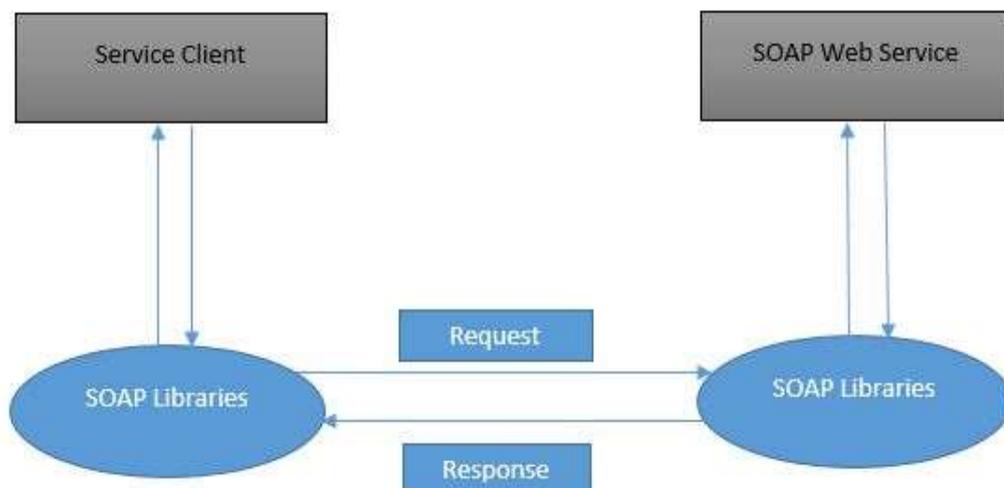


Figura 5: Web Service Soap

5.1.2 Servicios REST.

En primer lugar, conviene destacar que el término REST se refiere a una arquitectura en lugar de a un protocolo en concreto como es el caso de SOAP. A diferencia de SOAP utiliza directamente el protocolo HTTP. Siendo estrictos, la arquitectura REST no impone el uso de HTTP, no obstante en la práctica se entiende que un servicio web basado en REST, es aquel que se implementa directamente sobre la web.

La arquitectura REST está compuesta por clientes y servidores, es decir peticiones y respuestas que son construidos en base a la idea de transferir representaciones de recursos. Los clientes contienen representaciones y los servidores contienen los recursos en sí [13].

Los servicios Rest se caracterizan por ofrecer sencillez al momento de realizar cualquiera de sus funciones, que es realizada mediante una url que define la dirección del recurso al que se desea acceder.

Un Servicio Web REST tiene las siguientes características:

- Las interfaces deben construirse sobre HTTP. Las siguientes funciones son definidas:
 - ✓ HTTP GET: Usado para obtener una representación de un recurso. Un consumidor lo utiliza para obtener una representación desde una URI. Los servicios ofrecidos a través de este interfaz no deben contraer ninguna obligación respecto a los consumidores.
 - ✓ HTTP DELETE: Se usa para eliminar representaciones de un recurso.
 - ✓ HTTP POST: Usado para actualizar o crear las representaciones de un recurso.
 - ✓ HTTP PUT: Se usa para crear representaciones de un recurso. (Ver figura 6)
- La mayoría de los mensajes son XML, definidos por un esquema XML.
- Mensajes simples se pueden codificar en las URL.
- Los servicios y los proveedores de servicios deben ser recursos, mientras que los consumidores pueden ser un recurso.

Las ventajas de REST derivan de su simplicidad. Entre estas podemos destacar mejores tiempos de respuesta y disminución de sobrecarga tanto en cliente como en servidor. Mayor estabilidad frente a futuros cambios. Y también, una gran sencillez en el desarrollo de clientes, estos solo ha de ser capaces de realizar interacciones HTTP y codificar información en XML.

Como inconveniente cabe indicar que, al igual que ocurre con el protocolo HTTP, no se mantiene el estado. Es decir cada solicitud es tratada por el servidor de forma independiente sin recordar solicitudes anteriores [23].

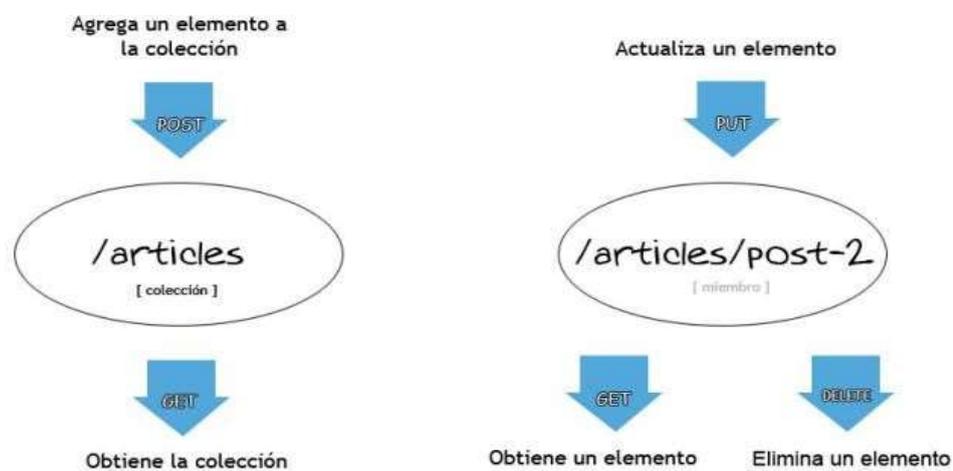


Figura 6: Web Service Api Rest

5.1.3 Service GraphQL

Cuando se crearon las primeras APIs allá por el 2000, un simple CRUD era suficiente en muchos casos, pero ya no es así, haciendo de las APIS RESTful algo cada vez más tedioso de mantener. REST ha sido un claro caso de éxito, pero su concepción CRUD basada en recursos, verbos y códigos de respuesta HTTP la hacen tener cada vez más limitaciones debido a su inflexibilidad.

Uno de los principales problemas que nos encontramos con RESTful es que utilizamos una URI para leer o escribir un único recurso, también denominado endpoint o punto de acceso [24].

Por cada tipo de recurso distinto que queramos manejar necesitaremos un punto de acceso distinto; ahora bien, en la mayoría de los casos reales actuales, cuando se pide

información, no solamente se pide un único recurso, lo cual implica varias llamadas a distintos puntos de acceso, las cuales no solamente nos devuelven la información que pedimos, sino que nos devuelven todo el conjunto de datos relativos al recurso que esté alojado en esa URI.

Por ejemplo, si queremos el DNI de una persona, el recurso nos traerá más información extra la cual no nos sirve para nada en nuestra consulta como fecha de nacimiento, nombre, apellidos, etc. Estos datos tienen que ser manejados y filtrados, tarea que habrá que realizar para todos los distintos objetos que tengamos que llamar, algunas opciones para aliviar la carga de estos puntos de acceso es la creación de APIs adhoc en el Backend.

Con la creación de APIs adhoc se crea una pseudo RESTful, saltándose varias reglas principales de RESTful, haciendo que lo que parece una buena idea se convierta en un caos de código duplicado, abstracciones y falta de retro compatibilidad, con un sinfín de pequeños puntos de acceso para cada dato del objeto y luego un punto de acceso extra para el total del objeto.

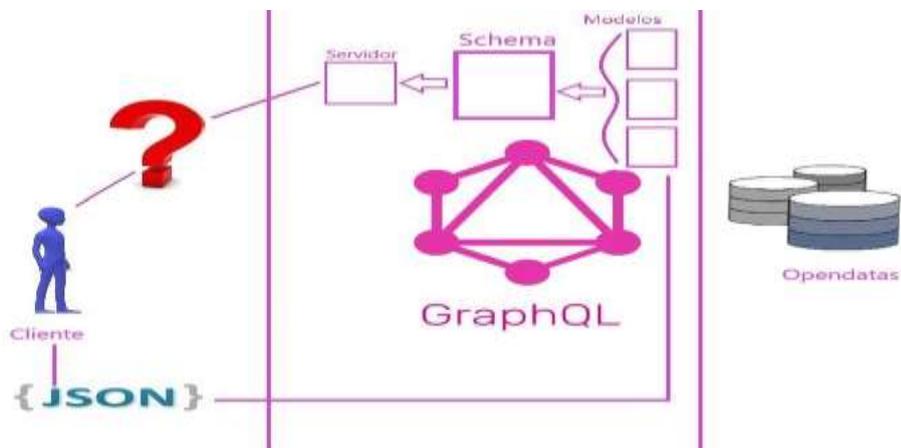


Figura 7: Service GraphQL [25]

Debido a que GraphQL tuvo mucha exposición recientemente, hay implementaciones de GraphQL en las más populares lenguajes de programación, incluidos PHP, Java, C /C++ y Haskell. Hay una gran cantidad de herramientas para interactuar con GraphQL o conectarlo con tecnologías de back-end existentes.

El **servidor GraphQL** actúa como una capa unificada para el acceso y la manipulación de datos. En un sistema distribuido, se encuentra en la misma capa como REST, SOAP y XML – RPC, lo que significa que se usa como una abstracción de capa para ocultar las bases de datos internas. Similar a estos patrones arquitectónicos, GraphQL no soporta Consultas ad hoc. Eso significa que, a diferencia de SQL, la estructura general de las posibles consultas está definida por el proveedor de servidor GraphQL por adelantado [26].

El **cliente GraphQL** dado que una API GraphQL tiene más estructura subyacente que una API REST, hay clientes más poderosos como Relay que pueden manejar automáticamente el procesamiento por lotes, el almacenamiento en caché y otras características. Pero no necesita un cliente complejo para llamar a un servidor GraphQL. Con express-graphql, solo puede enviar una solicitud HTTP POST al punto final en el que montó su servidor GraphQL, pasando la consulta GraphQL como el query campo en una carga útil JSON [26].

A diferencia de un Api Rest el servidor GraphQL tienen diferentes métodos que cumplen las diferentes funciones del Api Rest de manera más eficiente los mismos objetivos por ejemplo:

Query: que cumple la función de un GET el cual retorna solo los datos consultados.

Mutation: que supe al UPDATE POST y DELETE al permitir modificar los datos y al mismo tiempo devolver un estado con datos acerca del resultado.

Suscripción: es una de las grandes ventajas sobre los Api Rest por que permiten al cliente realizar una consulta y recibir notificaciones de nuevos resultados en el caso de eventos específicos del servidor todo en tiempo real.

5.1.4 JSON.

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero. Es fácil para los seres humanos a leer y escribir. Es fácil para las máquinas para analizar y generar. Se basa en un subconjunto del lenguaje de programación JavaScript, estándar ECMA-262 3ª Edición - Diciembre de 1999 [15].

5.1.5 SQL

Una base de datos SQL es una base de datos relacional que es estrictamente basado en la relación (tablas) para almacenar datos. Una relación (tabla) en una base de datos relacional se divide en Conjunto de filas y columnas. Cada columna representa un campo y cada fila representa un registro [27].

5.1.6 NoSQL

Las bases de datos NoSQL proporcionan la última característica llamada como fragmentación que lleva la partición de la base de datos a un nuevo nivel como en la forma de escalabilidad horizontal y disponibilidad. Los inconvenientes de las bases de datos NoSQL son falta de soporte RDBMS para consultas de usuario final, limitado restricciones de integridad como clave externa en la estructura Nivel y soporte limitado para el procesamiento de transacciones [27].

Una de las características más importantes de Bases de datos NoSQL es que no tienen un esquema fijo estructura, los registros pueden tener diferentes campos según los requisitos, y esto se llama como un esquema dinámico.

5.1.6.1 Bases de datos NoSQL.

Bases de datos de Key-Value Store:

Una base de datos de Key-Value Store es uno de los tipos de Bases de datos NoSQL que están teniendo fácil e Interfaz de programación de aplicaciones fácil de usar (API). No hay esquema fijo (Esquema menos) para Bases de datos KV. Como su nombre indica, los datos KV tienda consta de dos partes, una cadena que muestra clave y los datos reales que muestran valor. El kv Las bases de datos son como tablas hash donde están las claves. Utilizado como índices. Se encuentra que las bases de datos KV son comparativamente más rápido que RDBMS. Bases de datos KV Se puede encontrar en sitios web de compras en línea, etc.

Amazon DynamoDB.

Uno de los recién lanzados y totalmente gestionados KV. La base de datos es Amazon DynamoDB, que es ofrecida por Amazon e implementado utilizando el Dynamo de Amazon Modelo. Es un diseño especialmente a escala de aplicaciones de internet.

Proporciona alta fiabilidad, rapidez y Servicio rentable de base de datos NoSQL. Se almacena datos sobre unidades de estado sólido. Así que proporciona un acceso rápido a los datos. En lugar de almacenar datos en disco duro tradicional unidades, almacena datos en unidades de estado sólido, por lo que los datos El acceso es más rápido.

Bases de datos orientadas a columnas:

Datos almacenados en bases de datos orientadas a columnas en la Forma de columna entera en lugar de una fila. Esta Disminuye el acceso al disco en comparación con el relacional, que consta de columna y filas con Campos de tamaño uniforme para cada registro.

Cassandra

Como HBase, Cassandra también fue desarrollada usando Java. Fue formado por el software Apache. Fundaciones y estrenadas en 2008. Se trata de la Conceptos de tiendas de valor-clave y almacén de columnas. Base de datos, ya que se basa en el modelo Dynamo de Amazon. Algunas de las características de Cassandra son esquema dinámico, tolerancia de partición, Alta disponibilidad, persistencia, alta escalabilidad etc. Las aplicaciones que utilizan Cassandra son bancarias y Finanzas, sitios web de redes sociales y tiempo real. Análisis de datos, etc.

Bases de datos del almacén de documentos:

Una base de datos de documentos es un tipo de base de datos no relacional que ha sido diseñada para almacenar y consultar datos como documentos de tipo JSON. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el mismo formato de modelo de documentos que emplean en el código de aplicación. La naturaleza flexible, semiestructurada y jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las aplicaciones. El modelo de documentos funciona bien con casos de uso como catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo. Las bases de datos de documentos permiten una indexación fácil, potentes consultas ad hoc y análisis de colecciones de documentos.[23]

MongoDB

MongoDB es propuesto por la compañía 10gen para manejar las crecientes necesidades de almacenamiento de datos. Es una base de datos de almacenamiento de documentos NoSQL de código abierto que Está escrito en C ++. MongoDB usa JavaScript como su lenguaje de consulta. Los datos se almacenan en forma de colecciones. Cada colección consta de documentos. MongoDB almacena los documentos en Formato BSON que es una forma binaria de JSON. BSON admite diferentes tipos de datos, como enteros, flotadores, string, booleano, fecha, etc. MongoDB no tiene esquema tiene estructura documental. Para distribuir las colecciones a través de múltiples nodos ofrece una técnica de fragmentación. MongoDB automáticamente redistribuye los datos a través de los nodos, por lo que la carga es Equilibrado y distribuido equitativamente sobre los nodos. Permite la replicación maestro-esclavo. Técnica. Aquí los esclavos son los nodos que contienen Las copias de los nodos maestros y se utiliza para copia de seguridad. Procesar y leer operaciones [27].

6. Metodologías de desarrollo de software

Por metodología se entiende como un conjunto de procedimientos, técnicas, herramientas y documentos complementarios que ayudan a los desarrolladores de aplicaciones web y móviles a implementar sus ideas [28].

Existen muchas metodologías tradicionales para el desarrollo de software, que son de ayuda para los desarrolladores de aplicaciones web y móviles, hoy en día se han adquirido nuevas metodologías específicas para el desarrollo de aplicaciones móviles como son las denominadas y ya conocidas metodologías ágiles.

6.1 Metodología de desarrollo robusta o tradicional.

Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: MSF (por sus siglas en inglés Microsoft Solution Framework), MÉTRICA 3 y RUP (siglas de Rational Unified Process) [29].

6.2 Metodologías de desarrollo ágil.

Las metodologías ágiles para el desarrollo de software, surgieron como una solución inmediata, garantizando la realización de proyectos en corto plazo, las cuales se pueden ajustar a los Time-toMarket de los servicios móviles, como es el caso de las metodologías Mobile-D y Rapid7 Sin embargo, no garantizan el potencial de éxito de estos, éxito que depende en gran parte del conocimiento electrónico y de telecomunicaciones, del entorno específico donde se ejecuta la aplicación (terminales y redes), además de la caracterización del cliente [29].

6.2.1 Scrum

Scrum es una metodología ágil que se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de la toma de decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

El equipo Scrum se compone de un responsable del producto (dueño del producto, Product Owner), el equipo de desarrollo (Development Team o Scrum Team) y el Scrum Master, que es responsable de asegurar que Scrum es entendido y adoptado correctamente.

Scrum supone los siguientes artefactos: lista del producto (Product Backlog), lista del sprint (Sprint Backlog) y un incremento.

Un sprint es cada iteración del proceso de desarrollo.

La lista del producto contiene todas las funcionalidades previstas para el producto y la lista del sprint contiene el listado de aquellas funcionalidades que serán incluidas en la iteración siguiente (sin considerar ni definir en cuál sprint serán incluidas las restantes).

El incremento es “la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada”, independientemente de si finalmente el Product Owner decida realizar la entrega al cliente o destinatario del mismo.

Los sprints comienzan con una reunión donde se planifica el mismo (Sprint Planning Meeting) y finalizan con una revisión del sprint en cuanto al producto

(Sprint Review) y una retrospectiva (Sprint Retrospectiva), que implica una revisión del sprint en cuanto a personas, relaciones, procesos y herramientas. Durante el sprint se llevan a cabo reuniones diarias (Daily Meeting) donde se comparte información sobre el estado de la iteración y las dificultades presentes [30].

6.2.2 Metodología Extreme Programan (XP).

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [28].

6.2.2.1. Fases de la metodología extreme programing

La metodología extreme programing (XP) posee las fases: planificación, diseño, desarrollo y pruebas.

Planificación

- Se redactan las historias de usuarios.
- Se crea un plan de entregas.
- Se controla la velocidad del proyecto.
- Se divide el proyecto en iteraciones.
- Al comienzo de cada iteración se traza el plan de iteración
- Se rota al personal
- Cada día se convoca una reunión de seguimiento
- Corregir la propia metodología XP cuando falla

Desarrollo

- El cliente está siempre disponible
- Se debe escribir código de acuerdo a los estándares
- Desarrollar la unidad de pruebas primero
- Todo el código debe programarse por parejas
- Sólo una pareja se encargará de integrar el código
- Actualizar las versiones de los módulos lo más rápido posible
- Todo el código es común a todos

- Dejar las optimizaciones para el final

Pruebas

- Todo el código debe ir acompañado de su unidad de pruebas
- Todo el código debe pasar las unidades de pruebas antes de ser implantado.
- Crear una unidad de pruebas para protegerse del mismo
- Se deben ejecutar pruebas de aceptación a menudo y publicar los resultados
- Las pruebas unitarias

6.2.3 Metodología Mobile D

Mobile-D consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema como se muestra en la figura 8. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas. En la primera fase, Explorar, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento actores, definición del alcance y el establecimiento de proyectos. Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos.

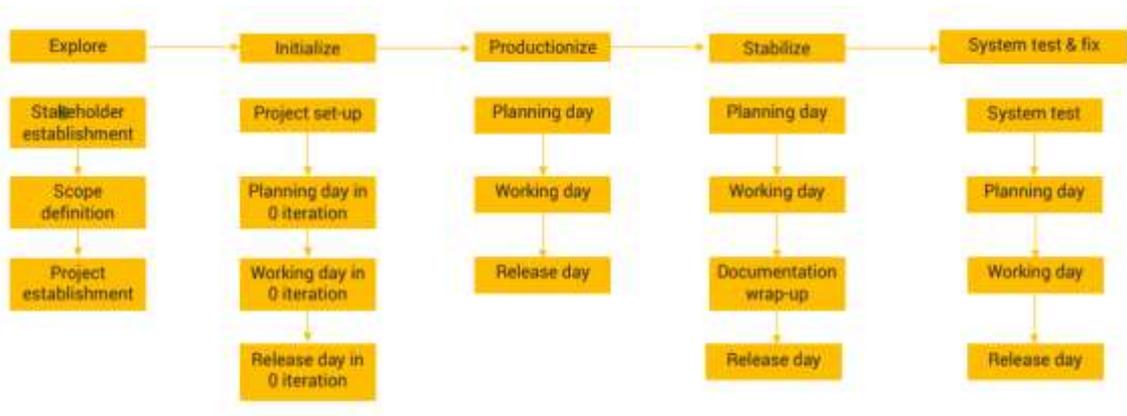


Figura 8: Fases de la metodología Mobile-D [31].

En la siguiente fase, iniciación, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico como los recursos físicos, tecnológicos y de comunicaciones

(incluyendo el entrenamiento del equipo de desarrollo). Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

En la fase de producción se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos. 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. Adicionalmente se puede considerar en esta fase la producción de documentación.

La última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados [32].

E. Materiales y métodos

De acuerdo a la pregunta de investigación se determinó que el tipo de investigación que se realizara es de tipo cualitativo.

Para el desarrollo del presente proyecto se utilizó varios métodos los cuales permitieron realizar un estudio detallado de los antecedentes del proyecto y conjuntamente con el empleo de técnicas se recolectó la información necesaria para el correcto desarrollo del presente TT.

1. Métodos de investigación

Según el tipo de investigación se usa los siguientes métodos:

1.1 Estudio de casos.

Este tipo de investigación fue utilizado para una exploración y análisis de casos de estudio sobre arquitecturas informativas, herramientas y metodologías de desarrollo (Ver sección revisión de literatura). De acuerdo a este tipo de investigación se la puede clasificar como de tipo etnográfico.

1.2 Observación activa:

La observación activa permitió conocer a fondo el tema de estudio de esta forma realizar un buen análisis que permita el correcto desarrollo de la arquitectura informática que permitirá la geo localización de los medicamentos en farmacias.

2. Técnicas de recolección de información.

Las técnicas que se utilizó en el desarrollo del presente trabajo son:

2.1 Encuesta

Las encuestas nos proporcionan una visión general mediante la recolección de información estandarizada de una población específica o una muestra representativa de la misma a través de cuestionarios.

Se realizó una encuesta para determinar la problemática existente sobre el problema de estudio también para establecer aquellas necesidades principales que tienen los usuarios de las farmacias al momento de localizar los medicamentos y la ubicación de los mismos dando como resultado los requerimientos de la aplicación para usuarios denominada Seo (Ver Anexo 1 apéndice 2).

Esta técnica también se utilizó para realizar las pruebas de aceptación de la arquitectura en un ambiente simulado (Ver Anexo 2).

2.2 Entrevista

Con esta técnica se recolectó información necesaria para determinar los requerimientos para la aplicación de farmacias denominada Pharma (Ver Anexo 1 apéndice 4).

2.3 Investigación Bibliográfica

Con esta técnica se sustentará la base teórica de la investigación, mediante consultas a: fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas entre otras (Ver sección Revisión de la Literatura).

3. Metodología de desarrollo de software

Para el desarrollo de la arquitectura informática como todo desarrollo de software es necesario hacer uso de una metodología la cual nos sirva de guía para llevar a cabo el correcto desarrollo en los tiempos adecuados.

Se planteó el uso de una metodología ágil que permita la elaboración de la aplicación móvil de una forma rápida y con un nivel medio de documentación. Es por ello, que se ha escogido la metodología Mobile-D (Ver sección de resultados 1.5). Consta de cinco fases que se emplearon de la siguiente forma:

Fase 1. Exploración: se realizó la planificación del proyecto. Aquí es donde se hizo una definición del alcance del proyecto y el establecimiento de las funcionalidades de la aplicación como las herramientas necesarias para su desarrollo (Véase sección de resultados, 2.1).

Fase 2. Inicialización: se aseguró el éxito de las próximas fases del proyecto mediante la preparación y verificación de todo el desarrollo y todos los recursos que se necesitarían además de determinarse los requerimientos de la arquitectura, donde se realizó un análisis de requisitos, para finalmente llevar a cabo el documento de especificación de requerimientos (IEEE 830) (Véase sección de resultados, 2.2).

Fase 3. Producción: se realizó la implementación de las funcionalidades requeridas del producto, mediante la aplicación de un ciclo de desarrollo iterativo e incremental. En esta fase podemos decir que se lleva a acabo todo el desarrollo e implementación (Véase sección de resultados, 2.3).

Fase 4. Estabilización: esta fase se llevó a cabo las últimas acciones de integración donde se verificó el completo funcionamiento de la aplicación. De toda la metodología, esta es la fase más importante de todas ya que es la que nos asegura la estabilización del desarrollo (Véase sección de resultados, 2.4).

Fase 5. Pruebas: por último, se ejecutó las pruebas necesarias para tener una versión estable y final del proyecto, para ello se realizaron varias pruebas para validar que la aplicación cumple con los objetivos para la cual fue creada, entre las cuales se mencionan las pruebas de carga, pruebas de interfaz, pruebas de datos ingresados y pruebas funcionales (Véase sección de resultados, 2.4).

4. Metodología para la revisión de literatura

En el proceso de desarrollo de la revisión de literatura se llevaron a cabo los siguientes pasos, los cuales están basados en la metodología de Bárbara Kitchenham [33].

4.1. Paso Uno: Definición de fuentes bibliográficas

Las fuentes bibliográficas donde se realizó las búsquedas de artículos, tesis son:

- ✓ Google Scholar
- ✓ IEEE
- ✓ ACM
- ✓ SCOPUS

4.2. Paso Dos: Definición de palabras clave

Las palabras clave utilizadas fueron las relacionadas con el tema del presente proyecto. Las palabras clave relacionadas son: Metodologías de desarrollo ágil, Metodologías comparación, sistemas operativos móviles, arquitectura informática, farmacias, farmacias y la tecnología, framework FrontEnd, tecnologías BackEnd, framework BackEnd, GraphQL, NodeJs, bases de datos no relacionales, que permitieron realizar las búsquedas en las fuentes bibliográficas definidas.

4.3. Paso Tres: Búsqueda de artículos

La búsqueda de artículos se delimitó a través de los siguientes criterios.

- Fecha de publicación: estudios a partir del año 2014.

- Idioma: español e Inglés
- Tipos de estudio: artículos, tesis, conferencias.
- Páginas web oficiales de: NodeJs, Ionic Framework, GraphQL Server, GraphQL Client, Android, Angular

4.4. Paso Cuatro: Criterios de exclusión

Los estudios que no tenían relación con la temática o información ambigua, no fueron analizados.

4.5 Paso Cinco: Análisis de Resultados

Se realizó un análisis del material bibliográfico encontrado, donde se aplicó los criterios de inclusión y exclusión. Posteriormente en la revisión de literatura se analizó los conceptos considerados útiles, los cuales fueron incluidos en esta sección permitiendo formar una estructura ordenada.

F. Resultados

Para el presente proyecto se establecieron tres fases los cuales buscan la implementación de una arquitectura informática como solución óptima al tema de estudio.

La primera fase donde busca definir una adecuada arquitectura informática y las diferentes herramientas y tecnologías necesarias para el desarrollo.

La segunda fase que se centró en el desarrollo de la arquitectura definida basados en la metodología definida.

La tercera fase que sirvió para evaluar la arquitectura desarrollada.

1. Definición de una arquitectura informática.

1.1 Análisis de enfoques de desarrollo de aplicaciones.

Para el presente trabajo se tomó tres enfoques de desarrollo de aplicaciones como son las aplicaciones web, aplicaciones nativas y aplicaciones híbridas estudiadas en la revisión de la literatura (Ver sección revisión de la literatura).

Para la elección de un enfoque se analizó las ventajas y desventajas de cada una de ellas como se muestra en la tabla I.

TABLA I: CUADRO COMPARATIVO DE APLICACIONES

	Aplicaciones Nativas	Aplicaciones Web	Aplicaciones Híbridas
Acceso a hardware	Completo	Parcial	Completo
Velocidad	Muy Rápida	Rápida	Muy Rápida
Costo	Elevado	Bajo	Moderado
Efectos táctiles	Disponibile	Parcialmente	Disponibile
Instalación	Requerida	No	No

Tienda	Si	No	Si
Multiplataforma	No	Si	Si
Internet	No Necesario	Necesario	No necesario
Interfaz	Excelente	Moderado	Buena
Tiempo de desarrollo	Mucho Tiempo	Poco tiempo	Moderado

Luego de haber analizado el cuadro comparativo sobre las diferentes tipos de aplicaciones (Ver tabla I) se determinó que el tipo de aplicación más conveniente para el presente proyecto es el tipo de aplicación híbrida por ser el que menos desventajas presenta como el tiempo de desarrollo y el costo, además de ser multiplataforma una característica tendencia entre los desarrolladores en la actualidad.

1.2 Análisis comparativo de Web Services.

Como se mencionó en la revisión de la literatura los Web Services son piezas fundamentales de las modernas arquitecturas informáticas. A continuación se presenta una tabla comparativa entre los principales tipos de Web Services revisados en la sección de revisión de la literatura.

TABLA II: CUADRO COMPARATIVO DE WEB SERVICES

	Soap	Api Rest	GraphQL
Desuso	Si	No	No
Seguridad	Seguro	Medianamente	Seguro
Documentación	Desuso	Buena	Moderado
Tipado fuerte	No	No	Si

Tipo de respuestas	XM	XML, Json	Json
Instalación	Difícil	Medianamente fácil	Fácil
Respuestas filtradas	No	No	Si
Tiempo real	No	No	Si
Frameworkos	Pocos	Varios	Varios
Comunidad	En decadencia	Se mantiene	Creciente

Luego de realizar la comparativa entre los diferentes Web Services y analizar esta información detalladamente se determinó que el Web Service más adecuado para el presente proyecto es el servidor GraphQL por ser el que está en auge a diferencia de los otros que se están quedando arraigados y presentar más ventajas como el filtro de las respuestas evitando así el colapso por información innecesaria y sobre todo por ser el único Web Service que maneja las peticiones en tiempo real característica sobresaliente en la arquitectura informática evitando el uso de tecnología para cumplir con esta característica importante.

1.3 Análisis comparativo de Frameworks.

Al término de haber definido un enfoque sobre el tipo de aplicación móvil a desarrollar se procede a definir un framework para su desarrollo a continuación una tabla con los diferentes frameworks y sus principales características analizador en la sección de revisión de la literatura (Ver sección de revisión de la literatura).

TABLA III: ANÁLISIS DE FRAMEWORKS DE DESARROLLO

	IONIC	PHONEGAP	REACTJS
Plataformas Soportadas	Android iOS	Android iOS Windows Phone	Android iOS Blackberry
Lenguaje de Programación	HTML5, CSS, JavaScript	HTML, CSS, JavaScript, Propio + Nativo	HTML, CSS, JavaScript + Nativo
Acceso a los recursos del dispositivo	Acceso completo a las funcionalidades del dispositivo	Acceso a la mayoría de las funcionalidades del dispositivo	Acceso completo de las funcionalidades del dispositivo
Soporte en línea	Soporte inmediato de la comunidad	Soporte tardío de una comunidad decreciente.	Soporte casi inmediato de una comunidad creciente
Experiencia con el framework	Mucha	Ninguna	Poca
Documentación	Si	Si	Si
Tipo de software	Código Abierto	Código Abierto	Código Abierto
Instalación	Sencilla	Medianamente Sencilla	Sencilla

Luego de hacer un análisis minucioso a la tabla sobre los frameworks y sus diferentes características (Ver tabla III) se concluye que el framework idóneo para trabajar en el presente proyecto es Ionic por las diferentes ventajas que muestra sobre los otros frameworks analizados y por el dominio con el que cuenta el equipo de desarrollo sobre este framework.

Además dicho framework permite crear interfaces intuitivas y minimalistas, por medio de la programación tradicional web (HTML5, CSS, JavaScript) basado en el modelo vista controlador.

1.4 Análisis comparativo sobre los mapas Google Maps y Mapbox.

Al tratarse de un proyecto acerca de geo localización es vital establecer una tecnología que nos permita cumplir este objetivo, existen varias plataformas que nos permiten hacer uso, manipulación y personalización de mapas tanto en aplicaciones web como móviles como son el caso de la famosa herramienta conocida a nivel mundial Google Maps y la nueva tecnología que cada día cobra más terreno en el ámbito de desarrollo Mapbox utilizada por famosas empresas como Uber, Pinterest entre otras (Ver sección revisión de la literatura)

TABLA IV: CUADRO COMPARATIVO DE MAPAS

Parámetros	GoogleMaps	MapBox
Costo	Cuenta con una versión gratuita y una de pago	MapBox es libre de usar, además cuenta con una versión de pago.
Accesibilidad	Mapas poco editables	Mapas editables.
Regiones	La descarga de mapas sin conexión no está disponible en algunas regiones debido a las limitaciones contractuales.	Las aplicaciones creadas con Mapbox Maps SDK para Android o iOS pueden descargar mapas de regiones preseleccionadas
Inicio	Desde el 8 de febrero de 2005.	Desde 2010
Tecnología	Principalmente basado en AJAX	Usa NodeJs que es JavaScript
Planes	Ofrece un crédito de 200\$ cada inicio de mes pero cabe destacar que se debe registrar una tarjeta de	Ofrece un plan gratuito que contiene:

	<p>crédito Con el crédito mencionado se puede obtener :</p> <ul style="list-style-type: none"> • 28.500 cargas web de mapas dinámicos • 100.000 cargas web de mapas estáticos • 25.000 cargas web estáticas de Street View 	<ul style="list-style-type: none"> • 50,000 usuarios activos mensuales • 50,000 solicitudes de geocodificación / mes • 50,000 solicitudes de direcciones / mes • 50,000 elementos matriciales / mes
--	---	---

Como se puede visualizar en la tabla IV, Mapbox se destaca por ser orientado a desarrolladores y manejar una versión gratuita bastante completa para sistemas pequeños. Esto fue base para la elección de este Mapa lo cual permitió acoplarse con facilidad a la estructura del mismo además de la experiencia que cuenta el equipo de desarrollo usando esta herramienta.

1.5 Análisis comparativo de metodologías.

Es importante realizar un análisis sobre las diferentes metodologías que existen para seleccionar la que mejor se adapte a las necesidades y características del presente proyecto.

TABLA V: ANÁLISIS COMPARATIVO DE METODOLOGÍAS TOMADA DE [23]

	XP	Mobile D	SCRUM
Comunicación con el cliente	Retroalimentación continúa entre el cliente y el equipo de desarrollo.	La comunicación con el cliente es menos que Scrum.	El Product Owner maneja la comunicación con el cliente.

Programación	Programación en parejas, en jornadas largas; Revisión de código mutuo.	Es la más importante, y que mayor tiempo posee junto con las pruebas.	El tiempo de programación se determina de acuerdo a la puntuación dada a cada tarea
Documentación	Lo principal son las historias de usuario y tarjeta CRC.	Lo principal es el Modelo del dominio, Storycards y storyboards.	Lo principal lo construyen las historias de usuario.
Pruebas	Se realizan unidades de pruebas, pruebas de aceptación y pruebas unitarias.	Se realizan pruebas unitarias, pruebas de interfaz, pruebas de datos ingresados y pruebas funcionales.	Se realizan pruebas unitarias, pruebas de integración, pruebas de aceptación.
Fases	Planificación de Proyectos, Diseño, Codificación, Pruebas	1. Exploración. 2. Iniciación. 3. Producción. 4. Estabilización. 5. Pruebas	Se define por Sprints de dos semanas aproximadamente y de un producto entregable.

Al término del análisis acerca de las diferentes metodologías se concluye que las mencionadas en el cuadro comparativo (Ver tabla V) tienen muchas similitudes en sus

características y algunas en sus fases. Pero al tener pocas diferencias se toman en cuenta pequeños detalles los cuales reduce la elección a la metodología Mobile-D, que es el que mayor se adapta a las necesidades del proyecto debido a que es enfocado a proyectos de software con un número pequeño de personal.

Para una mejor definición de la arquitectura se apoyó en el modelo diseñado por Philippe Kruchten que sirve para la descripción de la arquitectura de sistemas software, basados en el uso de múltiples vistas concurrente [34].

A continuación el documento para la definición de la arquitectura informática basado en el modelo 4 + 1.

1.6 Arquitectura de Software

Proyecto:

ARQUITECTURA INFORMÁTICA PARA GEO LOCALIZACIÓN DE
MEDICAMENTOS EN FARMACIAS

Documento de Arquitectura de Software

1.6.1 Introducción.

La definición de la arquitectura de software es parte esencial del presente TT ya que se trata del desarrollo de una arquitectura informática, además de ser indispensable para que cualquier producto de software sea construido exitosamente, puesto que permite determinar el comportamiento e interacción del sistema, y tener una visión clara de lo que se desea construir.

1.6.2 Propósito.

El presente documento describe el diseño de la arquitectura a través del modelo 4+1, donde se detalla la lógica y el comportamiento de las aplicaciones móviles con el Web Service.

1.6.3 Alcance

Este documento detalla la arquitectura de software que es utilizado para las aplicaciones móviles, mediante las vistas del modelo 4+1 como: la vista de escenarios (Casos de Uso), vista lógica, vista de procesos, vista física y vista de despliegue.

1.6.4 Definiciones, Siglas y abreviaturas

TABLA VI: DEFINICIÓN, SIGLAS Y ABREVIATURAS.

Nombre	Descripción
Cliente	Persona que va a buscar medicamentos a través de la aplicación de cliente.
Farmacia	Aplicación dirigida a las farmacias.
ERS	Especificación de requerimientos.
WS	Web Service
APP	Aplicación móvil.
Js	JavaScript
Vsc	Entorno de desarrollo visual studio code.

1.6.5 Referencias

TABLA VII: REFERENCIAS DEFINICIÓN DE ARQUITECTURA

Referencia	Título del documento
Anexo 1	Especificación de requisitos de software
K. Philippe and R. S. Corp., “Architectural Blueprints—The ‘4+1’ View Model of Software Architecture,” vol. 12, pp. 42–50, 1995.	Arquitectura de Software 4+1

1.6.6 Vista Global

Este documento muestra de manera concreta como está organizado la arquitectura de la aplicación, donde se detalla el desarrollo del modelo 4+1 que especifica por medio de las vistas la funcionalidad de la aplicación, las vistas están compuestas por: la vista de escenarios que describe los casos de uso; la vista lógica que detalla el modelo del negocio y su funcionalidad; la vista física que define los componentes físicos que interactúan con el sistema; la vista de despliegue muestra cómo están interactuando los componentes software del sistema; la vista de procesos que describe la funcionalidad de la aplicación.

1.6.7 Representación de la Arquitectura

En el presente proyecto la arquitectura de software utilizada es la Arquitectura Cliente Servidor, la cual está constituida por las vistas 4+1 de Kruchten, que se basa en el Lenguaje Unificado de Modelado UML, las vistas son las siguientes: vista de escenarios, vista lógica, vista de procesos, vista de despliegue y la vista física.

TABLA VIII: ARQUITECTURA DE VISTAS 4+1

Vista	Elemento Modelado	Descripción
Vista de escenarios	Casos de Uso	Muestra la interacción que tiene los diferentes actores en el sistema
Vista Lógica	Diagrama de clases Modelo Conceptual Modelo Relacional	Representa las funcionalidades y el servicio que proporciona a los usuarios.
Vista Física	Diagrama de despliegue	Muestra todos los componentes físicos del sistema.
Vista de Despliegue	Diagrama de componentes	Describe los componentes del sistema con el fin de que el programador entienda las interacciones que existen.
Vista de procesos	Diagrama de actividad	Describe los proceso de la funcionalidad de sistema

1.6.8 Objetivos y Restricciones de la Arquitectura

El desarrollo de la arquitectura cumplirá con las siguientes características:

- **Rendimiento:** Las peticiones hacia el servicio web serán eficiente debido a su ligero rendimiento del Framework Lumen
- **Disponibilidad:** El servicio web y las aplicaciones móviles estarán disponibles las 24 horas los siete días de la semana.
- **Portabilidad:** La aplicación móvil será construida en ionic por lo que será multiplataforma e hibrida, aunque las pruebas solo se realizaran en el sistema operativo Android de acuerdo a la planificación.
- **Calidad:** Las aplicaciones móviles serán de calidad cumpliendo todas las funcionalidades descritas en el documento ERS (Ver Anexo 1).

1.6.9 Restricciones

- **Restricciones de Tecnología:** La metodología a implementar es la Mobile-D; la aplicación móvil será compatible con la versión mínima de Android 5.1; las pruebas serán realizadas exclusivamente para la plataforma Android debido a que no se cuenta con la tecnología necesaria para las pruebas en la plataforma IOS como el programa Xcode que únicamente puede ser instalado en sistema operativo MacOS.

1.6.10 Vista de Escenarios.

En esta vista se detalla el diagrama de casos de uso, donde se muestra la interacción entre el usuario y las aplicaciones móviles.

1.6.10.1 Diagrama de Casos de Uso

El siguiente diagrama consta de dos actores: Farmacia y Cliente, cada uno de estos actores tiene diferentes funciones y para ejecutarlas deberán estar autenticados en sus respectivas aplicaciones móviles.

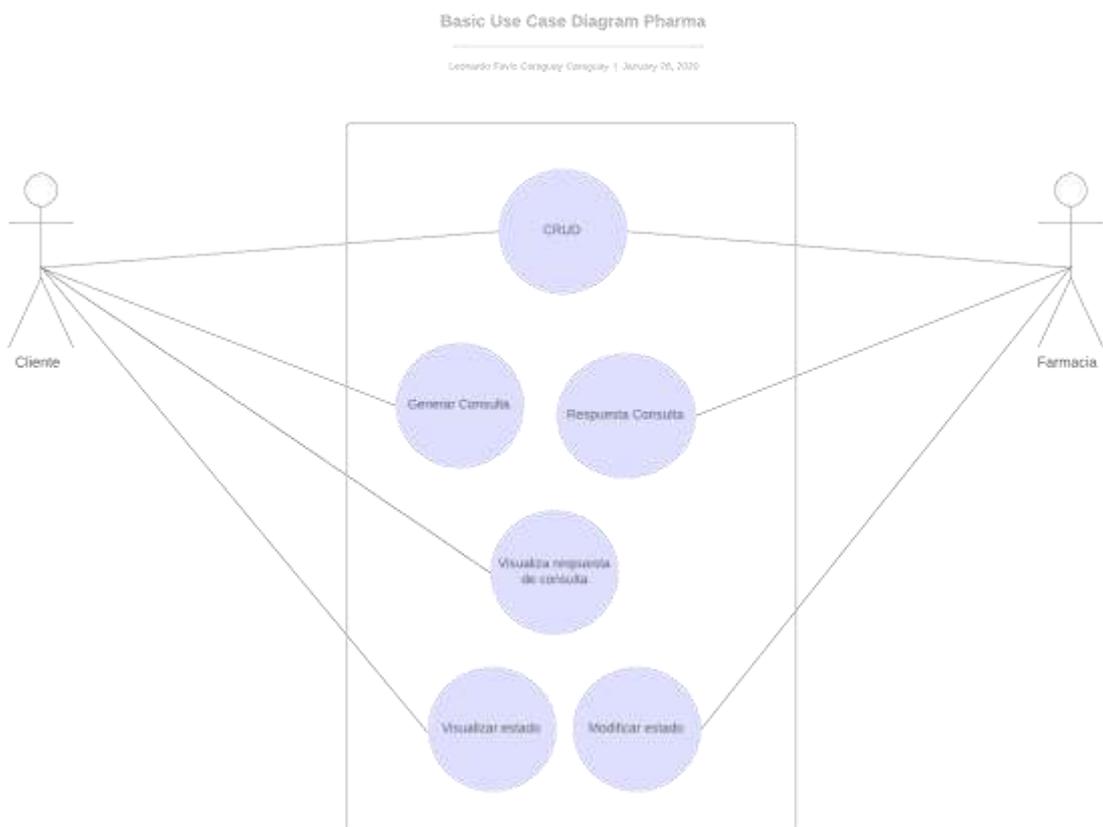


Figura 9: Diagrama de Casos de Uso

1.6.10.2 Especificación de Casos de Uso

TABLA IX: CASO DE USO CRUD

Autenticación, Registro	
Actores	Cliente y farmacia
Objetivos	Registro y autenticación en las aplicaciones tanto como cliente y farmacia.
Precondiciones	
<ul style="list-style-type: none"> • Haber ejecutado la aplicación. 	
Pasos:	
<ul style="list-style-type: none"> • El usuario hace clic en el botón mi ubicación disponible en la sección inicio de la aplicación. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Visualizar en el mapa un marcador con la ubicación del usuario. 	

Tabla X: CASO DE USO GENERAR CONSULTA

Generar Consulta	
Actores	Cliente
Objetivos	Generar una consulta de medicamento.
Precondiciones	
<ul style="list-style-type: none"> • Haber ejecutado la aplicación. 	
Pasos:	
<ul style="list-style-type: none"> • El usuario hace clic en el botón buscar medicamento y escribe el nombre del medicamento deseado. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Visualizar las farmacias que cuentan con este medicamento en stock. 	

TABLA XI: CASO DE USO RESPUESTA DE CONSULTAS

Respuesta de consulta	
Actores	Farmacia
Objetivos	Responder las consultas generadas en tiempo real.
Precondiciones <ul style="list-style-type: none"> • Haber ejecutado la aplicación. • Estar en estado disponible. 	
Pasos: <ul style="list-style-type: none"> • El usuario al visualizar las consulta deberá responder con las diferentes opciones disponibles. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar las consultas nuevas y elegir una respuesta a las mismas. 	

TABLA XII: CASO DE USO VISUALIZAR RESPUESTA

Visualizar respuestas	
Actores	Cliente
Objetivos	Visualizar las respuestas de la consulta generada
Precondiciones <ul style="list-style-type: none"> • Haber generado una consulta 	
Pasos: <ul style="list-style-type: none"> • Haber generado una consulta previamente • Esperar respuestas 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar los diferentes locales con sus respectivas respuestas. 	

TABLA XIII: CASO DE USO MODIFICAR ESTADO

Modificar estado	
Actores	Farmacia
Objetivos	Cambiar estado de la farmacia a convenir.
Precondiciones <ul style="list-style-type: none"> • Haber autenticado en la aplicación 	
Pasos: <ul style="list-style-type: none"> • Autenticación en la aplicación de la farmacia • Elegir un estado para la farmacia de entre las disponibles. 	
Resultado esperado: <ul style="list-style-type: none"> • Respuesta sobre el cambio del estado. 	

TABLA XIV: CASO DE USO VISUALIZAR ESTADO

Visualizar estado	
Actores	Cliente
Objetivos	Visualizar el estado de las farmacias.
Precondiciones <ul style="list-style-type: none"> • Haber autenticado en la aplicación 	
Pasos: <ul style="list-style-type: none"> • Autenticación en la aplicación de la farmacia • Elegir una farmacia para ver detalles haciendo clic en el icono. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar información de la farmacia como el estado de esta. 	

1.6.11 Vista Lógica.

En esta vista se hace referencia a los requerimientos funcionales de la aplicación, primeramente con el modelo conceptual y el diagrama de clases.

1.6.11.1 Modelo Conceptual.

Este es un modelo estático de objetos que componen la solución del problema en la geo localización de medicamentos en farmacias.

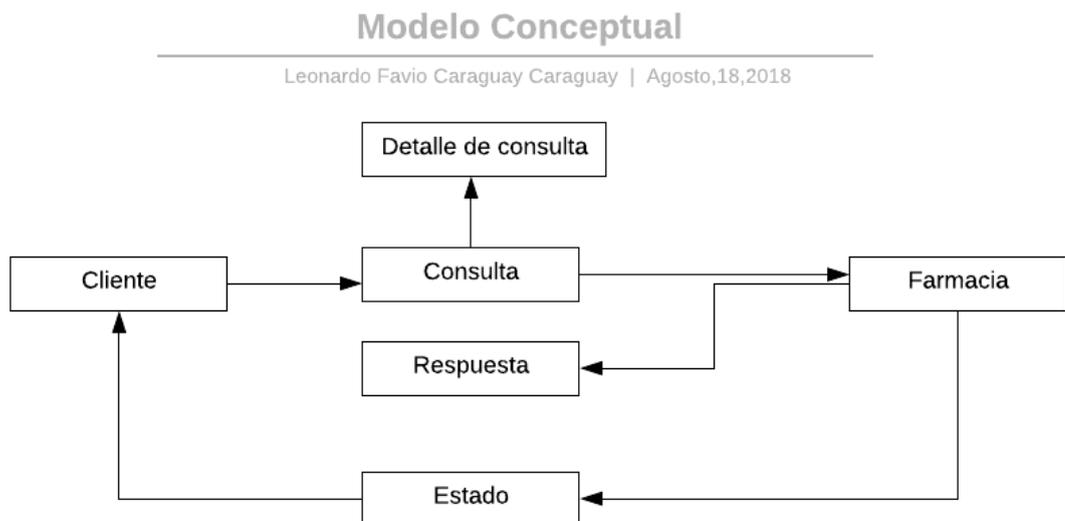


Figura 10: Modelo Conceptual

1.6.11.2 Diagrama de Clases.

El diagrama de clases del proyecto está definido por 6 clases cada una con sus respectivos atributos y relaciones.

Consulta: clase que contiene atributos como una descripción y cliente

Cuenta: clase que tiene atributos como usuario y contraseña

Cliente: en esta clase se indica que el usuario puede tener de una cuenta y generar una consulta

Medicamento: clase que contiene los detalles de un medicamento.

Respuesta: contiene las respuestas a las consultas.

Farmacia: clase que puede tener una cuenta y generar respuestas.

UML Class Diagram Pharma

Leonardo Favio Carriguy Carriguy | January 28, 2020

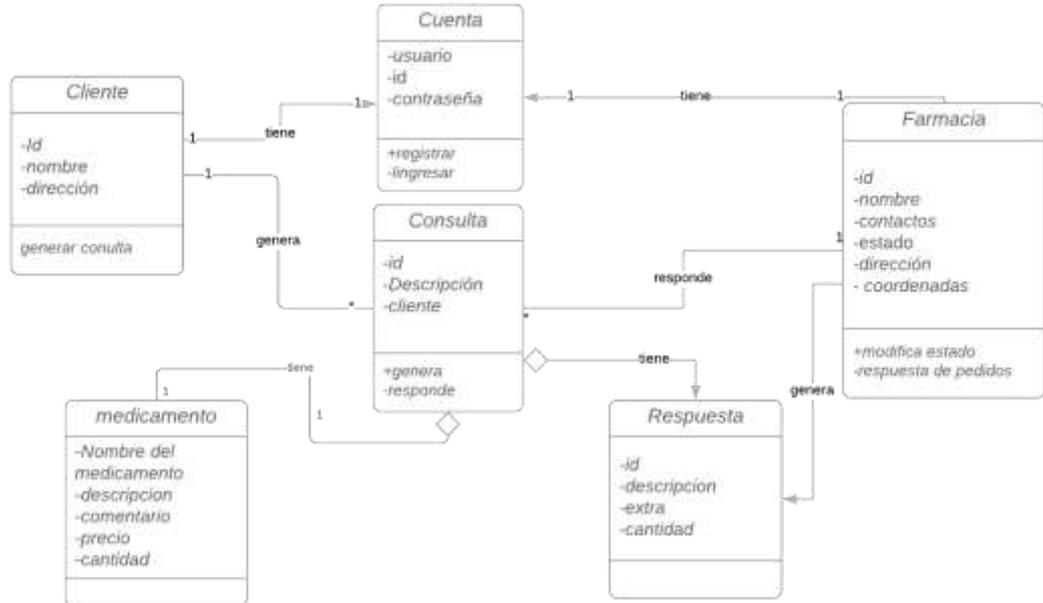


Figura 11: Diagrama de Clases Definición de Arquitectura

1.6.11.3 Modelo Relacional

Modelo de organización y gestión de bases de datos consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos. Se distingue de otros modelos, como el jerárquico, por ser más comprensible para el usuario inexperto, y por basarse en la lógica de predicados para establecer relaciones entre distintos datos. Surge como solución a la creciente variedad de los datos que integran las base de datos, podemos resumir el concepto como una colección de tablas (relaciones).

En el presente proyecto no se utiliza este modelo debido a que las herramientas de gestión de base de datos a utilizar no son relacionales.

1.6.12 Vista de Procesos.

En esta vista se detalla el diagrama de actividades de las aplicaciones y su proceso a desarrollarse.

1.6.12.1 Diagrama de actividades.

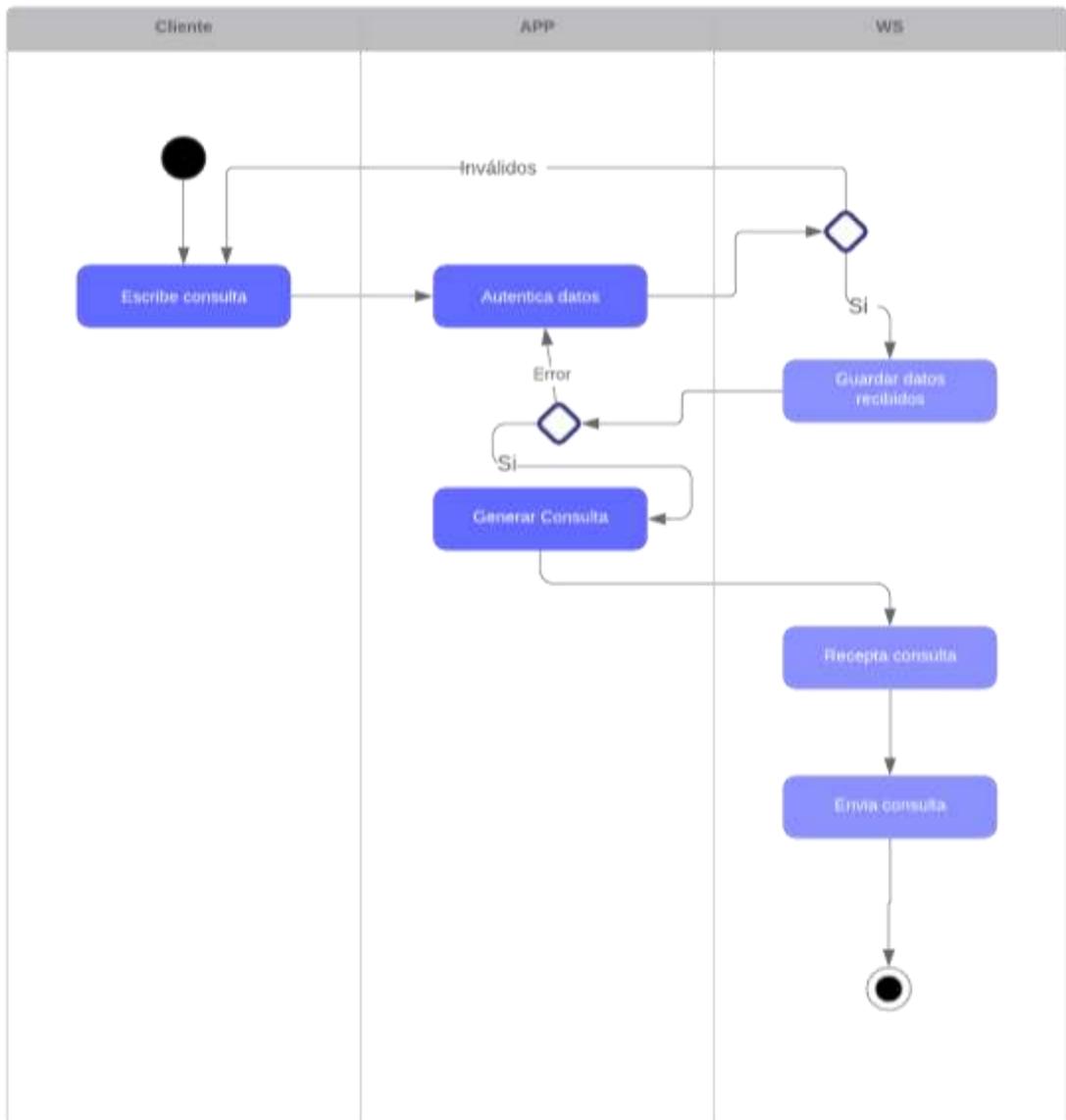


Figura 12: Diagrama de actividad Generar Consulta

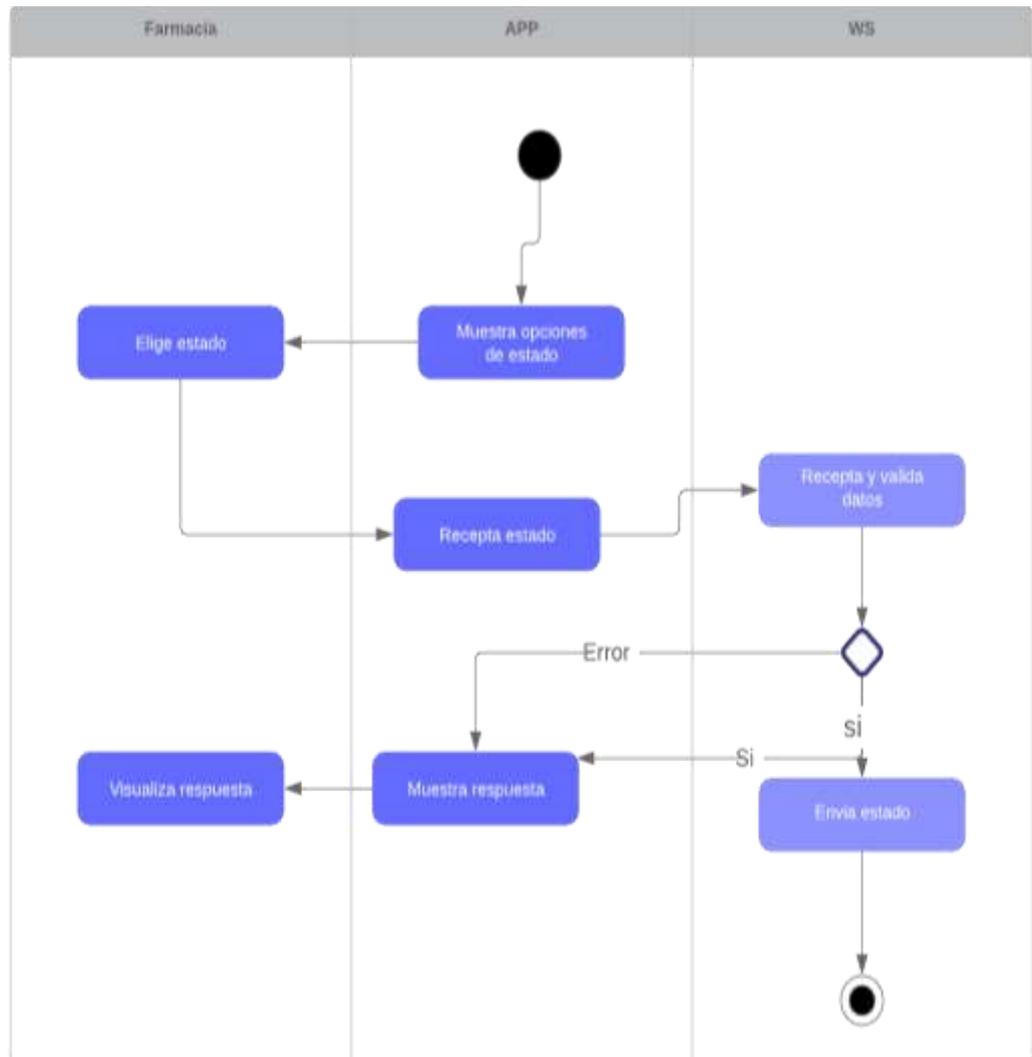


Figura 13: Diagrama de Actividad Elegir Estado

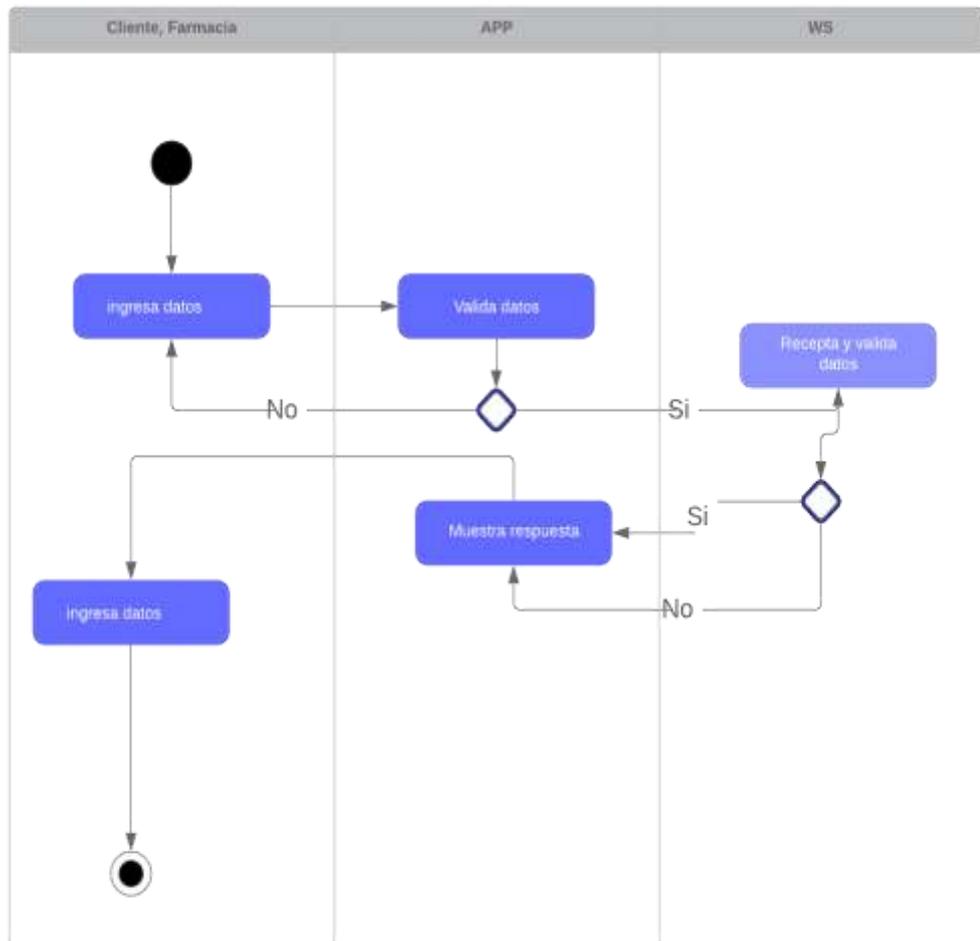


Figura 14: Diagrama de actividad Autenticación y Registro

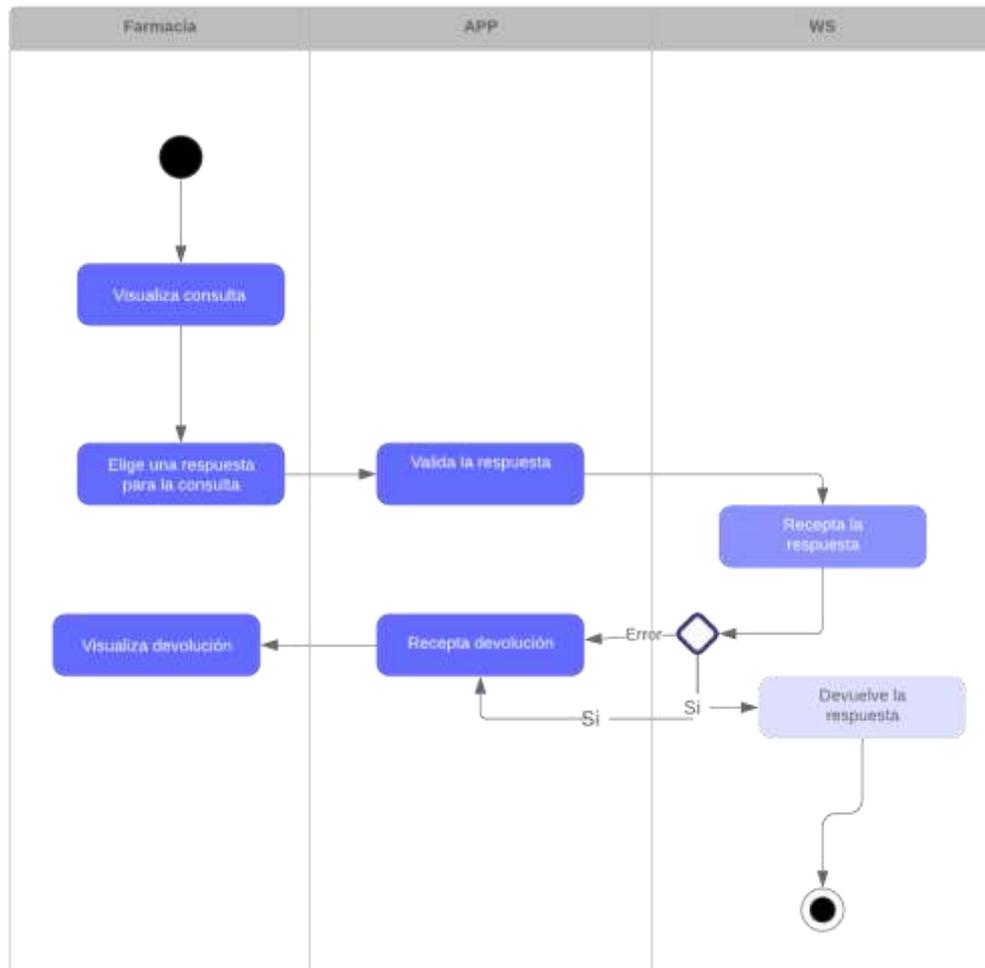


Figura 15: Diagrama de actividad Generar Respuesta

1.6.13 Vista de despliegue y Física

Estas vistas definen la organización de los diferentes componentes de la arquitectura indicando los nodos finales que conforman la solución, mostrando la arquitectura final. Estos componentes tienen la siguiente estructura.

Servicio Web: en esta parte está desarrollado el back-end donde se define la lógica del negocio (GraphQL) con el framework Express (Js), el cual tiene definido rutas y métodos que interactúan con la base de datos no relacional Mongo, con la ayuda de mongoose. Cada ruta está enlazado a un controlador y responde a las peticiones que son realizadas por los clientes (Aplicaciones Móviles Cliente y Farmacia) mediante el protocolo HTTP.

Aplicación Farmacia: esta componente está dirigido para los administradores de farmacias donde podrán visualizar las consultas de los clientes en tiempo real además de elegir su estado a conveniencia.

Aplicación Cliente: parte de la arquitectura donde el cliente de la farmacia podrá generar las consultas de los diferentes medicamentos deseados además de conocer la información de las farmacias como su estado ubicación y demás información básica.

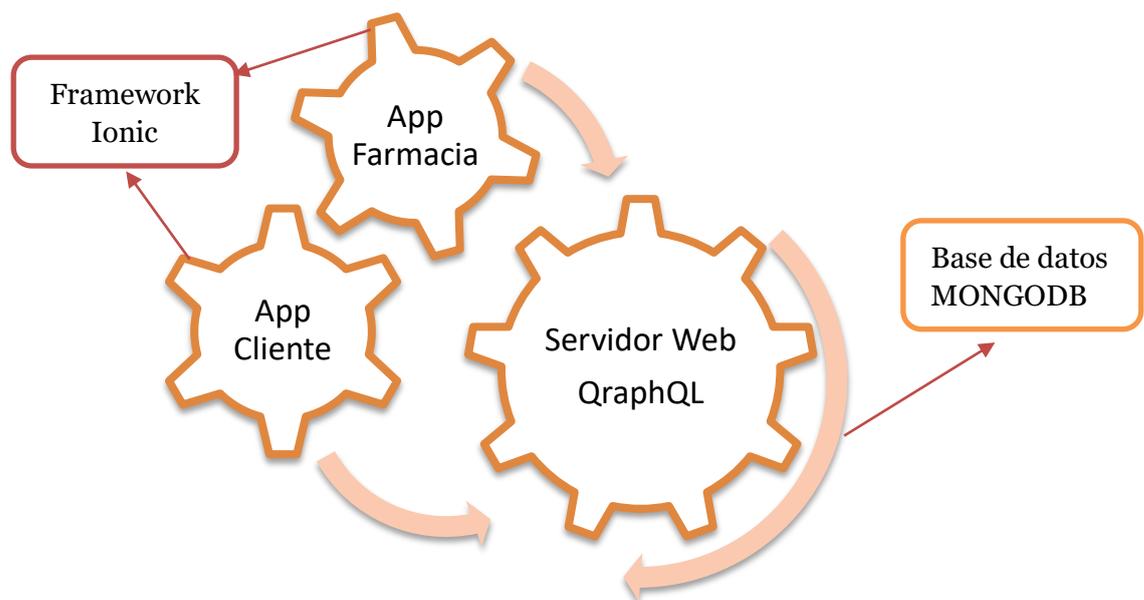


Figura 16: Diagrama de Despliegue

1.6.14 Arquitectura de la aplicación

Finalmente se muestra la arquitectura de la aplicación, donde se detalla técnicamente como está formada la arquitectura con sus diferentes componentes, sus funcionalidades y sus tecnologías.

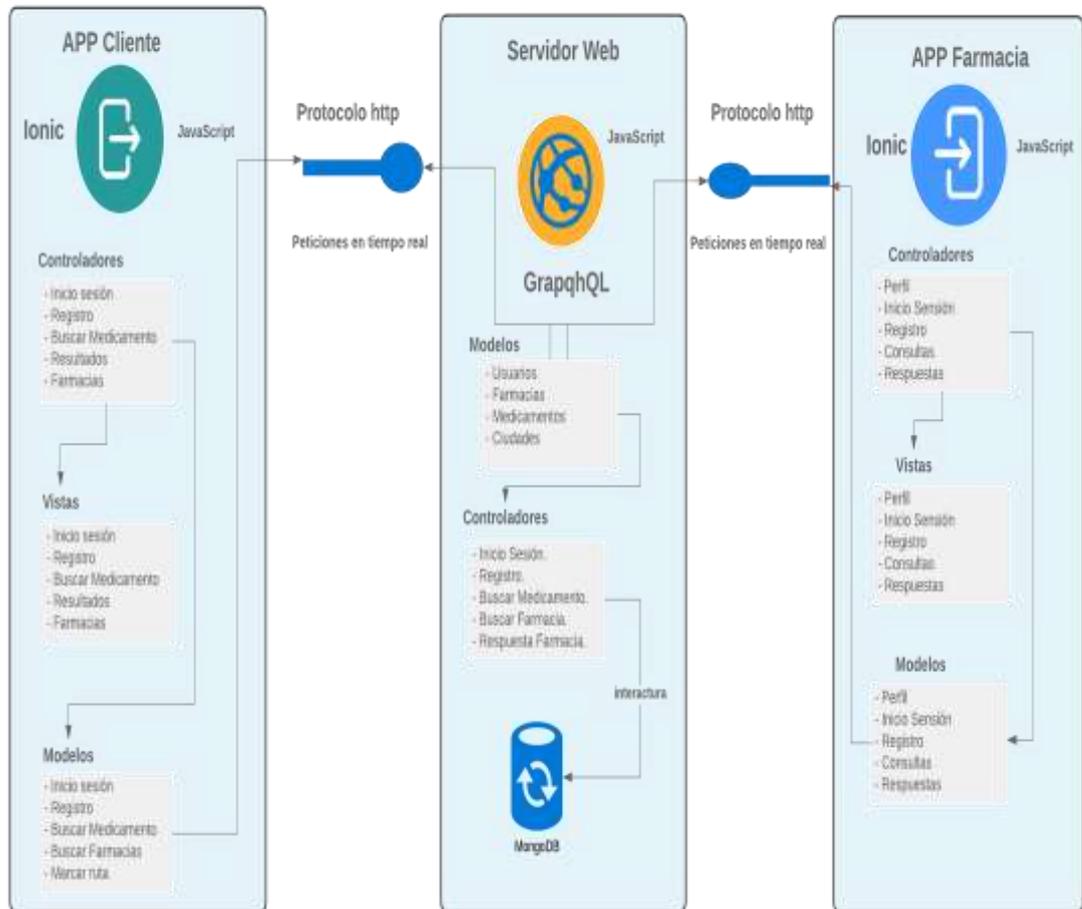


Figura 17: Arquitectura de Software

En breve resumen la arquitectura queda definida de la siguiente forma:

FrontEnd: dos aplicaciones móviles híbridas desarrolladas con el framework Ionic que despliega un mapa usando mapBox.

BackEnd: Servidor desarrollado en **GraphQL** que una a las dos aplicaciones mediante el protocolo http.

Todo desarrollado bajo la metodología Mobile-D, quedando así establecido la arquitectura del presente TT.

2. Desarrollo de la arquitectura definida

Luego de haber definido una metodología para el desarrollo se precedió al desarrollo de la arquitectura informática basados estrictamente a las condiciones de la metodología y sus fases.

2.1 Exploración

En esta etapa se define los requerimientos y alcance del proyecto, bases fundamentales para un adecuado desarrollo y exitoso producto.

2.1.1 Establecimiento de los Grupos de Interés o Stakeholders.

Los grupos de personas interesadas en la realización del presente proyecto son los siguientes.

- **Desarrollador:** es la persona encargada del análisis, desarrollo y pruebas de la aplicación.
- **Cliente:** es la persona que solicitó el producto y/o dueña del producto.
- **Usuarios:** son todas las personas que utilizan los servicios de la arquitectura en este caso particular los usuarios de las farmacias.

2.1.2 Requerimientos iniciales.

Se pretende realizar una arquitectura informática que permita geo localizar farmacias y la existencia de medicamentos en las mismas. En base a ello se ha recolectado la información que permita conocer las necesidades de los usuarios. A continuación un resumen de los principales puntos sobre los requerimientos para más detalles ver Anexo 1.

2.1.2.1 Requerimientos Funcionales.

Los requisitos funcionales son aquellos que indican qué debe hacer la aplicación, dichos requerimientos se presentan a continuación en la tabla VI.

TABLA XV: REQUERIMIENTOS FUNCIONALES

REQUERIMIENTO	PERTENECE	REQUISITO	DESCRIPCIÓN	CATEGORÍA
RF001	Aplicación de farmacéutico y usuario normal.	Obtener ubicación del usuario	Permite obtener la ubicación del celular en longitud y latitud.	Evidente
RF002	Aplicación farmacéutico y usuario normal.	Ingresar a la aplicación.	Permite ingresar mediante un correo y contraseña.	Evidente
RF003	Aplicación farmacéutico y usuario normal.	Registrarse	Permite registrarse en la aplicación ingresando datos personales.	Evidente
RF004	Aplicación usuario normal.	Marcar la ubicación de las farmacias en el mapa	Mostrará todos los marcadores de las diferentes farmacias en el mapa.	Evidente
RF005	Aplicación usuario normal.	Buscar farmacias por su nombre	Busca las farmacias por nombre	Evidente
RF006	Aplicación usuario normal.	Visualizar información de la farmacia.	Al usuario le permitirá visualizar la información de una farmacia	Evidente
RF007	Aplicación usuario normal	Mostrar ruta a la farmacia	Traza la ruta a la farmacia que el usuario eligió.	Evidente
RF008	Aplicación usuario normal	Buscar medicamento	Buscar medicamentos por nombres.	Evidente

RF009	Aplicación farmacéutico.	Seleccionar estado	Permite seleccionar un estado entre abierto y cerrado.	Evidente
RF010	Aplicación farmacéutico	Notificaciones sobre consultas.	Permite al usuario conocer sobre nuevas consultas con notificaciones	Evidente
RF011	Servicio web	Configuración.	Creación de los diferentes EndPoints para la interacción de las aplicaciones.	Evidente

2.1.2.2 Requerimientos no funcionales.

Los requerimientos no funcionales, son todos aquellos que no definen la funcionalidad de la aplicación, pero son necesarios para su correcto funcionamiento, dichos requerimientos se encuentran en la tabla VII.

TABLA XVI: REQUERIMIENTOS NO FUNCIONALES

Código	Descripción
RNF01	Rendimiento, el tiempo de respuesta de las consultas debe ser el mínimo.
RNF02	Soporte, estados de sin respuesta de las consultas y desconectado cuando no exista conectividad a internet.
RNF03	Amigable, el sistema utilizará interfaz amigable e intuitiva para el usuario.
RNF04	Disponibilidad, la aplicación debe estar disponible en un porcentaje aceptable.

2.1.2.3 Análisis de los requerimientos.

En base a los Requerimientos establecidos se ha podido determinar los procesos a realizar y los diferentes módulos iniciales.

TABLA XVII: MÓDULOS INICIALES

MÓDULO	CÓDIGO	PROCESO	REQUERIMIENTOS
Módulo de Visualización de Farmacias	P001	Visualizar las farmacias cercanas.	RF001, RF002
	P002	Visualizar información de una farmacia.	RF006
	P003	Visualizar la ruta hacia la farmacia.	RF007
Módulo de consultas	P004	Realizar la consulta de medicamentos y farmacias	RF008, RF005
	P005	Mostrar los resultados de las consultas	RF008,RF005
Módulo de estado	P006	Seleccionar el estado de la farmacia.	RF009
	P007	Visualización de las nuevas consultas.	RF010

2.1.3 Definición del alcance.

El alcance del proyecto de titulación se determina a través de su categoría, limitaciones y de los supuestos y dependencias.

2.1.2.4 Establecimiento de categoría

Los supuestos y dependencias de la aplicación son:

- La aplicación recoge datos de las farmacias a través de un Web Service
- Los datos obtenidos de la aplicación son únicamente acerca del Ecuador específicamente para la ciudad de Loja.

- Los usuarios manejan como idioma principal el español, siendo éste el lenguaje que se opera en la interfaz de la aplicación.

2.1.2.5 Limitaciones

Las limitaciones de la aplicación son:

2.3.1.1 La aplicación sólo puede ser ejecutada en dispositivos con el Sistema Operativo Android e IOS.

2.3.1.2 Para la ejecución de la aplicación es necesario tener una conexión a Internet o Datos móviles.

2.3.1.3 Para un correcto funcionamiento de la aplicación el dispositivo debe tener activado el GPS.

2.1.4 Establecimiento del proyecto

En esta etapa se determinan los recursos físicos y técnicos necesarios para el desarrollo del proyecto. Las herramientas a utilizar son las siguientes:

Justinmind: es una herramienta de creación de web y aplicación móvil de prototipos y de alta fidelidad. Ofrece capacidades se encuentran típicamente en herramientas como diagramas de arrastrar y soltar, modificación del tamaño, el formato y la exportación /importación de widgets.

Framework Ionic: proporciona herramientas y servicios para el desarrollo de aplicaciones móviles híbridas utilizando tecnologías web como CSS, HTML 5, y Sass. [34] Se utilizó para la interfaz gráfica de la aplicación en conjunto con

Apache Cordova: para compilar a las diferentes plataformas a desarrollar.

GraphQL: es un lenguaje de consulta para su API y un tiempo de ejecución del lado del servidor para ejecutar consultas mediante el uso de un sistema de tipos que usted define para sus datos.

Visual Studio Code: es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del

editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario.

Mongoose: es un marco de JavaScript que se usa comúnmente en una aplicación Node.js con una base de datos MongoDB.

NodeJs: es un entorno de ejecución para javascript multiplataforma de código abierto además de una enorme librería.

Git: es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

Robo 3T: Es el id para profesionales de MongoDB que permite hacer las pruebas necesarias de las bases de datos permitiendo manipular completamente las bases de datos.

2.2 Iniciación.

En esta etapa se realizan las actividades relacionadas a la configuración del ambiente de desarrollo y al diseño de la aplicación.

2.2.1 Configuración del ambiente de desarrollo.

En esta actividad se debe realizar la configuración de los elementos físicos y técnicos, en dejar claro que herramientas se va a utilizar en el trascurso del proyecto de manera que, al iniciar el desarrollo del mismo, todo transcurra sin ninguna novedad y mucho menos con retrasos. Para ello se instaló lo siguiente:

Backend:

- **NodeJs.-** es un entorno de ejecución para javascript multiplataforma de código abierto además de una enorme librería.
- **MongoDB.-** Base de datos NOSQL.
- **Robo 3T.-** Id de base de datos para MongoDB.
- **Mongoose.-** es un marco de JavaScript que se usa comúnmente en una aplicación Node.js con una base de datos MongoDB.

FrontEnd:

- **Ionic.**- framework
- **Apache Cordova.**- es un marco de desarrollo móvil de código abierto
- **Visual Studio Code.**- es un editor de código fuente desarrollado por Microsoft con un fuerte soporte para múltiples lenguajes de programación.
- **Git.**- es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

2.2.1.1 Capacitaciones.

Capacitación técnica al equipo de desarrollo sobre las tecnologías mencionadas anteriormente utilizando la información recolectada en la sección de Revisión de la Literatura del presente TT adicionalmente en las documentaciones oficiales de las herramientas disponibles en las páginas web oficiales.

2.2.1.2 Plan de comunicación.

En cuanto a la comunicación con los interesados, se resolvió realizar una reunión por cada entregable que se concluya con el fin de encontrar y corregir errores, realizar mejoras y de esta manera satisfacer los requerimientos antes mencionados.

2.2.2 Planificación Inicial

Se planifica el tiempo de duración de cada uno de los entregables y se asignan los recursos necesarios para que el desarrollo de cada uno de ellos tenga éxito y no ocurra ningún altercado en el desarrollo.

2.2.2.1 Exposición del plan del proyecto y la arquitectura de la aplicación

La arquitectura informática está orientada a geo referencia y consulta en tiempo real por lo que el proyecto consta de 3 partes como se muestra en la figura 9 (Ver Anexo 3):

- Aplicación móvil para farmacéutico.
- Aplicación móvil para usuario cliente.
- Web Service.

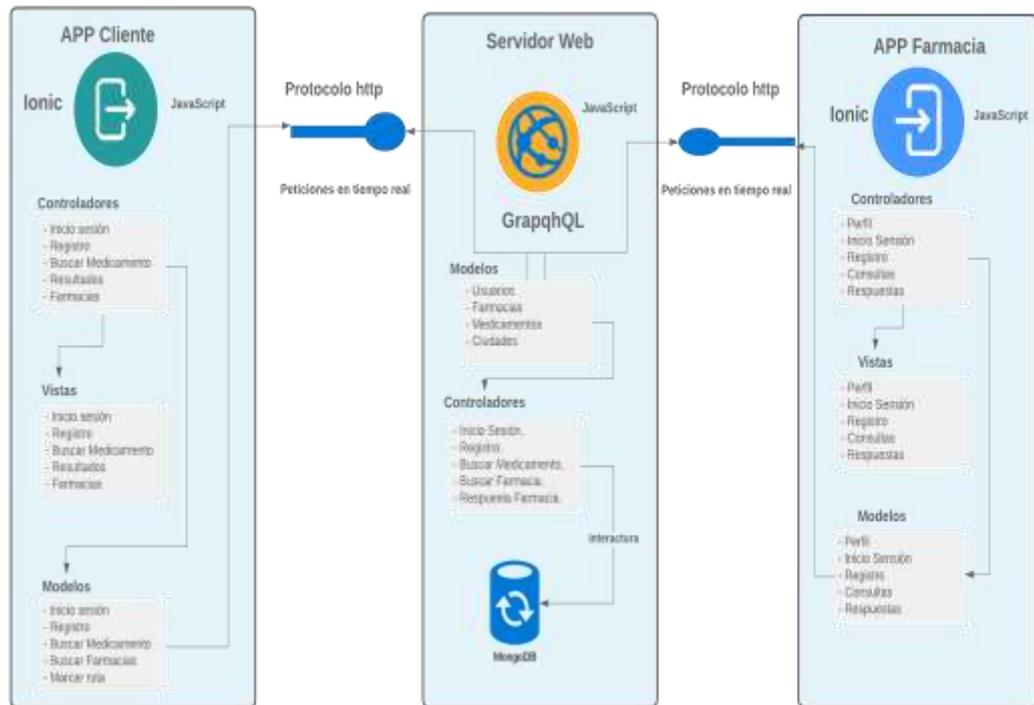


Figura 18: Diseño de la arquitectura

2.2.2.2 Análisis de procesos.

Una vez analizados los requerimientos iniciales se proponen los siguientes módulos y sus funcionalidades:

P001: Visualizar las farmacias cercanas.

- Registro y consulta de farmacias.
- Marcar coordenadas de las farmacias en el mapa.
- Consulta y registro de la ubicación del usuario

P002: Visualizar información de una farmacia.

- Marcar coordenadas de las farmacias en el mapa.
- Establecer información de farmacias a ser mostrada.

P003: Visualizar la ruta hacia la farmacia

- Marcar coordenadas de las farmacias en el mapa.
- Ubicación del usuario.
- Mostrar la ruta hacia la farmacia en el mapa.

P004: Buscar farmacias por su nombre.

- Búsqueda de farmacias por su nombre.
- Marcar coordenadas de las farmacias en el mapa.

P005: Buscar medicamentos por su nombre.

- Comunicación en tiempo real entre las dos aplicaciones móviles.
- Limitar las consultas a farmacias en estado activo.

2.2.2.3 Diseño de diagrama de clases.

Un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. A continuación el diseño del diagrama de clases.

UML Class Diagram Pharma

Leonardo Fabio Corregido Caraguby

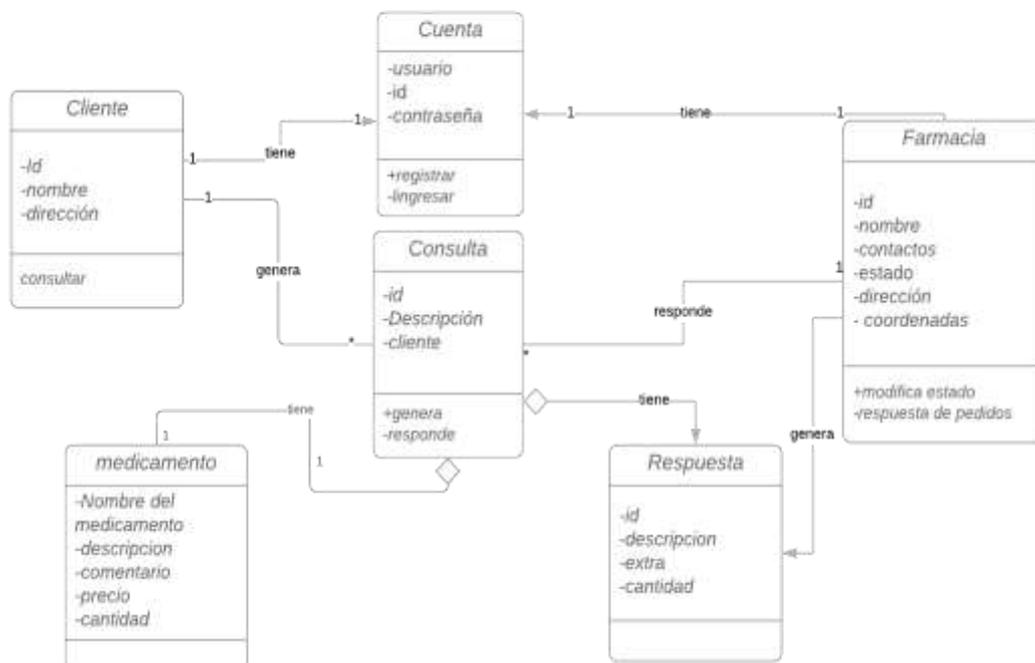


Figura 19: Diagrama de clases

Cliente: Clase cliente contiene una cuenta además de generar las consultas.

Cuenta: Clase cuenta que es el método padre y contiene información básica como usuario y contraseña.

Farmacia: Clase farmacia contiene una cuenta y genera las respuestas de las consultas.

Medicamento: Clase que describe los detalles del medicamento consultado.

Consulta: Clase consulta contiene información como medicamento y el cliente.

Respuesta: Clase respuesta contiene la información sobre las respuestas y sobre farmacias.

2.2.2.4 Definición de modelo de casos de usos.

Es una forma de diagrama de comportamiento UML mejorado. El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso. A continuación el diagrama definido para la arquitectura.

Basic Use Case Diagram Pharma

Leonardo Farió Cárquez Caraguay | January 21, 2022

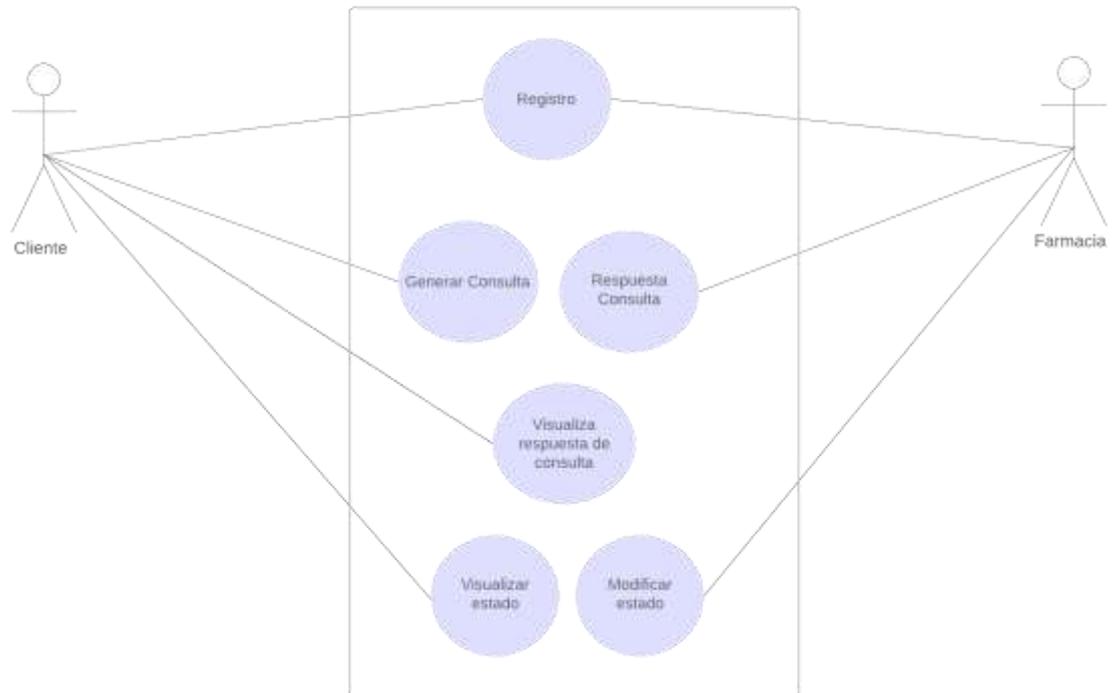


Figura 20: Diagrama de casos de uso

2.2.2.5 Elaboración de Prototipos de mediana fidelidad

Las pantallas planteadas buscan cumplir con los requerimientos de la arquitectura informática a continuación el diseño de las posibles pantallas. Cabe mencionar que los prototipos son una idea de las pantallas estas pueden cambiar o adaptarse luego a las necesidades que se presenten.

- **Iniciar Sesión.**

Prototipo de la pantalla inicio de sesión donde receptara los datos correo y contraseña para ingresar a la aplicación farmacéutico como usuario cliente.

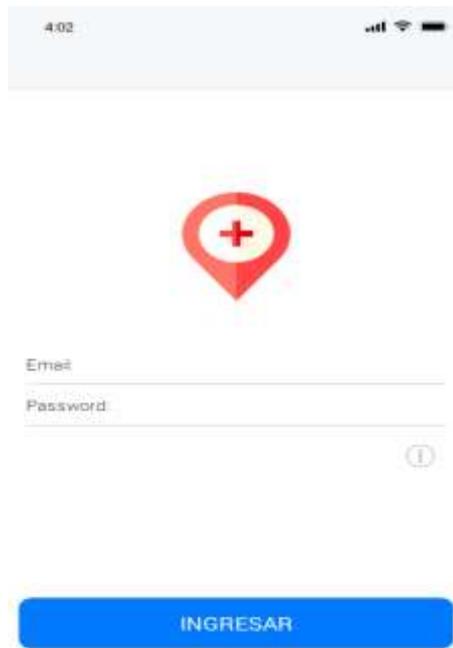


Figura 21: Prototipo pantalla Inicio de Sesión

- **Inicio**

Pantalla principal para aplicación farmacéutico donde puede elegir su estado.



Figura 22: Prototipo pantalla Inicio

- **Modal Nueva Consulta**

Modal con la información de las nuevas consultas y sus diferentes opciones para las respuestas.

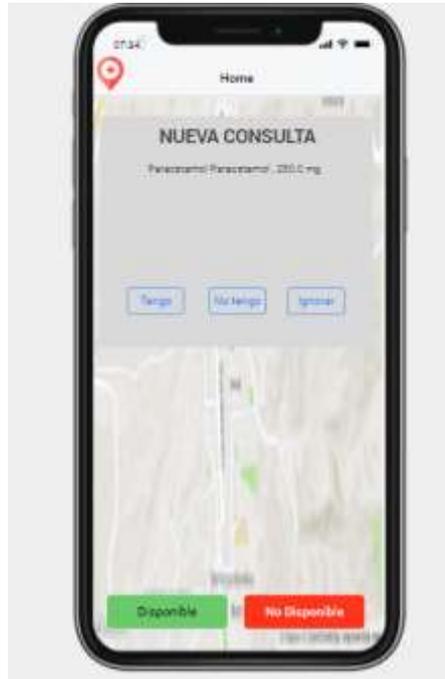


Figura 23: Prototipo pantalla Nueva Consulta

- **Menú Lateral.**

Menú Lateral con las diferentes opciones de la aplicación.

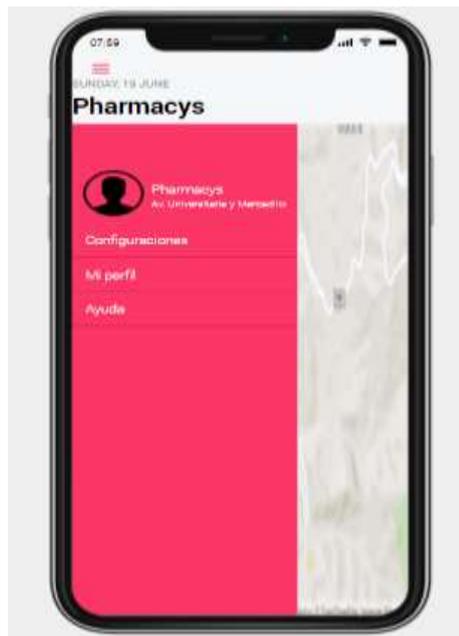


Figura 24: Prototipo Menú Lateral

- **Perfil**

Pantalla perfil con los principales detalles de la cuenta del usuario o farmacia.



Figura 25: Prototipo Mi Perfil

- **Registro**

Módulo de registro donde recibe los datos personales del usuario.



Figura 26: Prototipo Pantalla Registro

- **Información de Farmacia.**

Sección para visualizar información de la farmacia seleccionada en aplicación usuario.

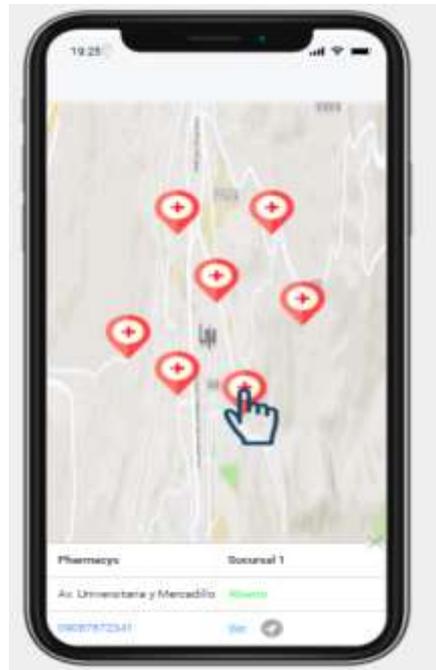


Figura 27: Prototipo Información Farmacia

- **Inicio**

Sección inicial para aplicativo usuario.

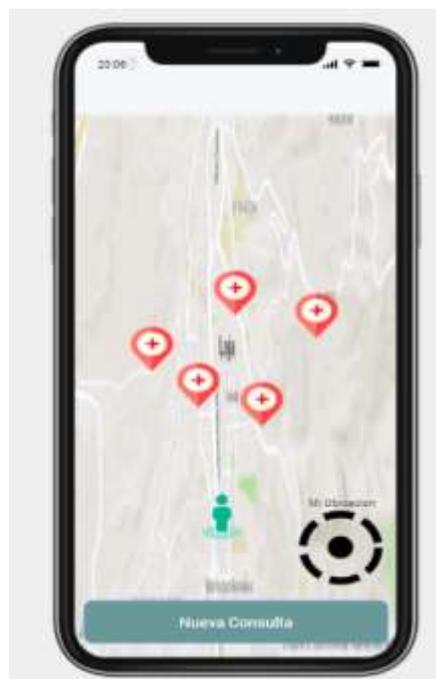


Figura 28: Prototipo Inicio Usuario

- **Nueva Consulta.**

Modal para generar nueva consulta ingresando el nombre del medicamento.

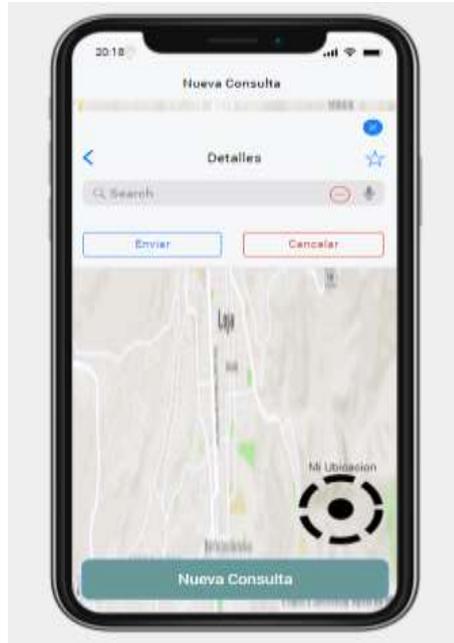


Figura 29: Prototipo Nueva Consulta

- **Resultados**

Modal de resultados de la consulta realizada.

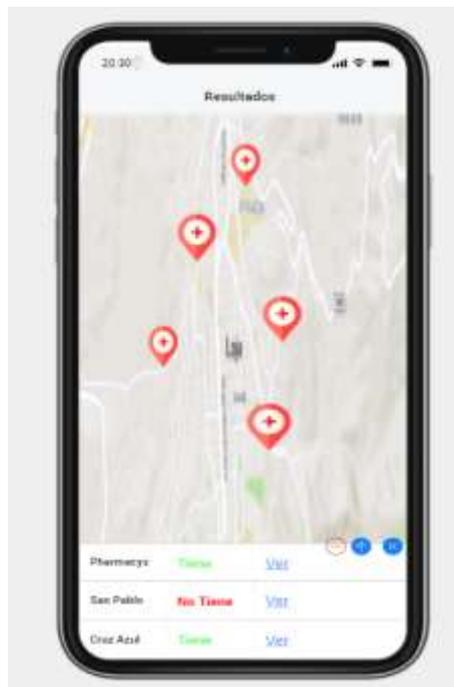


Figura 30: Prototipo Resultados

2.3 Producción

En la producción se realiza la codificación de los procesos diseñados y la posterior integración entre cada una de ellas para obtener el total funcionamiento de la aplicación.

Los objetivos de esta fase son:

- ✓ Implementar funcionalidades indicadas en los requerimientos.
- ✓ Obtener una documentación acerca de lo sucedido en esta fase denominado manual de programador.

2.3.1 Primera Iteración.

2.3.1.1 Día de Planificación.

La intención de este día es planificar todo acerca de la primera iteración como sus recursos, su alcance, su personal, tiempo, etc.

El objetivo de la primera iteración es instalar y verificar todas las herramientas mencionadas en la fase de iniciación en resumen se busca instalar y verificar lo siguiente:

- NodeJs
- Git
- Visual Studio Code
- Ionic
- MongoDB

2.3.1.2 Día de Trabajo

La prioridad principal de esta fase es culminar todo el trabajo planeado, tomando en cuenta los requisitos mencionados en el día de planificación para que al culminar la primera iteración los resultados sean los esperados.

Antes de la instalación de las diferentes herramientas se establece las características de la instalación, para la presente instalación de uso el Sistema Operativo Linux Debian en su versión 9 Stretch.

NodeJs

Para la instalación de NodeJs se sigue el procedimiento proporcionado por la documentación oficial encontrada en la web en resumen los pasos seguidos son:

Se usa el apt (administrador de paquetes de Linux) primero se actualiza su índice con:

```
$ sudo apt update
```

Luego se instala NodeJs desde los repositorios:

```
$ sudo apt install nodejs
```

Para verificar qué versión de Node.js ha instalado después de estos pasos iniciales, escriba:

```
$ nodejs -v
```

Ionic

Para la instalación de ionic primero se instala Cordova se usa el gestor de paquetes NPM proporcionado por NodeJs:

```
$ sudo npm install -g cordova
```

Luego utilizando el mismo gestor de paquetes NPM se instala Ionic:

```
$ sudo npm install -g ionic
```

Para verificar la instalación se usa:

```
$ ionic -v
```

Entonces se crea el primer proyecto usando:

```
$ ionic start HelloWorld blank
```

Donde HelloWorld es el nombre del proyecto.

Visual Estudio Code.

Para la instalación de Visual Code se siguen los pasos similares anteriores usando apt

Primero se actualiza el índice y luego se procede a instalar:

```
sudo apt-get install apt-transport-https  
sudo apt-get update  
sudo apt-get install code # or code-insiders
```

Git.

Para la instalación de git usamos los pasos encontrados en gitlab.com cabe mencionar que debemos contar con una cuenta en este sitio.

Entonces nos ubicamos en la carpeta principal del proyecto e iniciamos git con el comando:

```
git init
```

Creamos un proyecto nuevo en nuestra cuenta de gitlab lo nombramos y obtenemos la ruta del proyecto y agregamos la ruta en el proyecto local.

```
git remote add origin git@gitlab.com:LeonardCaraguay/pharm**.git
```

```
git add .
```

Agregamos los cambios y los guardamos junto con comentario para luego empujar todos los cambios agregados al repositorio de gitlab.

```
git commit -m "Initial commit"  
git push -u origin master
```

MongoDB.

De la misma manera instalaremos mongodb utilizando apt actualizamos el índice y procedemos a instalar

```
$ sudo apt-get install mongodb-org
```

2.3.1.3 Día de lanzamiento

Una vez culminado el día de trabajo, llega el día de presentación del primer entregable y para ello es necesario hacer una revisión acerca de lo descrito en el día de planificación y verificar si se cumplió lo establecido. Para proceder a realizar un resumen en donde se especifican las deficiencias encontradas durante la primera iteración, en este caso, no se hallaron ningún tipo de deficiencias por lo cual no se realiza el cuadro de resumen de deficiencias.

Luego de la revisión del día de planificación se concluye que se cumplió con todo lo descrito en este día, como evidencias las siguientes figuras.

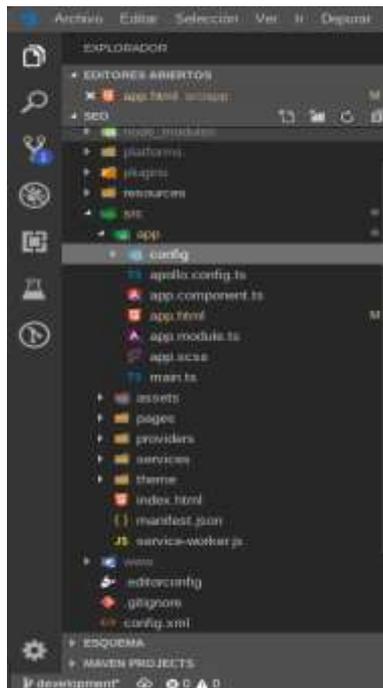


Figura 31: Directorio de proyecto Ionic en Visual Studio Code.

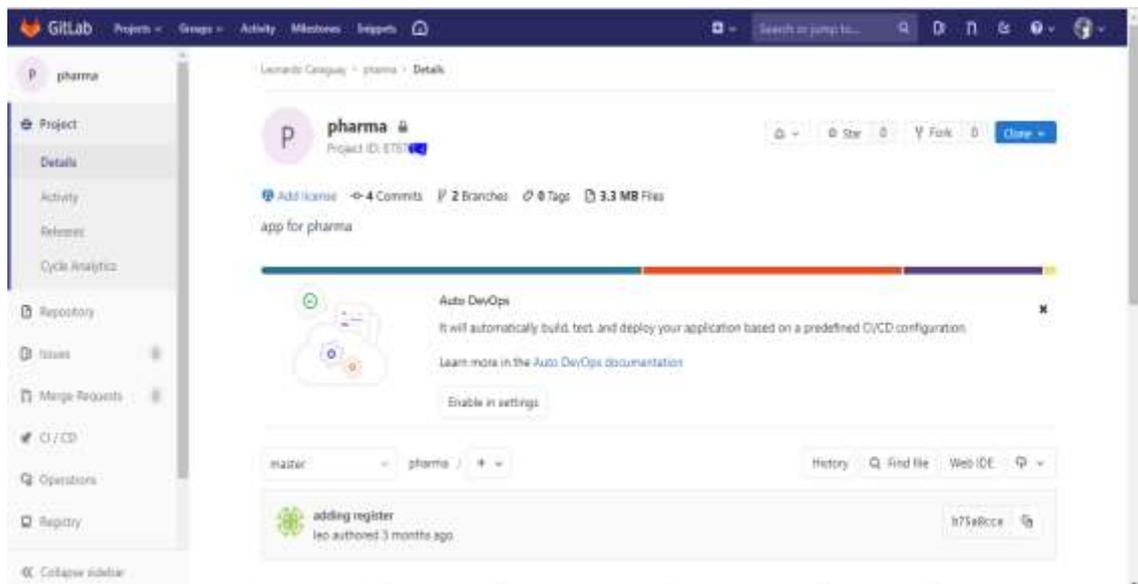


Figura 32: Proyecto en repositorio Git

2.3.2 Segunda Iteración.

2.3.2.1 Día de Planificación.

La intención de este día es planificar todo acerca de la segunda iteración como sus recursos, su alcance, su personal, tiempo, etc.

El objetivo de la segunda iteración es crear los principales módulos y funcionalidades de la primera parte del proyecto como es la aplicación para las farmacias en base a lo establecido en la fase de Iniciación.

Se basara en los prototipos diseñados y requerimientos mencionados anteriormente.

Para esta iteración se tomaran en cuenta los requerimientos establecidos únicamente para la aplicación para farmacias.

Se planifica terminar en 20 días laborables con los objetivos establecidos para esta iteración.

Se cuenta con el personal y los recursos necesarios para culminar exitosamente esta iteración a tiempo.

En resumen se busca los siguientes objetivos para esta iteración.

- Registro
- Inicio de Sesión
- Seleccionar Estado
- Mostrar y contestar consultas.

2.3.2.2 Día de Trabajo.

Para la codificación se basó en los objetivos planteados en el día de planificación.

Para la generación de nuevas pantallas se usa la siguiente línea de código `ionic generate page <nombre-pagina>` proporcionado por IonicCli donde se genera un archivo con extensión `.html` que es la vista del módulo, un archivo con extensión `.ts` que es el controlador y un archivo con extensión `.scss` que es el encargado del diseño y apariencia de la vista.

En la presente sección se muestran extractos de códigos usados, para mayores detalles revisar el Manual de Programador.

- **Registro.**

Para la codificación del módulo de registro es necesario establecer los datos a ser recolectados. Quedando como datos importantes a recolectar los siguientes:

- ✓ Nombre
- ✓ Dirección
- ✓ Slogan
- ✓ Teléfono
- ✓ Correo Electrónico.
- ✓ Y Ubicación en coordenadas.

A continuación un resumen del código utilizado.

```
16 <ion-content class="background-login" padding>
17   <div class="container main-container">
18     <div class="row userInfo">
19       <div class="col-xs-12 col-sm-6 col-md-offset-3">
20         <h3 class="text-center">CREAR CUENTA </h3>
21         <form role="form" class="" #userForm="ngForm" (ngSubmit)="register()"
22           <ion-label>
23             <input type="text"
24               required
25               name="Name"
26               [(ngModel)]="local.name"
27               (ngModelChange)="modelChanged($event)"
28               #name="ngModel">
29             <div class="label-text">Nombre</div>
30             <div *ngIf="name.errors && (name.dirty || name.touched)"
31               class="error">
32               <div [hidden]="!name.errors.required">
33                 El nombre es requerido.
34             </div>
35           </div>
36         </ion-label>
37         <ion-label>
38           <input type="text"
39             required
40             name="slogan"
41             [(ngModel)]="local.slogan"
42             (ngModelChange)="modelChanged($event)"
43             #slogan="ngModel">
44           <div class="label-text">Eslogan</div>
45           <div *ngIf="slogan.errors && (slogan.dirty || slogan.touched)"
46             class="error">
47             <div [hidden]="!slogan.errors.required">
48               El eslogan es requerido.
49           </div>
50         </div>
51 </ion-label>
```

Figura 33: Código vista de Registro de Usuario

- **Inicio de Sesión**

Para la codificación del Inicio de Sesión se establece que se requiere recolectar el correo electrónico o email y una contraseña.

En la siguiente figura un extracto del código para la vista.

```
21 <ion-content class="background-login" padding>
22 <div class="container main-container center-content">
23 <div>
24 <div class="col-xs-12 col-md-6 col-md-offset-3">
25 <form role="form" #loginForm="ngForm" (ngSubmit)="doLogin()">
26 <div class="">
27 <label>
28 <input type="text"
29 [(ngModel)]="account.email"
30 name="username"
31 #username="ngModel"
32 required
33 autocapitalize="none">
34 <div class="label-text">Correo Electrónico</div>
35 <div *ngIf="username.errors && (username.dirty || username.touched)"
36 class="error">
37 <div [hidden]="!username.errors.required">
38 Ingrese su Correo electrónico
39 </div>
40 </div>
41 </label>
42 </div>
43 <div class="">
44 <label>
45 <input type="password"
46 [(ngModel)]="account.password"
47 name="password"
48 #password="ngModel"
49 required>
50 <div class="label-text">Contraseña</div>
51 <div *ngIf="password.errors && (password.dirty || password.touched)"
52 class="error">
53 <div [hidden]="!password.errors.required">
54 Ingrese su clave.
55 </div>
56 </div>
57 </label>
58 </div>
```

Figura 34: Código de la vista de Inicio de Sesión.

- **Seleccionar Estado**

La selección de estado se realizara en la pantalla de inicio que es la que se muestra por defecto. Existen dos estados para seleccionar entre Disponible y No Disponible.

A continuación la figura con el código para la selección de estado.

```
<div class="buttons">
<button ion-button round class="available" color="secondary" (click)="updateStatus(true)">Disponible</button>
<button ion-button round class="busy" color="danger" (click)="updateStatus(false)">No disponible</button>
</div>
```

Figura 35: Código vista de Selección de Estado

- **Mostrar y contestar consultas.**

Para mostrar las consultas nuevas y dar contestación se usara un modal que se despliega desde la pantalla de inicio. Para la contestación de las consultas se tiene las opciones:

- No Tengo

- Tengo
- Ignorar
- Tengo genérico, con la opción de descripción acerca del mismo.
- Opción ingresar precio.

En la siguiente figura una parte del código del modal.

```

8 <ion-content class="conten-modal">
9 <div class="header">
10 <span class="label">Detalles de consulta</span>
11 <span class="query" *ngIf="query"><ion-icon name="list"></ion-icon> {{ query.text }}</span>
12 </div>
13 <div *ngIf="!showGeneric" class="button-group">
14 <button ion-button class="responsive-button" (click)="send('Tengo')" color="secondary">Tengo</button>
15 <button ion-button class="responsive-button" (click)="send('No Tengo')" color="danger">No Tengo</button>
16 <button ion-button class="responsive-button" (click)="showGenericSection()" color="generic">Tengo Genérico</button>
17 <button ion-button class="responsive-button" (click)="cancel()" color="gray">Ignorar</button>
18 </div>
19 <div class="generic" *ngIf="showGeneric">
20 <span class="label-input">Ingrese detalles del genérico</span>
21 <ion-input class="input-detail" type="text" [(ngModel)]="details" name="title"></ion-input>
22 <button ion-button class="responsive-button" (click)="sendGeneric('Tengo')" color="secondary">Enviar</button>
23 <button ion-button class="responsive-button" (click)="hideGeneric()" color="danger">Cancelar</button>
24 </div>
25 </ion-content>

```

Figura 36: Código de vista de Modal de Nuevas Consultas

2.3.2.3 Día de lanzamiento.

En base a lo establecido en el día de planificación se puede constatar que se cumplió con todos los objetivos sin percances mayores.

2.3.3 Tercera Iteración.

2.3.3.1 Día de Planificación.

El objetivo de este día es planificar todo acerca de la tercera iteración como sus recursos, su alcance, su personal, tiempo, etc.

El objetivo de la tercera iteración es crear los principales módulos y funcionalidades de la segunda parte del proyecto como es la aplicación para los clientes en base a lo establecido en la fase de Iniciación.

Se basara en los prototipos diseñados y requerimientos mencionados anteriormente.

Para esta iteración se tomaran en cuenta los requerimientos establecidos únicamente para la aplicación para los usuarios clientes.

Se planifica terminar en 30 días laborables con los objetivos establecidos para esta iteración.

Se cuenta con el personal y los recursos necesarios para culminar exitosamente esta iteración a tiempo.

2.3.3.2 Día de Trabajo

Como se estableció en el día de planificación se busca desarrollar los requerimientos de la aplicación usuario.

En esta sección para efectos de resumen se mostrara solo secciones de código con una breve descripción del mismo todos los detalles se encuentran en el Manual de Programador.

- **Pantalla Inicio.**

Donde se despliega un mapa con la ubicación de las diferentes farmacias y su estado.

A continuación el método para generar un mapa interactivo con MapBox cabe recalcar que se debe contar con un token proporcionado por el sitio oficial y la ubicación donde se debe mostrar el mapa por defecto.

```
createMap() {
  const loading = this.loadingCtrl.create({spinner: 'crescent'});
  loading.present();
  mapboxgl.accessToken = this.config.mBoxAccessToken;
  this.map = new mapboxgl.Map({
    container: 'map',
    pitch: 40,
    zoom: this.config.default_zoom,
    bearing: -15,
    center: [this.currentPosition.longitude, this.currentPosition.latitude],
    attributionControl: false,
    style: this.config.mBoxStyle
  });
};
```

Figura 37: Código para crear un mapa con MapBox.

Para obtener la ubicación actual del usuario se usa el siguiente método.

```
97 public getCurrentPosition() {
98     return Observable.create((observer) => {
99         if (window.navigator && window.navigator.geolocation) {
100             window.navigator.geolocation.getCurrentPosition(
101                 (position) => {
102                     const { latitude, longitude } = position.coords;
103                     const location = { latitude, longitude };
104
105                     window.localStorage.setItem('lastLocation', JSON.stringify(location));
106                     observer.next(location);
107                     observer.complete();
108                 },
109                 (error) => {
110                     switch (error.code) {
111                         case 1:
112                             observer.error(GEOLOCATION_ERRORS['errors.location.permissionDenied']);
113                             break;
114                         case 2:
115                             observer.error(GEOLOCATION_ERRORS['errors.location.positionUnavailable']);
116                             break;
117                         case 3:
118                             if (this._currentLocation && this._currentLocation.latitude) {
119                                 observer.next(this._currentLocation);
120                                 observer.complete();
121                             } else {
122                                 observer.error(GEOLOCATION_ERRORS['errors.location.timeout']);
123                             }
124                             break;
125                     }
126                 }, {
127                     timeout: 5000
128                 });
129             } else {
130                 observer.error(GEOLOCATION_ERRORS['errors.location.unsupportedBrowser']);
131             }
132         });
133     });
134 }
```

Figura 38: Código para obtener Ubicación Actual

- **Nueva Consulta**

Para generar una consulta se despliega un modal donde podemos ingresar los detalles de la consulta con las opciones de enviar y cancelar como se muestra en el siguiente código.

```

8 = <ion-content class="conten-modal">
9 =   <form role="form" class="" #userForm="ngForm" (ngSubmit)="sendConsult()">
10 =     <ion-grid>
11 =       <ion-row>
12 =         <ion-col class="label-input">
13 =           <label>Ingrese los detalles de su búsqueda</label>
14 =         </ion-col>
15 =       </ion-row>
16 =       <ion-row>
17 =         <ion-col size="1">
18 =           <ion-icon name="search"></ion-icon>
19 =         </ion-col>
20 =         <ion-col size="11">
21 =           <input placeholder="Ejm: ibuprofeno 400mg" class="input-sms" type="text"
22 =             required
23 =             name="text"
24 =             [(ngModel)]="message.text"
25 =             #text="ngModel">
26 =         </ion-col>
27 =       </ion-row>
28 =     </ion-grid>
29 =     <button class="cancel" ion-button type="submit" color="secondary">Enviar Consulta</button>
30 =   </form>
31 =   <button class="cancel" ion-button (click)="cancel()" color="cancel">Cancelar</button>
32 </ion-content>

```

Figura 39: Código Nueva Consulta

- **Sección Resultados.**

Para mostrar los resultados de las consultas se desplegara una sección en la página de inicio mostrando los detalles como nombre de farmacia dirección y respuesta.

Un extracto del código para mostrar los resultados.

```

16 <div *ngIf="showResults" class="results-container">
17 <div ion-row lass="row heading">
18 <ion-col class="text-left" col-10 col-sm-10 col-md-10 col-xl-10>
19 <h4>Resultados</h4>
20 </ion-col>
21 <ion-col class="text-left" col-2 col-sm-2 col-md-2 col-xl-2>
22 <ion-icon class="close-icon" ios="ios-close-circle" md="md-close-circle" (click)="closeResults()"></ion-icon>
23 </ion-col>
24 </div>
25 <div class="results-items">
26 <div *ngFor="let local of results" class="results-item" id="{{ local.name }}" (click)="goToLocal(local.location)">
27 <div class="row">
28 <div class="col-md-10">
29 <span class="fa fa-user"> </span> {{ local.name }}
30 <br>
31 <span class="fa fa-map-pin"></span>Dirección: {{local.address}}
32 </div>
33 <div class="col-md-12">
34 <i class="fas fa-check-double"></i> {{local.text}}
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>

```

Figura 40: Código para mostrar los resultados de las consultas

Donde los métodos `closeResults()` cerrara esta sección y `goToLocal()` nos lleva a la ubicación del local en el mapa.

2.3.3.3 Día de lanzamiento

En base a lo establecido en el día de planificación se puede constatar que se cumplió con todos los objetivos con algunos percances detallados a continuación.

- Error al crear el mapa.

Para crear un mapa en MapBox se debe contar con un token proporcionado y una configuración inicial de un estilo del mapa si no mostrara errores.

2.3.4 Cuarta Iteración.

2.3.4.1 Día de Planificación.

La intención de este día es planificar todo acerca de la cuarta iteración como sus recursos, su alcance, su personal, tiempo, etc.

El objetivo de la cuarta iteración es crear los principales módulos y funcionalidades de la tercera parte del proyecto como es el servidor web en GraphQL.

Se planifica terminar en 30 días laborables con los objetivos establecidos para esta iteración.

Se cuenta con el personal y los recursos necesarios para culminar exitosamente esta iteración a tiempo.

En resumen se busca los siguientes objetivos para esta iteración.

- Método para Inicio de Sesión.
- Método para Registro.
- Método para las consultas en tiempo Real.

2.3.4.2 Día de Trabajo

En esta sección para efectos de resumen se mostrara solo secciones de código con una breve descripción del mismo todos los detalles se encuentran en el Manual de Programador.

- **Inicio de Sesión**

El método de inicio de sesión se estableció en dos versiones una para la sección de farmacias y otra para clientes siento la lógica la misma con la excepción que los datos se busca en sus respectivos modelos.

En la siguiente figura se ilustra parte del método utilizado para el inicio de sesión de los usuarios donde se utiliza a email y password como parámetros principales.

```
login: async (email, password, User, SECRET, SECRET2) => {
  console.log(email, 'email');
  const user = await User.findOne({email})
  if(!user){
    return {
      success:false,
      errors:[{path:'email', message:'email does not exist'}]
    }
  }
  const validPassword = await bcrypt.compare(password, user.password)
  if(!validPassword){
    return {
      success:false,
      errors:[{path:'password', message:'Invalid Password'}]
    }
  }
  // console.log("user:[auth.login]",user);
  const refreshTokenSecret = user.password + SECRET2;
  // console.log('refreshTokenSecret:[auth.login]',user.password,SECRET2,refreshTokenSecret);
  const [token,refreshToken] = auth.getTokens(user, SECRET, refreshTokenSecret)

  return {
    success: true,
    token,
    id: user._id,
    refreshToken,
    errors: []
  }
},
```

Figura 41: Código Inicio de Sesión (Servidor)

- **Registro**

El registro también está dividido en dos versiones tanto como para farmacias como para clientes.

A continuación la figura con parte del código para registrar usuarios donde se guardan los datos principales establecidos en las iteraciones anteriores.

```

createUser: async (parent, args, {models:{User}, SECRET, SECRET2}) => {
  const otherErrors = [];
  const password = args.user.password;
  let login=null
  console.log(args, 'arguments');
  try{
    if(password.length<8){
      otherErrors.push({path: 'password', message:'Password must be greater than 8 characters!})
    }
    if(otherErrors.length){
      throw otherErrors;
    }
    const hashPassword = await bcrypt.hash(password, 10);
    const user = await User.create({...args.user, password: hashPassword});
    if(user && user._id){
      login = await auth.login(user.username, password, User, SECRET, SECRET2);
    }
    return {
      success: user && user._id,
      errors: []
    };
  }catch(error){
    return {
      success: false,
      errors: formatErrors(error,otherErrors)
    };
  }
}

```

Figura 42: Código para Registrar Usuarios (Servidor)

- **Consultas**

El método de consultas en tiempo real es controlado por un tipo de operación que ofrece graphql denominado Subscription que son una forma de enviar datos desde el servidor a los clientes que eligen escuchar mensajes en tiempo real desde el servidor.

En la siguiente figura se muestra la configuración de esta operación para manejar las consultas.

```

1  import { withFilter } from 'graphql-subscriptions';
2  import { pubsub } from '../subscriptions';
3  const MESSAGE_ADD = 'messageAdded';
4  const FOO = 'statusUpdated';
5
6  export default {
7    Subscription: {
8      messageAdded: {
9        subscribe: withFilter(
10         () => pubsub.asyncIterator(MESSAGE_ADD),
11         (payload, args) => {
12           return payload.category === args.category;
13         }
14       )},
15      statusUpdated: {
16        subscribe: withFilter(
17         () => pubsub.asyncIterator(FOO),
18         (payload, args) => {
19           return payload.category === args.category;
20         }
21       )}
22    }
23  }

```

Figura 43: Configuración de Suscripción para mensajes en tiempo real.

Esto creara una operación de suscripción llamado **messageAdded** ahora solo queda llamar a esta operación cada vez que se genere un nuevo mensaje para notificar a todos los clientes subscriptos a esta operación.

```
Mutation: {
  addMessage: async (parent, args, { models }) => {
    return models.Message.create({
      text: args.message.text,
      by: args.message.by,
      category: args.message.category,
    })
    .then(message => {
      return message;
    })
    .then(message => {
      pubsub.publish(MESSAGE_ADD, {
        category: args.message.category,
        messageAdded: {
          _id: message._id,
          text: args.message.text,
          client: args.message.client,
          local: args.message.local,
          by: args.message.by,
          category: args.message.category
        }
      });
    });
  }
};
```

Figura 44: Código de Mensaje en tiempo real

Entonces cada vez que se genere un mensaje este evento se publicara en la operación MESSAGE_ADD.

2.3.4.3 Día de lanzamiento

Luego de revisar lo establecido en el día de planeación se concluye que se cumplió con todos los objetivos en el tiempo establecido con algunos percances detallados a continuación.

- Error al guardar los registros en la base de datos.

Se debe levantar el servidor de mongo utilizando mongoose en un puerto definido estableciendo la variable useMongoClient en true y luego levantar el servidor graphql. Como se muestra a continuación.

```
//SERVER
mongoose.connect(process.env.MONGODB_URI, {useMongoClient: true}).then(
  () => {
    console.log('Conectado a Mongo!!!!')
    server.listen(process.env.PORT, ()=>{
      console.log('Corriendo GRAPHQL server...');
    });
  });
```

```
});  
}  
)
```

- Por cada modelo creado se debe establecer un tipo detallando los tipos de datos a guardar.

2.4 Estabilización.

El propósito de esta fase es integrar todos los módulos y comprobar el correcto funcionamiento de la aplicación que se realizara en una sola iteración.

Los objetivos de esta fase son:

- ✓ Finalización de la aplicación.
- ✓ Finalizar la documentación.
- ✓ Mejorar y garantizar la calidad de la aplicación

Las entradas de la fase de producto son:

- Funcionalidad implementada
- Artefactos de desarrollo relacionado.

Los entregables de esta fase son:

- ✓ Funcionalidad implementada de la aplicación.
- ✓ Documentación del producto finalizado.

2.4.1 Iteración 1

2.4.1.1 Día de Planificación.

El propósito de este día es planificar toda la actividad a realizarse en el día de trabajo además de sus objetivos y complementos a utilizar.

El objetivo para esta iteración es obtener la funcionalidad implementada.

Se estima contar con los recursos necesarios.


```

  }
}

/**
 * Send a JWT request to our login endpoint with the data
 * the user entered on the form.
 */
register(args) {
  let req = this.apollo.mutate({
    mutation: mutations.createUser,
    variables: {user: args},
  }).promise();
  if (args) {
    req.subscribe(({ data }) => {
      console.log(data, "registro de éxito");
    }, (error) => {
      console.log("error en registro", error);
    });
  }
  return req;
}

getUser(id: string) {
  let req = this.apollo.query({
    query: queries.getUser,
    variables: { id: id },
  });
  req.subscribe(({ data }) => {
    console.log(data, "obtencion usuario exitosa");
  }, (error) => {
    console.log("error en obtencion usuario", error);
  });
  return req;
}

```

Figura 46: Código de métodos disponibles en el servidor.

2.4.1.3 Día de Lanzamiento.

Una vez desarrollados y probados de forma individual los procesos que componen la aplicación, se procede a integrar y probar en el navegador para verificar el funcionamiento y poder llegar a una versión estable se repite este proceso las veces necesarias hasta obtener el objetivo planteado.

Se concluye que se termina con éxito la presente iteración cumpliendo con el objetivo planteado el día de planificación.

Con la redacción del presente documento se cumplen con totalidad todos los objetivos de la fase de estabilización.

3. Evaluar que la arquitectura permita la geo localización de los medicamentos en las diferentes farmacias.

Al término de las fases de desarrollo y estabilización se procedió con la fase de pruebas que comprende el tercer objetivo del presente TT.

3.1 Pruebas

Para la realización de las pruebas se tomó como referencia el estándar ieee 829 que sirve para pruebas del sistema y la realización de los documentos de la misma, la norma específica el formato de los documentos, pero no estipula si todos deben ser producidos ni incluye ningún criterio sobre el contenido adecuado para estos documentos por lo que cualquier modificación o adaptación sobre presente proyecto es válida.

3.1.1 Identificador único del documento.

DOC-001

3.1.2 Introducción.

El documento a continuación describe el alcance, la aproximación, los recursos y la planificación y las actividades necesarias. Identifica elementos de prueba, las características que deben probarse, las tareas de prueba, lo que hará cada tarea.

Para las pruebas fue necesario alojar el servidor en la nube se hizo uso de una plataforma en la nube Heroku que es un servicio en la nube que admite varios lenguajes de programación como java, ruby, php y el usado en el presente proyecto NodeJs.

Además se alojaron los instaladores en la nube para mejor experiencia de los usuarios que van a ser parte de las pruebas.

Links:

- ✓ Seo(Cliente):
https://drive.google.com/drive/folders/1CRQXgdHwtUL_YmBIBg_6nCr3A3P8qCpA?usp=sharing
- ✓ Pharma(Farmacias):
https://drive.google.com/drive/folders/1GOPx_sBoxKI0VIQ9GiiP9FJ5NYaxY8jX?usp=sharing
- ✓ Servidor:

<https://seo-be.herokuapp.com/graphiql>

3.1.3 Elementos

Los elementos a ser probados son tres:

- ✓ Aplicación móvil de Farmacias denominada **Pharma** en su versión 001.
- ✓ Aplicación móvil de Clientes denominada **Seo** en su versión 001.
- ✓ Servidor Web.

3.1.4 Características a probar.

Las características a ser probadas están relacionadas con:

- Interfaz
- Funcionalidad
- Soporte

Interfaz.- Que cuente con todas las pantallas definidas en la especificación de requerimientos (Ver Anexo 1).

Soporte.- Que sea tolerante a fallos e informe de los mismos a los usuarios.

Funcionalidad.- De acuerdo a lo especificado en los requerimientos las funciones a ser probadas son las siguientes.

TABLA XVIII: CASOS DE PRUEBAS

Código	Caso de prueba	Detalle	Requerimiento
CP001	Obtener ubicación del usuario	Muestra la ubicación actual del usuario	RF001
CP002	Inicio de Sesión	Ingreso mediante un usuario y contraseña	RF002

CP003	Registro	Registro de usuario nuevo	RF003
CP004	Mostrar farmacias en el mapa	Mostrar la ubicación de las diferentes farmacias en el mapa	RF004
CP005	Búsqueda	Buscar farmacias por nombre	RF005
CP006	Información de la farmacia	Información detalla de la farmacia	RF006
CP007	Ruta hacia farmacia	Marcar la Ruta hacia la farmacia	RF007
CP008	Búsqueda de medicamento	Buscar medicamento mediante el nombre	RF009
CP009	Cambiar estado	Cambiar el estado de una farmacia y visualizar inmediatamente en su información	RF010
CP010	Notificación de consultas	Notificación de una consulta nueva.	RF011

3.1.5 Características que no se deben probar.

Las características que no serán probadas están relacionadas con:

- Errores relacionados con el tiempo.
- Condiciones de errores no detectadas.
- Condiciones especiales de los datos.

- Fallos de configuración/compatibilidad con el software.
- Incapacidad de soportar el volumen de carga o fallos de software.

3.1.6 Enfoque.

De acuerdo al criterio de los involucrados en el presente TT las pruebas se realizaran en un ambiente simulado y debido a la capacidad operativa que implica las pruebas en un ambiente real.

Se tomara como usuarios de prueba a estudiantes de 10mo ciclo paralelo “B” de la carrera de ingeniería en sistemas que cursan el presente año en la Universidad Nacional de Loja, dicha población es de 28 estudiantes los cuales se dividirán en dos secciones para probar la aplicación Pharma y la aplicación Seo.

El éxito de las pruebas estará determinada por esta población y de ser necesario se repetirá varias veces las pruebas hasta obtener el cien por ciento de éxito en las pruebas.

La forma de medir el éxito de las pruebas será mediante una encuesta contestada por los involucrados en las pruebas.

3.1.7 Criterio de aprobación.

Esta sección describe los criterios de éxito para evaluar los resultados de las pruebas.

TABLA XIX: CASO DE PRUEBA MI UBICACIÓN

Caso de prueba: CP001	
Código: CR001	Requerimiento: RF01
Descripción: Verificar que el usuario puede visualizar la ubicación de su dispositivo en el mapa.	

<p>Condición de ejecución:</p> <ul style="list-style-type: none"> • Haber ejecutado la aplicación. • Haber iniciado sesión.
<p>Pasos:</p> <ul style="list-style-type: none"> • El usuario hace clic en el botón mi ubicación disponible en la sección inicio de la aplicación.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Visualizar en el mapa un marcador con la ubicación del usuario.

TABLA XX: CASO DE PRUEBA INICIO DE SESIÓN

Caso de prueba: CP002	
Código: CR002	Requerimiento: RF02
<p>Descripción: Comprobar que el usuario puede ingresar a la aplicación mediante un usuario y contraseña.</p>	
<p>Condición de ejecución:</p> <ul style="list-style-type: none"> • Haber ejecutado la aplicación. • Tener una cuenta registrada. 	
<p>Pasos:</p> <ul style="list-style-type: none"> • Ingresar un usuario y contraseña. Hacer clic en ingresar. 	
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Navegar hacia el modulo principal de la aplicación automáticamente luego de ingresar. 	

TABLA XXI: CASO DE PRUEBA REGISTRO.

Caso de prueba: CP003	
Código: CR003	Requerimiento: RF03
Descripción: Comprobar que el usuario pueda registrarse llenando sus datos principales.	
Condición de ejecución:	
Pasos: <ul style="list-style-type: none"> • Clic en la opción registrarme. • Llenar los campos solicitados. • Clic en registrar. 	
Resultado esperado: <ul style="list-style-type: none"> • Navegar automáticamente al módulo principal de la aplicación. 	

TABLA XXII: CASO DE PRUEBA VISUALIZAR LAS FARMACIAS.

Caso de prueba: CP004	
Código: CR004	Requerimiento: RF04
Descripción: Comprobar que el usuario pueda visualizar las diferentes farmacias en el mapa.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Navegar en el mapa para observar. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar las farmacias en el mapa. 	

TABLA XXIII: CASO DE PRUEBA INFORMACIÓN DE FARMACIA.

Caso de prueba: CP006	
Código: CR006	Requerimiento: RF06
Descripción: Comprobar que el usuario pueda visualizar la información de la farmacia.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Clic en un marcador de farmacia que esté disponible. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar los datos de la farmacia como nombre dirección teléfono, etc. 	

TABLA XXIV: CASO DE PRUEBA BÚSQUEDA DE FARMACIA.

Caso de prueba: CP005	
Código: CR005	Requerimiento: RF05
Descripción: Buscar farmacia por nombre.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Clic en la opción buscar farmacia. • Ingresar el nombre de una farmacia. • Clic en el botón enviar. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar las coincidencias encontradas con el nombre ingresado. 	

TABLA XXV: CASO DE PRUEBA RUTA

Caso de prueba: CP007	
Código: CR007	Requerimiento: RF07
Descripción: Marcar la ruta hacia la farmacia.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Clic en un marcador de farmacia. • Clic en detalles • Clic en el botón marcar ruta. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar la ruta hacia la farmacia además de distancia y tiempo. 	

TABLA XXVI: CASO DE PRUEBA BUSCAR MEDICAMENTO

Caso de prueba: CP008	
Código: CR008	Requerimiento: RF08
Descripción: Buscar medicamento por nombre.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Clic en la opción buscar medicamento. • Ingresar el nombre de un medicamento. • Clic en el botón enviar. 	
Resultado esperado:	

- Visualizar las coincidencias encontradas con el nombre ingresado con el nombre del respectivo local.

TABLA XXVII: CASO DE PRUEBA CAMBIO DE ESTADO DE FARMACIA

Caso de prueba: CP009	
Código: CR009	Requerimiento: RF09
Descripción: Cambiar el estado de la farmacia	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. 	
Pasos: <ul style="list-style-type: none"> • Clic en una de las opciones de estado entre disponible y no disponible. 	
Resultado esperado: <ul style="list-style-type: none"> • Visualizar el cambio de estado en la pantalla. 	

TABLA XXVIII: CASO DE PRUEBA NUEVAS CONSULTAS

Caso de prueba: CP010	
Código: CR010	Requerimiento: RF10
Descripción: Visualizar las nuevas consultas.	
Condición de ejecución: <ul style="list-style-type: none"> • Haber iniciado sesión. • Establecer el estado en disponible. 	

Pasos: <ul style="list-style-type: none">• Clic en las diferentes opciones disponibles para contestar las consultas.
Resultado esperado: <ul style="list-style-type: none">• Visualizar la respuesta enviada en la aplicación Seo.

3.1.8 Criterio de suspensión.

Los motivos por los cuales se suspendan las pruebas contemplan:

- No disponibilidad del usuario.
- Errores relacionados con el hardware.

3.1.9 Prueba de entregables.

Los resultados de la prueba son los documentos que entregará el equipo de pruebas al final del proceso de prueba.

- ✓ Informe de pruebas de aceptación.

3.1.10 Necesidades ambientales.

En cuanto a:

SOFTWARE y HADWARE:

- Teléfono celular inteligente con conexión a internet.
- Sistema operativo Android versión superior a 5.1.

3.1.11 Responsabilidades.

Leonardo Caraguay: Pruebas de software y documentación.

3.1.12 Necesidad del personal y formación.

Familiarizado con la manipulación de un celular inteligente.

3.1.13 Riesgos y contingencias.

Posibles riesgos:

- No contar con el equipo necesario para las pruebas.
- No contar con conexión a internet.

Contingencias:

- ✓ Se proveerá al usuario un equipo temporal para las pruebas.
- ✓ Se proveerá al usuario conexión a internet estable.

3.1.14 Aprobaciones.

Las pruebas están avaladas por el encargado de las pruebas y su documentación.

Leonardo Caraguay

José Guamán

Encargado de las pruebas

Director del presente TT

3.2 Reporte de pruebas.

Como se estableció en el documento anterior las pruebas están vinculadas a los ejes de funcionalidad, soporte e interfaz.

3.2.1 Pruebas de interfaz.

Este tipo de pruebas tienen por objetivo comprobar que el diseño y conexión de los prototipos de pantalla es igual al obtenido en la aplicación desarrollada.

3.2.1.1 Verificación de pantallas.

A continuación, se describe una tabla en la que se puede verificar que la aplicación posee las mismas pantallas que se definió en los prototipos de la fase de inicialización.

TABLA XXIX: PRUEBAS DE INTERFAZ

Prototipos de Pantalla	Pantallas de Aplicación	Cumplimiento
Registro	Registro	✓
Pantalla Principal	Inicio	✓
Menú Lateral	Menú Lateral	✓
Búsqueda de medicamentos	Modal de búsqueda	✓
Resultados	Sección de resultados	✓
Búsqueda de farmacias	Modal de búsqueda	✓
Inicio de sesión	Inicio de sesión.	✓
Información de farmacias	Modal información de farmacias	✓

3.2.2 Pruebas de soporte.

Para comprobar que la aplicación realiza un control adecuado sobre los errores comunes y presentarlos al usuario.

PS01: Error de datos de entrada.

Para comprobar que la aplicación realiza un adecuado control sobre la información ingresada hacia esta, se desarrollaron pruebas específicas sobre las pantallas que poseen entrada de datos.

En la aplicación las pantallas que poseen un ingreso de datos son: inicio de sesión, registro, búsqueda de farmacias, búsqueda de medicamentos.

A continuación un ejemplo del control que se realiza para verificar que los campos de entradas sean llenados correctamente.



Figura 47: Prueba de soporte de datos de entrada

PS02: Error de ubicación.

Es necesario tener activado la ubicación para el correcto funcionamiento de las aplicaciones y cuando éste no se encuentre activo debe ser informado.

En la siguiente figura se muestra el control realizado ante este error.



Figura 48: Prueba de soporte ubicación

Ps03: Error de conexión.

Es indispensable la conexión a internet pero cuando ésta no se encuentra disponible es necesario que el usuario sea notificado de este error.

En la figura a continuación se muestra el soporte ante este posible percance.



Figura 49: Prueba de soporte no conexión

3.2.3 Pruebas de funcionalidad.

En cuanto a las pruebas funcionales se procedió a verificar los ítems descritos en los criterios de aceptación del plan de pruebas.

3.2.3.1 CR001: Obtener ubicación del usuario

La ejecución de la prueba realizada se muestra en la Figura 50.

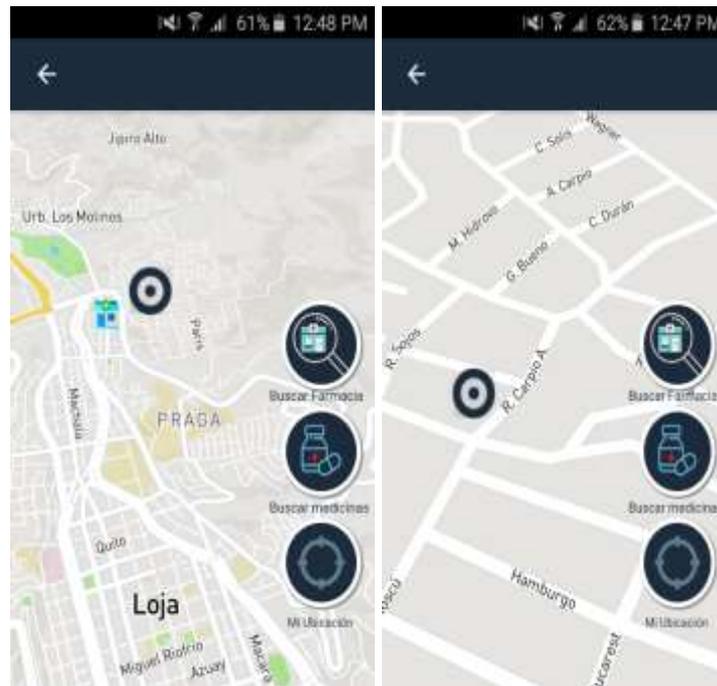


Figura 50: Caso de prueba mi ubicación

3.2.3.2 CR002: Inicio de sesión.

La ejecución de la prueba se muestra en el figura 51.

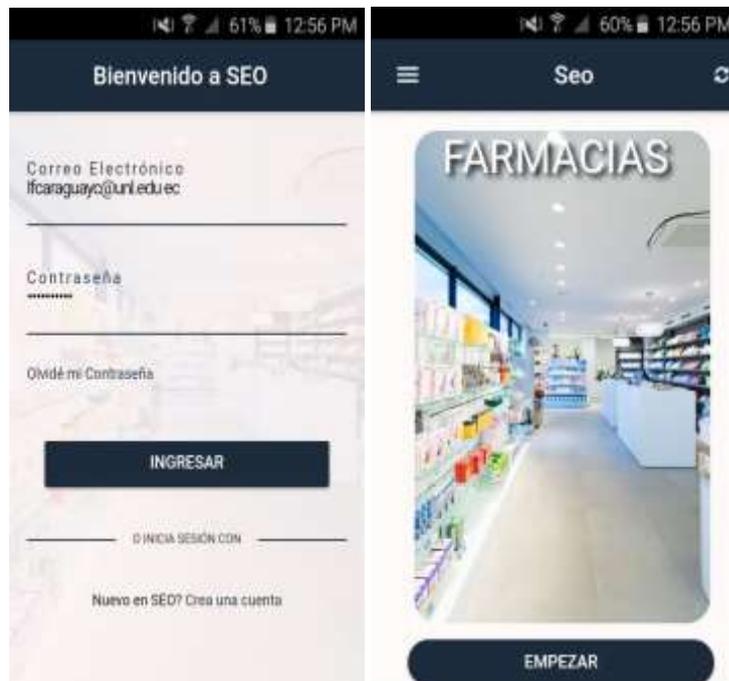


Figura 51: Caso de prueba inicio de sesión

3.2.3.3 CR003: Registro

En la figura 52 se muestra la ejecución de prueba.



Figura 52: Caso de prueba de registro.

3.2.3.4 CR004: Mostrar farmacias.

La ejecución de la prueba se muestra en la figura 53.

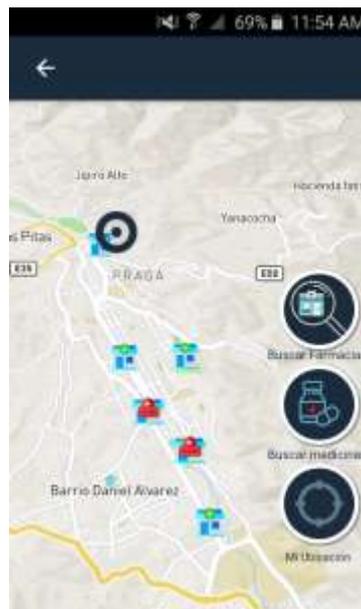


Figura 53: Caso de prueba mostrar farmacias

3.2.3.5 CR005: Buscar farmacia por nombre

En la siguiente figura se muestra la ejecución de la prueba.

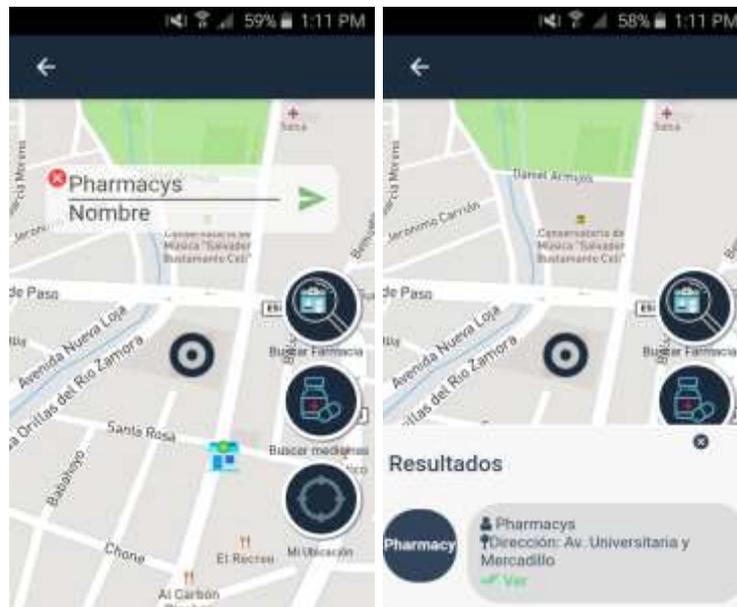


Figura 54: Caso de prueba búsqueda de farmacia

3.2.3.6 CR006: Caso de prueba detalles de farmacia.

En la siguiente figura se muestra la ejecución de la prueba.

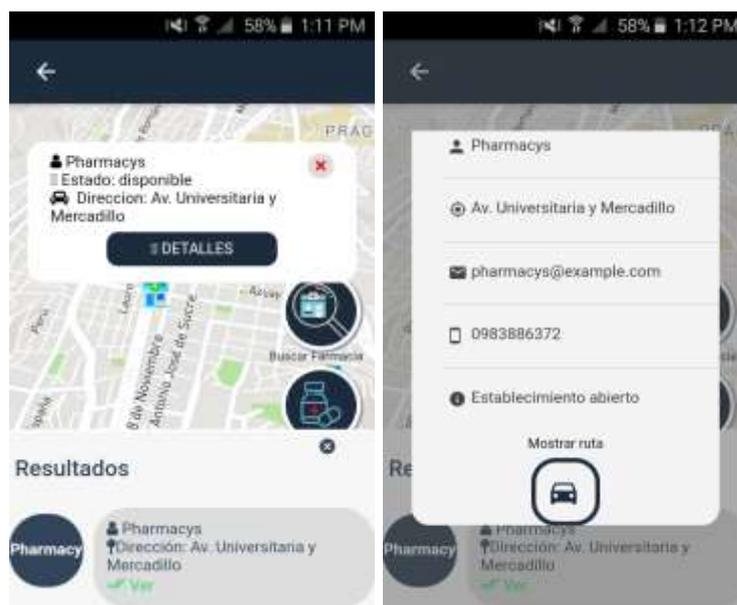


Figura 55: Caso de prueba detalles de farmacia

3.2.3.7 CR007: Ruta hacia la farmacia.

En la siguiente figura se muestra la ejecución de la prueba.

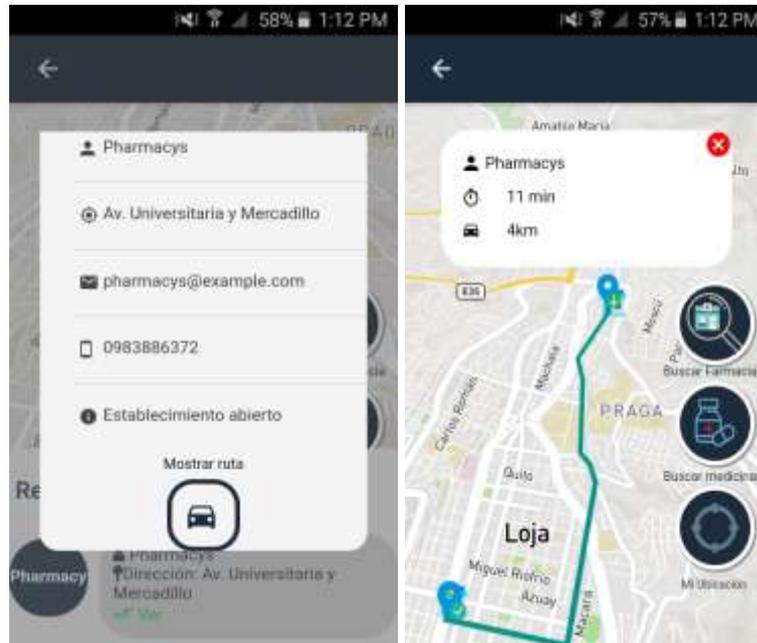


Figura 56: Caso de prueba ruta hacia la farmacia

3.2.3.8 CR008: Búsqueda de medicamento.

En la siguiente figura se muestra ejecución de la prueba.

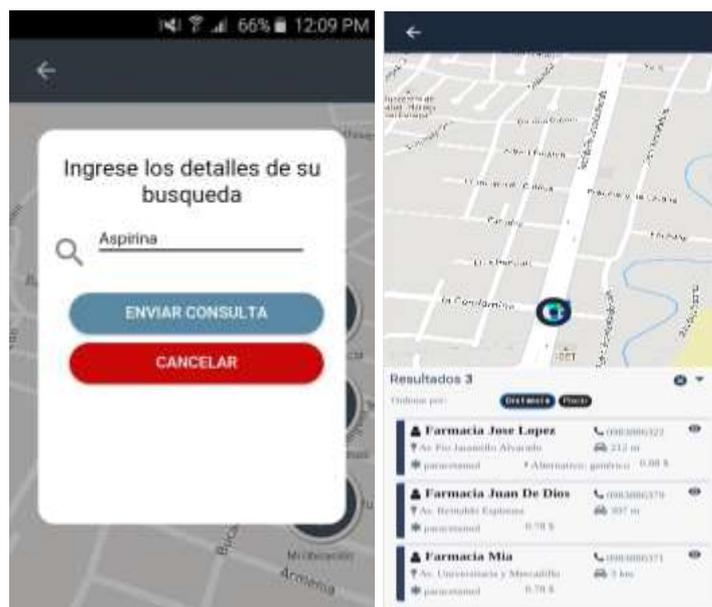


Figura 57: Caso de prueba búsqueda de medicamento

3.2.3.9 CR009: Cambio de estado.

En la siguiente figura se muestra la ejecución de la prueba.

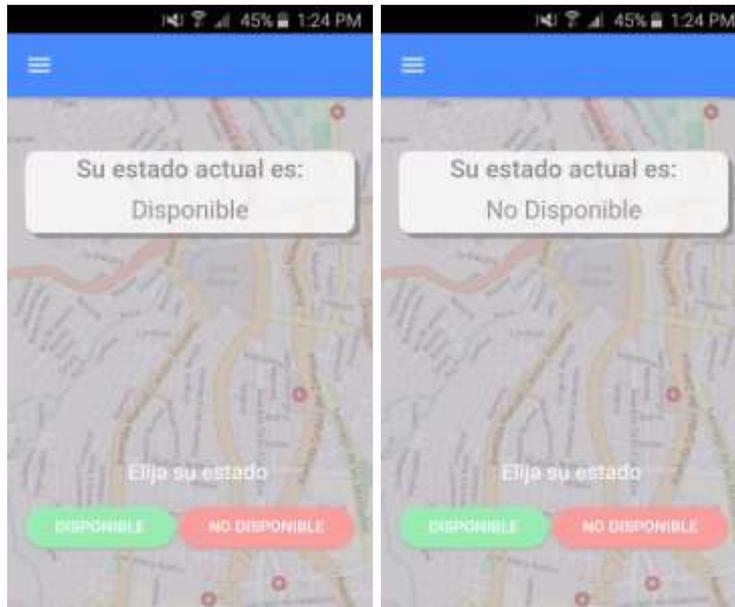


Figura 58: Caso de prueba cambio de estado

3.2.3.10 Notificación de nueva consulta.

En la siguiente figura se muestra la ejecución de la prueba.



Figura 59: Caso de prueba nueva consulta

3.2.3.11 Resultados de las pruebas

En la siguiente tabla se detallan los resultados de las pruebas ejecutadas según los casos de pruebas.

TABLA XXX: RESULTADOS CASOS DE PRUEBAS

Caso de prueba	Requerimiento	Criterio de aceptación	Resultado
CP001	RF001	CR001	Exitoso
CP002	RF002	CR002	Exitoso
CP003	RF003	CR003	Exitoso
CP004	RF004	CR004	Exitoso
CP005	RF005	CR005	Exitoso
CP006	RF006	CR006	Exitoso
CP007	RF007	CR007	Exitoso
CP008	RF008	CR008	Exitoso
CP009	RF009	CR009	Exitoso
CP010	RF010	CR010	Exitoso

Luego de haber ejecutado todos los casos de pruebas se concluye que las pruebas se terminan con un éxito total en todos los ítems dando así como resultado en una versión estable de la arquitectura informática.

3.2.4 Pruebas de aceptación.

Las pruebas de aceptación tienen como objetivo mostrar el cumplimiento de los requisitos garantizando la calidad del producto validando que cumpla las especificaciones para las que fue diseñado.

Los resultados de las pruebas de aceptación arrojan el total de éxito en los casos de pruebas planteado verificando los datos descritos en la tabla 21. Para detalles ver Anexo 2.

3.2.5 Pruebas de Usabilidad

Una forma de definir si una aplicación de software es usable, es evaluando si cumple con características de usabilidad. La usabilidad se define como el alcance de un producto o servicio usado por usuarios específicos para lograr un objetivo con efectividad, eficiencia y satisfacción en un contexto específico[35]. Las pruebas de usabilidad comprenden tres técnicas que permiten medir la efectividad eficiencia y satisfacción.

Pruebas con usuarios: permite que ciertos tipos de usuarios realicen tareas específicas para que a partir de sus ejecuciones evalúen la usabilidad y se identifiquen errores.

Observación: Percibe la actividad de los usuarios mediante la grabación de sus ejecuciones u observación dedicada.

Preguntar a los usuarios: mediante el uso de cuestionarios o entrevistas sobre la satisfacción y opinión en el uso de la herramienta.

Basados en lo antes mencionado se procede a evaluar la usabilidad de la arquitectura considerando los parámetros eficiencia y eficacia.

Para ello como primer paso es establecer la muestra a ser evaluada.

Para realizar el cálculo del tamaño de la muestra que servirá como referencia para obtener el número de personas que deberán ser encuestadas para poder comprobar el problema central que se planteó en el apartado de la problemática, se hará uso de la fórmula que se muestra en la siguiente formula.

$$n = (Z^2 pqN) / (Ne^2 + Z^2 pq)$$

Donde:

n: muestra: es el número representativo del grupo de personas que queremos estudiar (población) y, por tanto, el número de encuestas que debemos realizar, o el número de personas que debemos encuestar.

N: población: es el grupo de personas que vamos a estudiar, en este caso la población total es de 171 748 según el censo del 2010 de población y vivienda que comprende los individuos de una e edad entre 14 y 39 años de edad que se consideran las más familiarizadas con las tecnologías [36].

Z: nivel de confianza: mide la confiabilidad de los resultados. Lo usual es utilizar un nivel de confianza de 95 % (1.96) o de 90 % (1.65). En el presente caso se utilizara 1.28 que equivale a un 80% de confianza.

e: grado de error: mide el porcentaje de error que puede haber en los resultados. Lo usual es utilizar un grado de error de 5 % o de 10 %. Mientras menor margen de error, mayor validez tendrán los resultados, pero, por otro lado, mayor será el número de la muestra, es decir, mayores encuestas tendremos que realizar.

p: probabilidad de ocurrencia: probabilidad de que ocurra el evento. Lo usual es utilizar una probabilidad de ocurrencia del 50 %.

q: probabilidad de no ocurrencia: probabilidad de que no ocurra el evento. Lo usual es utilizar una probabilidad de no ocurrencia del 50 %. La suma de "p" más "q" siempre debe dar 100 %.

Entonces aplicando la formula tenemos:

$$n = (1.28^2 * 0.5 * 0.5 * 171748) / (171748 * 0.08^2 + 1.28^2 * 0.5 * 0.5)$$

Entonces n= 64 que es el número de encuestados a realizar.

Luego de definir la muestra se procedió a diseñar la encuesta para ello se utilizó la herramienta en línea gmail que nos permite realizar encuestas totalmente gratis.

Los casos de pruebas a utilizar son los especificados en las pruebas de funcionalidad ya que para probar características de usabilidad no se requieren de casos de pruebas específicos.

Para ello se definieron las siguientes preguntas.

- 1) ¿La aplicación permite ubicar medicamentos en las diferentes farmacias?
- 2) ¿La aplicación permite ubicar medicamentos en farmacias según la ubicación más cercana?
- 3) ¿La aplicación permite ubicar medicamentos en farmacias según el precio más bajo?
- 4) ¿Qué tiempo requiere para ubicar un medicamento en una farmacia sin el uso de la aplicación?
- 5) ¿Qué tiempo requirió para ubicar un medicamento mediante la aplicación?
- 6) ¿Considera usted que la aplicación le permite el ahorro de tiempo al momento de ubicar un medicamento?

La encuesta se encuentra disponible en la siguiente dirección:

<https://forms.gle/xxKxYu8uCJfhsUZRA>

Luego de aplicar la encuesta a 65 personas se obtuvieron los siguientes resultados.

- 1) ¿La aplicación permite ubicar medicamentos en las diferentes farmacias?

66 respuestas



Figura 60: Ubicación de medicamentos

2) ¿La aplicación permite ubicar medicamentos en farmacias según la ubicación más cercana?

66 respuestas

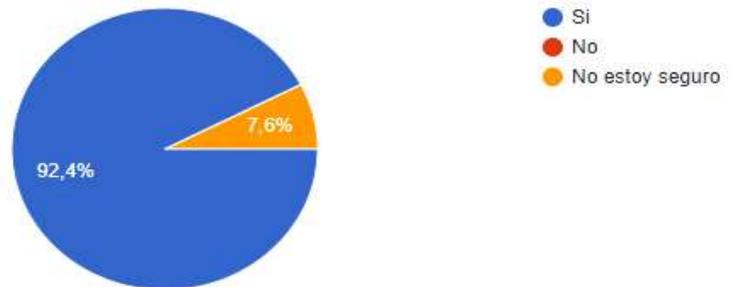


Figura 61: Ubicación de medicamentos según distancia

3) ¿La aplicación permite ubicar medicamentos en farmacias según el precio más bajo?

66 respuestas

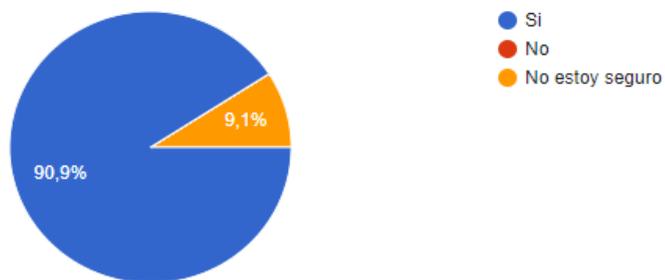


Figura 62: Ubicación de medicamentos según precio

4) ¿Qué tiempo requiere para ubicar un medicamento en una farmacia sin el uso de la aplicación?

66 respuestas

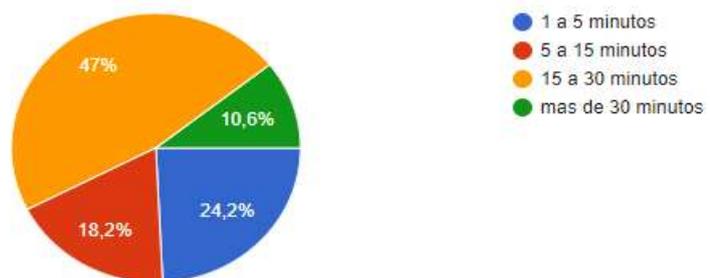


Figura 63: Ubicación de medicamentos sin usar aplicación

5) ¿Qué tiempo requirió para ubicar un medicamento mediante la aplicación?

66 respuestas

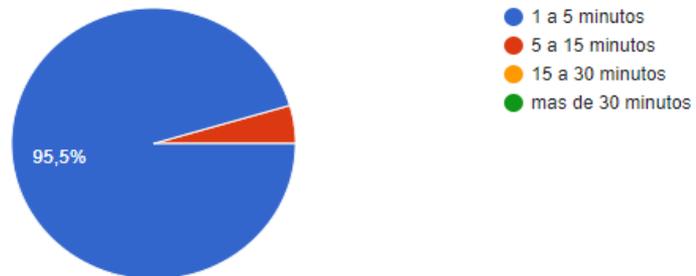


Figura 64: Tiempo de ubicación de un medicamento mediante la aplicación

6) ¿Considera usted que la aplicación le permite el ahorro de tiempo al momento de ubicar un medicamento?

66 respuestas

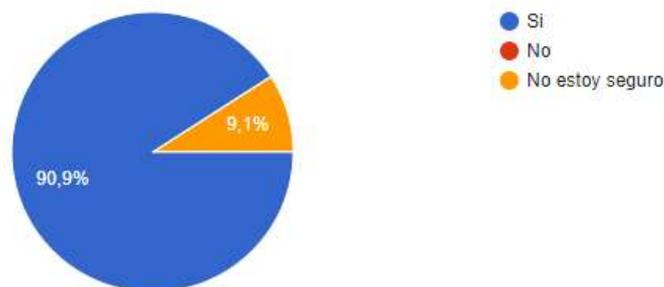


Figura 65: Ahorro de tiempo al momento de ubicar los medicamentos mediante la aplicación

Análisis de resultados.- Como se mencionó anteriormente el objetivo de estas pruebas es evaluar a la aplicación seo en términos de eficiencia y eficacia y así poder analizar si la arquitectura informática optimiza o no la ubicación de medicamentos en las diferentes farmacias.

Las evaluaciones se aplicaron a 66 personas diferentes que cumplieron a función de un cliente de farmacia que busca un medicamento. La evaluación se realizó en dos grupos un grupo de 33 personas que cumplen el rol de farmacias y el grupo restante cumple el rol de cliente de la misma manera en una segunda instancia se cambiaron los roles para que toda la muestra sea parte de la evaluación completa a la arquitectura.

Luego de aplicar la técnica de recolección de información encuesta se llega a las siguientes deducciones:

El 92.4% de los encuestados consideran que la aplicación si permite ubicar los medicamentos en las diferentes farmacias según la ubicación más cercana y solo el 7.6% no está seguro.

El 90.9% aseveran que la aplicación si permite la ubicación de los medicamentos en las diferentes farmacias según el precio más bajo y solo 9.1% no está seguro.

El 75.8% de los encuestados aseguran que sin el uso de la aplicación se demoran más de 5 min en ubicar un medicamento en las diferentes farmacias.

El 95.5% de los encuestados aseveran que usando la aplicación les toma entre 1 y 5 minutos ubicar los medicamentos en las diferentes farmacias.

Finalmente el 90.9% consideran que la aplicación seo si permite el ahorro de tiempo al momento de ubicar un medicamento en las diferentes farmacias y el 9.1% no está seguro.

G. Discusión

El objetivo de este apartado es evidenciar un análisis acerca de los resultados obtenidos con el desarrollo del presente TT.

1. Desarrollo de la propuesta alternativa

El desarrollo del presente trabajo de titulación consta de 3 fases las cuales corresponden a un objetivo específico las cuales componen un solo objetivo general.

1.1 Definir una arquitectura informática que permita la geo localización de medicamentos en farmacias.

Para el cumplimiento de este objetivo se realizó varios análisis y cuadros comparativos para elegir los diferentes elementos idóneos y herramientas que conformaran la arquitectura informática, como es el caso enfoques de desarrollo, frameworks, servidores web, mapas, metodologías. (Ver sección de resultados subsección 1)

Luego basados en los resultados de estos análisis se procedió a elegir los mejores elementos que definen una arquitectura idónea que permitió el cumplimiento del objetivo principal del presente TT.

Para una mejor definición se apoyó en el modelo 4+1 de Kruchten que indican como está estructurado la arquitectura (Ver anexo 3). Como resultado queda definida una arquitectura compuesta por dos aplicaciones móviles desarrolladas en ionic que interactúan en tiempo real a través de un servidor web desarrollado con la tecnología GraphQL.

1.2 Desarrollar una arquitectura informática que permita la geo localización de medicamentos en farmacias.

En el desarrollo del presente objetivo se basó en la metodología mobile-D definida en el objetivo anterior que consta de cuatro fases:

1.2.1 En la primera fase denominada **exploración** que fue la encargada de la planificación y deducción de requisitos del proyecto, donde pudimos tener la visión completa del alcance del proyecto y también todas las funcionalidades del producto (Ver sección resultados subsección 2, fase exploración).

1.2.2 En la segunda fase denominada **inicialización** donde se configuro todo el ambiente para el desarrollo asegurando así un éxito en las etapas posteriores, se verifico que las herramientas necesarias estén instaladas o se precedió a

instalar aquellas que faltaban (Ver sección resultados subsección 2, fase inicialización).

1.2.3 En la tercera fase **desarrollo** donde se codificó todos los elementos definidos en la fase de exploración, como se define en la metodología esta fase se divide en iteraciones (Ver sección resultados subsección 2, fase desarrollo).

Para la primera iteración se desarrolló todas las pantallas y funcionalidades de la primera parte de la arquitectura como es la aplicación para farmacias.

En la segunda iteración se desarrolló todas las funcionalidades y módulos de la segunda parte de la arquitectura que comprende la aplicación para usuarios de farmacias.

En la tercera iteración se desarrolló las funciones requeridas en el web Service que es la tercera pieza en la arquitectura.

1.2.4 En la fase de **estabilización** se desarrolló en una sola iteración donde se integró todas las partes de la arquitectura (Ver sección resultados subsección 2, fase estabilización).

1.2.5 La cuarta fase de **pruebas** donde se evaluó la funcionalidad de todo lo codificado basados en el estándar **ieee 829** (Ver sección de resultados subsección 3).

1.3 Evaluar que la arquitectura permita la geo localización de los medicamentos en las farmacias.

Con la ayuda del estándar ieee-829 se procedió con la cuarta fase de la metodología y tercer objetivo.

El primero paso fue establecer un plan de pruebas donde se define todo lo que se va a evaluar con la ayuda de los requerimientos se estableció evaluar en tres aspectos como es el de soporte, interfaz y funcionalidad.

Se define que el ambiente de pruebas es un escenario simulado que servirá como primer escenario para la evaluación.

Luego de haber definido el plan de pruebas se procede a evaluar la arquitectura según lo propuesto en este plan dando como resultado en un total de éxito en los ítems evaluados sin percances mayores dando así como resultado una versión estable del producto (Ver sección 3 subsección 1).

2. Valoración técnica económica ambiental

En base a los resultados del presente TT se concluye que contribuye positivamente en los aspectos económicos, técnicos y ambientales.

En el aspecto económico la puesta a producción de la arquitectura informática desarrollada ayudaría a mejorar las ventas a los locales farmacéuticos además de un ahorro de tiempo a los usuarios al momento de buscar medicinas lo que implica un ahorro de dinero además de aportar al capital tecnológico local.

En el aspecto técnico el presente trabajo de titulación puede servir como fuente de consulta o guía para aquellos desarrolladores que empiecen a explorar el mundo de la programación especialmente para aquellos relacionados con la programación móvil y programación de servicios web.

Finalmente la puesta a producción de la arquitectura evita que los usuarios tengan que transitar hacia las farmacias utilizando muchas veces medios de transportes que contaminan el medio ambiente como los autos, autobuses, etc. Además de que su ejecución no utiliza papel ni realiza mecanismos o procesos industriales que afecten el medio ambiente.

El presente trabajo de titulación se realizó con la ayuda de recursos humanos, materiales técnicos y tecnológicos mencionados a continuación.

La siguiente tabla menciona los recursos humanos empleados en el desarrollo del TT, donde consta el autor del presente TT y el director del mismo quien con su predisposición fue un gran aporte a la culminación exitosa el proyecto.

TABLA XXXI: RECURSOS HUMANOS

Recurso	Rol	Horas	Valor Unitario	Valor Total
Leonardo Caraguay	Analista Diseñador Programador Evaluador	400	\$10	\$4000
Total:				\$ 4000

En la siguiente tabla se presentan los recursos materiales utilizados en el presente TT.

TABLA XXXII: RECURSOS MATERIALES

Recursos	Cantidad	Valor Unitario	Valor Total
Materiales de Oficina (lápiz, borradores, perfiles, etc.)	-	-	\$40.00
Resmas de papel A4	2	\$4	\$8.00
Impresiones	4	\$15.00	\$60.00
Anillado	3	\$2.00	\$6,00
Empastado	3	\$8.00	\$24.00
Total:			\$138.00

En la siguiente tabla se detalla los recursos técnicos y tecnológicos empleados para el cumplimiento de los objetivos planteados.

TABLA XXXIII: RECURSOS TÉCNICOS Y TECNOLÓGICOS

Recursos	Cantidad	Valor Unitario	Valor total
Recursos de Hardware			
Portátil	1	\$1000	\$1000
Flash USB	1	\$10	\$10
Celular con Android	1	\$200	\$200
Recursos de Software			
Visual Studio Code	1	\$0	\$0
Sistema Operativo Debian	1	\$0	\$0
Ionic	1	\$0	\$0
Angular	1	\$0	\$0
Recursos de Telecomunicaciones			
Internet	400 horas	\$0.50	\$200
Subtotal:			\$ 1410.00

Finalmente en la siguiente tabla se muestra un resumen de todos los costes incurridos anteriormente y su monto total.

TABLA XXXIV: TOTAL DE RECURSOS USADOS

Descripción	Total \$
Recursos Humanos	4000.00
Recursos Materiales	138.00
Recursos Técnicos	1410.00
Total:	5548.00

H. Conclusiones

Al término del presente trabajo de titulación se ha llegado a las siguientes conclusiones:

- El modelo 4+1 es un modelo bastante útil para documentar arquitectura de software, ya que todas las personas interesadas en el proyecto, desde el usuario final hasta los desarrolladores pueden entender el software que se esté desarrollando desde sus diferentes perspectivas gracias a las diferentes vistas que este modelo presenta.
- El desarrollo del Web Service con la tecnología GraphQL ayudó el manejo de la interactividad en tiempo real entre el servidor y cliente evitando usar más tecnologías para esta funcionalidad como se debe hacer con los Api Rest tradicionales.
- La metodología mobile-D fue la mejor opción para el proyecto, ya que, se adaptó a sus características como el número pequeño de integrantes y su necesidad de obtener resultados en poco tiempo.
- Los servicios ofrecidos por MapBox ayudó el cumplimiento de algunos requerimientos para el presente proyecto tales como: marcar ruta, obtener ubicación, obtener distancias entre dos puntos y la creación de mapas interactivos.
- Las pruebas de usabilidad permitieron evaluar la arquitectura en términos de eficiencia y eficacia.
- El desarrollo de la arquitectura informática optimiza la búsqueda y geo localización de medicamentos en las diferentes farmacias permitiendo al usuario un ahorro de tiempo y dinero. Como se muestra en las pruebas de usabilidad.
- El desarrollo de la arquitectura informática ayuda a los locales a potenciar su mercado con el uso de las nuevas tendencias tecnológicas.

I. Recomendaciones

Una vez terminado el presente TT se emiten las siguientes recomendaciones.

- Realizar un correcto análisis de la tecnología y herramientas para tener una visión clara del proyecto y evitar percances que retrasen el proyecto.
- Elegir de preferencia herramientas con las cuales se está familiarizado previamente o aquellas que están en auge ya que contarán con una buena documentación e información que facilitara la comprensión del mismo.
- Establecer objetivos reales con respecto a las funcionalidades que va a realizar el software esto ayudara retrasos o fracaso en el proyecto.
- Realizar el mayor número de pruebas asegura la calidad de un producto de software.
- Contar con una base de conocimiento previo sobre las tecnologías a utilizar agiliza la fase desarrollo de la arquitectura.
- Realizar investigación sobre las nuevas tecnologías de desarrollo de tal manera que se amplíe el área de conocimiento, además esto ayuda a realizar proyectos mayormente interesantes.
- Para trabajos futuros se puede aplicar este tipo de tecnología en otros ámbitos como el de repuestos automotrices, parqueaderos, cajeros automáticos y todo aquel ámbito donde se necesite conocer su ubicación. También se puede implementar la opción para ordenar resultados en base a los precios, cercanías de los locales etc

J. Referencias

- [1] T. Keith, "Industria farmacéutica," *Encicl. salud y Segur. en el Trab.*, p. 79.2/10, 1998.
- [2] E. Ortiz, C. Galarza, F. Cornejo, and J. Ponce, "Acceso a medicamentos y situación del mercado farmacéutico en Ecuador," *Rev. Panam. salud pública = Pan Am. J. public Heal.*, vol. 36, no. 1, pp. 57–62, 2014.
- [3] V. Crespo, "Arquitectura," *Actualmente, los Sist. informáticos están Present. en casi todos los ámbitos la vida. Sin embargo, es muy difícil garantizar un Determ. Niv. Segur. e Inocuidad. Políticas débiles o mal implementadas pueden conducir a enormes pérdidas*, 2007.
- [4] Sommerville Ian, *Ingeniería del Software*, 6th ed. 2005.
- [5] E. de Clasificaciones, "Definición de geolocalización," 2017. [Online]. Available: <https://www.tiposde.org/tecnologia/1014-definicion-de-geolocalizacion/>. [Accessed: 11-Dec-2018].
- [6] P. Educativo, "No Title," 2017. [Online]. Available: <https://www.portaleducativo.net/quinto-basico/537/Mapas-y-planos>. [Accessed: 12-Nov-2018].
- [7] J. E. Pérez, *Introducción a AJAX*. 2008.
- [8] E. Comercio, "No Title," 2018. [Online]. Available: <https://elcomercio.pe/tecnologia/actualidad/google-maps-siete-funciones-usuario-debe-conocer-noticia-493444>. [Accessed: 13-Nov-2018].
- [9] G. Cloud, "No Title," 2018. [Online]. Available: <https://cloud.google.com/maps-platform/?hl=es-419>. [Accessed: 14-Nov-2018].
- [10] D. Laia, C. R. Joan, C. C. Jordi, and P. N. Antoni, "Rutas turísticas personalizadas en dispositivos móviles sin necesidad de conexión a Internet: Itiner@," 2012.
- [11] F. OSM, "Openstreetmap," 2018. [Online]. Available: openstreetmap.org. [Accessed: 24-Dec-2018].
- [12] MapBox, "About," 2018. [Online]. Available: <https://www.mapbox.com/about/>. [Accessed: 15-Nov-2018].
- [13] W. Zaman and P. N. Basu, "A Framework of Web Service Based Architecture for M-learning of a University Consortium Education System."

- [14] L. Delía, N. Galdamez, P. Thomas, L. Corbalan, and P. Pesado, "Análisis Experimental de desarrollo de Aplicaciones Móviles Multiplataforma," *XX Congr. Argentino Ciencias la Comput. (Buenos Aires, 2014)*, 2014.
- [15] C. X. R. MUÑOZ, "DESARROLLO DE UNA APLICACIÓN HIBRIDA-MÓVIL PARA DETERMINAR LOS TIPOS DE ESPECIES DE VEGETACIÓN INDICADORAS SEGÚN LOS PISOS ZOOGEOGRÁFICOS DEL ECUADOR, MEDIANTE EL USO DE LAS HERRAMIENTAS PHONEGAP Y JQUERY MOBILE PARA LA PLATAFORMA ANDROID," UNIVERSIDAD TÉCNICA DEL NORTE, 2015.
- [16] S. S. Salgado Escobar, "Estudio de la eficiencia de los frameworks híbridos y nativos en el desarrollo de aplicaciones móviles, basados en bechmark para el consorcio informega," Universidad de las Fuerzas Armadas, 2018.
- [17] Ionic, "About," 2018. [Online]. Available: <https://ionicframework.com/>. [Accessed: 16-Nov-2018].
- [18] M. A. Alvarez, "Que es MVC," 2014. [Online]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html>. [Accessed: 18-Nov-2018].
- [19] EcuRed, "Ionic," 2017. [Online]. Available: <https://www.ecured.cu/Ionic>. [Accessed: 20-Nov-2018].
- [20] F. Inc, "About," 2018. [Online]. Available: <https://reactjs.org/>. [Accessed: 22-Nov-2018].
- [21] D. Garcia, "Características de React," 2016. [Online]. Available: <https://desarrolloweb.com/articulos/caracteristicas-react.html>. [Accessed: 23-Nov-2018].
- [22] Vicente Pelechano, "Servicios Web. Estandares, extensiones y perspectivas de futuro," no. December, 2005.
- [23] J. C. Sánchez Cuenca, "Aplicación móvil para georreferenciación y búsqueda de farmacias utilizando tecnología multiplataforma," 2017.
- [24] F. Nogatz and D. Seipel, "Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts," *Electron. Proc. Theor. Comput. Sci.*, vol. 234, pp. 42–56, 2017.
- [25] Á. G. Román, "Escuela Politécnica Trabajo Fin de Grado Integrando fuentes heterogéneas de datos en Web con GraphQL Escuela Politécnica Grado en Ingeniería Informática en Ingeniería del Software Trabajo Fin de Grado

- Integrando fuentes heterogéneas de datos en Web con Gr,” Universidad de Extremadura, 2018.
- [26] O. Hartig and J. Pérez, “An initial analysis of facebook’s GraphQL language,” *CEUR Workshop Proc.*, vol. 1912, 2017.
- [27] S. S. Pore and S. B. Pawar, “Comparative Study of SQL & NoSQL Databases,” *Ijarcet*, vol. 4, no. 5, pp. 1747–1753, 2015.
- [28] M. C. G. Mantilla, L. L. C. Ariza, and B. M. Delgado, “Metodología para el desarrollo de aplicaciones móviles,” *Tecnura*, vol. 18, pp. 20–35, 2014.
- [29] Y. D. Amaya Balaguera, “Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual,” *J. Technol.*, pp. 111–124, 2013.
- [30] J. T. Herrera, “Aplicación móvil para consultas académicas en la Universidad Nacional de Loja,” p. 156, 2014.
- [31] J. S. C. BUITRÓN, “Pantalla de Inicio (launcher) Android, para Personas No Videntes,” UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL, 2015.
- [32] C. Ramon Vásquez, “La Universidad Católica de Loja,” 2017.
- [33] B. Kitchenham, *Procedures for Performing Systematic Reviews*. {Department of Computer Science, Keele University, UK, 2004.
- [34] K. Philippe and R. S. Corp., “Architectural Blueprints—The ‘4+1’ View Model of Software Architecture,” vol. 12, pp. 42–50, 1995.
- [35] S. M. G, “Proceso de Pruebas de Usabilidad de Software Software Usability Testing Process Introducción,” pp. 27–32, 2018.
- [36] Instituto nacional de estadística y censos, “FASCÍCULO PROVINCIAL LOJA,” 2010. .

K. Anexos

1. Anexo 1: Especificación de requisitos de software ERS ESPECIFICACIÓN DE REQUERIMIENTOS SOFTWARE

Proyecto: Arquitectura informática para geo
localización de medicamentos en farmacias

Historia del Documento

Fecha	versión	Comentarios	Autor
10 Octubre 2018	0.1	Versión inicial	Leonardo
....	..	Revisada...	si

1.1 INTRODUCCIÓN

La especificación de los requerimientos de software (RS) provee una imagen clara del producto software. Contiene los requerimientos del cliente del punto de vista de entradas y salidas. Contiene diagramas, ecuaciones, figuras, y cualquier otro elemento que ayuda a clarificar los requerimientos del cliente. Este documento no especifica los procesos u otra información que no es perceptible por el cliente, es decir el sistema es considerado como una caja negra.

Esta especificación ha sido elaborada tomando en cuenta las directrices establecidas por el estándar "IEEE (Recommended Practice Software Requirements Specification) ANSI/IEEE 830-1998".

1.2 Propósito

Se redacta este documento con el objetivo de tener una visión mejorada sobre el proyecto para los involucrados.

Contiene las especificaciones funcionales no funcionales restricciones del sistema que se desea construir. El mismo que está integrada por dos aplicaciones móviles interconectadas por un servicio web en la nube junto a otros componentes y herramientas.

1.3 Alcance

La arquitectura informática para búsqueda de medicamentos en farmacias tiene con objetivo optimizar la ubicación de los diferentes medicamentos en las farmacias. La información sobre la existencia de los medicamentos se maneja en tiempo real, y la información sobre las farmacias se actualiza mensualmente para evitar contratiempos con los usuarios.

Algunas de sus principales funcionalidades son:

- Para ubicar medicinas en farmacias
- Permite la búsqueda de farmacias por nombre y en un mapa.
- Muestra la disponibilidad de las farmacias en tiempo real.
- Muestra la información principal de farmacia como nombre, dirección, teléfono etc.
- Marca la ruta hacia la farmacia con respecto a mi ubicación.

1.4 Personal involucrado

Nombre	Leonardo Caraguay
Rol	Analista y Desarrollador
Categoría profesional	Estudiante de Ingeniería en Sistemas
Responsabilidades	Análisis, Diseño, Desarrollo, Pruebas
Información	lfcaraguayc@unl.edu.ec
Aprobación	Aprobado

Nombre	José Guamán
Rol	Analista y Desarrollador
Categoría profesional	Superior
Responsabilidades	Supervisor del proyecto
Información	jose.o.guaman@unl.edu.ec

Aprobación	Aprobado
-------------------	-----------------

1.5 Definiciones, Acrónimos y Abreviaturas.

Término	Significado
Aplicación móvil.	Aplicación informática que puede ejecutarse en teléfonos inteligentes y otros dispositivos compatibles
Interfaz	Permite la interacción del usuario con la aplicación
ERS	Especificación de requisitos de Software
RF	Requisito funcional
RNF	Requisito no funcional
Localización	Determinación del lugar donde se encuentra una persona u objeto.
WS	Web Service
Usuario	Persona que hace uso de un software

1.6 Visión General del Documento.

Este Documento tiene como objetivo conocer las especificaciones del software llevando a cabo un análisis de la información. En la primera sección del documento, ámbito del sistema, los acrónimos y referencias. La siguiente sección se realiza una descripción global del ERS, la cual incluye perspectivas, funciones características y restricciones del producto. Y la última sección presenta la matriz de requerimientos funcionales y no funcionales con la información detallada de cada requisito de usuario. Una vez realizada la recolección y análisis de la información de manera satisfactoria el documento proporcionara la manera de como iniciar el desarrollo de la arquitectura, garantizando que el resultado final sea el mejor.

1.7 Descripción General

1.7.1 Perspectiva del producto.

El proyecto está dividido en dos aplicaciones móviles y un servicio web

La primera que permite buscar y ubicar el medicamento o medicamentos además de visualizar si los diferentes locales están disponibles o no para atención referente a nuestra ubicación en el mapa.

La segunda que permite emitir una respuesta ante una o varias consultas realizadas por la aplicación de buscador además de establecer un estado de disponible o no disponible.

1.7.2. Funcionalidad del producto.

Funciones:

Registro

- Visualizar un mapa de los diferentes locales afiliados y su disponibilidad.
- Marcar ruta desde la ubicación del usuario y farmacia.
- Mostrará los datos de un local seleccionado.
- Búsqueda por nombre de local.
- Búsqueda de medicamento por nombre.

1.7.3 Características de los usuarios.

Tipo de Usuario	Usuario
Formación	Ninguna Formación específica
Habilidades	Conocimiento básico en el manejo de un dispositivo móvil.
Actividades	Utilización completa de la aplicación.

1.7.4 Restricciones

- Las aplicaciones se desarrollara bajo las tecnologías como HTML 5, JavaScript, Scss, MapBox.
- Las aplicaciones deberán tener un diseño e implementación sencilla.
- Las aplicaciones deben ser multiplataforma.
- Se desarrollaran bajo el framework Ionic.

1.7.5 Suposiciones y Dependencias.

Basados en la encuesta aplicada (Ver apéndice 2) se determina que los dispositivos donde funcionaran deberán contar con el Sistema Operativo Android. Recalcando que los dispositivos en los que se instale el software debe cumplir con las condiciones mínimas de funcionamiento.

1.8 Requisitos específicos

1.8.1 requisitos comunes de interfaces

Las interfaces de usuario deberán permitir visualizar toda la información especificada anteriormente.

1.8.2 interfaces de usuarios.

La interfaz gráfica de las aplicaciones será diseñada de manera que sea intuitiva y facilite su utilización.

1.8.3 interfaces de hardware

Las aplicaciones se ejecutaran en los diferentes dispositivos que cuenten con los SO. Antes mencionados y deberán cumplir con las siguientes especificaciones.

- Procesador mínimo de 600MHz.
- 1 GB mínimo de memoria interna
- GPS.
- Conexión a datos o internet.

1.8.4 interfaces de software

Se requiere que el dispositivo cuente con los Sistema Operativo Android.

1.9 Requisitos funcionales

1.9.1 Requisito funcional 1 (Farmacia, Cliente)

Referencia	REF-001
Nombre	Ingresar a la aplicación
Descripción	La aplicación móvil permitirá al Usuario ingresar a la aplicación mediante un usuario y contraseña.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que al ejecutar la aplicación móvil se muestre en la pantalla principal
Por verificar	Verificar si al presionar el botón [Ingresar] con uno o los dos campos vacíos la aplicación presenta un mensaje indicativo.
Por verificar	Verificar si cuando se ingresa la cédula o la clave incorrecta, la aplicación presenta un mensaje indicando la razón por la que no se puede ingresar al sistema.

Por verificar	Verificar si al ingresar los datos de la cuenta correctamente, la aplicación presenta una pantalla con los datos personales del usuario
---------------	---

1.9.2 Requisito funcional 2 (Farmacia, Cliente).

Referencia	REF-002
Nombre	Visualizar los datos personales de usuario
Descripción	La aplicación móvil permitirá al Usuario visualizar sus datos personales (cédula, nombres, email, teléfono, dirección)
Prioridad	Media
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que al ingresar a la aplicación se muestran los datos personales del Usuario (nombre, cédula, email, teléfono, celular, dirección).

1.9.3 Requisito funcional 3 (Cliente)

Referencia	REF-003
Nombre	Visualizar farmacias
Descripción	La aplicación móvil permitirá al Usuario visualizar las distintas farmacias en un mapa.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que al ingresar a la aplicación se muestra la ubicación de las farmacias en el mapa.

1.9.4 Requisito funcional 4 (Cliente)

Referencia	REF-004
Nombre	Buscar farmacia
Descripción	La aplicación móvil permitirá al usuario buscar una farmacia mediante el nombre.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación permite consultar la ubicación de una farmacia mediante el nombre

1.9.5 Requisito funcional 5 (Cliente)

Referencia	REF-005
Nombre	Buscar medicamento
Descripción	La aplicación móvil permitirá al usuario buscar un medicamento mediante el nombre.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación permite consultar la ubicación de un medicamento mediante el nombre

1.9.6. Requisito funcional 6 (Cliente)

Referencia	REF-006
Nombre	La aplicación mostrará la ruta hacia el local.
Descripción	La aplicación móvil permitirá visualizar la ruta hacia la farmacia seleccionada además de la distancia y tiempo.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación muestre la mejor y correcta ruta.

1.9.7. Requisito funcional 7 (Usuario)

Referencia	REF-007
Nombre	La aplicación mostrará los datos del local seleccionado.
Descripción	La aplicación móvil permitirá visualizar los datos de la farmacia seleccionada y toda la información del mismo con una opción de visualizar ruta.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación muestre la información correcta del local.

1.9.8. Requisito funcional 8 (Farmacia)

Referencia	REF-009
Nombre	Seleccionar estado
Descripción	La aplicación móvil permitirá seleccionar un estado de disponibilidad entre disponible o no disponible

Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación permite seleccionar el estado
Por verificar	Verificar que en la aplicación de buscador se muestre como no disponible.

1.9.9 Requisito funcional 9 (Farmacia)

Referencia	REF-010
Nombre	Configurar notificaciones
Descripción	La aplicación móvil permitirá seleccionar si desea o no recibir notificaciones de las búsquedas
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación permite seleccionar el estado

1.9.10 Requisito funcional 10 (Farmacia)

Referencia	REF-011
Nombre	Notificación de nuevas consultas
Descripción	La aplicación móvil permitirá recibir una notificación de las diferentes consultas que lleguen aun cuando la aplicación se encuentre cerrada pero en un estado de disponibilidad.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación

Por verificar	Verificar que la aplicación notifica los pedidos nuevos
---------------	---

1.9.11 Requisito funcional 11 (Farmacia, Cliente)

Referencia	REF-012
Nombre	Registro
Descripción	La aplicación móvil permitirá registrar las nuevas farmacias para mostrarse en el mapa y los nuevos usuarios.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que la aplicación guarda los nuevos registros

1.9.12 Requisito funcional 12 (Web Service)

Referencia	REF-013
Nombre	Creación de los EndPoints
Descripción	Creación de los métodos y funciones en el servicio web para gestionar las diferentes funcionalidades de las aplicaciones.
Prioridad	Alta
Versión	0.1
Estado	Criterios de aceptación
Por verificar	Verificar que las diferentes EndPoints devuelven los datos deseados.

1.10 Requisitos no funcionales.

- Rendimiento

Referencia	RNF-001
Nombre	Rendimiento
Descripción	- El tiempo de respuesta de la aplicación debe ser el mínimo.
Prioridad	Alta
Versión	0.1

- Soporte

Referencia	RNF-002
Nombre	Soporte
Descripción	- La aplicación móvil de consulta mostrará a los usuarios cuando no existe respuesta y cuando se encuentra en un estado de desconectado.
Prioridad	Alta
Versión	0.1

- Amigable

Referencia	RNF-003
Nombre	Amigable
Descripción	Utilizará una interfaz intuitiva y amigable para el usuario.
Prioridad	Alta
Versión	0.1

- Disponibilidad

Referencia	RNF-004
-------------------	---------

Nombre	Disponible 24/7
Descripción	Deberá estar disponible continuamente con un nivel de servicio de 7 días y por 24 horas.
Prioridad	Alta
Versión	0.1

1.11 Apéndices

Apéndice 1: Selección de la Muestra.

Para realizar el cálculo del tamaño de la muestra que servirá como referencia para obtener el número de personas que deberán ser encuestadas para poder comprobar el problema central que se planteó en el apartado de la problemática, se hará uso de la fórmula que se muestra en la figura (1)

$$n = (Z^2 pqN) / (Ne^2 + Z^2 pq) \quad (1)$$

Donde:

n: muestra: es el número representativo del grupo de personas que queremos estudiar (población) y, por tanto, el número de encuestas que debemos realizar, o el número de personas que debemos encuestar.

N: población: es el grupo de personas que vamos a estudiar, en este caso la población total es de 171 748 según el censo del 2010 de población y vivienda que comprende los individuos de una e edad entre 14 y 39 años de edad que se consideran las más familiarizadas con las tecnologías [1].

Z: nivel de confianza: mide la confiabilidad de los resultados. Lo usual es utilizar un nivel de confianza de 95 % (1.96) o de 90 % (1.65). En el presente caso se utilizara 1.28 que equivale a un 80% de confianza.

e: grado de error: mide el porcentaje de error que puede haber en los resultados. Lo usual es utilizar un grado de error de 5 % o de 10 %. Mientras menor margen de error, mayor validez tendrán los resultados, pero, por otro lado, mayor será el número de la muestra, es decir, mayores encuestas tendremos que realizar.

p: probabilidad de ocurrencia: probabilidad de que ocurra el evento. Lo usual es utilizar una probabilidad de ocurrencia del 50 %.

q: probabilidad de no ocurrencia: probabilidad de que no ocurra el evento. Lo usual es utilizar una probabilidad de no ocurrencia del 50 %. La suma de “p” más “q” siempre debe dar 100 %.

Entonces aplicando la formula tenemos:

$$n = (1.28^2 * 0.5 * 0.5 * 171748)/(171748 * 0.1^2 + 1.28^2 * 0.5 * 0.5)$$

Entonces n= 40 que es el número de encuestados a realizar.

Apéndice 2: Referencias

[1] INEC. (2010, Nov 28). Censo2010 [En Linea]. Disponible: <http://www.ecuadorencifras.gob.ec/wp-content/descargas/Manu-lateral/Resultados-provinciales/loja.pdf>

Apéndice 3: Encuesta Dirigida a la muestra seleccionada.

Universidad Nacional de Loja

Área de la Energía, las Industrias y los Recursos naturales No Renovables CARRERA DE INGENIERIA EN SISTEMAS Encuesta

El objetivo de la presente encuesta es para determinar las diferentes necesidades y problemas de la aplicación a desarrollarse, para facilitar la ubicación de las farmacias en el país.

- 1. De las siguientes tecnologías identifique cuales usted usa con frecuencia**
 - a) Celular**
 - b) Computadora**
 - c) Internet**
 - d) Ninguna**
 - e) Otros**
- 2. Si usted cuenta con un teléfono celular inteligente que sistema operativo usa el mismo**
 - a) Android**
 - b) los**

- c) Desconozco
 - d) Otro.
3. Con que frecuencia visita usted una farmacia para conseguir de un medicamento
- a) Diario
 - b) Semanal
 - c) Mensual
 - d) Anual
4. De las siguientes opciones de qué forma usted se transporta a una farmacia
- a) En automóvil propio
 - b) En autobús
 - c) En taxi
 - d) Caminando
 - e) Otros
5. De las siguientes opciones como conoce usted si una farmacia se encuentra atendiendo
- a) Internet
 - b) Aplicación móvil
 - c) Voy a averiguar en la farmacia
 - d) Otro
6. De las siguientes opciones como conoce usted si una farmacia tiene un medicamento
- a) En Internet
 - b) Aplicación móvil
 - c) Voy a la farmacia a preguntar
 - d) Otro
7. Para conseguir todos los medicamentos de interés usted tiene que visitar:
- a) Una sola farmacia
 - b) Varias farmacias
 - c) Demasiadas farmacias
8. Cuando ha visitado una farmacia para conseguir un medicamento le ha sucedido que ésta no cuenta con el mismo
- a) No me ha sucedido

- b) Me ha sucedido algunas de veces
 - c) Me ha sucedido muchas veces
9. ¿Considera que una aplicación móvil o web que le informe sobre la existencia de medicamentos en dichas farmacias le ayudaría ahorrar tiempo?
- a) Si
 - b) No

Apéndice 4: Resultados de la Encuesta

4. De las siguientes tecnologías identifique cuales usted usa con frecuencia

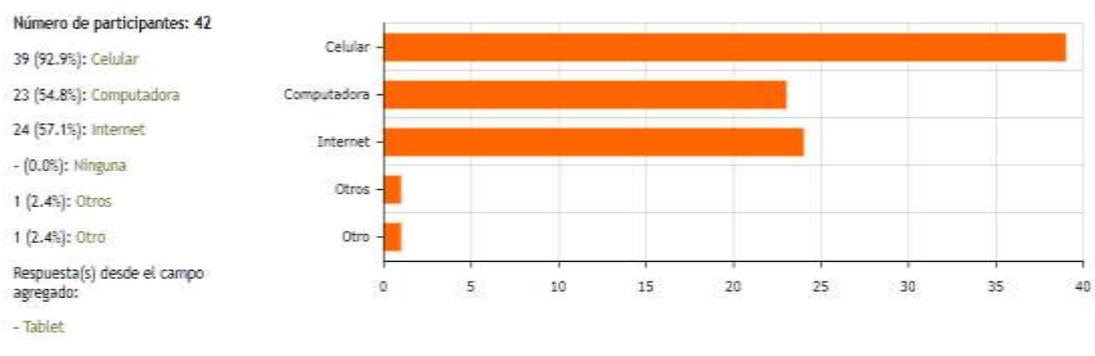


Figura 1: Población que afirma tener celular como su principal aparato tecnológico

5. Si usted cuenta con un celular inteligente que sistema operativo cuenta este

Número de participantes 42

- 40 (97%): Android
- 1 (1.5%): ios
- 1 (1.5%): Desconozco
- 0 (0.0%): Otros

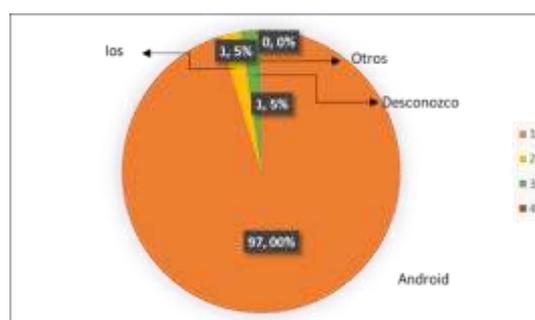


Figura 2: Población que afirma tener como principal sistema operativo en su celular Android.

6. Con que frecuencia visita usted una farmacia para conseguir de un medicamento

Número de participantes: 42

1 (2.4%): Diario
5 (11.9%): Semanal
27 (64.3%): Mensual
9 (21.4%): Anual

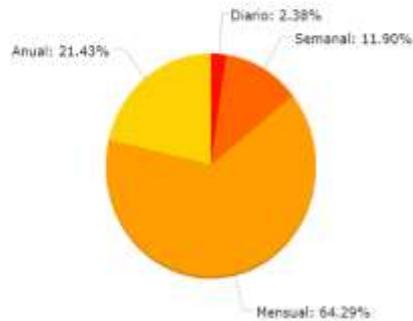


Figura 3: Población que afirma visitar una farmacia mensualmente

7. De las siguientes opciones de qué forma usted se transporta a una farmacia

Número de participantes: 42

13 (31.0%): En automóvil propio
21 (50.0%): En autobús
4 (9.5%): En taxi
30 (71.4%): Caminando
- (0.0%): Otros

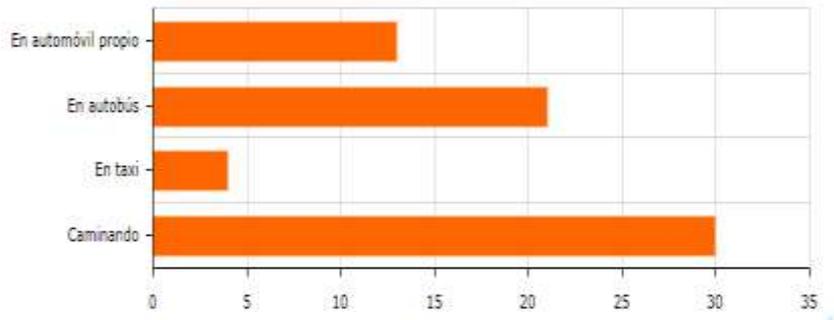


Figura 4: Población que afirma transportarse a una farmacia caminando

8. De las siguientes opciones como conoce usted si una farmacia se encuentra atendiendo

Número de participantes: 42

5 (11.9%): Internet
- (0.0%): Aplicación móvil
35 (83.3%): Voy a averiguar en la farmacia
1 (2.4%): Otro
1 (2.4%): Otro

Respuesta(s) desde el campo agregado:

- Farmacia de turno por familiares

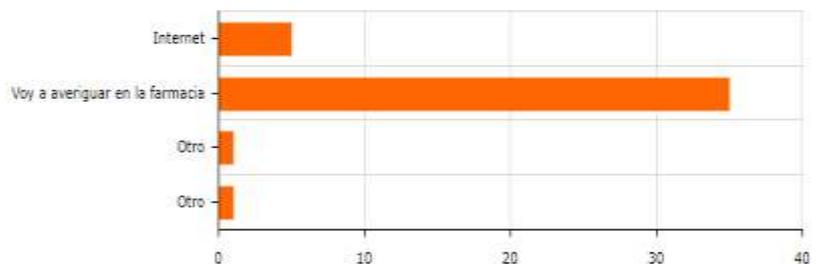


Figura 5: Población que asevera que va a una farmacia para conocer si esta se encuentra atendiendo

9. De las siguientes opciones como conoce usted si una farmacia tiene un medicamento

Número de participantes: 42

1 (2.4%): En Internet

1 (2.4%): Aplicación móvil

40 (95.2%): Voy a la farmacia a preguntar

-(0.0%): Otro

-(0.0%): Otro

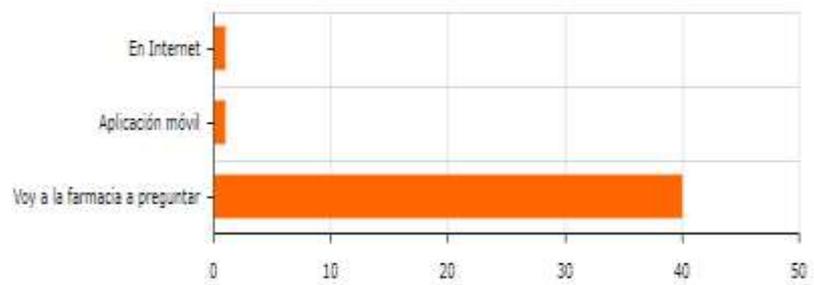


Figura 6: Población que asevera que va a la farmacia para conocer de la existencia de un medicamento

10. Para conseguir todos los medicamentos de interés usted tiene que visitar:

Número de participantes: 42

8 (19.0%): Una sola farmacia

32 (76.2%): Varias farmacias

2 (4.8%): Demasiadas farmacias

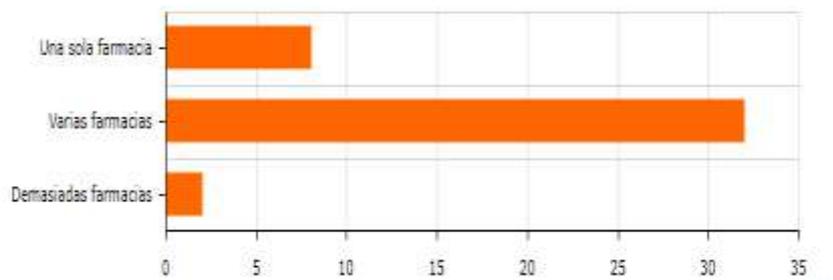


Figura 7: Población que afirma que visita varias farmacias para satisfacer su receta.

11. Cuando ha visitado una farmacia para conseguir un medicamento le ha sucedido que ésta no cuenta con el mismo.

Número de participantes: 42

4 (9.5%): No me ha sucedido

31 (73.8%): Me ha sucedido algunas de veces

7 (16.7%): Me ha sucedido muchas veces

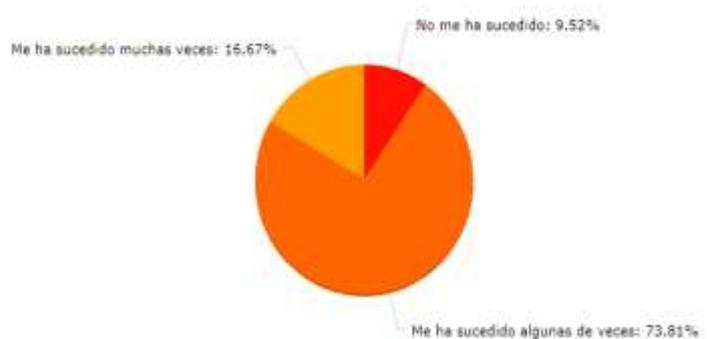


Figura 8: Población que afirma que cuando visita una farmacia esta no cuenta con el medicamento deseado

12. ¿Considera que una aplicación móvil o web que le informe sobre la existencia de medicamentos en dichas farmacias le ayudaría ahorrar tiempo?

Número de participantes: 42

41 (97,6%): sí

1 (2,4%): no

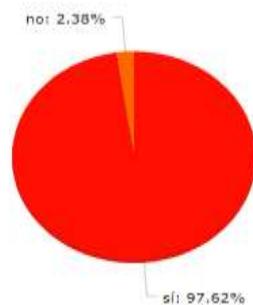


Figura 9: Población que afirma que la creación de una aplicación móvil para búsqueda de medicamentos es factible.

Apéndice 5: Entrevista

La entrevista utilizada para levantamiento de requerimientos fue la misma utilizada para el desarrollo del anteproyecto que fue realizado a cinco farmacias ubicadas en diferentes sectores la ciudad de Loja. Estas entrevistas se llevaron a cabo con el fin de recabar información acerca de estado actual de las farmacias en nuestra localidad con relación a la tecnología y su aporte en el mercado farmacéutico, luego de la recolección mediante esta técnica y el análisis de la información se resume en los siguientes puntos importantes:

- Los horarios no son constantes y pueden variar diariamente.
- Cuentan con las tecnologías como computadora celular e internet.
- El número de empleados varía entre uno y tres empleados en todas las farmacias mencionadas.
- En dos farmacias de las cinco pertenecen a pequeñas cadenas locales donde si cuentan con un servicio de venta en línea (portal web).
- El capital que manejan es muy pequeño con respecto a las grandes cadenas.
- Les es difícil competir con las grandes cadenas farmacéuticas porque su inventario es pequeño y necesitan vender a mayor precio para poder percibir ganancias. A diferencia de las grandes cadenas que venden a precio de costo y ganan en regalías por vender grandes cantidades.
- Sus principales usuarios son aquellas personas que buscan recetas pequeñas o medicinas unitarias.


UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE ENERGÍA LA INDUSTRIA Y LOS RECURSOS NO RENOVABLES
CARRERA DE INGENIERIA EN SISTEMAS


1858

ENTREVISTA REALIZADA POR: Caraguay Caraguay Leonardo Favio

Evidencia de entrevistas

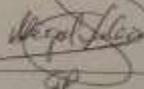
Nombre	Institución/Empresa	Firma
Boris Dúmas	Farmacia Norte	
Mario Quena	Farmacia San Carlos de Guano	
Mireya Medina	Farmacia María Paula	
Diego Latorre C.	Farmacia San Fernando	
Tania Castillo	Farmacia Pichincha	

Figura 10: Firmas de las entrevistas

2. Anexo 2: Pruebas de Aceptación.

Las presentes pruebas se realizan con la finalidad verificar las funcionalidades de la arquitectura informática creada en el presente TT, arquitectura que busca optimizar la búsqueda y localización de medicamentos en las diferentes farmacias.

Las pruebas se realizaron en un escenario simulado por lo que se contó con la colaboración de estudiantes de la carrera de ingeniería en sistemas decimo ciclo paralelo b del año en curso quienes se dividieron en dos grupos para hacer los roles de farmacias en un grupo de 13 estudiantes y un grupo de 15 estudiantes para el rol de clientes de farmacias.

Para poder medir cuantitativamente los resultados se aplicó encuestas a los participantes de las pruebas donde se puede determinar la funcionalidad total de la arquitectura.

2.1 PRUEBAS DE ACEPTACIÓN SEO

Los parámetros a medir en estas pruebas están establecidos en el plan de pruebas (Ver sección Resultados subsección pruebas).

CASOS DE PRUEBA SEO

Por favor dígnese llenando la siguiente encuesta según los resultados de su prueba de la aplicación Seo:

Marque su respuesta en la casilla correspondiente, teniendo en cuenta los siguientes valores.

- **Exitoso:** El resultado de la prueba ha sido el esperado.
- **Detenido:** Presenta dificultades para la correcta ejecución de la prueba
- **Fallido:** El resultado no ha sido el esperado
- **No ejecutado:** La prueba no se ha llevado a cabo.

a) Registro

La aplicación permite registrarme ingresando los datos principales

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

b) Inicio de Sesión

La aplicación permite iniciar sesión mediante un correo y contraseña.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

c) Mi ubicación

La aplicación muestra mi ubicación en el mapa

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

d) Farmacias en el mapa

La aplicación permite visualizar las diferentes farmacias en el mapa.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

e) Información de las farmacias

La aplicación permite visualizar la información de las farmacias como dirección, nombre etc.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

f) Buscar farmacias

La aplicación permite buscar farmacias mediante el nombre y muestra los resultados.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

g) Buscar farmacias

La aplicación permite buscar farmacias mediante el nombre y muestra los resultados.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

h) Buscar medicamentos

La aplicación permite buscar medicamentos mediante el nombre y muestra los resultados.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

i) Mostrar ruta

La aplicación marca la ruta hacia la farmacia desde la ubicación del usuario

- Exitoso
- Detenido
- Fallido
- No ejecutado

Comentario o Sugerencia

2.2 Resultados de la encuesta de pruebas de aceptación Seo

1) Registro

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

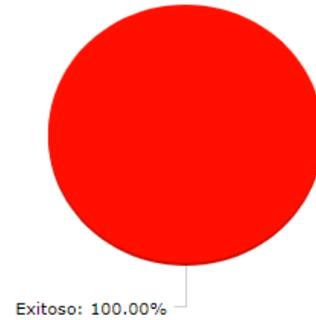


Figura 66: Resultado de caso de prueba registro

2) Inicio de Sesión

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

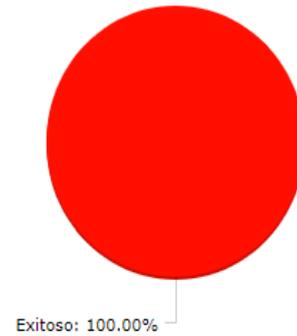


Figura 67: Resultado caso de prueba inicio de sesión

3) Mi ubicación

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

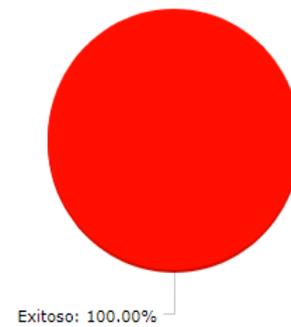


Figura 68: Resultado caso de prueba mi ubicación

4) Farmacias en el mapa

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

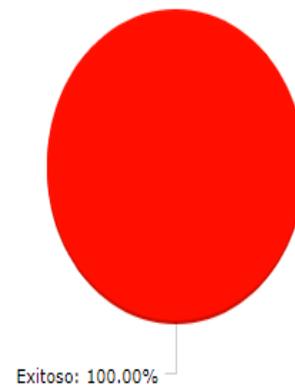


Figura 69: Resultado caso de prueba farmacias en el mapa

5) Información de las farmacias

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

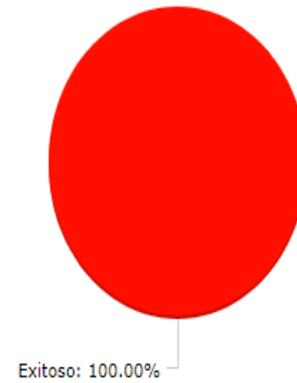


Figura 70: Resultado caso de prueba información de las farmacias

6) Buscar farmacias

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

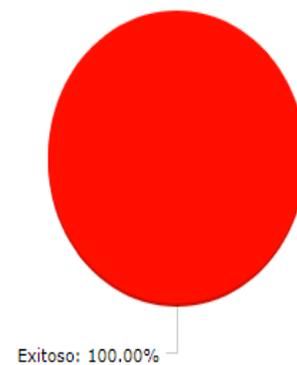


Figura 71: Resultado caso de prueba buscar farmacias

7) Buscar medicamentos

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

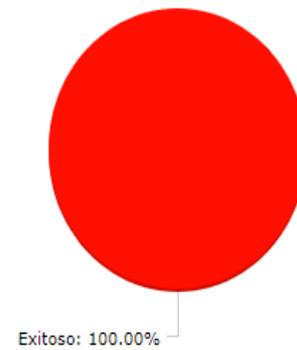


Figura 72: Resultado caso de prueba buscar medicamentos

8) Mostrar ruta

Número de participantes: 15

15 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

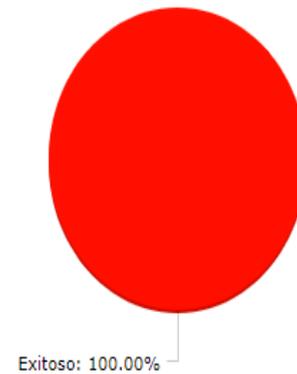


Figura 73: Resultado caso de prueba mostrar ruta

2.3 Pruebas de Aceptación Pharma

CASOS DE PRUEBA PHARMA

Por favor dígnese llenando la siguiente encuesta según los resultados de su prueba de la aplicación Seo:

Marque su respuesta en la casilla correspondiente, teniendo en cuenta los siguientes valores.

- **Exitoso:** El resultado de la prueba ha sido el esperado.
- **Detenido:** Presenta dificultades para la correcta ejecución de la prueba
- **Fallido:** El resultado no ha sido el esperado
- **No ejecutado:** La prueba no se ha llevado acabo.

1) Registro *

La aplicación debe permitir registrar ingresando los datos de la farmacia.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

2) Inicio de Sesión *

La aplicación debe permitir ingresar mediante el correo y contraseña además de notificar cuando exista un error.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

3) Seleccionar el estado *

La aplicación permite elegir el estado entre disponible y no disponible y notificara que el cambio se realizó con éxito. Además no se puede notificar de las consultas en un estado no disponible

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

4) Visualizar las nuevas consultas *

La aplicación muestra las nuevas consultas generadas

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

5) Contestar las nuevas consultas *

La aplicación permite elegir las diferentes opciones para contestar

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

6) Notificación de las nuevas consultas *

La aplicación notifica las nuevas consultas aun cuando la aplicación este ejecutando en segundo plano o con el celular bloqueado.

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

7) Visualización de información *

La aplicación muestra los datos de la cuenta registrada

- Exitoso
- Detenido
- Fallido
- No ejecutado

Sugerencia u observación

2.4 Resultados de pruebas de aceptación Pharma.

1) Registro

Número de participantes: 13

12 (92.3%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

1 (7.7%): No ejecutado

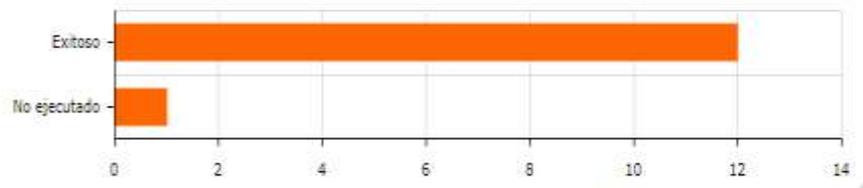


Figura 74: Resultado caso de prueba registro Pharma

2) Inicio de Sesión

Número de participantes: 13

13 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

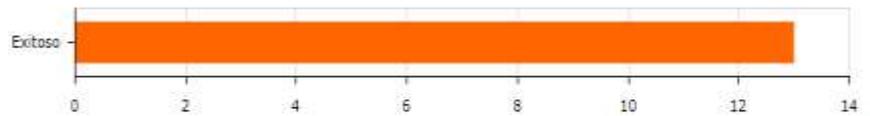


Figura 75: Resultado caso de prueba inicio de sesión Pharma

3) Seleccionar el estado

Número de participantes: 13

13 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado

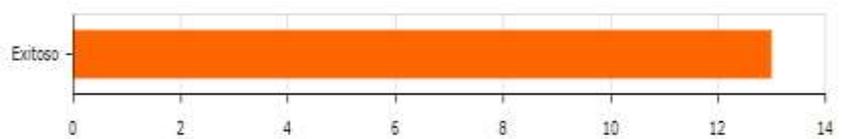


Figura 76: Resultado caso de prueba seleccionar estado Pharma

4) Visualizar las nuevas consultas

Número de participantes: 13

13 (100.0%): Exitoso

- (0.0%): Detenido

- (0.0%): Fallido

- (0.0%): No ejecutado



Figura 77: Resultado caso de prueba nuevas consultas Pharma

5) Contestar las nuevas consultas

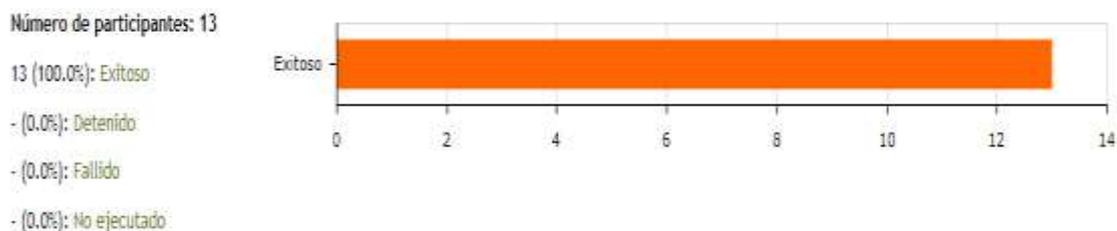


Figura 78: Resultado caso de prueba contestar nuevas consultas Pharma

6) Notificación de las nuevas consultas

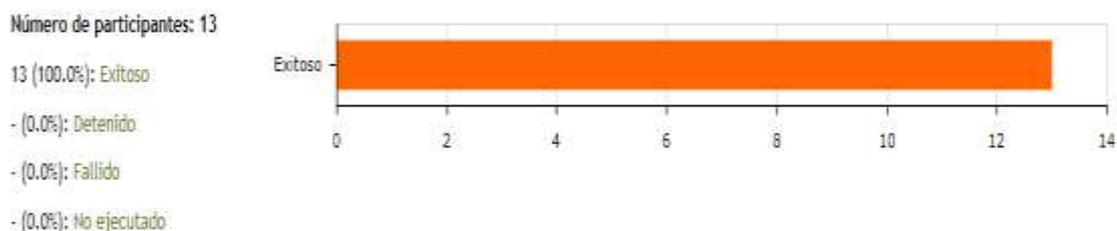


Figura 79: Resultado caso de prueba notificación nueva consulta Pharma

7) Visualización de información

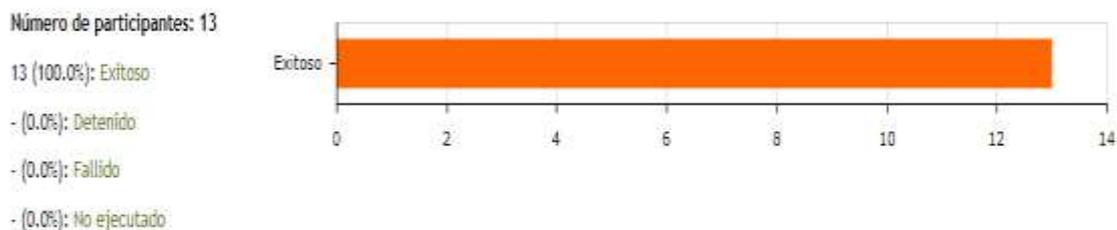


Figura 80: Resultado caso de prueba visualización de información Pharma

2.5 Interpretación de los resultados.

Basados en los resultados obtenido se puede determinar que la arquitectura funciona correctamente en los parámetros evaluados terminando con éxito la fase de pruebas y dando paso así a una versión estable y funcional de la arquitectura.

3. Anexo 3: Certificado de traducción del Resumen.

Loja, 20 de febrero del 2020

CERTIFICADO

Yo **MARIA TERESA GUAMAN CHAMBA** con cedula **1104187024** registro senescyt número **1008-12-1134669** certifico que se ha realizado la traducción del resumen del trabajo de titulación denominado "**Arquitectura informática para geo localización de medicamentos en farmacias**", realizado por el egresado **LEONARDO FAVIO CARAGUAY CARAGUAY**, previo a la obtención del título de Ingeniero en Sistemas.

Esto todo en cuanto puedo informar sobre el particular y autorizo al interesado hacer uso del presente certificado de la manera que estime conveniente.

Atentamente,



.....
MARIA TERESA GUAMAN CHAMBA

**LICENCIADA EN CIENCIAS DE LA EDUCACION EN LA ESPECIALIDAD DE
IDIOMA INGLES**

4. Anexo 4: Licencia.



Arquitectura informática para geo localización de medicamentos en farmacias por Leonardo Favio Caraguay Caraguay está bajo una licencia [Atribución-NoComercial 4.0 Internacional](#).

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

Adaptar — remezclar, transformar y construir a partir del material

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia](#).