



Desarrollo de una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de covid-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga

Guaila Chulco Darwin Jair, Rosas Salguero Byron Manuel

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería en Software

Trabajo de titulación, previo a la obtención del título de Ingeniero en Software

Ing. Carrillo Medina, José Luis, PHD.

Latacunga

19 de marzo del 2021



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE**

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Desarrollo de una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de covid-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga”** fue realizado por los señores **Guaila Chulco, Darwin Jair y Rosas Salguero, Byron Manuel** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga 19 de marzo de 2021

Firma:

.....

PHD. Carrillo Medina José Luis

C. C.: 0501553788



Document Information

Analyzed document	Tesis_Guaila_Rosas_SW_18-03-2021.docx (D98843178)
Submitted	3/19/2021 5:47:00 AM
Submitted by	Jose Carrillo
Submitter email	jlcarrillo@espe.edu.ec
Similarity	2%
Analysis address	jlcarrillo.espe@analysis.arkund.com



JOSE LUIS
CARRILLO

Sources included in the report

W	URL: https://repositorio.espe.edu.ec/bitstream/21000/13115/1/T-ESPEL-SOF-0015.pdf Fetched: 12/27/2019 4:41:05 AM		1
W	URL: https://repository.ucc.edu.co/bitstream/20.500.12494/7898/1/SISTEMA%20DE%20CONTROL... Fetched: 3/16/2021 8:26:15 PM		2
W	URL: https://repositorio.espe.edu.ec/bitstream/21000/13114/1/T-ESPEL-SOF-0014.pdf Fetched: 1/4/2021 8:04:34 AM		2
SA	T_Final_F.docx Document T_Final_F.docx (D51865056)		1
W	URL: https://docplayer.es/82854452-Universidad-mayor-de-san-andres-facultad-de-ciencias... Fetched: 9/11/2020 7:20:29 PM		1
SA	PARA URKUND - Domínguez Orrala Nixon Johnny - Quillupangui Orquera Jefferson Bryan ... Document PARA URKUND - Domínguez Orrala Nixon Johnny - Quillupangui Orquera Jefferson Bryan ... (D97661394)		1
SA	TESIS - CALVA GUERRERO.docx Document TESIS - CALVA GUERRERO.docx (D64595615)		1
W	URL: http://www.dspace.uce.edu.ec/bitstream/25000/19476/1/T-UCE-0011-ICF-170.pdf Fetched: 11/29/2020 8:34:53 PM		1



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE**

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Guaila Chulco, Darwin Jair y Rosas Salguero, Byron Manuel**, con cédulas de ciudadanía n° 1600666026 y 1723943393 respectivamente, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Desarrollo de una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de covid-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga 15 de marzo de 2021

Firmas

Darwin Guaila

Guaila Chulco, Darwin Jair

C.C.: 1600666026

Byron Rosas

Rosas Salguero, Byron Manuel

C.C.: 1723943393



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros **Guaila Chulco, Darwin Jair** y **Rosas Salguero, Byron Manuel**, con cédulas de ciudadanía n° **1600666026** y **1723943393** respectivamente, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Desarrollo de una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de covid-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga 15 de marzo de 2021

Firmas

Darwin Guaila

Guaila Chulco, Darwin Jair

C.C.: 1600666026

Byron Rosas

Rosas Salguero, Byron Manuel

C.C.: 1723943393

Dedicatoria

A Dios por darme la fuerza y sabiduría para seguir adelante frente a los obstáculos y guiarme para cumplir con mis objetivos.

A mi muy amado padre Manuel Rosas Jiménez y a mi amada madre Isabel Salguero Sinchiri quienes han estado presentes en cada paso que doy siendo mi apoyo incondicional durante mi camino hacia el conocimiento y por brindarme sus consejos y confianza a pesar de las dificultades, siendo mi fuerza e inspiración para continuar día a día.

A mis tíos cercanos Juan Salguero, Javier Salguero y Carlos Salguero quienes estuvieron apoyándome con sus consejos y fuerza para saber afrontar las adversidades que se me presentaron.

A mis tías Katherine Salguero, Cintia Salguero y Lourdes Salguero por estar siempre junto a mi brindándome el ánimo para continuar en los malos momentos.

A mis primos Michael Estrella, Mishell Figueroa, Johanna Loaiza, Anthony Figueroa, David Salguero y Nahomi Salguero por acompañarme en esta etapa de mi vida y de quienes quiero ser un ejemplo de superación en conocimiento, en valores y de vida, para que ustedes los tomen como base y se esfuercen como personas de bien y en búsqueda de la excelencia.

A mi enamorada Estefanía quien me dio consejos cuando los malos momentos se presentaban.

A esta prestigiosa Universidad que me abrió las puertas hacia un mundo de conocimiento, ayudándome en la formación profesional pero aún más importante ser un buen ser humano que aporte a la sociedad.

Sr. Byron Rosas

Dedicatoria

A mi madre y a mi padre por acompañarme durante todo el proceso, estar pendiente de mí y apoyarme incondicionalmente con su ejemplo y guía.

A mi familia que estuvo brindándome su apoyo y ánimo en los momentos de adversidad y lo cual me permitió superar cada uno de los obstáculos.

A la prestigiosa Universidad de las Fuerzas Armadas ESPE por darme el conocimiento necesario para lograr este proyecto.

Sr. Darwin Guilla.

Agradecimiento

Agradezco a Dios por brindarme la oportunidad de crecer y aprender, de darme la fuerza necesaria para superar las adversidades que se me presentaron.

Un especial agradecimiento a mis padres por ser el pilar fundamental en mi camino del conocimiento y de vida, gracias por ser mi ejemplo a seguir como personas de valores y de conocimiento, también agradezco por sus consejos y apoyo incondicional.

A mis tíos y tías por haber aportado con su ejemplo, ánimo y buenos consejos para que yo pudiera cumplir esta anhelada meta. También agradezco a mis tíos políticos Ivan Figueroa, Byron Loaiza y a mi tía Mayra Marmol por aportar a que uno de mis sueños se haga realidad.

A mis primos donde encontré un hombro donde apoyarme en los malos momentos y de quienes espero sigan mis pasos.

A mi amigo Darwin por su arduo trabajo para hacer posible este proyecto, gracias por aportar con conocimiento, responsabilidad y amistad.

A la Universidad de las Fuerzas Armadas ESPE Sede Latacunga por permitirme estudiar, conocer y aprender para formarme como un profesional de excelencia.

Al Doctor José Luís Carrillo por ser nuestro guía científico para hacer posible este proyecto, gracias por su confianza y respaldo.

A nuestra estimada Ingeniera Ximena López, quien siempre estuvo pendiente de nosotros sus estudiantes, gracias por sus consejos y ánimo.

Un agradecimiento a todos mis docentes quienes aportaron con su conocimiento y experiencias para que yo me pudiera formar como profesional.

Sr. Byron Rosas.

Agradecimiento

Agradezco a mis padres por ser la fuerza y guía que me impulso a lograr mis metas, gracias por estar siempre para mí y ser mi ejemplo a seguir.

A la Universidad de las Fuerzas Armadas ESPE Sede Latacunga que proporciono el espacio y recursos para lograr completar mi formación profesional y como un buen ser humano.

A mis docentes por aportar con su conocimiento para que mi formación como profesional sea la mejor.

A mi tutor de tesis el Doctor José Luis Carrillo que fue un guía para lograr este proyecto.

Sr. Darwin Guilla.

Tabla de contenidos

Carátula	1
Certificación	2
Reporte Urkund.....	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria.....	6
Agradecimiento	8
Tabla de contenidos	10
Índice de tablas	15
Índice de figuras.....	18
Resumen	24
Abstract.....	25
Generalidades.....	26
Introducción	26
Planteamiento y Formulación del Problema.....	29
Justificación e Importancia.....	31
Objetivos	32
<i>Objetivo General</i>	32
<i>Objetivos Específicos</i>	32
Hipótesis	33
Variables de la investigación.....	33
<i>Indicadores</i>	33
Marco teórico.....	34
Introducción del capítulo	34
Antecedentes históricos	34

Antecedentes Conceptuales y Referenciales	36
<i>Conceptos relacionados al problema.....</i>	<i>36</i>
Pandemia.....	36
Coronavirus.....	36
Covid-19.....	36
Tele asesoría.....	37
Videollamadas.....	37
<i>Conceptos relacionados a la metodología.....</i>	<i>37</i>
Software.....	37
Ingeniería de software.....	38
Metodología.....	38
Método.....	38
Proceso.....	38
Ciclo de Vida del software.....	39
Modelo de proceso incremental.....	39
Metodología Ágil.....	40
SCRUM.....	42
Valores de Scrum.....	43
Equipo en Scrum.....	43
Eventos de Scrum.....	44
Artefactos de Scrum.....	46
Arquitectura de software.....	46
Paradigmas de Programación.....	47
Conceptos Base de la Programación Orientada a Objetos(POO).....	48
Principios SOLID.....	49
Arquitectura limpia.....	51
Arquitectura Hexagonal.....	51
BCE.....	52
Onion Architecture.....	52
Características de una arquitectura limpia:.....	52
Capas de una Arquitectura limpia.....	53
<i>Conceptos relacionados a las tecnologías utilizadas.....</i>	<i>53</i>
Aplicación Software.....	54

Aplicación web.....	54
Backend Software.....	54
Frontend Software.....	54
API.....	55
API Rest.....	55
API graphql.....	56
Diferencias entre API REST y API GraphQL.....	57
Agora.io.....	58
Javascript.....	58
Typescript.....	58
Html.....	59
Css.....	59
Node.js.....	60
Angular.....	60
Mongodb.....	60
Firebase.....	60
Mongoose.....	61
Typegoose.....	61
TypeGraphQL.....	61
Antecedentes Contextuales.....	61
Desarrollo de la aplicación web.....	64
Introducción.....	64
Análisis.....	64
<i>Épicas</i>	67
<i>Historias de usuario</i>	68
<i>Product Backlog (Backlog de producto)</i>	83
<i>Sprints</i>	86
Diseño.....	91
<i>Diagrama de arquitectura física</i>	91
<i>Diagrama de arquitectura web</i>	92

<i>Diagrama de arquitectura lógica</i>	94
<i>Diagrama de arquitectura limpia</i>	95
<i>Diagrama de clases</i>	96
<i>Diagrama entidad relación</i>	98
<i>Mockups(Maquetas)</i>	99
Desarrollo por Sprint	104
<i>Capa de Dominio</i>	104
Tipos de datos propios	104
Enums	104
Entidades	109
Abstracciones	114
Interactors	117
Sprint 1: Interactors	118
Sprint 2: Interactors	120
Sprint 3: Interactors	122
Sprint 4: Interactors	125
Sprint 5: Interactors	127
Sprint 6: Interactors	130
Sprint 7: Interactors	131
Sprint 8: Interactors	134
<i>Capa de detalle</i>	136
Servicios	136
Persistencia	137
API resolvers	139
<i>Cliente: Front end</i>	143
Pruebas e Implementación	149
<i>Pruebas</i>	149
<i>Implementación</i>	162
Validación	163
Introducción	163

Análisis de resultados.....	164
Análisis de satisfacción respecto a la aplicación web.....	173
Conclusiones y recomendaciones.....	177
Conclusiones	177
Recomendaciones	178
Bibliografía.....	179
Anexos	185

Índice de tablas

Tabla 1 <i>Análisis y planificación de Scrum</i>	65
Tabla 2 <i>Historias de usuario pertenecientes a la Épica 1</i>	69
Tabla 3 <i>Historias de usuario pertenecientes a la Épica 2</i>	71
Tabla 4 <i>Historias de usuario pertenecientes a la Épica 3</i>	73
Tabla 5 <i>Historias de usuario pertenecientes a la Épica 4</i>	77
Tabla 6 <i>Historias de usuario pertenecientes a la Épica 5</i>	78
Tabla 7 <i>Historias de usuario pertenecientes a la Épica 6</i>	80
Tabla 8 <i>Detalle de cada Sprint</i>	87
Tabla 9 <i>Estado de la entidad (Enum)</i>	105
Tabla 10 <i>Estado del usuario (Enum)</i>	105
Tabla 11 <i>Roles(Enum)</i>	105
Tabla 12 <i>Motivo de alta (Enum)</i>	105
Tabla 13 <i>Género del usuario (Enum)</i>	106
Tabla 14 <i>Institución que aisló al paciente (Enum)</i>	106
Tabla 15 <i>Tipo de espacio (Enum)</i>	107
Tabla 16 <i>Dificultad para respirar (Enum)</i>	107
Tabla 17 <i>Tipo de Examen (Enum)</i>	107
Tabla 18 <i>Estado diario del paciente (Enum)</i>	108
Tabla 19 <i>Diagnóstico actual (Enum)</i>	108
Tabla 20 <i>Estado de la notificación(Enum)</i>	108
Tabla 21 <i>Entidades – Entidad Base</i>	109
Tabla 22 <i>Entidades – Entidad Usuario</i>	110
Tabla 23 <i>Entidades – Entidad Datos de Paciente</i>	111
Tabla 24 <i>Entidades – Entidad Hospital</i>	112
Tabla 25 <i>Entidades – Entidad Espacio</i>	112

Tabla 26 <i>Entidades – Entidad Seguimiento</i>	113
Tabla 27 <i>Entidades – Entidad Notificación</i>	114
Tabla 28 <i>Lista de chequeo N° 1</i>	149
Tabla 29 <i>Lista de chequeo N° 2</i>	149
Tabla 30 <i>Lista de chequeo N° 3</i>	150
Tabla 31 <i>Lista de chequeo N° 4</i>	150
Tabla 32 <i>Lista de chequeo N° 5</i>	150
Tabla 33 <i>Lista de chequeo N° 6</i>	151
Tabla 34 <i>Lista de chequeo N° 7</i>	151
Tabla 35 <i>Lista de chequeo N° 8</i>	151
Tabla 36 <i>Lista de chequeo N° 9</i>	152
Tabla 37 <i>Lista de chequeo N° 10</i>	152
Tabla 38 <i>Lista de chequeo N° 11</i>	152
Tabla 39 <i>Lista de chequeo N° 12</i>	153
Tabla 40 <i>Lista de chequeo N° 13</i>	153
Tabla 41 <i>Lista de chequeo N° 14</i>	153
Tabla 42 <i>Lista de chequeo N° 15</i>	154
Tabla 43 <i>Lista de chequeo N° 16</i>	154
Tabla 44 <i>Lista de chequeo N° 17</i>	154
Tabla 45 <i>Lista de chequeo N° 18</i>	155
Tabla 46 <i>Lista de chequeo N° 19</i>	155
Tabla 47 <i>Lista de chequeo N° 20</i>	155
Tabla 48 <i>Lista de chequeo N° 21</i>	156
Tabla 49 <i>Lista de chequeo N° 22</i>	156
Tabla 50 <i>Lista de chequeo N° 23</i>	156
Tabla 51 <i>Lista de chequeo N° 24</i>	157

Tabla 52 <i>Lista de chequeo N° 25</i>	157
Tabla 53 <i>Lista de chequeo N° 26</i>	157
Tabla 54 <i>Lista de chequeo N° 27</i>	158
Tabla 55 <i>Lista de chequeo N° 28</i>	158
Tabla 56 <i>Lista de chequeo N° 29</i>	158
Tabla 57 <i>Lista de chequeo N° 30</i>	159
Tabla 58 <i>Lista de chequeo N° 31</i>	159
Tabla 59 <i>Lista de chequeo N° 32</i>	159
Tabla 60 <i>Lista de chequeo N° 33</i>	160
Tabla 61 <i>Lista de chequeo N° 34</i>	160
Tabla 62 <i>Lista de chequeo N° 35</i>	160
Tabla 63 <i>Lista de chequeo N° 36</i>	161
Tabla 64 <i>Lista de chequeo N° 37</i>	161
Tabla 65 <i>Lista de chequeo N° 38</i>	161
Tabla 66 <i>Lista de chequeo N° 39</i>	162
Tabla 67 <i>Resumen general de la información de los pacientes</i>	164
Tabla 68 <i>Porcentaje de seguimientos realizados por los pacientes</i>	170
Tabla 69 <i>Resultados tabulados de las encuestas de satisfacción</i>	173
Tabla 70 <i>Preguntas sobre el nivel de satisfacción para el experto</i>	175

Índice de figuras

Figura 1 <i>Tendencia acumulada nacional de casos confirmados COVID-19</i>	28
Figura 2 <i>Marco Scrum</i>	43
Figura 3 <i>Épicas con su indicador de duración</i>	67
Figura 4 <i>Historia de usuario 1</i>	69
Figura 5 <i>Historia de usuario 2</i>	69
Figura 6 <i>Historia de usuario 3</i>	69
Figura 7 <i>Historia de usuario 4</i>	70
Figura 8 <i>Historia de usuario 5</i>	70
Figura 9 <i>Historia de usuario 6</i>	71
Figura 10 <i>Historia de usuario 7</i>	71
Figura 11 <i>Historia de usuario 8</i>	71
Figura 12 <i>Historia de usuario 9</i>	72
Figura 13 <i>Historia de usuario 10</i>	72
Figura 14 <i>Historia de usuario 11</i>	72
Figura 15 <i>Historia de usuario 12</i>	73
Figura 16 <i>Historia de usuario 13</i>	73
Figura 17 <i>Historia de usuario 14</i>	74
Figura 18 <i>Historia de usuario 15</i>	74
Figura 19 <i>Historia de usuario 16</i>	74
Figura 20 <i>Historia de usuario 17</i>	75
Figura 21 <i>Historia de usuario 18</i>	75
Figura 22 <i>Historia de usuario 19</i>	75
Figura 23 <i>Historia de usuario 20</i>	76
Figura 24 <i>Historia de usuario 21</i>	76
Figura 25 <i>Historia de usuario 22</i>	77

Figura 26 <i>Historia de usuario 23</i>	77
Figura 27 <i>Historia de usuario 24</i>	77
Figura 28 <i>Historia de usuario 25</i>	78
Figura 29 <i>Historia de usuario 26</i>	78
Figura 30 <i>Historia de usuario 27</i>	79
Figura 31 <i>Historia de usuario 28</i>	79
Figura 32 <i>Historia de usuario 29</i>	80
Figura 33 <i>Historia de usuario 30</i>	80
Figura 34 <i>Historia de usuario 31</i>	80
Figura 35 <i>Historia de usuario 32</i>	81
Figura 36 <i>Historia de usuario 33</i>	81
Figura 37 <i>Historia de usuario 34</i>	81
Figura 38 <i>Historia de usuario 35</i>	82
Figura 39 <i>Historia de usuario 36</i>	82
Figura 40 <i>Historia de usuario 37</i>	82
Figura 41 <i>Historia de usuario 38</i>	83
Figura 42 <i>Historia de usuario 39</i>	83
Figura 43 <i>Backlog de producto(Parte 1)</i>	84
Figura 44 <i>Backlog de producto(Parte 2)</i>	85
Figura 45 <i>Backlog de producto(Parte 3)</i>	85
Figura 46 <i>Backlog de producto(Parte 4)</i>	86
Figura 47 <i>Historias de usuario – Sprint 1</i>	87
Figura 48 <i>Historias de usuario – Sprint 2</i>	87
Figura 49 <i>Historias de usuario – Sprint 3</i>	88
Figura 50 <i>Historias de usuario – Sprint 4</i>	88
Figura 51 <i>Historias de usuario – Sprint 5</i>	89

Figura 52 <i>Historias de usuario – Sprint 6</i>	90
Figura 53 <i>Historias de usuario – Sprint 7</i>	90
Figura 54 <i>Historias de usuario – Sprint 8</i>	91
Figura 55 <i>Arquitectura física Cliente - Servidor</i>	92
Figura 56 <i>Arquitectura web</i>	93
Figura 57 <i>Diagrama de arquitectura lógica</i>	94
Figura 58 <i>Diagrama de arquitectura limpia</i>	95
Figura 59 <i>Diagrama de clases general</i>	96
Figura 60 <i>Diagrama de clases específico de una historia de usuario</i>	97
Figura 61 <i>Diagrama entidad - relación</i>	98
Figura 62 <i>Mockup(Maqueta) de Inicio de Sesión</i>	99
Figura 63 <i>Mockup(Maqueta) de registro de usuario</i>	100
Figura 64 <i>Mockup(Maqueta) de Estado de Seguimientos</i>	100
Figura 65 <i>Mockup(Maqueta) de tabla</i>	101
Figura 66 <i>Mockup(Maqueta) de datos de paciente</i>	101
Figura 67 <i>Mockup(Maqueta) del resumen de seguimientos de un paciente</i>	102
Figura 68 <i>Mockup(Maqueta) de la atención de un seguimiento</i>	102
Figura 69 <i>Mockup(Maqueta) de la vista de mapa</i>	103
Figura 70 <i>Mockup(Maqueta) de la vista de gráficos</i>	103
Figura 71 <i>Repositorio de Hospital</i>	115
Figura 72 <i>Repositorio de Seguimiento</i>	115
Figura 73 <i>Repositorio de Notificación</i>	116
Figura 74 <i>Repositorio de Espacio</i>	116
Figura 75 <i>Repositorio de usuario</i>	116
Figura 76 <i>Servicio de Correo</i>	117
Figura 77 <i>Servicio de Encriptado</i>	117

Figura 78 <i>Sprint 1 – Interactor Crear usuario administrador</i>	118
Figura 79 <i>Sprint 1 – Interactor Crear espacio</i>	118
Figura 80 <i>Sprint 1 – Interactor Filtrar espacio</i>	119
Figura 81 <i>Sprint 1 – Interactor Ver espacio</i>	119
Figura 82 <i>Sprint 2 – Interactor Crear hospital</i>	120
Figura 83 <i>Sprint 2 - Interactor Filtrar hospital</i>	120
Figura 84 <i>Sprint 2 – Interactor Crear cuenta de usuario</i>	121
Figura 85 <i>Sprint 2 – Interactor Iniciar sesión de usuario</i>	121
Figura 86 <i>Sprint 2 – Interactor Asignar roles</i>	122
Figura 87 <i>Sprint 3 – Interactor Mostrar perfil de usuario</i>	122
Figura 88 <i>Sprint 3 – Interactor Solicitar seguimiento</i>	123
Figura 89 <i>Sprint 3 – Interactor Cancelar seguimiento</i>	123
Figura 90 <i>Sprint 3 – Interactor Filtrar seguimiento</i>	124
Figura 91 <i>Sprint 3 – Interactor Agendar seguimiento</i>	124
Figura 92 <i>Sprint 4 – Interactor Atender seguimiento</i>	125
Figura 93 <i>Sprint 4 – Interactor Mostrar seguimiento completo</i>	125
Figura 94 <i>Sprint 4 - Interactor Ver un seguimiento</i>	126
Figura 95 <i>Sprint 4 – Interactor Editar un seguimiento</i>	126
Figura 96 <i>Sprint 4 – Interactor Crear una notificación</i>	127
Figura 97 <i>Sprint 5 – Interactor Obtener notificaciones recibidas</i>	127
Figura 98 <i>Sprint 5 – Interactor Ver una notificación</i>	128
Figura 99 <i>Sprint 5 – Interactor Colocar notificación como vista(leída)</i>	128
Figura 100 <i>Sprint 5 – Interactor Obtener notificaciones enviadas</i>	129
Figura 101 <i>Sprint 6 – Interactor Ver mapa</i>	130
Figura 102 <i>Sprint 6 – Ver usuarios sin seguimientos</i>	131
Figura 103 <i>Sprint 7 – Interactor Ver contadores</i>	132

Figura 104 <i>Sprint 7 – Interactor Actualizar hospital</i>	132
Figura 105 <i>Sprint 7 – Interactor Actualizar espacio</i>	133
Figura 106 <i>Sprint 7 – Interactor Listar usuarios</i>	133
Figura 107 <i>Sprint 7 – Interactor Dar de alta a un usuario</i>	134
Figura 108 <i>Sprint 8 – Interactor Cambiar contraseña</i>	134
Figura 109 <i>Sprint 8 – Interactor Actualizar perfil de usuario</i>	135
Figura 110 <i>Sprint 8 – Interactor Recuperar contraseña</i>	135
Figura 111 <i>Sprint 8 – Interactor Resetear contraseña</i>	136
Figura 112 <i>Detalle – Clase concreta del servicio de encriptado</i>	137
Figura 113 <i>Detalle – Clase concreta del servicio de correo</i>	137
Figura 114 <i>Detalle – Clase concreta del repositorio de seguimiento</i>	138
Figura 115 <i>Detalle – Clase concreta del repositorio de notificación</i>	138
Figura 116 <i>Detalle – Clase concreta del repositorio de hospital</i>	138
Figura 117 <i>Detalle - Clase concreta del repositorio de espacio</i>	139
Figura 118 <i>Detalle - Clase concreta del repositorio de usuario</i>	139
Figura 119 <i>Detalle – Resolver usuario</i>	140
Figura 120 <i>Detalle – Resolver seguimiento</i>	140
Figura 121 <i>Detalle – Resolver notificación</i>	141
Figura 122 <i>Detalle – Resolver hospital</i>	141
Figura 123 <i>Detalle – Resolver espacio</i>	142
Figura 124 <i>Detalle – Resolver Data(Para interactor de mapa y contadores)</i>	142
Figura 125 <i>Detalle - API</i>	142
Figura 126 <i>Cliente – Vista de inicio de sesión</i>	143
Figura 127 <i>Cliente – Vista de registro de usuario</i>	143
Figura 128 <i>Cliente – Vista de tabla (pacientes)</i>	144
Figura 129 <i>Cliente – Vista de seguimientos</i>	144

Figura 130 <i>Cliente – Vista de resumen de seguimientos</i>	145
Figura 131 <i>Cliente - Vista de seguimientos</i>	145
Figura 132 <i>Cliente – Vista de pacientes sin seguimientos</i>	146
Figura 133 <i>Cliente – Vista de atención a través de una video llamada</i>	146
Figura 134 <i>Cliente – Vista de atención de un seguimiento formulario</i>	147
Figura 135 <i>Cliente – Vista de ficha médica de un paciente</i>	147
Figura 136 <i>Cliente – Vista de gráficas</i>	148
Figura 137 <i>Cliente – Vista de mapa</i>	148
Figura 138 <i>Porcentaje de pacientes con alguna enfermedad</i>	165
Figura 139 <i>Porcentaje de pacientes fumadores</i>	166
Figura 140 <i>Comparación de las temperaturas de los pacientes</i>	167
Figura 141 <i>Porcentaje de pacientes por diagnóstico</i>	168
Figura 142 <i>Porcentaje de pacientes atendidos</i>	169
Figura 143 <i>Gráfica de seguimientos</i>	171
Figura 144 <i>Seguimientos realizados y esperados</i>	172
Figura 145 <i>Porcentaje de satisfacción de usuarios</i>	174

Resumen

El presente proyecto está orientado al desarrollo de una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de covid-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga, está enfocado en apoyar a los profesionales de la salud para realizar una atención médica sin exponerse al contagio de covid-19, también provee a las personas contagiadas o sospechosas una forma de comunicarse con un doctor para recibir atención médica. La aplicación web tiene como principal funcionalidad la comunicación a través de video y audio entre un paciente y doctor, con el fin de realizar un seguimiento del estado de salud del paciente, conocer su evolución respecto a sus signos vitales y síntomas. Posee un módulo de estadística que proporciona gráficas de datos de los pacientes respecto a sus seguimientos. Y un módulo de mapa de calor que permite observar la cantidad de pacientes dentro de un área física. El desarrollo de la aplicación web está basado en buenas prácticas de desarrollo de software, programación orientada a objetos, principios SOLID y arquitectura limpia para mejorar el desacoplamiento de componentes y proveer escalabilidad de funcionalidad al software. También se usó metodologías de desarrollo ágil como Scrum, de la cual se tomaron los conceptos base para adaptarlo al desarrollo del proyecto.

PALABRAS CLAVES:

- **APLICACIÓN WEB**
- **TELE ASESORÍA MÉDICA**
- **SCRUM**
- **ARQUITECTURA LIMPIA.**

Abstract

This project is oriented to the development of a web application that supports the monitoring of the health status of people exposed to covid-19 infection through tele medical assistance at the Universidad de las Fuerzas Armadas ESPE Sede Latacunga, is focused on supporting health professionals to perform medical care without being exposed to covid-19 infection, also provides infected or suspected people a way to communicate with a doctor to receive medical care. The main functionality of the web application is the video and audio communication between a patient and a doctor, in order to monitor the patient's health status and to know the evolution of the patient's vital signs and symptoms. It has a statistics module that provides graphs of patient data regarding their follow-ups. And a heat map module that allows to observe the number of patients within a geographical area. The development of the web application is based on good software development practices, object-oriented programming, SOLID principles and clean architecture to improve the decoupling of components and provide scalability of functionality to the software. Agile development methodologies such as Scrum were also used, from which the base concepts were taken to adapt it to the development of the project.

KEYWORDS:

- **WEB APPLICATION**
- **TELE MEDICAL ASSISTANCE**
- **SCRUM**
- **CLEAN ARCHITECTURE.**

Capítulo I

Generalidades

1.1. Introducción

Los coronavirus (CoV) son una amplia familia de virus que pueden causar diversas afecciones, desde el resfriado común hasta enfermedades más graves, como ocurre con el coronavirus causante del síndrome respiratorio de Oriente Medio (MERS-CoV) y el que ocasiona el síndrome respiratorio agudo severo (SRAS-CoV), originada también en China, en noviembre de 2002, la epidemia de SRAS eventualmente se extendió por 26 países, con 8.098 casos confirmados y 774 muertos (Wallace, 2020).

La OMS señala que la mayoría de variantes de este tipo de enfermedad pueden ser transmitidas a los seres humanos a través de los animales (transmisión zoonótica). Se sabe que hay otros tipos de coronavirus que se encuentran circulando entre animales, que todavía no han infectado al ser humano (Organización Mundial de la Salud[OMS], 2020). A finales del año 2019 se registró una nueva mutación de esta enfermedad (denominada COVID-19, coronavirus de tipo 2 causante del síndrome respiratorio agudo severo (SRAS-CoV-2)), que es una nueva cepa de coronavirus, que no se había encontrado antes en el ser humano (OMS, WHO, 2020).

La COVID-19 es una enfermedad infecciosa que se transmite mediante vía respiratoria, causada por el coronavirus. Teniendo como principal brote la ciudad de Wuhan, república de China, en diciembre de 2019. La OMS en el mes de marzo de 2020 declara al COVID-19 como una pandemia ya que esta se propago por todas partes del mundo. (OMS, WHO, 2020)

Al presentarse este nuevo virus, en china inicialmente se lo nombro neumonía por coronavirus nuevo (NCP), para luego nombrarse por la OMS como COVID-19 y por el Comité Internacional de Taxonomía de Virus como SARS-CoV-2. A finales del mes de enero la OMS lo declaro como una emergencia sanitaria mundial, siendo el epicentro de este brote en curso la ciudad de Wuhan, en la provincia de Hubei, centro de China, también se pensó que el mercado mayorista de productos de mar Huanan es el lugar donde el SARS-CoV-2 cruzo la barrera desde un animal e infectar un humano. Al 24 de febrero se han reportado más de 80.000 casos confirmados, incluidas más de 2.700 muertes en todo el mundo que afectan al menos a 37 países. (Yuen, Ye, Fung, Chan, & Jin, 2020).

En septiembre de 2020 a nivel mundial se han reportado 29.444.198 número de casos confirmados, 20.1 millones de personas recuperadas y 931.321 personas fallecidas por este nuevo virus. (OMS, World Health Organization, 2020).

Cada generación aproximadamente, ocurre un evento similar, una pandemia global que causa estragos en la población mundial, dejando vulnerable los sistemas de salud y la economía. El COVID-19 es el tercer caso documentado de un coronavirus animal transmitido a humanos que está causando una pandemia. (Portnoy, Waller, & Elliott, 2020).

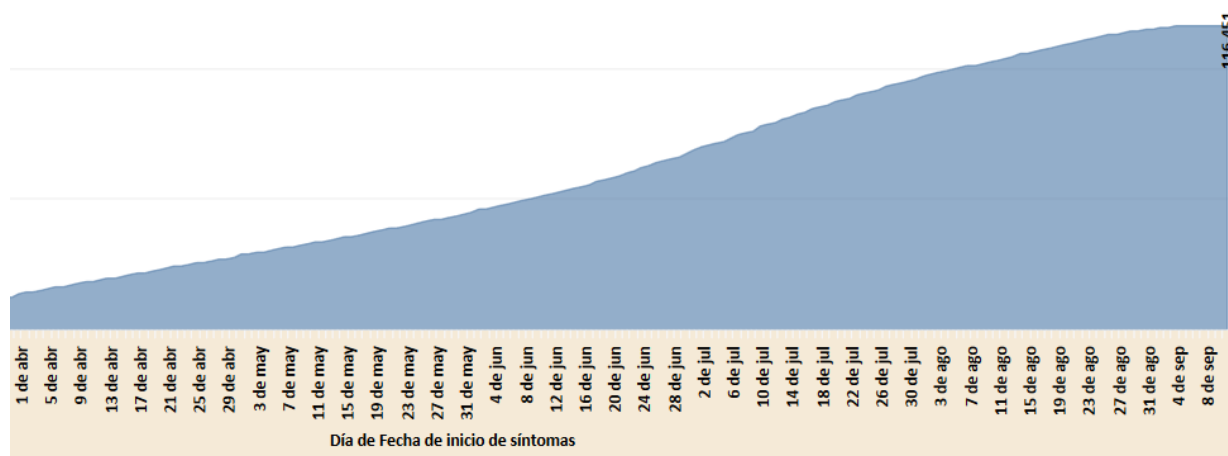
Al ser un virus nuevo no se tiene una vacuna, la respuesta inicial es tomar medidas de tal forma que se retrase la propagación con el fin de evitar contagios y el colapso del sistema de atención médica. Este virus es mucho más infeccioso que el síndrome respiratorio agudo severo, lo que lleva a diez veces más casos en una cuarta parte del tiempo. (Portnoy, Waller, & Elliott, 2020).

Debido a los avances tecnológicos en aplicaciones software que funcionan en tiempo real, son interactivas y hacen uso de audio y video, representan un apoyo en este tiempo de pandemia, siendo su uso en la telemedicina (TM) con el potencial de ayudar al permitir que los pacientes con enfermedades leves reciban la atención de apoyo que necesitan y, al mismo tiempo, minimizar su exposición a otros pacientes con enfermedades agudas. Siendo la TM una alternativa eficaz y más segura en las circunstancias actuales. (Portnoy, Waller, & Elliott, 2020)

En el Ecuador se confirmó el primer caso de coronavirus el día 29 de febrero de 2020, el 13 de marzo de 2020 se activó el COE Nacional para la coordinación de la emergencia y el 11 de marzo de 2020 por la Ministra de Salud, se declara el Estado de Emergencia Sanitaria en el Sistema Nacional de Salud. (Ministerio de Salud Pública, 2020).

Figura 1

Tendencia acumulada nacional de casos confirmados COVID 19



Nota. (Ministerio de Salud Pública, 2020)

Los datos relacionados con la atención médica o control a través de medios tecnológicos según la página del ministerio de Salud Pública del Ecuador son: 828.914 llamadas al 171 relacionadas a COVID-19, 302.246 seguimiento telefónico, llamadas a pacientes con diagnóstico confirmado, 110.041 tele consultas, ciudadanos atendidos a través de APP SALUDEC y por un médico del 171, consultado el 20 de septiembre del 2020.

Hasta el día 16 de sep. de 2020 en Ecuador se han reportado 119.553 número de casos confirmados, 97.063 personas recuperadas y 10.963 personas fallecidas por este nuevo virus. (OMS, World Health Organization, 2020)

1.2. Planteamiento y Formulación del Problema.

Hasta el 17 de septiembre de 2020 se contabilizaban a nivel mundial 29.737.453 millones de personas contagiadas, 937.391 personas fallecidas por COVID-19, se registran 292.307 nuevos casos las últimas 24 horas y también 6057 nuevas muertes (Organización Mundial de la Salud, 2020).

En América la situación, no es menos alarmante, según la Organización Panamericana de la Salud (2020) hasta el 18 de septiembre de 2020 se notificaron 15.228.769 de personas contagiadas y 522.114 decesos por COVID-19, Siendo el continente más golpeado por la pandemia a nivel mundial.

En nuestro país hasta el momento el Instituto Nacional de Investigación en Salud Pública (INSPI) reporta 122.257 casos confirmados y 10.864 personas fallecidas entre confirmados y probables de COVID-19. A nivel de nacional, pichincha es la provincia

más golpeada por la pandemia con 27.575 casos confirmados y 1.445 personas fallecidas entre confirmados y probables (Ministerio de Salud Pública, 2020).

Desde el fin del estado de excepción hasta el 22 de septiembre a nivel nacional se registraron 4.490 nuevos casos, es decir un promedio diario de 499 infectados. Los cantones que concentran más casos de COVID-19 son: Quito, Guayaquil, Cuenca, Loja, Santo Domingo, Ambato y Portoviejo. Mientras tanto en la provincia de Cotopaxi se contabilizan 3.355 infectados y 254 defunciones entre confirmados y probables (MSP, 2020).

Luego de la suspensión de actividades académicas en el marco del coronavirus a mediados de marzo y posterior inicio de clases virtuales, el comité de operaciones de Emergencia (COE) nacional autorizó el pasado 1 de octubre a cinco universidades del Ecuador el retorno progresivo de clases presenciales, en base a un plan piloto. Según la secretaria de educación superior, ciencia, tecnología e innovación (SENESCYT) hay 850.000 personas que conforman el sistema de educación superior de los cuales 750.000 son estudiantes y el resto docentes y personal administrativo (SENESCYT, 2020).

De su lado la Universidad de las Fuerzas Armadas ESPE en el 2019 contaba con 32.383 estudiantes matriculados y 1.616 docentes y personal administrativo, solo la Sede Latacunga contaba 305 servidores públicos entre docentes y personal administrativo y 4.510 estudiantes (Universidad De Las Fuerzas Armadas ESPE, 2019).

Dada la cantidad de personas vinculadas a la Universidad de las Fuerzas Armadas Sede Latacunga es de vital importancia tener una aplicación web que ayude al

personal médico al registro y detección temprana de personas con posibles síntomas de COVID-19 debido a que podría ser clave para evitar una expansión del virus entre su personal y a su vez permitiría tener una evaluación diaria del paciente poniendo especial atención en los pacientes que tengan síntomas relacionados al COVID-19 logrando así una optimización de los recursos disponibles puesto que el personal médico solo se enfocaría en personas que realmente necesiten una atención y además protegería al personal de salud de un posible contagio.

1.3. Justificación e Importancia.

Actualmente, por la situación que atraviesa el mundo, los sistemas de atención médica están saturados, este dilema ocurre a nivel mundial, la capacidad de brindar servicio no solo a los afectados por el COVID-19, sino también a pacientes que padecen otras enfermedades agudas y crónicas, al mismo tiempo que se protege a los médicos, enfermeras y otro personal de salud aliado. Es por ello que en países como Estados Unidos, los sistemas de salud recurren a la telemedicina, aplicaciones software que permiten la comunicación entre un paciente y el doctor, facilitando la atención de las personas desde sus hogares, lo que ha evitado que las personas se contagien del COVID-19. (Bashshur, Doarn, Frenk, Kvedar, & Woolliscroft, 2020).

La Universidad de las Fuerzas Armadas ESPE Sede Latacunga, a menudo busca contribuir con conocimiento, tecnología y el desarrollo de aplicaciones que permitan mitigar los problemas relacionados con la crisis sanitaria producida por el virus, el presente proyecto contribuye, a través de una aplicación web, con el registro y seguimiento de personas con sospecha de coronavirus, mediante un test de signos y síntomas que permiten determinar el estado de salud de los pacientes desde donde se encuentren (su hogar), además, el sistema permitirá realizar una tele asesoría medica

por medio de video llamadas, entre la persona que solicita atención y el doctor, evitando el contacto físico entre personas con el fin de no propagar el virus, la aplicación también proporcionará datos para la toma de decisiones, como: evolución de los síntomas de los pacientes, datos estadísticos y un mapa de calor para proveer de indicadores del avance del virus.

La comunidad universitaria (docentes, alumnos, administrativos, militares, directivos etc.) podrá beneficiarse del presente proyecto (aplicativo) al detectar personas con sospecha de coronavirus (detección temprana), lo que permitirá la toma de decisiones de forma rápida y oportuna para evitar la cadena de contagio de la enfermedad. Además, una vez implantado, el valor agregado del aplicativo será contribuir en la creación de una base de datos (dataset) sobre el seguimiento y evolución de pacientes con y sin coronavirus, teniendo como finalidad servir de base para la realización de pruebas, simulaciones, predicciones, etc., con datos reales de la enfermedad COVID-19.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de COVID-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.

1.4.2. Objetivos Específicos

- Elaborar el marco teórico sobre el estado del arte en general sobre el COVID-19.
- Elaborar el marco teórico sobre el uso de una arquitectura limpia y buenas prácticas de desarrollo de software.

- Desarrollar una aplicación web utilizando tecnologías modernas, con arquitectura limpia y buenas prácticas de desarrollo de software para permitir la escalabilidad y el mantenimiento.
- Validar la aplicación web mediante pruebas de funcionalidad y la valoración de expertos.

1.5. Hipótesis

¿Si se desarrolla una aplicación web se apoya el seguimiento del estado de salud de personas expuestas al contagio de COVID-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga?

1.6. Variables de la investigación.

- Variable Independiente: Se desarrolla una aplicación web.
- Variable Dependiente: Apoyo al seguimiento del estado de salud de personas expuestas al contagio de COVID-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.

1.6.1. Indicadores

- Porcentaje de personas atendidas(asesoradas) vía remota.
- Porcentaje de seguimientos realizados.
- Índice de satisfacción del usuario.

Capítulo II

Marco teórico

2.1. Introducción del capítulo

En el capítulo siguiente se presenta información relevante que fundamentará teóricamente el trabajo de investigación y se lo ha dividido en los siguientes apartados: antecedentes históricos, antecedentes conceptuales, antecedentes referenciales y antecedentes contextuales. En estos apartados se tratan diversos temas, que van desde los coronavirus y su origen, pasando por las principales arquitecturas y los principales elementos de las arquitecturas limpias, las buenas prácticas de desarrollo de software, hasta cuestiones generales concernientes a la pandemia, la metodología utilizada con sus principales elementos, las tecnologías que se utilizarán en este proyecto y las aplicaciones software que se han implementado en el país para enfrentar la pandemia.

2.2. Antecedentes históricos

Los coronavirus son una familia de virus que producen las enfermedades que afectaron a la población mundial en noviembre del 2002 con el síndrome respiratorio agudo severo (SARS) en China y en el 2012 con síndrome respiratorio de Oriente Medio (MERS) el cual apareció por primera vez en Arabia Saudita. (Stadler, y otros, 2003). SARS fue notificado a la OMS por primera vez como un brote de neumonía desconocido en Guangdong provincia del sur de China, sin embargo, entre el 16 de noviembre de 2002 y el 9 de febrero de 2003 el virus se había extendido a diferentes partes del mundo, arrasó 29 países, afectó a 8.098 personas, dejó 774 pacientes muertos y casi paralizó la economía asiática en los 114 días de pandemia, las altas medidas de cuarentena impuestas lograron una pausa temporal de la pandemia. (Stadler, y otros, 2003).

A pesar de ser rápidamente contenido, logro traspasar las fronteras y convertirse en una pandemia a nivel mundial, transmitiéndose mediante las partículas de saliva que están presentes en el aire cuando una persona infectada tose, estornuda o habla. No se ha producido ninguna nueva transmisión del virus desde 2004. (Mayo Clinic, 2020).

MERS virus zoonótico, transmitido por dromedarios infectados en África, Asia Meridional y Oriente Medio, aunque también es transmitido de persona a persona, en total, 27 países han notificado casos desde 2012. Cabe señalar que el 35% de pacientes con MERS han muerto. (OMS, World Health Organization, 2020).

La enfermedad del coronavirus(SARS-CoV-2) denominada COVID-19 que surgió en el 2019, en la ciudad de Wuhan, China, provocando un brote de neumonía atópico. Se expandió rápidamente por el mundo a causa de su capacidad de transmisión y por los diferentes medios de transporte superando las cifras del SARS y MERS en número de personas infectadas y áreas de epidemia.(Hu, Guo, Zhou, & Shi, 2020).

A finales de diciembre del 2019, en el mercado de Huanan Seafood Wholesale Market en Wuhan que se encargaba de la venta de mariscos y otros animales tales como aves de corral y vida silvestre, se encontraron los primeros casos de COVID-19. (Hu, Guo, Zhou, & Shi, 2020).

Hasta el 17 de marzo de 2021, se reportan 120,383,919 casos confirmados de COVID-19, incluidas 2,664,386 víctimas mortales en todo el mundo. (OMS, 2021)

2.3. Antecedentes Conceptuales y Referenciales

El presente proyecto para cumplir con su objetivo, hace uso de diferentes tecnologías, métodos y técnicas que en conjunto permiten obtener el resultado esperado, con lo cual se desea resolver el problema planteado. A continuación, se describe tanto la parte del problema como la técnica, para un mejor entendimiento del desarrollo del proyecto.

2.3.1. *Conceptos relacionados al problema*

Pandemia. Según la OMS una pandemia se da cuando se propaga mundialmente una nueva enfermedad (Organización mundial de la Salud, 2010).

Coronavirus. Los coronavirus pertenecen una gran familia de virus que pueden provocar enfermedades en animales inclusive hasta en seres humanos. Se conoce que ciertos coronavirus pueden causar infecciones respiratorias en humanos, que van desde un simple resfriado hasta enfermedades mucho más graves como el síndrome respiratorio agudo severo (SRAS) y el síndrome respiratorio de Oriente Medio (MERS). La enfermedad causada por coronavirus COVID-19 se provocada por el coronavirus descubierto más recientemente. (Organización Mundial de la Salud[OMS], 2020).

Covid-19. La enfermedad infecciosa provocada por el coronavirus se le conoce como la COVID-19. Los síntomas más comunes son fiebre, tos seca y cansancio, aunque se pueden presentar otros como náuseas, vómito, diarrea, escalofríos, dolor de cabeza entre otros. Aproximadamente el 80% de personas contagiadas se recuperan sin recibir tratamiento hospitalario, el 15% presenta síntomas graves y el 5% desarrollan un cuadro clínico crítico. (Organización Mundial de la Salud[OMS], 2020; Yuen, Ye, Fung, Chan, & Jin, 2020).

Tele asesoría. Se distinguen dos tipos de tele asesoría, la primera es la tele asesoría **asincrónica** para monitorear y entregar retroalimentación por medios de mensajería tales como el correo, correo automatizado, teléfono celular u otros dispositivos que no requieran el contacto cara a cara. La segunda es la tele asesoría **sincrónica** que implica el contacto cara a cara, en tiempo real a través de video llamadas (audio y video), con esto se permite la comunicación del paciente con el médico, uno a uno, o incluso varios simultáneamente. (Verhoeven, Tanja-Dijkstra, Nijland, Eysenbach, & Van Gemert-Pijnen, 2010).

Videollamadas. “La videoconferencia es una tecnología que proporciona un sistema de comunicación, diseñado para el intercambio de información mediante la interacción visual, auditiva y verbal en tiempo real con personas que se encuentran geográficamente distantes.” (Cuevas Valencia & Adame Añorve, 2013).

2.3.2. Conceptos relacionados a la metodología

En el presente proyecto se utilizó una metodología de software que toma como base Scrum, los principios de la programación orientada a objetos(POO), el principio de la arquitectura limpia de Robert Cecil Martin entre otros. A continuación, se describen conceptos acerca del proceso de ingeniería de software:

Software. Son los programas con su documentación y configuración asociada que permite que estos funcionen de la forma esperada, son desarrollados para un cliente en particular o para un mercado general. (Sommerville, 2005).

Ingeniería de software. Según (Pressman, 2005): “La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad.”

Metodología. Una metodología según la Real Academia Española(RAE) es un: “Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.”.

En la ingeniería de software, una metodología es considerada como un proceso que establece el orden de las actividades, organización y como se deben realizar las tareas e incluso la parte intelectual. Para lograr mejores aplicaciones, proceso de desarrollo y estandarizar la forma en la que se va a trabajar en una organización. (García Peñalvo, García Holgado, & Vázquez Ingelmo, 2020).

Método. “Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.” (Oxford Languages, 2020).

Proceso. “Un proceso define quién hace qué, cuándo y cómo, para alcanzar cierto objetivo.” (Rumbaugh, como se citó en (Pressman, 2005)).

Ciclo de Vida del software. “El ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo. Esto sí lo debe indicar la metodología”. (García Peñalvo, García Holgado, & Vázquez Ingelmo, 2020).

Modelo de proceso incremental. Busca proporcionar un incremento al final de cada iteración, la cual está compuesta por fases: comunicación, planeación, análisis y diseño, codificación y pruebas, despliegue. (Pressman, 2005)

Se aplica este modelo cuando se requieren entregas tempranas de un software, esto aporta valor al cliente y permite administrar riesgos técnicos. (Pressman, 2005) A continuación se enumeran las fases del modelo del proceso incremental:

Comunicación: La comunicación tiene que llevarse a cabo con el acto de escuchar y entender la necesidad del usuario a través de notas, video llamadas para fomentar el cara a cara, etc. Esto se lo hace con el fin de establecer las metas y objetivos del software a realizar. (Pressman, 2005)

Planeación: Se tiene dos enfoques de planeación, la minimalista que indica que un plan no sería necesario, por el cambio que se produce en el desarrollo de software. Tenemos la tradicional que indica la necesidad de un plan para enrumbar al equipo, mientras más detallado mejor. Por otro lado, también tenemos la ágil que indica que un plan rápido es necesario y que se ajustará a medida que avance el trabajo real. (Pressman, 2005)

Modelado: En esta fase se realiza el **análisis** como modelos de requerimientos donde se presentan las necesidades del cliente. También en esta fase se incluye el **diseño** que ayuda al equipo de profesionales a entender las características del software a través de arquitectura, interfaz de usuario y detalle a nivel de componentes. (Pressman, 2005)

Construcción: En esta fase se realiza la **codificación** donde se escribe el código fuente en un lenguaje de programación acorde con las necesidades del proyecto. También tenemos las **pruebas** que se realizan a un componente específico como prueba unitaria, las de integración para probar cómo funcionan los componentes juntos, pruebas de validación que evalúan si los requerimientos fueron satisfechos y las pruebas de aceptación que son efectuadas por el cliente. (Pressman, 2005)

Despliegue: Esta fase consiste en la entrega de incrementos y del producto final, para luego brindar la asistencia y retroalimentación. (Pressman, 2005)

Metodología Ágil. Es un conjunto de métodos y metodologías que permiten a un equipo pensar de forma más efectiva para trabajar más eficientemente y tomar las mejores decisiones. (Stellman & Greene, 2015).

Una filosofía con lineamientos de desarrollo:

Filosofía:

- Satisfacción del cliente
- Entrega rápida de software
- Incremental
- Equipos pequeños y motivados
- Productos del trabajo con mínima ingeniería de software
- Sencillez en el desarrollo

Lineamientos de desarrollo de una metodología ágil:

- Énfasis en la entrega sobre el análisis y el diseño
- Comunicación activa entre desarrolladores y con el cliente(s)

(Pressman, 2005)

La intervención del cliente es fundamental en el desarrollo ágil, el prioriza y propone los nuevos requerimientos que van a ser incluidos en cada incremento, así también evalúa los mismos en cada iteración. Es necesario conocer las habilidades del equipo de desarrollo para que se aprovechen de la mejor manera bajo sus propias formas de trabajo sin procesos establecidos. Esperar a que cambien los requerimientos, de tal forma que el diseño se adapte a estos cambios. Eliminar la complejidad y mantener la simplicidad tanto en el software como en el proceso. (Sommerville, 2005)

SCRUM. Scrum es marco en el cuál un equipo multidisciplinario aborda problemas complejos, y a su vez entregan productos de forma continua creando el mayor valor posible. (Scrum, 2020).

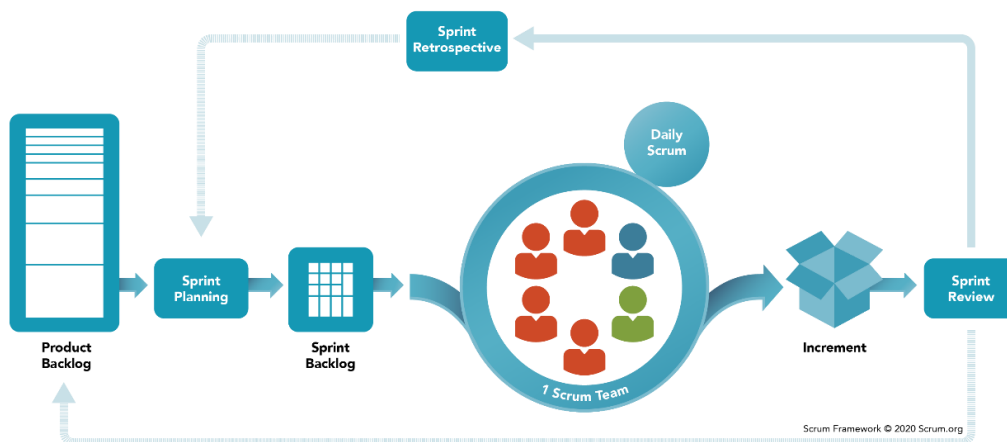
Scrum se basa en la experiencia y en la observación por lo que sus pilares son la inspección, adaptación y transparencia.

Como se puede observar en la Figura 2, Scrum requiere un Scrum Master para fomentar un entorno donde:

- El Product Owner es el encargado principalmente de ordenar el trabajo de un problema en un Product Backlog.
- El Equipo Scrum transforma una parte específica del trabajo en un Incremento de valor al finalizar el Sprint.
- El equipo Scrum y sus partes interesadas evalúan los resultados obtenidos en el sprint y se preparan para el siguiente.

- Repetir

(Scrum, 2020)

Figura 2*Marco Scrum**Nota.* (Scrum, 2020)**Valores de Scrum**

- **Compromiso:** Lograr objetivos y el apoyo mutuo.
- **Enfoque:** En trabajar en el sprint.
- **Apertura:** Abiertos para el trabajo y desafíos.
- **Respeto:** Mutuo sobre la capacidad e independencia.
- **Coraje:** Para el trabajo difícil.

Equipo en Scrum. Equipo pequeño, multifuncional, auto gestionable que son responsables de todas las actividades relacionadas con el producto, a través de sprints para crear incrementos útiles y está compuesto por desarrolladores, product owner (dueño del producto) y scrum master.

- **Desarrolladores:** Crear incrementos útiles en cada sprint y adaptar su plan hacia el objetivo del sprint.

- **Product Owner:** Desarrollar y comunicar el producto objetivo, crear y comunicar los elementos que se incluirán en el product backlog¹ y asegurarse de que sea transparente, visible y entendible.
- **Scrum Master:** Lidera el equipo hacia la gestión ágil con Scrum dentro del proyecto.

“Asegurarse de que todos los eventos de Scrum tengan lugar y sean positivos, productivos y se mantengan dentro del plazo.”

(Scrum, 2020)

Eventos de Scrum. Son la oportunidad de realizar la inspección con el fin de ser transparentes y reducir la complejidad y son los siguientes sprint, sprint planning, daily scrum, sprint review y sprint retrospective. (Scrum,2020)

Sprint: Un sprint es el contenedor de los eventos y es el núcleo del Scrum.

- Las ideas se convierten en valor, con duración fija.
- Cada sprint comienza al finalizar el anterior.
- “Todo el trabajo necesario para lograr el objetivo del producto, incluida la planificación de Sprint, Scrums diarios, Revisión de Sprint y Retrospectiva de Sprint, ocurre dentro de Sprints.” (Scrum, 2020)
- Un sprint se vuelve obsoleto y se cancela por el Product Owner si su objetivo cambia.

Sprint Planning:

- Es creado en colaboración con todo el equipo.

¹ El product backlog es una lista de tareas que se tienen que realizar durante todo el desarrollo de un proyecto.

- Se establece el trabajo para un sprint.
- Se define que elementos del backlog pasan a ser ejecutados o desarrollados en el sprint y que será entregado como incremento resultante.

Daily Scrum:

- Reunión diaria fija, donde el equipo de desarrollo se reúne por 15 minutos.
- De ser posible, es mejor en el mismo lugar y la misma hora.
- Se revisa sobre lo que se logró el día anterior, lo que se realizará el día actual y que problemas u obstáculos impiden avanzar.

Sprint Review:

- Inspección del incremento.
- Planificar formas de aumentar la calidad y la eficacia.
- La información obtenida de esta reunión afecta al backlog.

Sprint Retrospective:

- El Equipo Scrum analiza el último Sprint, se enfoca en las personas, las interacciones, los procesos, las herramientas y su definición de terminado. El Equipo Scrum examina qué sus aciertos durante el Sprint, qué obstáculos y problemas encontró y cómo fueron resueltos (o no). (Scrum, 2020)

Artefactos de Scrum. Representan trabajo o valor, su compromiso es garantizar que se proporcione información que mejore la transparencia y el enfoque frente al cual se pueda medir el progreso. Y son: product backlog, sprint backlog e incremento.

- **Product Backlog:** El progreso se mide a través del producto objetivo.
 - Fuente principal de información de un producto.
 - Es gestionado y priorizado por el product owner.
 - Se realizan actividades de refinamiento (Dividir y definir los elementos a elementos más pequeños).
- **Sprint Backlog:** El progreso se mide a través del sprint objetivo.
 - Elementos tomados del product backlog que se trabajaran en un sprint.
 - Es gestionado por el equipo de desarrollo
 - Se visualiza que elementos no se han empezado a desarrollar, los que están en desarrollo y cuales han sido finalizados.
- **Incremento:** El progreso se mide a través de la definición de terminado.
 - Es el resultado de un sprint.
 - Debe ser utilizable, funcional.
 - Puede haber varios incrementos en un Sprint.

(Scrum, 2020)

Arquitectura de software. Es la forma en que los distintos componentes se integran y coexisten, relacionándose de tal forma que permiten: Analizar el diseño y si este logra cumplir con los requerimientos, realizar cambios en una etapa temprana, reducir riesgos asociados con el software. (Pressman, 2005)

“La arquitectura es un modelo para un sistema. Proporciona una abstracción asociada para gestionar la complejidad del sistema y establecer un mecanismo de comunicación y coordinación entre los elementos. Define una resolución estructurada para satisfacer todas las necesidades técnicas y operativas, optimizando los atributos de calidad comunes como rendimiento y seguridad. Además, implica una colección de selecciones serias con respecto a la organización asociada con el desarrollo del sistema de software y cada una de esas selecciones tendrá un impacto sustancial en la calidad, la capacidad de mantenimiento, el rendimiento y el éxito general del producto final.” (Jaiswal, 2019).

La arquitectura de software está estrechamente relacionada con los paradigmas de programación ya que un paradigma implica estructura y el patrón de programación. Tenemos tres paradigmas de programación estos son: la programación estructurada, programación funcional y la programación orientada a objetos. Estos tres paradigmas se alinean con las tres preocupaciones de toda arquitectura, función, separación de componentes y administración de datos. (Martin, 2018).

Paradigmas de Programación. Según la Real Academia de la Lengua un paradigma es un ejemplar o modelo base para resolver un problema, con esto un paradigma de programación es un modelo de desarrollo de programas. A continuación, se definen paradigmas de programación:

Programación Estructurada. Se basa en dividir un problema en problemas más pequeños por medio de funciones o subrutinas. Esto quiere decir que se toma una declaración de problema grande y descomponerla en funciones probables y estas a su vez se pueden descomponer en funciones de nivel inferior y así sucesivamente. (Martin, 2018)

Programación Funcional. Utilizar las funciones para abstraer los flujos de control y operaciones en los datos con la finalidad de disminuir los cambios de estado en la aplicación. Esto se logra cuando las funciones se combinan unas con otras para brindar comportamiento adicional. (Atencio, 2016)

Programación Orientada a Objetos. Un camino para modelar el mundo real a través de la combinación de datos y funciones, basándose en tres conceptos: Encapsulación, herencia y polimorfismo. (Martin, 2018)

Conceptos Base de la Programación Orientada a Objetos(POO). Entre los conceptos base que forman parte fundamental de la programación orientada a objetos tenemos:

Encapsulación. Ocultar funciones y datos del resto, limitando el acceso o restringiendo el mismo. Un ejemplo de encapsulamiento es la clase que engloba métodos y propiedades. (Martin, 2018)

Herencia. “La herencia es simplemente volver a declarar un grupo de variables y funciones dentro de un ámbito circundante.” (Martin, 2018)

Polimorfismo. En programación orientada a objetos consiste en la habilidad de un objeto de actuar como si fuese otro. Por ejemplo: el objeto de una clase A que tiene subclases B, C, este objeto se comporte como si fuese un objeto de cualquiera de las subclases, puede ser B o C. (Hall, 2017)

Principios SOLID. Los principios SOLID son cinco y permiten escribir código limpio, robusto y de fácil mantenimiento para lograr crear software eficaz y escalable a funcionalidades. A continuación, se describen los cinco principios:

Single Responsibility Principle (Principio de Responsabilidad única). “Un módulo debería tener una, y solamente una razón para cambiar” (Martin, 2018)

Un módulo puede ser un archivo o un set de funciones y estructura de datos que tienen cohesión. Al tener cohesión significa que el código del módulo mantiene una relación con el actor siendo este un grupo de usuarios o stakeholders(interesados). (Martin, 2018)

Open/Close Principle (Principio Abierto/Cerrado). “Un artefacto de software debería ser abierto para extensiones, pero cerrado para modificaciones” (Martin, 2018)

Abierto a extensiones, un artefacto debe ser extendido en comportamiento a medida que los requerimientos cambien. Al ser cerrado a modificaciones no cambia el código fuente del código, este permanece intocable. (Martin, 2018)

Liskov Substitution Principle (Principio de Sustitución de Liskov). “Si para cada objeto o1 de tipo S existe un objeto o2 de tipo T tal que para todos los programas de tipo P definidos en términos de T, el comportamiento de P es no cambiado cuando o1 es sustituido por o2 entonces S es un subtipo de T” (Martin, 2018)

Dado T el tipo base, por ejemplo, una interface o clase abstracta con métodos que pueden ser sobre escritos y siendo S el subtipo que hereda del tipo T, un cliente tiene la referencia de T y no debe saber a qué subtipo está llamando, comportándose de forma independiente de la instancia de S que se le asigne. (Hall, 2017)

Interface Segregation Principle (Principio de Segregación de interfaces). “Una interfaz representa el límite entre un comportamiento requerido por un cliente y como se lo implementa.” (Hall, 2017)

Este principio busca que las interfaces sean pequeñas y que los métodos y propiedades a implementarse no queden sin su respectiva implementación. Por tal motivo se divide la funcionalidad con la creación de interfaces más

pequeñas y con una sola responsabilidad, así se implementa solo lo necesario.
(Hall, 2017)

Dependency Inversion Principle (Principio de Inversión de Dependencia).

Busca que el cliente dependa de abstracciones y no de clases concretas, esto se logra a través de interfaces, el cliente depende de las interfaces, y las clases o tipos que las implementan dependen de estas mismas interfaces. (Hall, 2017)

Arquitectura limpia. “El objetivo de una arquitectura de software es minimizar los recursos humanos requeridos para construir y mantener el sistema requerido.” (Martin, 2018)

El concepto de Arquitectura Limpia nace de Robert C Martin, como una arquitectura orientada a las reglas de negocio e influenciada por otras arquitecturas como son:

Arquitectura Hexagonal. (también conocida como Puertos y Adaptadores), desarrollada por Alistair Cockburn, y adoptado por Steve Freeman y Nat Pryce en su maravilloso libro Desarrollo de software orientado a objetos con pruebas.

La Arquitectura Hexagonal plantea que el dominio sea el núcleo central de las capas y que esté desacoplado de las capas externas. Haciendo uso de interfaces o puertos en lugar de utilizar directamente el principio de inversión de dependencias. (Salguero, 2018).

BCE. presentado por Ivar Jacobson de su libro Ingeniería de Software Orientado a Objetos: un enfoque basado en casos de uso. BCE (Limite – Controlador - Entidad), es una variación del patrón Modelo-Vista-Controlador. Entidades son objetos que representan datos del sistema, Limites son objetos que interactúan con los actores del sistema, Controladores son objetos que trabajan entre límites y entidades. (Pearce, s.f.)

Onion Architecture. por Jeffrey Palermo. En esta arquitectura todo el acoplamiento es hacia dentro, todo el código depende de capas más centrales, pero no depende de las capas más alejadas del centro. Siendo la capa de dominio la más central. En los bordes vemos interfaces visuales, Infraestructura y Pruebas. La capa exterior está reservada para cosas que cambian con frecuencia. (Palermo, 2008)

Características de una arquitectura limpia:

- **Independiente de los frameworks:** No ser dependiente, o tener limitaciones por frameworks.
- **Testable:** Permitir realizar pruebas a las reglas de negocio, sin depender de interfaces, bases de datos u otros componentes que no sean esenciales.
- **Independiente de la interfaz:** La interfaz debe ser fácilmente intercambiable sin afectar a las reglas de negocio.
- **Independiente de la base de datos:** La base de datos debe ser fácilmente intercambiable sin afectar a las reglas de negocio.

- **Independiente de cualquier entidad externa:** Las reglas de negocio no deben verse afectadas por nada del exterior.(Martin, 2018)

Capas de una Arquitectura limpia. La arquitectura limpia separa dos capas principales, que son:

- **Dominio:** Entidades y casos de uso. Esenciales para las reglas de negocio.
 - Entidad.** Son objetos que encapsulan las reglas de negocio, son la parte central del software y de la cual dependen las demás capas.
 - Abstracción.** Son las interfaces o clases abstractas que requiere un interactor y tienen que ser implementadas por las clases concretas.
 - Interactor.** Encapsula e implementa las historias de usuario. Permite el flujo de información desde y hacia las entidades, el cambio en el interactor no afecta a las entidades.
- **Detalle:** Comprende frameworks, interfaz de usuario y base de datos. Componentes no esenciales para las reglas de negocio. Implementa las interfaces o clases abstractas que expone el dominio.

(Martin, 2018)

2.3.3. Conceptos relacionados a las tecnologías utilizadas

A continuación, se describen las herramientas software de apoyo utilizadas para la gestión del proyecto y los lenguajes de programación, frameworks, etc, para el desarrollo del producto software.

Aplicación Software. “En informática, se entiende por software de aplicación, programas de aplicaciones o en algunos casos aplicaciones, al conjunto de los programas informáticos generalmente instalados en el sistema por el usuario, y diseñados para llevar a cabo un objetivo determinado y concreto, de tipo lúdico, instrumental, comunicativo, informativo, etc.” (Raffino, 2020)

Aplicación web. Las aplicaciones web se ejecutan desde cualquier navegador y son accesibles en cualquier dispositivo y también algo que las caracteriza principalmente es que a diferencia de lo que sucede con las aplicaciones nativas, no requieren descarga o instalación alguna. (Yeeply, 2018)

Backend Software. Es la entidad de un producto software responsable de recibir las solicitudes de un cliente(frontend), y manejarlas en servicios en la nube o servidores dedicados. El backend de un producto puede ser alojado en diferentes servidores en la nube, tal es el caso de empresas que prestan estos servicios como: Google Cloud, Amazon, Azure de Microsoft, entre otros. El backend se encarga de recibir las solicitudes, procesarla llevando a cabo una operación y responder al cliente. (Filipova & Vilao, 2018)

Frontend Software. Corresponde a la parte visible para el usuario final, también es considerado como cliente,esto quiere decir cualquier aplicación sea que se ejecute en un navegador o en un dispositivo móvil y que exponga una interfaz gráfica o incluso una línea de comandos. El frontend envía peticiones al backend y muestra el resultado al usuario. (Filipova & Vilao, 2018)

API. “Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de aplicaciones. Suele considerarse como el contrato entre un proveedor de información y un usuario, donde se establece el contenido que se requiere del consumidor (la llamada) y el que necesita el productor (la respuesta).” (RedHat, 2020)

“Una API es la interfaz que un programa de software presenta a otros programas, a los humanos y, en el caso de las API web, al mundo a través de Internet. Siendo los bloques de construcción que permiten la interoperabilidad para las principales plataformas comerciales en la web.” (Jin, Sahni, & Shevat, 2018)

API Rest. “Una API de transferencia de estado representacional (REST), o API de RESTful, es una interfaz de programación de aplicaciones que se ajusta a los límites de la arquitectura REST.” (RedHat, 2020)

REST es un conjunto de principios de arquitectura que se puede implementar según sea necesario, se usa en una arquitectura cliente-servidor compuesta de clientes, servidores y recursos (Un recurso es una entidad que se puede identificar, nombrar, direccionar o manejar en la web.) (Jin, Sahni, & Shevat, 2018), con la gestión de solicitudes a través de HTTP (Protocolo de Transferencia de Hipertexto). (RedHat, 2020)

Expone datos como recursos y utiliza métodos HTTP estándar para representar transacciones de Crear (POST), Leer (GET), Actualizar (PATH,

PUT) y Eliminar (DELETE), que serían las operaciones en inglés Create, Read, Update, Delete (CRUD), estas se usan contra los recursos. Por ejemplo, la API de Stripe representa a los clientes, cargos, saldos, reembolsos, eventos, archivos y pagos como recursos. (Jin, Sahni, & Shevat, 2018)

Los recursos son parte de la url por lo tanto en el caso de un software que maneje clientes, para obtener un cliente el método sería: Método GET con la url: /cliente/5

Esto nos devuelve los recursos(datos) solicitados en este caso el cliente con valor o identificador 5.

API graphql. A GraphQL se lo puede definir como un lenguaje de consultas para API, que brinda una descripción completa y fácilmente comprensible de los datos dentro del API, además como brinda a los clientes la capacidad de solicitar únicamente lo que necesitan, facilita la escalabilidad de las API con el tiempo y brinda potentes herramientas de desarrollo. (Graphql, 2020)

“Ha ganado una tracción significativa recientemente. Fue desarrollado internamente por Facebook en 2012 antes de ser lanzado públicamente en 2015 y ha sido adoptado por proveedores de API como GitHub, Yelp y Pinterest.

GraphQL permite a los clientes definir la estructura de los datos requeridos y el servidor devuelve exactamente esa estructura.” (Jin, Sahni, & Shevat, 2018)

Una consulta de datos a un software que maneja clientes, se define de la siguiente manera:

```
query ($id:String!) {  
  cliente(login: $id){  
    nombre
```



```

        compania
        creadoEn
    }
}

```

Donde cliente es una función de consulta que recibe como parámetro login que es de tipo id, y retorna los datos {nombre, compañía, creado En}.

Diferencias entre API REST y API GraphQL

API REST:

- Expone los datos como recurso, usa verbos HTTP para representar operaciones CRUD.
- Verbos HTTP que utiliza (GET, POST, PUT, PATCH, DELETE).
- Métodos estándar, códigos de estado por ejemplo (200 petición exitosa).
- Fácil de mantener.
- Como desventajas tiene: Varios viajes de ida y vuelta HTTP (múltiples rutas url), grandes cargas de datos.
- Es usado para APIs con operaciones similares a un CRUD.

API GraphQL:

- Lenguaje de consulta para APIs.
- El lado del cliente define qué estructura en la respuesta desde que el servidor le envíe.
- Verbos HTTP que utiliza (GET(queries) – POST(mutations)).
- Ahorra viajes de ida y vuelta en las peticiones (una sola ruta).
- Tamaño de carga útil más pequeña.

- Trabaja con tipos de datos.
- Como desventajas tiene: Requiere procesamiento adicional para análisis de consultas, se dificulta la optimización de rendimiento del backend y para aplicaciones pequeñas es demasiado complicado.
- Es usado para APIs que necesitan ser flexibles y para el mantenimiento consistente.(Jin, Sahni, & Shevat, 2018)

Agora.io. “El Kit de desarrollo de video de Agora es usado para la integración de chat de video en tiempo real para aplicaciones web, móvil y nativo.” (Agora.io, 2021)

Javascript. “Es un lenguaje de programación ligero, interpretado, o compilado [...]. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB and Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo, programación funcional).” (Mozilla, 2020)

Typescript. TypeScript es un lenguaje de código abierto que se basa en JavaScript, al agregar definiciones de tipos estáticos. Proporciona una mejor documentación y validación de código. El código javascript también es válido en typescript pero sin el control de errores, aunque también existe un modo estricto. (TypeScript, 2020)

“El código TypeScript se transforma en código JavaScript a través del compilador TypeScript o Babel. Este JavaScript es un código limpio y simple que se ejecuta en cualquier lugar donde se ejecute JavaScript: en un navegador, en Node.JS o en sus aplicaciones.” (TypeScript, 2020)

Html. HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el elemento fundamental de la Web. Permite definir el significado y la estructura del contenido web. Hipertexto hace referencia a los enlaces que posibilitan la navegación entre varias páginas web, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son una característica esencial de la Web. (Mozilla, 2020)

Html utiliza marcas para indicar un texto, párrafos, enlaces u otros.

```
<h1>Este es un título</h1>
```

```
<p>Este es un párrafo</p>
```

Este contenido se mostrará en el navegador el primero “h1” como un título con letras de tamaño grande, y el segundo como un párrafo “p” con letras más pequeñas.

Css. “Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML).” (Mozilla, 2020)

“Es un mecanismo simple para agregar estilo (por ejemplo, fuentes, colores, espaciado) a los documentos web.” (World Wide Web Consortium, 2020)

Node.js. “es un entorno de ejecución multiplataforma en JavaScript que permite a los desarrolladores construir aplicaciones del lado del servidor y de red.” (Mozilla, 2020)

Angular. “Es un framework Javascript potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. El tipo de aplicación Javascript que se desarrolla con Angular es del estilo SPA (Single Page Application – Aplicaciones de una sola página) o también las denominadas PWA (Progressive Web App – Aplicaciones Web Progresivas que pueden ejecutar funciones nativas).” (DesarrolloWeb, 2020)

Mongodb. Es un motor de base de datos distribuida, documental y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Lo que significa que almacena datos en forma de documentos tipo JSON. (MongoDB, 2020)

Firebase. “Base de datos de documentos NoSQL completamente administrada para el desarrollo de aplicaciones web y móviles. Está diseñado para almacenar y sincronizar fácilmente los datos de la aplicación a escala global.” (Firebase, 2020)

Sus características clave son:

- Documentos y colecciones con consultas potentes.
- SDK de iOS, Android y Web con acceso a datos sin conexión.
- Sincronización de datos en tiempo real.
- SDK de servidor Node, Python, Go y Java (Firebase, 2020)

Mongoose. “Mongoose es un ODM (Object Document Mapper – Mapea documentos a objetos) proporciona una solución sencilla basada en esquemas para modelar los datos de una aplicación. Incluye conversión de tipos incorporada, validación, creación de consultas, enlaces de lógica empresarial y más, listo para usar.” (Mongoose, 2020)

Typegoose. Es una librería javascript que se instala en node.js como una dependencia externa y sirve como un "contenedor" para escribir fácilmente modelos de Mongoose con TypeScript. (Typegoose, 2020)

TypeGraphQL. TypeGraphQL es una librería cuyo proceso es definir el esquema de graphql usando solo clases.(TypeGraphQL, 2020)

2.4. Antecedentes Contextuales

El elevado número de contagios de coronavirus en el país y el lento avance del plan de vacunación del ministerio de salud pública que espera vacunar al 52 % de la población para noviembre del 2021, pero que según un informe de Economist Intelligence Unit, del 28 de enero, que estima que la mayoría de países de ingresos medios, como Ecuador, vacinará a grupos vulnerables hasta finales del 2022, mientras que los más ricos podrán hacerlo entre marzo y julio, hacen pensar que una vuelta a

normalidad no se avizora en el corto plazo. Por su parte la biotecnóloga Jhommara Bautista estima que si se cumplen los envíos de dosis de las farmacéuticas las inmunizaciones en el país culminarían en el 2022 (El Comercio, 2021).

Ante este escenario nada alentador y dada la gravedad de la situación de la cual la ciudadanía fue consiente desde la declaración del estado de excepción el 16 de marzo de 2020 por el gobierno Nacional, se ha generado un gran interés en el gobierno nacional, la empresa privada y la comunidad universitaria, por crear herramientas que ayuden en la lucha contra La COVID-19.

Es por ello que han surgido aplicaciones tanto web como móviles que permiten ayudar en la lucha contra el virus que provoca la COVID-19. Tal es el caso de la aplicación ASÍ Ecuador que fue desarrollada por la empresa ecuatoriana Link y fue implementada por el Ministerio de Salud y Ministerio de Telecomunicaciones, la cual tiene como principal funcionalidad notificar al usuario si una persona que estuvo cerca de él, dio positivo para Covid-19, guardando total anonimato de las personas que la utilizan (Primicias, 2020).

También tenemos la aplicación CovidEC que fue como lo explicó en su momento el ministro de Telecomunicaciones Andrés Michelena tiene como principal objetivo que la población conozca más sobre la sintomatología del coronavirus COVID-19, para evitar la saturación del número 171 y de los hospitales (Teleamazonas, 2020).

Por su parte el sector académico no se ha quedado atrás y ha desarrollado aplicaciones entre los cuales tenemos el proyecto “Chatbot en Telegram para consultas de casos COVID-19 en el Ecuador” el cual es una propuesta de configuración de un

chatbot en la aplicación Telegram, empleando la herramienta NodeRed con el objetivo de que los usuarios (gobierno, instituciones de salud pública y privada, personas en general) consulten estadísticas, por provincia del Ecuador, sobre la pandemia (Prieto-Reinoso & Cabrera-Mejía, 2020)

Capítulo III

Desarrollo de la aplicación web

3.1. Introducción

En el presente capítulo se expone el desarrollo de la aplicación web, partiendo de una lista de requerimientos obtenidos en las reuniones realizadas con un experto con el fin de conocer el problema y las características necesarias que se tienen que implementar en él software.

Como primer apartado se presenta el análisis en el cual se indica cómo se implementaron las bases del marco Scrum, el cual gracias a sus tres pilares transparencia, inspección y adaptación, nos permite trabajar con el experto como parte del equipo el cual suministra información constantemente, también inspeccionar continuamente para evitar el riesgo y adaptarnos a los cambios. A su vez para el diseño y codificación se utilizó los principios de una arquitectura limpia para mejorar el desacoplamiento entre componentes y permitir que el software sea adaptable a nuevos requerimientos a futuro. Por último, se presenta la implementación y pruebas del software en un entorno de producción y sus respectivas vistas en funcionamiento.

3.2. Análisis

En la etapa de análisis tenemos la aplicación de Scrum para gestionar las historias de usuario, los tiempos que requieren desarrollarlas y como llevar a cabo el desarrollo del proyecto de forma ágil. Para lo cual se ha establecido lo siguiente:

Tabla 1*Análisis y planificación de Scrum*

Elemento	Valor	Unidad	Detalle
Tiempo estimado de desarrollo	2	Meses	El tiempo de desarrollo fue establecido por acuerdo mutuo entre el equipo de desarrollo y el experto.
Duración de un sprint	1	Semana	La duración de una semana permite al equipo minimizar el riesgo, gracias a sprints cortos es posible mejorar la comunicación, entregar incrementos y recibir retroalimentación en tiempos cortos.
Velocidad de un sprint	30	Puntos	La velocidad de un sprint la decidió el equipo de desarrollo en base al número de historias y la capacidad técnica del equipo. También se consideró el tiempo desarrollo de 2 meses.
Número de sprints	8	Sprint	El número de sprints resultado de la asignación de puntos a cada historia, el total de puntos de todas las historias de usuario para un sprint tienen que estar dentro del valor de velocidad de un sprint, caso contrario las historias pasan al siguiente sprint.
Daily meetings(Reunión diaria)	15	Minutos	Las reuniones diarias se realizan entre el equipo de desarrollo y su duración es fija. Estas reuniones se realizan a través de video llamadas utilizando Google Meet.
Sprint Planning(Planificación de un sprint)	2	Horas	De acuerdo a la documentación de Scrum es recomendable que para un sprint de 4 semanas se

Elemento	Valor	Unidad	Detalle
			realice un sprint planning de 8 horas. Por lo tanto, al ser de una semana se realiza esta reunión en 2 horas por semana.
Equipo de desarrollo	2	Personas	El equipo se compone de dos desarrolladores uno para Backend y otro para Frontend.
Experto	1	Persona	El experto del proyecto es un profesional de la salud.

En la tabla 1 se indica los elementos utilizados para el desarrollo de todo el proyecto, entre estos elementos tenemos eventos de Scrum, el equipo de desarrollo, el experto y el tiempo total de desarrollo, cada uno de estos elementos tienen un valor que puede ser la duración, los sprints o el número de personas que intervienen, también tenemos la unidad y un detalle que indica el porqué del valor e información extra.

Se usó solo los eventos necesarios para llevar a cabo el proyecto, así como también se asume roles extra dentro del equipo de desarrollo, es decir un desarrollador puede ser el scrum master, y otro desarrollador ser el product owner, se llevó de esta manera por motivo de cantidad de personas dentro del scrum team.

Cómo primera parte del análisis se tomó los requerimientos del experto en formato de historias de usuario de forma general, motivo por el cual resultaron extensas e imposible de desarrollar según lo establecido en la tabla 1. Para resolver esto se las consideró como épicas y como segundo apartado del análisis se realiza la división de estas épicas en historias de usuario más pequeñas y posibles de desarrollar, al tener nuestras historias de usuario lo siguiente fue priorizarlas, por lo cual se creó el backlog

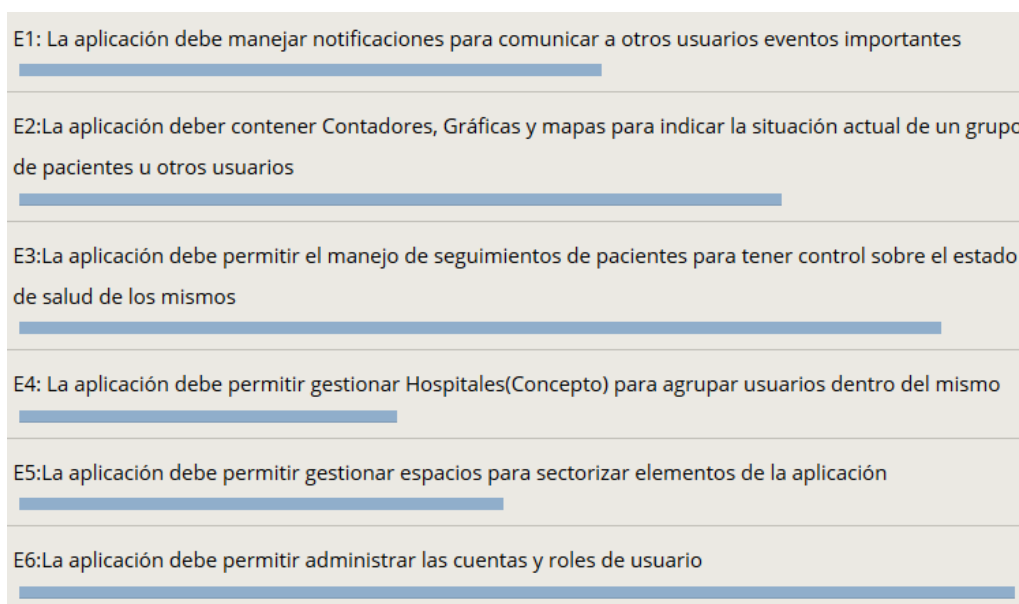
de producto donde las historias que aportan mayor valor al proyecto se encuentran en la cima de la tabla. Finalmente se otorga una puntuación de acuerdo a la secuencia de Fibonacci a cada historia para estimarlas y colocarlas dentro de un sprint.

3.2.1. Épicas

Como se mencionó en la introducción la información suministrada por el experto se traducen a historias de usuario. Las historias de usuario que son demasiado extensas o generales son consideradas como épicas. A continuación, se listan estas épicas:

Figura 3

Épicas con su indicador de duración



Gracias a la herramienta Pivotal tracker², en la parte inferior se puede apreciar una barra que representa el tamaño de cada épica en relación a las demás. Se puede

² Pivotal Tracker es una herramienta colaborativa que nos permite administrar proyectos de forma ágil.

visualizar en la Figura 3 que la épica de mayor tamaño es la de cuentas y roles de usuario y la que sigue es la de seguimientos, esto ocurre porque estas épicas representan la funcionalidad principal de la aplicación web.

Las épicas presentadas en la Figura 3 son historias de usuario extensas que forman parte del modelo de negocio el cual se traduce a la realización de seguimientos de un profesional de la salud a un paciente, la gestión de hospitales, espacios, cuentas de usuario, mapas y gráficas. Estas épicas se dividen en historias de usuario más pequeñas y son listadas a continuación.

3.2.2. *Historias de usuario*

Las tablas a continuación, permiten observar las historias de usuario por épica y sus criterios de aceptación, cada una de las historias está contenida en una épica.

E1: La aplicación debe manejar notificaciones para comunicar a otros usuarios eventos importantes.

Tabla 2

Historias de usuario pertenecientes a la Épica 1

Historias de Usuario

Figura 4

Historia de usuario 1



Como Root, administrador, doctor, paciente quiero consultar una notificación para acceder al enlace de la llamada o a su contenido
administrar notificaciones

Criterios de aceptación:

Al seleccionar la notificación me redirige a la vista de video llamada.

Figura 5

Historia de usuario 2



Como Root, administrador, doctor, quiero crear una notificación para informar al paciente de que se va a realizar una video llamada.
administrar notificaciones

Criterios de aceptación:

La notificación llega al paciente, luego de ser enviada.

Figura 6

Historia de usuario 3



Como ROOT, doctor, paciente, administrador quiero obtener/Listar notificaciones enviadas para mostrarlo en el api graphql (Esto es interno - servidor - y se muestra a través del API)
administrar notificaciones

Historias de Usuario

Criterios de aceptación:

Al consultarse sobre notificaciones enviadas al API GraphQL se obtiene la información de las mismas.

Figura 7

Historia de usuario 4



Como root, administrador, doctor, paciente quiero obtener/Listar mis notificaciones recibidas enviadas por otro usuario para estar informado respecto a lo que me envían otros usuarios.

administrar notificaciones

Criterios de aceptación:

Las notificaciones que recibe un usuario y que son enviadas por otro, pueden ser visualizadas.

Figura 8

Historia de usuario 5



Como root, doctor, paciente, administrador, quiero que las notificaciones se coloquen como vistas para cuando sean leídas estas cambien a ese estado. (Esto es interno - servidor - y se muestra a través del API)

administrar notificaciones

Criterios de aceptación:

El API GraphQL permite el cambio de estado de una notificación ha vista (leída).

Nota. Historias de usuario y criterios de aceptación pertenecientes a la época 1.

E2: La aplicación deber contener Contadores, Gráficas y mapas para indicar la situación actual de un grupo de pacientes u otros usuarios.

Tabla 3

Historias de usuario pertenecientes a la Épica 2

Historias de Usuario

Figura 9

Historia de usuario 6

★ Como Root y director quiero un contador de pacientes por diagnóstico para visualizarlo en la interfaz y como gráfica.
contadores gráficas y mapas

Criterios de aceptación:

Se visualiza en la interfaz un contador y una gráfica con el número de pacientes por diagnóstico.

Figura 10

Historia de usuario 7

★ Como Root y director quiero un contador de pacientes por día y diagnóstico para visualizarlo en la interfaz y como gráfica.
contadores gráficas y mapas

Criterios de aceptación:

Se visualiza en la interfaz una gráfica con el número de pacientes por día y por diagnóstico.

Figura 11

Historia de usuario 8

★ Como Root y director quiero un contador de pacientes por rol y hospital para visualizarlo en la interfaz.
contadores gráficas y mapas

Historias de Usuario

Criterios de aceptación:

Se visualiza en la interfaz un contador con el número de pacientes por rol y hospital.

Figura 12

Historia de usuario 9



Como Director quiero un listado de los usuarios sin seguimientos por día para tener conocimiento de pacientes que ya no están registrando su información.

contadores gráficas y mapas

Criterios de aceptación:

Se visualiza un listado con los pacientes que no han solicitado seguimientos el día actual

Figura 13

Historia de usuario 10



Como Root y director quiero obtener los datos de los pacientes por diagnóstico, para visualizar sus coordenadas en un mapa.

contadores gráficas y mapas

Criterios de aceptación:

En el mapa se visualizan las coordenadas de los pacientes agrupados por diagnóstico.

Figura 14

Historia de usuario 11



Como Root y director quiero obtener los datos de los pacientes por día y diagnóstico, para visualizar sus coordenadas en un mapa.

contadores gráficas y mapas

Historias de Usuario

Criterios de aceptación:

En el mapa se visualizan las coordenadas de los pacientes agrupados por día y diagnóstico.

Figura 15

Historia de usuario 12



Como Root y director quiero obtener los datos de los pacientes por rol y hospital, para visualizar sus coordenadas en un mapa.
contadores gráficas y mapas

Criterios de aceptación:

En el mapa se visualizan las coordenadas de los pacientes agrupados por rol y hospital.

Nota. Historias de usuario y criterios de aceptación pertenecientes a la épica 2.

E3: La aplicación debe permitir el manejo de seguimientos de pacientes para tener control sobre el estado de salud de los mismos.

Tabla 4

Historias de usuario pertenecientes a la Épica 3

Historias de Usuario

Figura 16

Historia de usuario 13



Como paciente quiero solicitar un seguimiento para enviar sus signos y obtener una valoración de un doctor.
administrar seguimientos de pacientes

Historias de Usuario

Criterios de aceptación:

El paciente está permitido solicitar 3 seguimientos por día.

Figura 17

Historia de usuario 14

★ Como doctor quiero agendar un seguimiento para que no existan conflictos con el seguimiento respecto a un doctor con otro, y evitar la saturación de llamadas al mismo tiempo para un doctor.
administrar seguimientos de pacientes

Criterios de aceptación:

El doctor puede agendar un seguimiento, si este es solicitado con llamada.

Figura 18

Historia de usuario 15

★ Como doctor quiero atender una solicitud de seguimiento, para realizar una video llamada con el paciente y realizar la respectiva toma de datos para el registro del seguimiento.
administrar seguimientos de pacientes

Criterios de aceptación:

El doctor puede atender un seguimiento, si este está agendado.

Figura 19

Historia de usuario 16

★ Como doctor, paciente quiero cancelar una solicitud de seguimiento para evitar ser atendido o para no agendar a un paciente.
administrar seguimientos de pacientes

Criterios de aceptación:

Historias de Usuario

El doctor y el paciente están permitidos cancelar una solicitud.

Figura 20

Historia de usuario 17

★ *Como doctor, paciente quiero consultar un seguimiento para conocer el estado del paciente ese día en ese seguimiento.*
administrar seguimientos de pacientes

Criterios de aceptación:

Se visualiza un seguimiento específico de un día específico.

Figura 21

Historia de usuario 18

★ *Como doctor quiero editar un seguimiento para agregar mis observaciones en los respectivos campos, de acuerdo a la información obtenida en una llamada.*
administrar seguimientos de pacientes

Criterios de aceptación:

El seguimiento no puede ser editado si:

- Todavía no es atendido.
 - No ha sido atendido por el mismo doctor.
 - No pertenece al día actual.
-

Figura 22

Historia de usuario 19

★ *Como paciente, doctor, administrador, director, quiero filtrar seguimientos para listar cuales fueron atendidos, agendados o están solicitados.*
administrar seguimientos de pacientes

Criterios de aceptación:

Historias de Usuario

Se visualiza los seguimientos distribuidos en atendidos, agendados o que están solicitados.

Figura 23

Historia de usuario 20



Como paciente, doctor, director quiero ver el total de seguimientos de un solo paciente para ver el historial de seguimientos tanto solicitados como atendidos, agrupados por días.

administrar seguimientos de pacientes

Criterios de aceptación:

Se visualiza en una tabla la clasificación por día de los seguimientos realizados por un solo paciente.

Figura 24

Historia de usuario 21



Como doctor quiero comunicarme con el paciente a través de una video llamada con el fin de obtener los datos y realizar el seguimiento.

administrar seguimientos de pacientes

Criterios de aceptación:

La video llamada entre el doctor y el paciente permite la comunicación y toma de datos entre ambos usuarios.

Nota. Historias de usuario y criterios de aceptación pertenecientes a la época 3.

E4: La aplicación debe permitir gestionar Hospitales(Concepto) para agrupar usuarios dentro del mismo.

Tabla 5

Historias de usuario pertenecientes a la Épica 4

Historias de Usuario

Figura 25

Historia de usuario 22

★ *Como usuario registrado en la aplicación quiero filtrar hospitales por campo espacio para obtener los hospitales que pertenezcan a un espacio en específico.*
administrar hospitales

Criterios de aceptación:

Se obtiene un listado de los hospitales que pertenecen a un espacio en específico.

Figura 26

Historia de usuario 23

★ *Como root quiero crear hospitales para agrupar un administrador, pacientes y doctores.*
administrar hospitales

Criterios de aceptación:

Se permite la creación de un hospital solo al usuario root.

Figura 27

Historia de usuario 24

★ *Como root quiero actualizar un hospital para realizar cambios en los campos.*
administrar hospitales

Criterios de aceptación:

Historias de Usuario

Se permite la actualización de un hospital solo al usuario root.

Nota. Historias de usuario y criterios de aceptación pertenecientes a la época 4.

E5: La aplicación debe permitir gestionar espacios para sectorizar elementos de la aplicación.

Tabla 6

Historias de usuario pertenecientes a la Épica 5

Historias de Usuario

Figura 28

Historia de usuario 25

★ *Como Root quiero crear espacios para sectorizar los hospitales en espacios físicos.*
administrar los espacios

Criterios de aceptación:

- Se permite la creación de espacio solo al usuario root.
- Un espacio contiene un campo de espacio que representa a un espacio padre.
- Un espacio contiene un campo tipo. (barrio, parroquia, cantón, provincia)

Figura 29

Historia de usuario 26

★ *Como root, administrador quiero filtrar los espacios para listarlos por tipo y espacio padre.*
administrar los espacios

Criterios de aceptación:

Se visualiza el listado de los espacios según el filtro aplicado.

Historias de Usuario

Figura 30

Historia de usuario 27



Como root quiero actualizar un espacio, para cambiar sus datos dentro de la aplicación.

administrar los espacios

Criterios de aceptación:

Se permite la actualización de un espacio solo al usuario root.

Figura 31

Historia de usuario 28



Como root, director, doctor quiero ver un espacio para conocer sus campos incluyendo latitud y longitud.

administrar los espacios

Criterios de aceptación:

Se visualiza los campos de un espacio.

Nota. Historias de usuario y criterios de aceptación pertenecientes a la épica 5.

E6: La aplicación debe permitir administrar las cuentas y roles de usuario.


Tabla 7

Historias de usuario pertenecientes a la Épica 6

Historias de Usuario

Figura 32

Historia de usuario 29


 *Como usuario root, quiero tener el rol de super administrador para tener control sobre la aplicación web y asignar roles de administrador.*
crear y asignar roles de usuario

Criterios de aceptación:

- La aplicación tiene un único rol de root.
- El root es el único que asigna el rol de administrador.

Figura 33

Historia de usuario 30


 *Como administrador quiero agregar usuarios a un hospital y asignar un role dentro del hospital*
crear y asignar roles de usuario

Criterios de aceptación:

El administrador pertenece a un hospital y está permitido agregar a otros usuarios y asignarles un rol dentro del hospital.

Figura 34

Historia de usuario 31

 *Como usuario quiero iniciar sesión para acceder a mi cuenta dentro de la aplicación.*
crear y asignar roles de usuario

Historias de Usuario

Criterios de aceptación:

El usuario puede iniciar sesión y visualizar la ventana de inicio y seleccionar un hospital donde actuar con su rol asignado por un administrador, en el caso de paciente no es necesario.

Figura 35

Historia de usuario 32



Como usuario quiero crear una cuenta para poder ser asignado a un hospital con mi rol.

crear y asignar roles de usuario

Criterios de aceptación:

- El usuario crea una cuenta dentro de la aplicación.
 - Solo en el caso de paciente, selecciona un hospital donde ser atendido.
-

Figura 36

Historia de usuario 33



Como usuario quiero un botón de olvide mi contraseña o similar para recuperar mi contraseña.

crear y asignar roles de usuario

Criterios de aceptación:

Al dar clic en el botón olvide mi contraseña, el usuario es redirigido a un formulario para que ingrese su correo.

Figura 37

Historia de usuario 34



Como usuario quiero recibir un enlace a mi correo en caso de recuperar mi contraseña para ser redirigido a una vista y colocar mi nueva contraseña.

crear y asignar roles de usuario

Historias de Usuario

Criterios de aceptación:

- El usuario recibe en su correo un enlace que lo redirigirá a un formulario para actualizar su contraseña.
- Este enlace es válido por 45min.

Figura 38

Historia de usuario 35

★ *Como usuario quiero un formulario para cambiar mi contraseña en la aplicación.*
crear y asignar roles de usuario

Criterios de aceptación:

El usuario puede cambiar su contraseña por una nueva.

Figura 39

Historia de usuario 36

★ *Como usuario doctor, paciente quiero un formulario o campo de motivo de alta, para colocar el motivo de alta de un paciente.*
crear y asignar roles de usuario

Criterios de aceptación:

El doctor puede dar de alta a un paciente, el paciente también puede darse de alta de la aplicación web.

Figura 40

Historia de usuario 37

★ *Como usuario root, administrador, doctor, paciente quiero una vista de perfil de usuario, para ver los datos de mi perfil y de otros usuarios, excepto en el caso de paciente, solo ve de su perfil.*
crear y asignar roles de usuario

Historias de Usuario

Criterios de aceptación:

- Se visualiza el perfil de un usuario.
- Solo el paciente y el director no pueden ver el perfil de otro usuario que no sea su propio perfil.

Figura 41

Historia de usuario 38



*Como root, administrador, doctor, quiero listar usuarios para ver los usuarios actualmente registrados en la aplicación.
crear y asignar roles de usuario*

Criterios de aceptación:

Se lista los usuarios que están registrados en la aplicación web.

Figura 42

Historia de usuario 39



*Como usuario quiero actualizar mis datos para que los cambios sean reflejados en la vista de perfil.
crear y asignar roles de usuario*

Criterios de aceptación:

Se visualiza los cambios realizados por el usuario, en la vista de perfil.

Nota. Historias de usuario y criterios de aceptación pertenecientes a la épica 6.

3.2.3. Product Backlog (Backlog de producto)

Del apartado anterior se obtiene una lista de historias de usuario desordenadas, es necesario saber que historias de usuario son las primeras en ser implementadas por tal motivo se crea un backlog de producto donde las historias de usuario son priorizadas de forma descendente de mayor prioridad a menor prioridad, de acuerdo al valor que

aporta al modelo de negocio, en este caso el aporte que tiene la historia de usuario a la solución del problema.

A continuación, se presenta el backlog de producto en partes, por motivo del tamaño de las imágenes.

Figura 43

Backlog de producto(Parte 1)

Icebox	
★	Como usuario root, quiero tener el rol de super administrador para tener control sobre la aplicación web y asignar roles de administrador. <i>crear y asignar roles de usuario</i>
★	Como Root quiero crear espacios para sectorizar los hospitales en espacios físicos. <i>administrar los espacios</i>
★	Como root, administrador quiero filtrar los espacios para listarlos por tipo y espacio padre. <i>administrar los espacios</i>
★	Como root, director, doctor quiero ver un espacio para conocer sus campos incluyendo latitud y longitud. <i>administrar los espacios</i>
★	Como root quiero crear hospitales para agrupar un administrador, pacientes y doctores. <i>administrar hospitales</i>
★	Como usuario registrado en la aplicación quiero filtrar hospitales por campo espacio para obtener los hospitales que pertenezcan a un espacio en específico. <i>administrar hospitales</i>
★	Como usuario quiero crear una cuenta para poder ser asignado a un hospital con mi rol. <i>crear y asignar roles de usuario</i>
★	Como usuario quiero iniciar sesión para acceder a mi cuenta dentro de la aplicación. <i>crear y asignar roles de usuario</i>
★	Como administrador quiero agregar usuarios a un hospital y asignar un rol dentro del hospital <i>crear y asignar roles de usuario</i>
★	Como usuario root, administrador, doctor, paciente quiero una vista de perfil de usuario, para ver los datos de mi perfil y de otros usuarios, excepto en el caso de paciente, solo ve de su perfil. <i>crear y asignar roles de usuario</i>
★	Como paciente quiero solicitar un seguimiento para enviar sus signos y obtener una valoración de un doctor. <i>administrar seguimientos de pacientes</i>

Nota. Para crear el Product Backlog se usó la herramienta Pivotal Tracker, el Product Backlog se llama IceBox dentro de la herramienta.

Figura 44

Backlog de producto(Parte 2)

★	Como doctor, paciente quiero cancelar una solicitud de seguimiento para evitar ser atendido o para no agendar a un paciente. <i>administrar seguimientos de pacientes</i>
★	Como paciente, doctor, administrador, director, quiero filtrar seguimientos para listar cuales fueron atendidos, agendados o están solicitados. <i>administrar seguimientos de pacientes</i>
★	Como doctor quiero agendar un seguimiento para que no existan conflictos con el seguimiento respecto a un doctor con otro, y evitar la saturación de llamadas al mismo tiempo para un doctor. <i>administrar seguimientos de pacientes</i>
★	Como doctor quiero atender una solicitud de seguimiento, para realizar una video llamada con el paciente y realizar la respectiva toma de datos para el registro del seguimiento. <i>administrar seguimientos de pacientes</i>
★	Como paciente, doctor, director quiero ver el total de seguimientos de un solo paciente para ver el historial de seguimientos tanto solicitados como atendidos, agrupados por días. <i>administrar seguimientos de pacientes</i>
★	Como doctor, paciente quiero consultar un seguimiento para conocer el estado del paciente ese día en ese seguimiento. <i>administrar seguimientos de pacientes</i>
★	Como doctor quiero editar un seguimiento para agregar mis observaciones en los respectivos campos, de acuerdo a la información obtenida en una llamada. <i>administrar seguimientos de pacientes</i>
★	Como Root, administrador, doctor, quiero crear una notificación para informar al paciente de que se va a realizar una video llamada. <i>administrar notificaciones</i>
★	Como root, administrador, doctor, paciente quiero obtener/Listar mis notificaciones recibidas enviadas por otro usuario para estar informado respecto a lo que me envían otros usuarios. <i>administrar notificaciones</i>
★	Como Root, administrador, doctor, paciente quiero consultar una notificación para acceder al enlace de la llamada o a su contenido <i>administrar notificaciones</i>
★	Como doctor quiero comunicarme con el paciente a través de una video llamada con el fin de obtener los datos y realizar el seguimiento. <i>administrar seguimientos de pacientes</i>

Nota. Para crear el Product Backlog se usó la herramienta Pivotal Tracker, el Product Backlog se llama IceBox dentro de la herramienta.

Figura 45

Backlog de producto(Parte 3)

★	Como root, doctor, paciente, administrador, quiero que las notificaciones se coloquen como vistas para cuando sean leídas estas cambien a ese estado. (Esto es interno - servidor - y se muestra a través del API) <i>administrar notificaciones</i>
★	Como ROOT, doctor, paciente, administrador quiero obtener/Listar notificaciones enviadas para mostrarlo en el api graphql (Esto es interno - servidor - y se muestra a través del API) <i>administrar notificaciones</i>
★	Como Root y director quiero obtener los datos de los pacientes por role y hospital, para visualizar sus coordenadas en un mapa. <i>contadores gráficos y mapas</i>
★	Como Root y director quiero obtener los datos de los pacientes por día y diagnóstico, para visualizar sus coordenadas en un mapa. <i>contadores gráficos y mapas</i>
★	Como Root y director quiero obtener los datos de los pacientes por diagnóstico, para visualizar sus coordenadas en un mapa. <i>contadores gráficos y mapas</i>
★	Como Director quiero un listado de los usuarios sin seguimientos por día para tener conocimiento de pacientes que ya no están registrando su información. <i>contadores gráficos y mapas</i>
★	Como Root y director quiero un contador de pacientes por rol y hospital para visualizarlo en la interfaz. <i>contadores gráficos y mapas</i>
★	Como Root y director quiero un contador de pacientes por día y diagnóstico para visualizarlo en la interfaz y como gráfica. <i>contadores gráficos y mapas</i>
★	Como Root y director quiero un contador de pacientes por diagnóstico para visualizarlo en la interfaz y como gráfica. <i>contadores gráficos y mapas</i>
★	Como root quiero actualizar un hospital para realizar cambios en los campos. <i>administrar hospitales</i>
★	Como root quiero actualizar un espacio, para cambiar sus datos dentro de la aplicación. <i>administrar los espacios</i>

Nota. Para crear el Product Backlog se usó la herramienta Pivotal Tracker, el Product Backlog se llama IceBox dentro de la herramienta.

Figura 46

Backlog de producto(Parte 4)

★	Como usuario quiero actualizar mis datos para que los cambios sean reflejados en la vista de perfil. <i>crear y asignar roles de usuario</i>
★	Como root, administrador, doctor, quiero listar usuarios para ver los usuarios actualmente registrados en la aplicación. <i>crear y asignar roles de usuario</i>
★	Como usuario doctor, paciente quiero un formulario o campo de motivo de alta, para colocar el motivo de alta de un paciente. <i>crear y asignar roles de usuario</i>
★	Como usuario quiero un formulario para cambiar mi contraseña en la aplicación. <i>crear y asignar roles de usuario</i>
★	Como usuario quiero un botón de olvide mi contraseña o similar para recuperar mi contraseña. <i>crear y asignar roles de usuario</i>
★	Como usuario quiero recibir un enlace a mi correo en caso de recuperar mi contraseña para ser redirigido a una vista y colocar mi nueva contraseña. <i>crear y asignar roles de usuario</i>

Nota. Para crear el Product Backlog se usó la herramienta Pivotal Tracker, el Product Backlog se llama IceBox dentro de la herramienta.

El motivo de crear un product backlog es el de obtener las historias priorizadas y listas para ser colocadas en un sprint, como se muestra en el siguiente apartado.

3.2.4. Sprints

Cada sprint contiene un número de historias de usuario tomadas del product backlog, también posee un detalle sobre la fecha de inicio, fecha de finalización y los puntos, estos puntos corresponden a la suma de los puntos asignados a cada historia de usuario. Con la herramienta Pivotal Tracker es posible puntuar las historias de usuario con valores de 0,1,2,3,5,8, secuencia de Fibonacci. Esto se lo hace con los indicadores en forma de barras que se encuentran en el lado izquierdo de cada historia.

La velocidad de cada sprint es constante para todos, esto representa el número total de puntos de historias de usuario que se pueden desarrollar en un sprint. Como se puede apreciar en la Tabla 8, algunos sprint superan el número de puntos, sin embargo, esto se ve compensado en otros sprints en los cuales el número de puntos es menor a la velocidad.

Tabla 8

Detalle de cada Sprint

Nombre del Sprint	Detalle
Sprint 1	
Figura 47	Fecha de inicio:
<i>Historias de usuario – Sprint 1</i>	09 - 03 - 2020
<p>★ ☰ Como usuario root, quiero tener el rol de super administrador para tener control sobre la aplicación web y asignar roles de administrador. crear y asignar roles de usuario</p>	Fecha Fin: 15 - 03 - 2020
<p>★ ☰ Como Root quiero crear espacios para sectorizar los hospitales en espacios físicos. administrar los espacios</p>	Puntos: 23
<p>★ ☰ Como root, administrador quiero filtrar los espacios para listarlos por tipo y espacio padre. administrar los espacios</p>	
<p>★ ☰ Como root, director, doctor quiero ver un espacio para conocer sus campos incluyendo latitud y longitud. administrar los espacios</p>	
Sprint 2	
Figura 48	Fecha de inicio:
<i>Historias de usuario – Sprint 2</i>	16 - 03 - 2020
<p>★ ☰ Como root quiero crear hospitales para agrupar un administrador, pacientes y doctores. administrar hospitales</p>	Fecha Fin: 22 - 03 - 2020
<p>★ ☰ Como usuario registrado en la aplicación quiero filtrar hospitales por campo espacio para obtener los hospitales que pertenezcan a un espacio en específico. administrar hospitales</p>	Puntos: 37
<p>★ ☰ Como usuario quiero crear una cuenta para poder ser asignado a un hospital con mi rol. crear y asignar roles de usuario</p>	
<p>★ ☰ Como usuario quiero iniciar sesión para acceder a mi cuenta dentro de la aplicación. crear y asignar roles de usuario</p>	
<p>★ ☰ Como administrador quiero agregar usuarios a un hospital y asignar un rol dentro del hospital crear y asignar roles de usuario</p>	

Nombre del Sprint	Detalle
Sprint 3	
Figura 49	Fecha de inicio:
<i>Historias de usuario – Sprint 3</i>	23 - 03 - 2020
<p>★ ☰ Como usuario root, administrador, doctor, paciente quiero una vista de perfil de usuario, para ver los datos de mi perfil y de otros usuarios, excepto en el caso de paciente, solo ve de su perfil. crear y asignar roles de usuario</p>	Fecha Fin: 29 - 03 - 2020
<p>★ ☰ Como paciente quiero solicitar un seguimiento para enviar sus signos y obtener una valoración de un doctor. administrar seguimientos de pacientes</p>	Puntos: 28
<p>★ ☰ Como doctor, paciente quiero cancelar una solicitud de seguimiento para evitar ser atendido o para no agendar a un paciente. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como paciente, doctor, administrador, director, quiero filtrar seguimientos para listar cuales fueron atendidos, agendados o están solicitados. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como doctor quiero agendar un seguimiento para que no existan conflictos con el seguimiento respecto a un doctor con otro, y evitar la saturación de llamadas al mismo tiempo para un doctor. administrar seguimientos de pacientes</p>	
Sprint 4	
Figura 50	Fecha de inicio:
<i>Historias de usuario – Sprint 4</i>	30 - 03 - 2020
	Fecha Fin:
	05 - 04 - 2020
	Puntos: 26

Nombre del Sprint	Detalle
<p>★ ☰ Como doctor quiero atender una solicitud de seguimiento, para realizar una video llamada con el paciente y realizar la respectiva toma de datos para el registro del seguimiento. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como paciente, doctor, director quiero ver el total de seguimientos de un solo paciente para ver el historial de seguimientos tanto solicitados como atendidos, agrupados por días. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como doctor, paciente quiero consultar un seguimiento para conocer el estado del paciente ese día en ese seguimiento. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como doctor quiero editar un seguimiento para agregar mis observaciones en los respectivos campos, de acuerdo a la información obtenida en una llamada. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como Root, administrador, doctor, quiero crear una notificación para informar al paciente de que se va a realizar una video llamada. administrar notificaciones</p>	

Sprint 5

Figura 51

Historias de usuario – Sprint 5

Fecha de inicio:

06 - 04 - 2020

Fecha Fin:

12 - 04 - 2020

Puntos: 35

<p>★ ☰ Como root, administrador, doctor, paciente quiero obtener/Listar mis notificaciones recibidas enviadas por otro usuario para estar informado respecto a lo que me envían otros usuarios. administrar notificaciones</p>	
<p>★ ☰ Como Root, administrador, doctor, paciente quiero consultar una notificación para acceder al enlace de la llamada o a su contenido administrar notificaciones</p>	
<p>★ ☰ Como doctor quiero comunicarme con el paciente a través de una video llamada con el fin de obtener los datos y realizar el seguimiento. administrar seguimientos de pacientes</p>	
<p>★ ☰ Como root, doctor, paciente, administrador, quiero que las notificaciones se coloquen como vistas para cuando sean leídas estas cambien a ese estado. (Esto es interno - servidor - y se muestra a través del API) administrar notificaciones</p>	
<p>★ ☰ Como ROOT, doctor, paciente, administrador quiero obtener/Listar notificaciones enviadas para mostrarlo en el api graphql (Esto es interno - servidor - y se muestra a través del API) administrar notificaciones</p>	
<p>★ ☰ Como Root y director quiero obtener los datos de los pacientes por rol y hospital, para visualizar sus coordenadas en un mapa. contadores gráficos y mapas</p>	

Nombre del Sprint	Detalle
Sprint 6	
Figura 52	Fecha de inicio:
<i>Historias de usuario – Sprint 6</i>	13 - 04 - 2020
<ul style="list-style-type: none"> <li data-bbox="315 470 1166 569"> ★ Como Root y director quiero obtener los datos de los pacientes por día y diagnóstico, para visualizar sus coordenadas en un mapa. contadores gráficos y mapas <li data-bbox="315 579 1166 678"> ★ Como Root y director quiero obtener los datos de los pacientes por diagnóstico, para visualizar sus coordenadas en un mapa. contadores gráficos y mapas <li data-bbox="315 688 1166 787"> ★ Como Director quiero un listado de los usuarios sin seguimientos por día para tener conocimiento de pacientes que ya no están registrando su información. contadores gráficos y mapas <li data-bbox="315 798 1166 896"> ★ Como Root y director quiero un contador de pacientes por rol y hospital para visualizarlo en la interfaz. contadores gráficos y mapas <li data-bbox="315 907 1166 1005"> ★ Como Root y director quiero un contador de pacientes por día y diagnóstico para visualizarlo en la interfaz y como gráfica. contadores gráficos y mapas 	Fecha Fin: 19 - 04 - 2020 Puntos: 31
Sprint 7	
Figura 53	Fecha de inicio:
<i>Historias de usuario – Sprint 7</i>	20 - 04 - 2020
<ul style="list-style-type: none"> <li data-bbox="315 1236 1166 1335"> ★ Como Root y director quiero un contador de pacientes por diagnóstico para visualizarlo en la interfaz y como gráfica. contadores gráficos y mapas <li data-bbox="315 1346 1166 1415"> ★ Como root quiero actualizar un hospital para realizar cambios en los campos. administrar hospitales <li data-bbox="315 1425 1166 1524"> ★ Como root quiero actualizar un espacio, para cambiar sus datos dentro de la aplicación. administrar los espacios <li data-bbox="315 1535 1166 1633"> ★ Como root, administrador, doctor, quiero listar usuarios para ver los usuarios actualmente registrados en la aplicación. crear y asignar roles de usuario <li data-bbox="315 1644 1166 1743"> ★ Como usuario doctor, paciente quiero un formulario o campo de motivo de alta, para colocar el motivo de alta de un paciente. crear y asignar roles de usuario 	Fecha Fin: 26 - 04 - 2020 Puntos: 28

Nombre del Sprint	Detalle
Sprint 8	
Figura 54	Fecha de inicio:
<i>Historias de usuario – Sprint 8</i>	27 - 04 - 2020
<ul style="list-style-type: none"> ★ ☰ Como usuario quiero un formulario para cambiar mi contraseña en la aplicación. crear y asignar roles de usuario 	Fecha Fin: 03 - 05 - 2020
<ul style="list-style-type: none"> ★ ☰ Como usuario quiero actualizar mis datos para que los cambios sean reflejados en la vista de perfil. crear y asignar roles de usuario 	Puntos: 21
<ul style="list-style-type: none"> ★ ☰ Como usuario quiero un botón de olvide mi contraseña o similar para recuperar mi contraseña. crear y asignar roles de usuario 	
<ul style="list-style-type: none"> ★ ☰ Como usuario quiero recibir un enlace a mi correo en caso de recuperar mi contraseña para ser redirigido a una vista y colocar mi nueva contraseña. crear y asignar roles de usuario 	

3.3. Diseño

En la etapa de diseño, se usó diagramas para representar como está estructurado el proyecto tanto en la parte física como lógica, también se usa diagramas UML para explicar el uso de la arquitectura limpia y como esta se adaptaría al proyecto.

Se utilizó arquitectura limpia porque permite escalar en funcionalidad y esto va acorde con el marco Scrum, en cada Sprint es necesario entregar un incremento, por lo cual al tener desacoplado el código se facilita las extensiones sin afectar al resto de código.

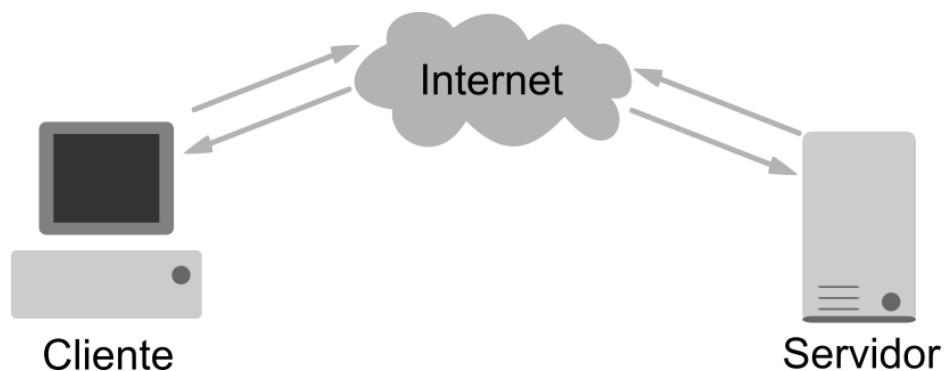
3.3.1. Diagrama de arquitectura física

Para la arquitectura física se usó la comunicación entre Cliente – Servidor, ambos conectados a través del internet. La comunicación es bidireccional un cliente con conexión a internet envía una petición al servidor, el servidor que también está

conectado a internet procesa esta petición y la responde, como se puede ver en la Figura 55 a continuación.

Figura 55

Arquitectura física Cliente - Servidor



3.3.2. Diagrama de arquitectura web

La arquitectura web que se presenta en la siguiente Figura 56, está compuesta por diferentes tecnologías, tanto del lado del cliente como del servidor.

Las tecnologías en la parte del cliente son las siguientes:

Angular js: Es un framework javascript que permite estructurar de mejor manera el código, facilita la reutilización de elementos de la interfaz dentro del desarrollo.

Firebase: Se utiliza firebase para el envío y recepción de notificaciones en tiempo real.

Agora.io: API que permite la comunicación bidireccional de audio y video, se utilizó para las video llamadas entre el paciente y el doctor.

Las tecnologías alojadas en el servidor son:

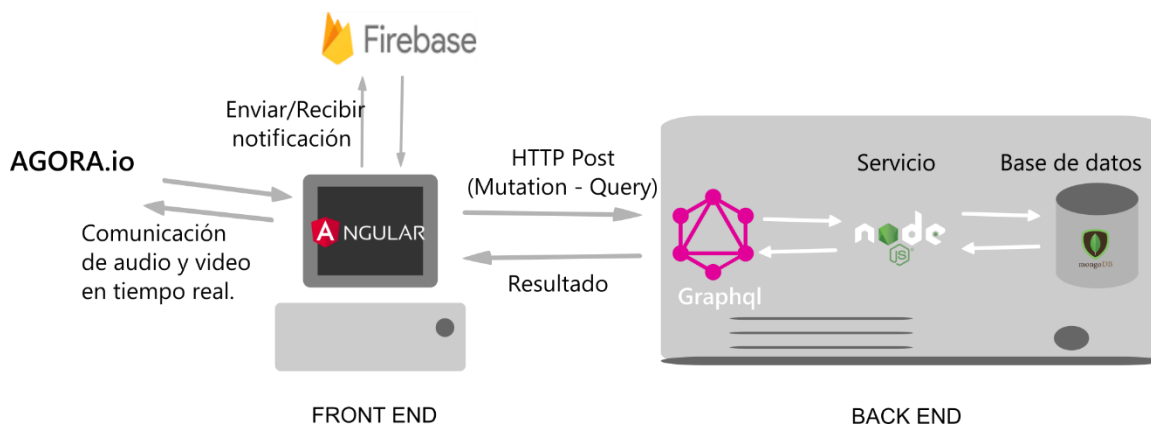
GraphQL: Recibe las peticiones y las envía a un servicio de node.js, también toma la respuesta del servicio y envía una respuesta al cliente. Tiene Mutaciones que son operaciones de actualización y Queries que son operaciones de consulta.

Node.js: Permite ejecutar javascript del lado del servidor, posee la lógica de negocio y procesa peticiones, además de mantener la comunicación con la base de datos.

MongoDB: Es una base de datos no relacional. Se usó para la persistencia de los datos.

Figura 56

Arquitectura web



En la Figura 56 se visualiza el flujo que siguen los datos, partiendo desde el frontend elaborado en Angular y llegando como petición a graphql el cual la recibe y selecciona un servicio de node.js que procese esta petición y en caso de requerirse ser almacenado en mongodb. El proceso de respuesta es similar se parte del servicio que toma los datos de mongodb y los envía a través de graphql con un formato específico.

Las apis de Agora.io y firebase son consumidas por el frontend y proporcionan funcionalidad extra, para realizar video llamadas y el envío de notificaciones en tiempo real.

3.3.3. Diagrama de arquitectura lógica

La arquitectura limpia busca la independencia de componentes, también separa el software en dos capas principales, la capa de dominio y la capa de detalle.

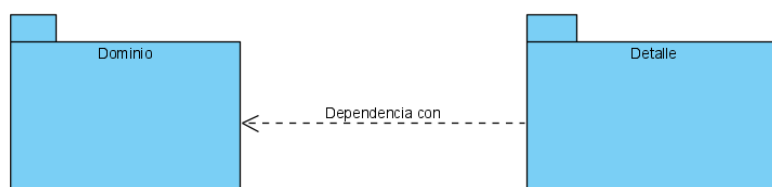
Dominio: Contiene la lógica de negocio.

Detalle: Contiene elementos externos a la lógica de negocio, como base de datos, vistas o frameworks.

La capa de detalle tiene que depender de la capa de dominio, porque la parte central y que contiene la implementación en código de las historias de usuario del proyecto es el dominio, todo lo demás puede cambiar a excepción de este.

Figura 57

Diagrama de arquitectura lógica



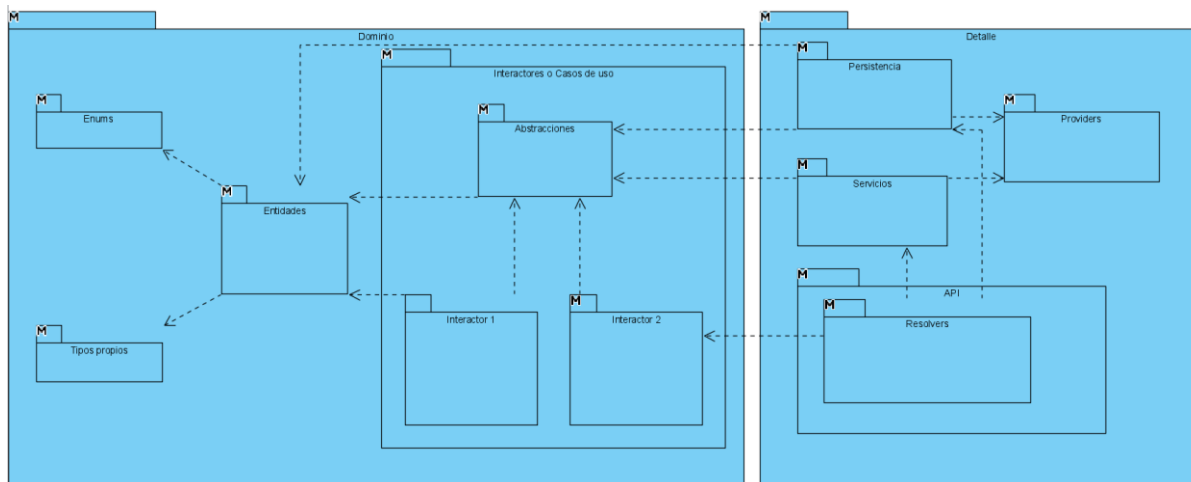
Como se observa en la Figura 57, la capa de detalle es dependiente de la capa de dominio, estas capas son representadas como directorios dentro del código.

3.3.4. Diagrama de arquitectura limpia

A continuación, se presenta un diagrama de cómo se llevó a cabo el uso de una arquitectura limpia dentro del presente proyecto.

Figura 58

Diagrama de arquitectura limpia



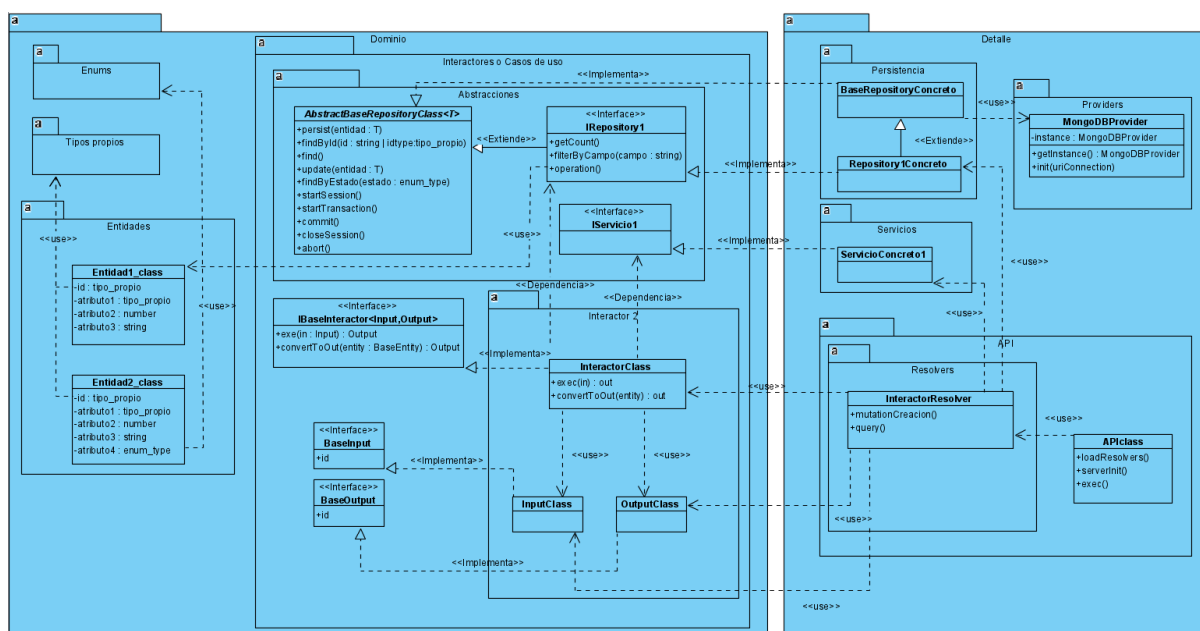
Al igual que en el diagrama de arquitectura lógica de la Figura 57, podemos apreciar la dependencia que existen entre los diferentes directorios, que conforman el proyecto, siendo en el dominio los enums, tipos de datos y las entidades la capa de la cual dependen las demás. La conexión entre capa de dominio y de detalle se realiza a través del interactor que son clases que realizan la implementación de una historia de usuario, este usa las abstracciones para exponer lo que requiere del detalle. Y a su vez en el detalle los resolvers que son clases que reciben peticiones del cliente usan a los interactores creando dependencia entre estos. Más adelante se presentan diagramas que muestran a detalle como fluye la información a través de las capas.

3.3.5. Diagrama de clases

El diagrama de clases a continuación permite observar como las clases interactúan dentro del proyecto de forma general.

Figura 59

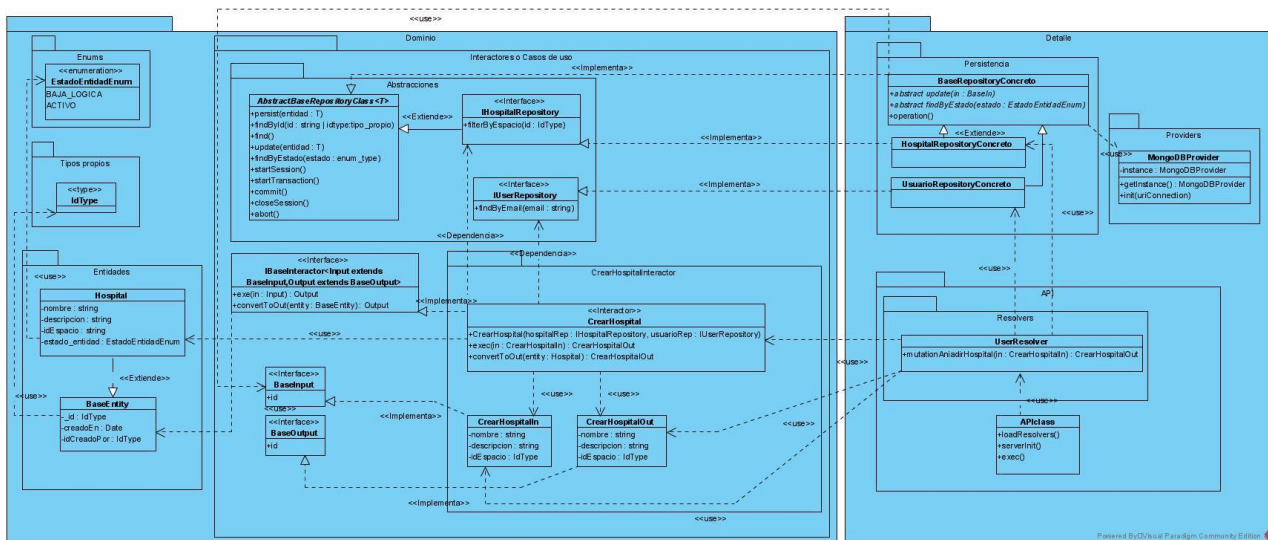
Diagrama de clases general



El diagrama de clases de la Figura 59, muestra la dependencia entre clases y el flujo de los datos, una petición del cliente es recibida por la clase InteractorResolver, esta a su vez crea una instancia del InteractorClass y pasa como argumentos las clases concretas que implementan las interfaces IRepository y IServicio que el InteractorClass expuso como dependencias. A su vez el InteractorClass usa clases de entrada que vienen de la petición y clases de salida de datos que son obtenidos de las entidades a través del método convertToOut.

En la siguiente Figura 60 se puede observar como el diagrama general de la Figura 60 se puede adaptar a una historia de usuario específica con sus dependencias y entidades.

Figura 60
Diagrama de clases específico de una historia de usuario



La Figura 60, implementa la arquitectura limpia para la historia de usuario de creación de hospitales, la cual es representada por una clase interactor CrearHospital, la misma que depende de sus interfaces IHospitalRepository para métodos de persistencia para la entidad Hospital e IUserRepository para métodos de persistencia para la entidad Usuario, al crear un hospital es necesario conocer el usuario que lo va a crear y si tiene los permisos para hacerlo y también es necesario conocer el Hospital y su clase de persistencia con los métodos requeridos para el guardado en una base de datos.

Se puede observar el desacoplamiento dentro de la capa de dominio entre las entidades, abstracciones y los interactores, los interactores pueden usar las abstracciones y entidades que sean necesarias para cumplir con la historia de usuario. También usa

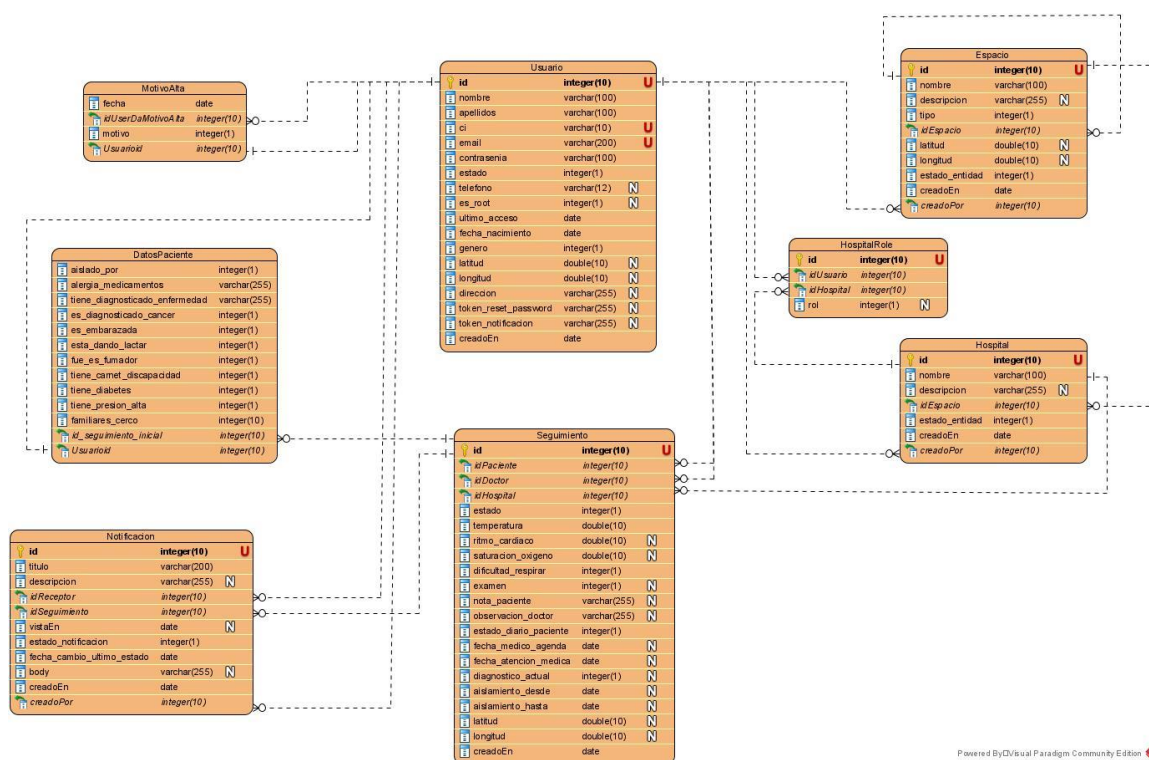
una clase de entrada y una clase de salida para permitir el ingreso y salida de información. La capa de dominio no depende del detalle, por lo tanto, esta desacoplada y no es afectada si se produce un cambio de las clases de detalle.

3.3.6. Diagrama entidad relación

MongoDB es una base de datos no relacional por lo que es una ventaja en él proyecto, al ser escalable la base de datos no mantiene estructuras fijas y es adaptable a las historias de usuario. Al ser no relacional solo maneja referencias entre colecciones de datos, donde cada colección sería una tabla en una base de datos relacional y las referencias serían similares a las relaciones, también maneja índices únicos y un objeto de id único. Por lo cual es necesario realizar un diagrama entidad relación para guiar y entender como persisten los datos dentro de la base de datos.

Figura 61

Diagrama entidad - relación

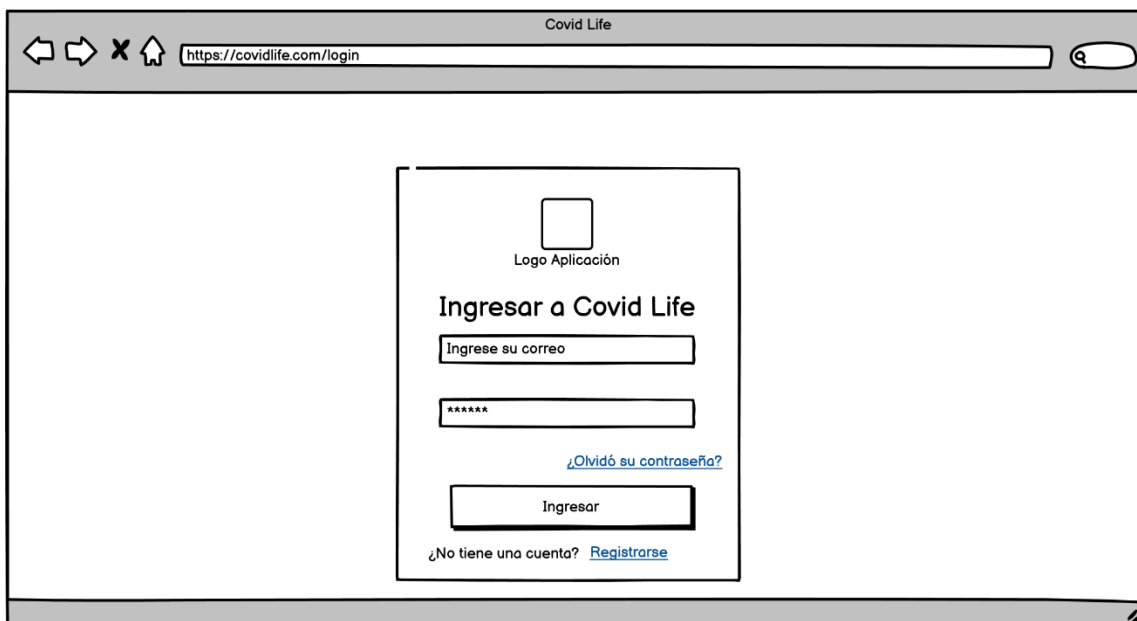


3.3.7. Mockups(Maquetas)

A continuación, se presenta las maquetas de las vistas principales que formarán parte del diseño visual final del proyecto. Esto permite observar cómo se distribuyen los diferentes componentes antes de ser implementados. Las maquetas a continuación se realizaron con la herramienta de maquetado Balsamiq Mockups³.

Figura 62

Mockup(Maqueta) de Inicio de Sesión



Nota. La maqueta indica cómo se vería el formulario de inicio de sesión, con sus campos y botón. Además, cuenta con una opción de olvido su contraseña y registro.

³ Balsamiq Mockups es una herramienta de maquetado, que permite realizar prototipos visuales de una aplicación software.

Figura 63

Mockup(Maqueta) de registro de usuario

Nota. La maqueta indica cómo se vería el formulario de registro, con sus campos y botón. También indica cómo sería el registro por pasos.

Figura 64

Mockup(Maqueta) de Estado de Seguimientos

Nota. La maqueta indica cómo visualiza el doctor los seguimientos antes de seleccionar uno para atenderlo.

Figura 65

Mockup(Maqueta) de tabla

Byron Rosas
Doctor
Seguimientos
Pacientes
Estadísticas

Pacientes
Pacientes actualmente registrados

Dr. Darwin Guaila
Agregar usuarios

#	Nombre	Apellido	Correo	Cédula	Teléfono
1	Byron	Rosas	xyz@hotmail.com	1725878798	0222222
2	Juan	Toa	xyz@hotmail.com	1725878798	0222222
3	Pedro	Guaila	xyz@hotmail.com	1725878798	0222222

Nota. La maqueta indica cómo se verían las tablas para mostrar información de las entidades.

Figura 66

Mockup(Maqueta) de datos de paciente

Atención de seguimiento - Video llamada

Byron Rosas

Correo Electrónico: [Redacted]

Teléfono: [Redacted]

Dirección: [Redacted]

Cédula: [Redacted]

Fecha de Nacimiento: [Redacted]

Género: [Redacted]

Último Acceso: [Redacted]

Ficha Médica

Aislado por: [Redacted]

Alergia a medicamentos: [Redacted]

Tiene diagnosticado alguna enfermedad: [Redacted]

Tiene diagnosticado cancer: [Redacted]

Esto embarazada: [Redacted]

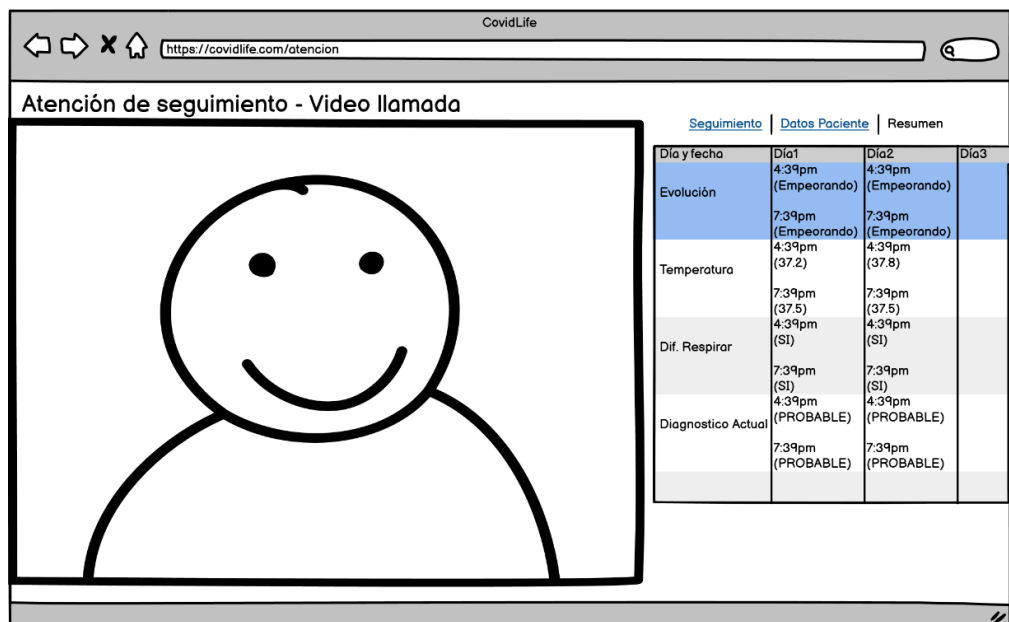
Esto dando de Lactar: [Redacted]

Video Call Placeholder

Nota. La maqueta indica cómo se verían los datos de los pacientes al momento de realizarse una video llamada.

Figura 67

Mockup(Maqueta) del resumen de seguimientos.



Nota. La maqueta indica cómo se verían los datos de seguimiento de los pacientes al momento de realizarse una video llamada.

Figura 68

Mockup(Maqueta) de la atención de un seguimiento

Byron Rosas

Evolucion(No editable): Temperatura No editable:

Ritmo Cardiaco(Opcional) Saturación de Oxígeno: (Opcional)

Dificultad para respirar: SI No Examen realizado:(Opcional)

Nota enviada por el paciente(No editable) Escriba una breve observación

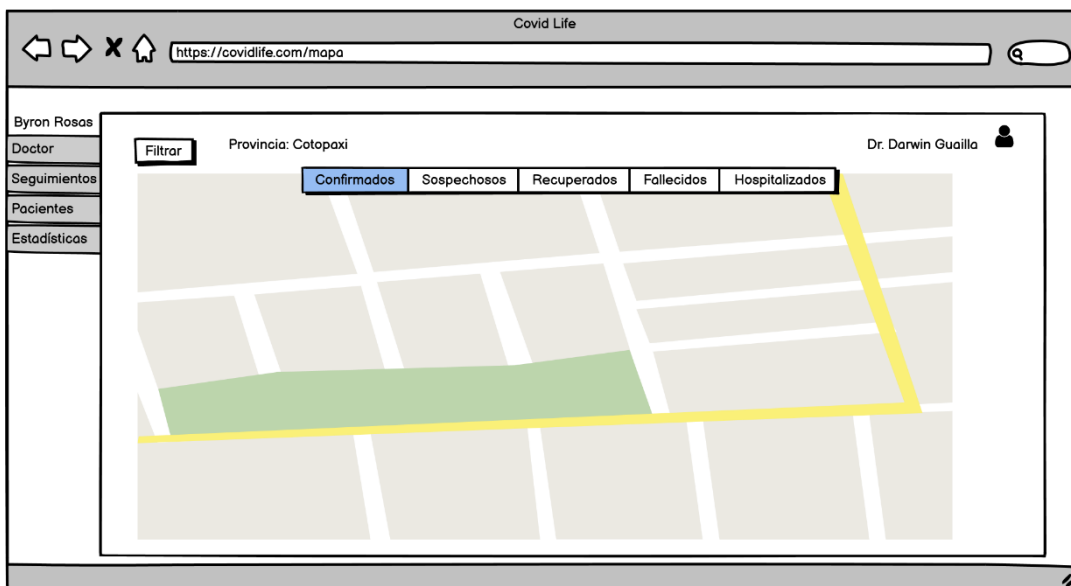
Aislado desde: Aislado Hasta:

Diagnóstico Actual: Atendido por:

Nota. La maqueta indica cómo se vería el formulario de atención de los pacientes al momento de realizarse una video llamada.

Figura 69

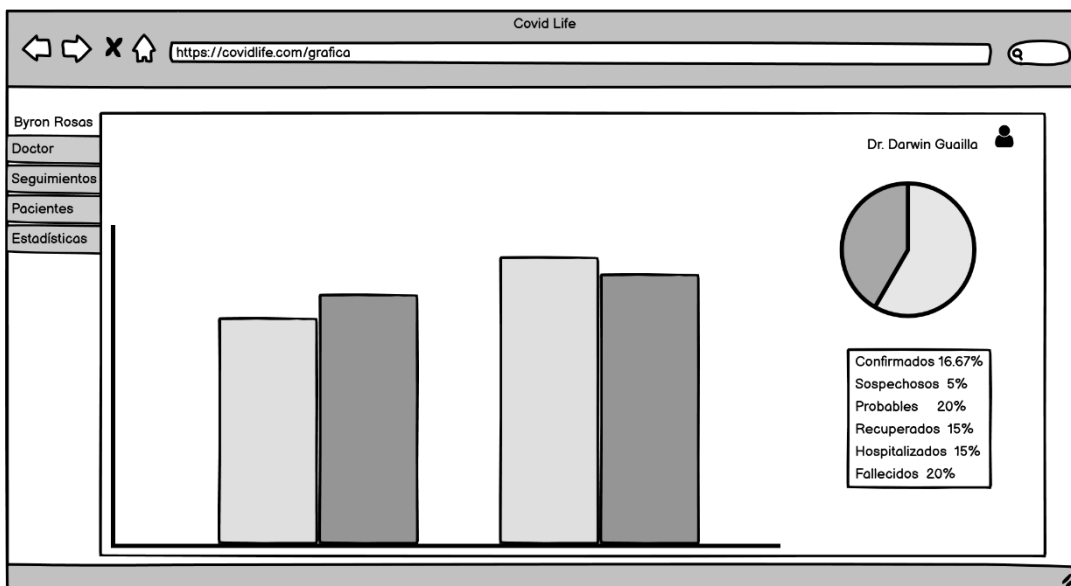
Mockup(Maqueta) de la vista de mapa



Nota. La maqueta indica cómo se vería el mapa y sus filtros dentro de la aplicación web.

Figura 70

Mockup(Maqueta) de la vista de gráficos



Nota. La maqueta indica cómo se verían los datos estadísticos dentro de la aplicación web.

3.4. Desarrollo por Sprint

El desarrollo de cada Sprint se basó en los diagramas de arquitectura mostrados en el apartado de diseño. Primero se comienza por las clases que pertenecen al dominio incluyendo los enums y datos propios. Estos irán aumentando dependiendo de la necesidad de cada historia de usuario. Al desarrollar una historia de usuario es necesario crear un interactor con sus interfaces, clases abstractas, entradas y salidas. Al final en el detalle se crea un resolver que instancie al interactor y lo provea de clases concretas, una entrada y reciba una salida para devolverla al cliente.

3.4.1. Capa de Dominio

Tipos de datos propios

La utilización de tipos de datos propios, creados exclusivamente para el proyecto, en este caso resultó útil para colocar un alias y de esta forma tener un solo punto de cambio para el tipo identificador ID, como se observa a continuación:

En el caso de mongodb se usa: "mongoose.ObjectId" , con el alias esto cambia a **IdType**, esto beneficia, porque al usar un tipo propio nos permite cambiar en un solo lugar y el resto de la aplicación tomará este cambio.

Enums

Los enums representan datos constantes del modelo de negocio dentro de la aplicación web.

Tabla 9*Estado de la entidad (Enum)*

Nombre	Valor
BAJA LÓGICA	0
ACTIVO	1

Tabla 10*Estado del usuario (Enum)*

Nombre	Valor
ACTIVO	0
INACTIVO	1

Tabla 11*Roles(Enum)*

Nombre	Valor
ROOT	0
ADMIN	1
PACIENTE	2
DOCTOR	3
DIRECTOR	4

Tabla 12*Motivo de alta (Enum)*

Nombre	Valor
HOSPITALIZADO	1

Nombre	Valor
FALLECIDO	2
RECUPERADO	3
SIN RESPUESTA	4

Tabla 13*Género del usuario (Enum)*

Nombre	Valor
M (Masculino)	0
F (Femenino)	1
PREFIERO NO DECIRLO	2

Tabla 14*Institución que aisló al paciente (Enum)*

Nombre	Valor
MPS	0
IESS	1
ISSFA	2
ISSPOL	3
PRIVADO	4

Tabla 15*Tipo de espacio (Enum)*

Nombre	Valor
PROVINCIA	0
CANTON	1
PARROQUIA	2
BARRIO	3

Tabla 16*Dificultad para respirar (Enum)*

Nombre	Valor
SI	0
NO	1

Tabla 17*Tipo de Examen (Enum)*

Nombre	Valor
NO REQUIERE	0
SOLICITADA	1
PRUEBA RAPIDA IGC POSITIVA	2
PRUEBA RAPIDA IGM POSITIVA	3
PRUEBA RAPIDA IGG E IGM NEGATIVA	4
PRUEBA RAPIDA IGG E IGM POSITIVA	5
PCR TR POSITIVA	6
PCR TR NEGATIVA	7

Nombre	Valor
PRUEBA RAPIDA INDETERMINADA	8
PCR INDETERMINADA	9

Tabla 18*Estado diario del paciente (Enum)*

Nombre	Valor
IGUAL EVOLUCION	0
MEJOR EVOLUCION	1
EMPEORANDO	2

Tabla 19*Diagnóstico actual (Enum)*

Nombre	Valor
PROBABLE	0
SOSPECHOSO	1
CONFIRMADO	2
RECUPERADO	3
FALLECIDO	4
HOSPITALIZADO	5

Tabla 20*Estado de la notificación(Enum)*

Nombre	Valor
RECHAZADA	0

Nombre	Valor
ACEPTADA	1
ENVIADA	2

Entidades

Las entidades son clases que están relacionadas con el negocio, en el presente proyecto estas clases son los elementos principales porque de estos dependen las demás capas.

Clase: Entidad Base

Descripción: Esta clase contiene campos comunes entre las entidades, de esta clase extienden las demás entidades.

Tabla 21

Entidades – Entidad Base

Campos	Tipo de Dato
_id	Identificador único de una entidad
Creado En	Fecha
Id creador	Tipo Identificador Foráneo para usuario

Clase: Usuario

Descripción: Esta clase representa a los actores de la aplicación con sus respectivos campos y roles.

Tabla 22*Entidades – Entidad Usuario*

Campos	Tipo de Dato
Nombre	Texto
Apellidos	Texto
CI (Número de identificación)	Texto
Email	Texto
Contraseña	Texto
Estado	Estado del usuario (Enum)
Teléfono	Texto
Roles	Objeto Role-Hospital (Arreglo)
Es Root (super administrador)	Booleano
Último acceso	Fecha
Motivo de alta	Objeto Motivo de Alta
Fecha de Nacimiento	Fecha
Género	Género del usuario (Enum)
Latitud	Numérico
Longitud	Numérico
Dirección	Texto
Datos del Paciente	Objeto de datos de paciente
Token para recuperar contraseña	Texto
Token para notificaciones	Texto

Clase: Datos de Paciente

Descripción: Esta clase representa los datos de un paciente, está orientado a información médica.

Tabla 23

Entidades – Entidad Datos de Paciente

Campos	Tipo de dato
Aislado por	Institución que aisló al paciente(Enum)
Medicamentos que causan alergia	Texto
Enfermedades diagnosticadas	Texto
Es diagnosticado con cáncer	Booleano
Está embarazada	Booleano
Está dando de lactar	Booleano
Es fumador	Booleano
Tiene carnet de discapacidad	Booleano
Tiene diabetes	Booleano
Tiene presión alta	Booleano
Familiares en cerco epidemiológico	Numérico
Id Seguimiento Inicial	Tipo Identificador foráneo seguimiento

Clase: Hospital

Descripción: La clase representa un hospital al que pertenecen y tienen acción un conjunto de usuarios.

Tabla 24*Entidades – Entidad Hospital*

Campos	Tipo de Dato
Nombre	Texto
Descripción	Texto
Id Espacio	Tipo Identificador Foráneo Espacio
Estado Entidad	Estado de la entidad (Enum)

Clase: Espacio

Descripción: La clase representa un espacio físico al que pertenecen hospitales.

Tabla 25*Entidades – Entidad Espacio*

Campos	Tipo de Dato
Nombre	Texto
Descripción	Texto
Tipo	Tipo de espacio (Enum)
Id Espacio	Tipo Identificador Foráneo para espacio padre
Latitud	Texto
Longitud	Texto
Estado Entidad	Estado de la entidad (Enum)

Clase: Seguimiento

Descripción: La clase seguimiento representa el estado de salud de un paciente en un día y tiempo específico.

Tabla 26

Entidades – Entidad Seguimiento

Campo	Tipo de Dato
Id Paciente	Tipo Identificador Foráneo paciente
Id Doctor	Tipo Identificador Foráneo doctor
Id Hospital	Tipo Identificador Foráneo hospital
Estado Entidad	Estado de la entidad (Enum)
Temperatura	Numérico
Ritmo cardiaco	Numérico
Saturación de Oxígeno	Numérico
Dificultad para respirar	Dificultad para respirar (Enum)
Examen	Tipo de Examen (Enum)
Nota del paciente	Texto
Observación del doctor	Texto
Estado diario del paciente	Estado diario del paciente (Enum)
Fecha agendado	Fecha
Fecha atención médica	Fecha
Diagnóstico Actual	Diagnóstico actual (Enum)
Aislamiento desde	Fecha
Aislamiento hasta	Fecha
Latitud	Numérico
Longitud	Numérico

Clase: Notificación

Descripción: Esta clase representa la información enviada como notificación entre usuarios.

Tabla 27

Entidades – Entidad Notificación

Campos	Tipo de Dato
Título	Texto
Descripción	Texto
Id Seguimiento	Tipo Identificador Foráneo seguimiento
Id Receptor	Tipo Identificador Foráneo usuario
Visto en (cuándo fue vista la notificación)	Fecha
Estado de la notificación	Estado de la notificación(Enum)
Fecha de cambio del último estado de la notificación	Fecha
Cuerpo de la notificación	Cualquier objeto (clave: valor) (Any)

Abstracciones

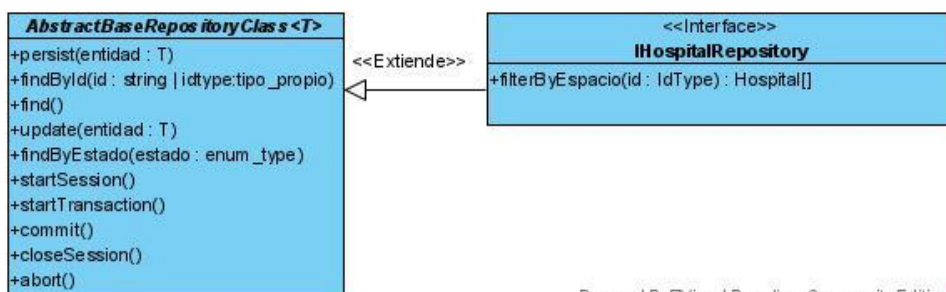
Las abstracciones son representadas por clases abstractas e interfaces, si el dominio ocupara clases concretas del detalle, existiría una dependencia del dominio hacia el detalle. Para invertir la dependencia, el dominio no ocupa clases concretas que pertenecen al detalle, solo abstracciones que son implementadas por clases concretas del detalle. A continuación, se listan estas abstracciones:

Los repositorios contienen métodos de persistencia que tienen que ser implementados por clases concretas dependiendo del motor de base de datos

que se utilice. También se puede observar que los repositorios extienden de una clase repositorio base, esto es para generalizar los métodos comunes entre clases.

Figura 71

Repositorio de Hospital



Powered By Visual Paradigm Community Edition

Figura 72

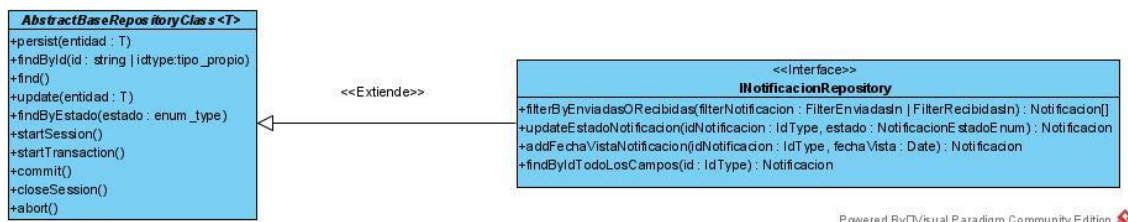
Repositorio de Seguimiento



Powered By Visual Paradigm Community Edition

Figura 73

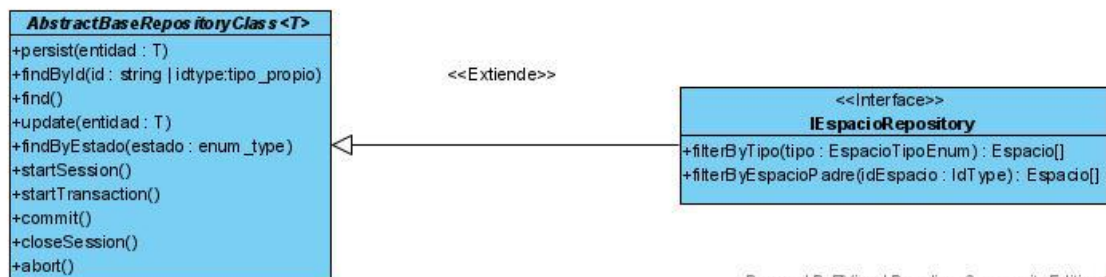
Repositorio de Notificación



Powered By DV/visual Paradigm Community Edition

Figura 74

Repositorio de Espacio



Powered By DV/visual Paradigm Community Edition

Figura 75

Repositorio de usuario



Powered By DV/visual Paradigm Community Edition

Los servicios al igual que los repositorios son abstracciones cuya función son proporcionar funcionalidades extra como encriptado y correo. A continuación, se muestran los servicios utilizados en el proyecto.

Figura 76

Servicio de Correo



Figura 77

Servicio de Encriptado



Interactors

Un interacto es una clase que pertenece a la capa de dominio y representa una historia de usuario, utiliza abstracciones para no depender de detalles. Esto quiere decir que depende de sus propias interfaces, las cuales son inyectadas en el constructor de la clase como dependencias. También depende de objetos y entidades que estén dentro del dominio.

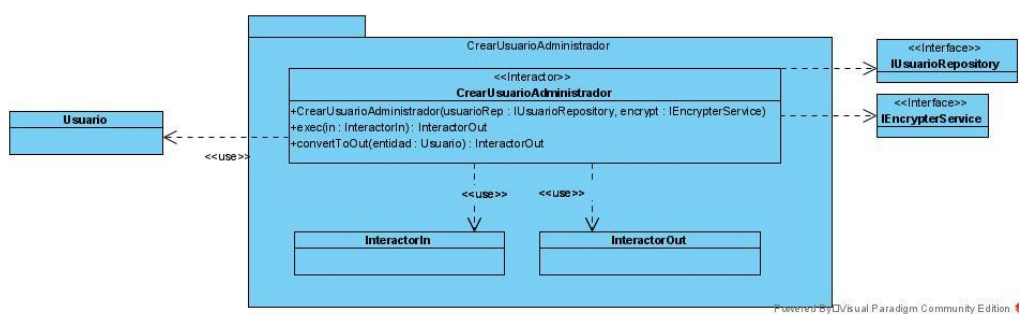
A continuación, se muestra el diseño de cada interacto que guía la codificación de cada historia de usuario por sprint.

Sprint 1: Interactors

1. Como usuario root, quiero tener el rol de super administrador para tener control sobre la aplicación web y asignar roles de administrador.

Figura 78

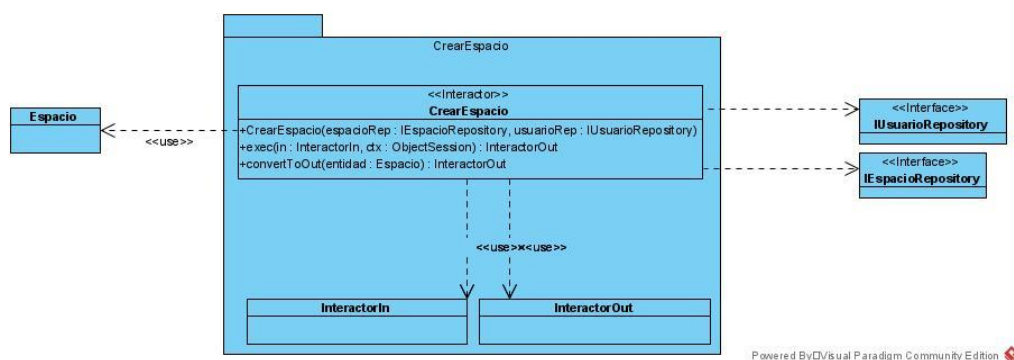
Sprint 1 – Interactor Crear usuario administrador



2. Como root quiero crear espacios para sectorizar los hospitales en espacios físicos.

Figura 79

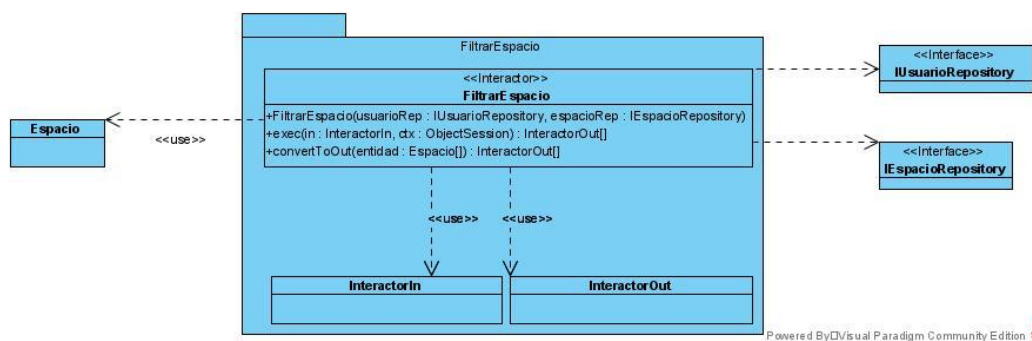
Sprint 1 – Interactor Crear espacio



3. Como root, administrador quiero filtrar los espacios para listarlos por tipo y espacio padre.

Figura 80

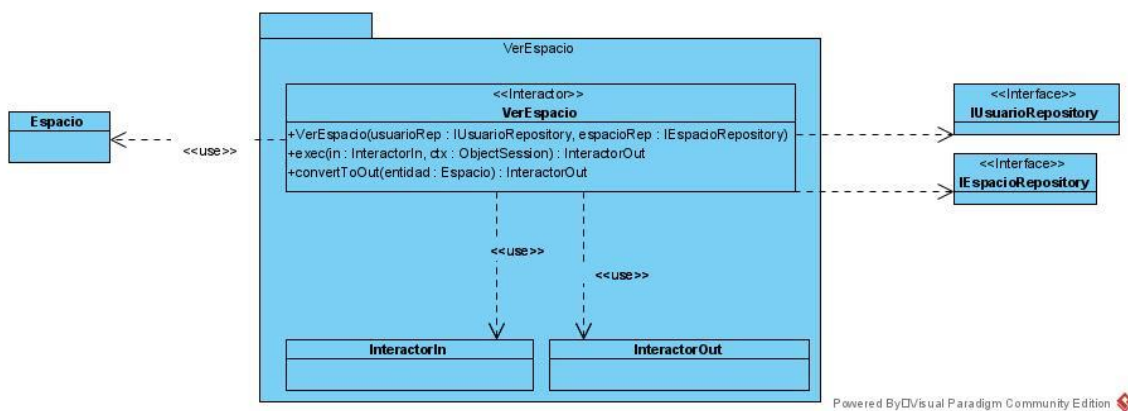
Sprint 1 – Interactor Filtrar espacio



4. Como root, director, doctor quiero ver un espacio para conocer sus campos incluyendo latitud y longitud.

Figura 81

Sprint 1 – Interactor Ver espacio

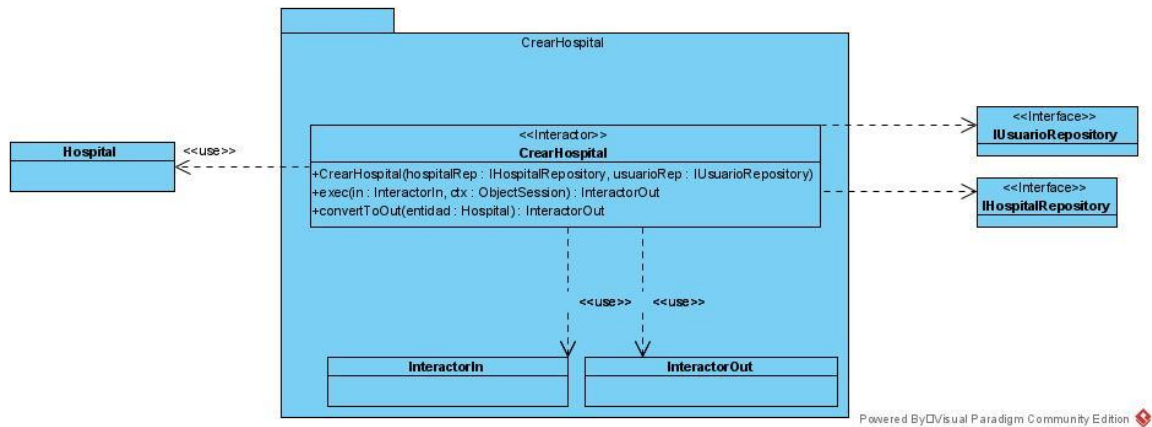


Sprint 2: Interactors

1. Como root quiero crear hospitales para agrupar un administrador, pacientes y doctores.

Figura 82

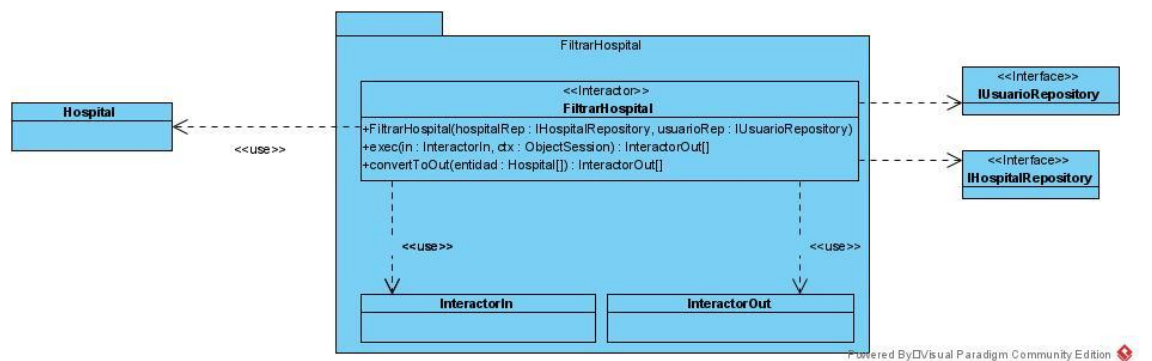
Sprint 2 – Interactor Crear hospital



2. Como usuario registrado en la aplicación quiero filtrar hospitales por campo espacio para obtener los hospitales que pertenezcan a un espacio en específico.

Figura 83

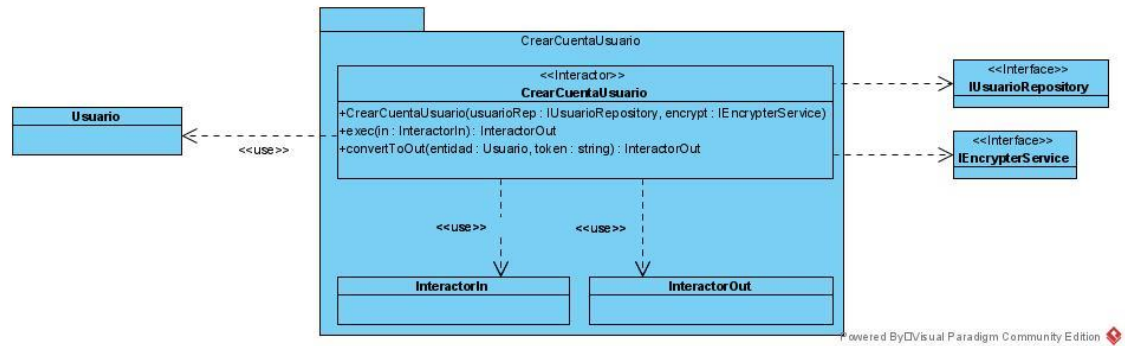
Sprint 2 - Interactor Filtrar hospital



- Como usuario quiero crear una cuenta para poder ser asignado a un hospital con mi rol.

Figura 84

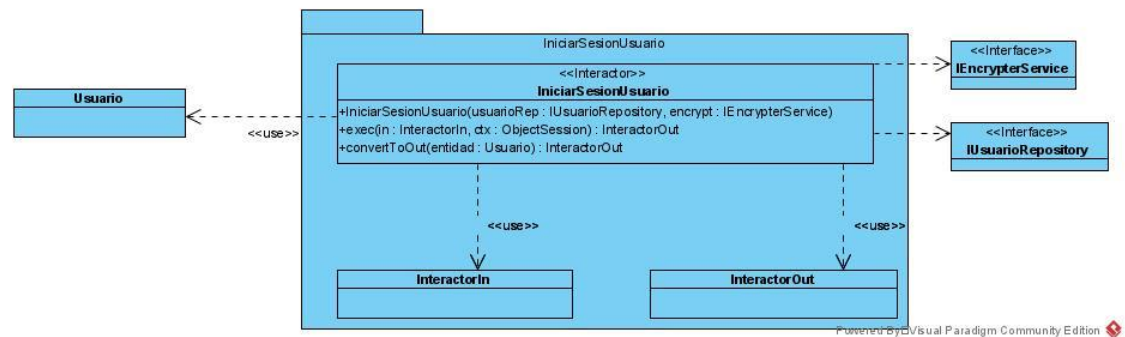
Sprint 2 – Interactor Crear cuenta de usuario



- Como usuario quiero iniciar sesión para acceder a mi cuenta dentro de la aplicación.

Figura 85

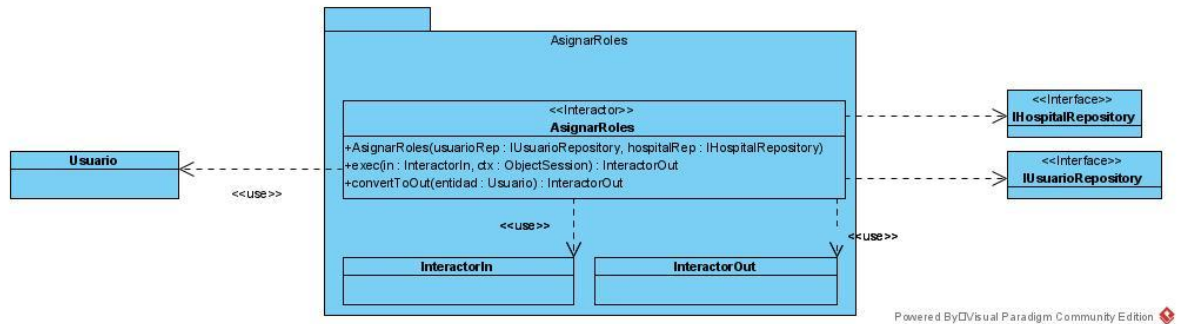
Sprint 2 – Interactor Iniciar sesión de usuario



- Como administrador quiero agregar usuarios a un hospital y asignar un rol dentro del hospital.

Figura 86

Sprint 2 – Interactor Asignar roles

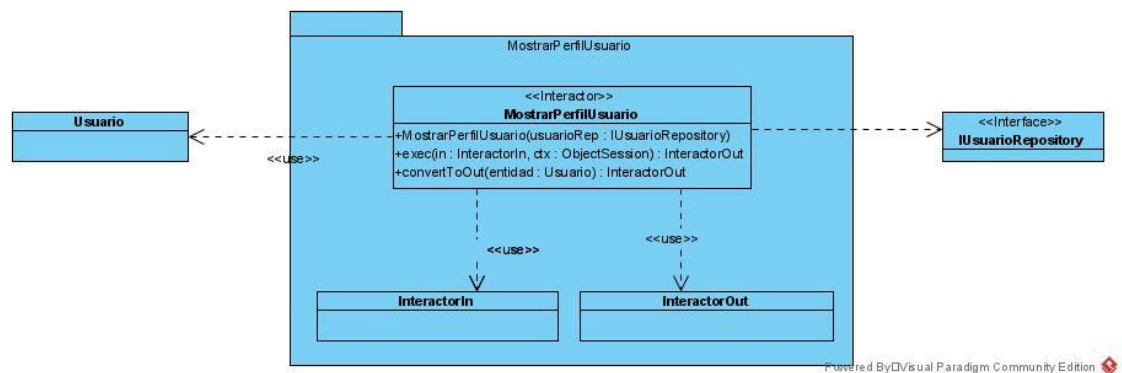


Sprint 3: Interactors

- Como usuario root, administrador, doctor, paciente quiero una vista de perfil de usuario, para ver los datos de mi perfil y de otros usuarios, excepto en el caso de paciente, solo ve de su perfil.

Figura 87

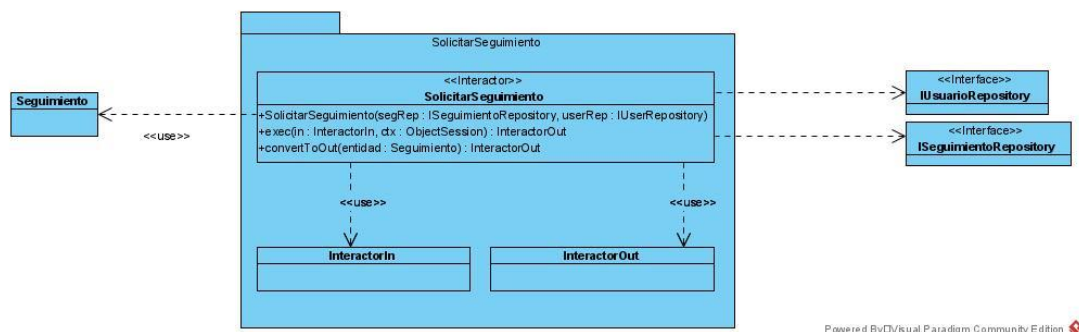
Sprint 3 – Interactor Mostrar perfil de usuario



- Como paciente quiero solicitar un seguimiento para enviar sus signos y obtener una valoración de un doctor.

Figura 88

Sprint 3 – Interactor Solicitar seguimiento

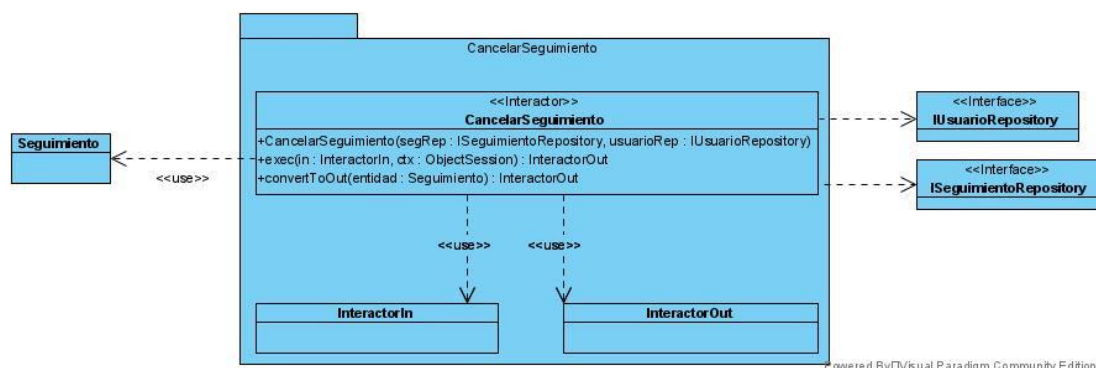


Powered By DV/Visual Paradigm Community Edition

- Como doctor, paciente quiero cancelar una solicitud de seguimiento para evitar ser atendido o para no agendar a un paciente.

Figura 89

Sprint 3 – Interactor Cancelar seguimiento

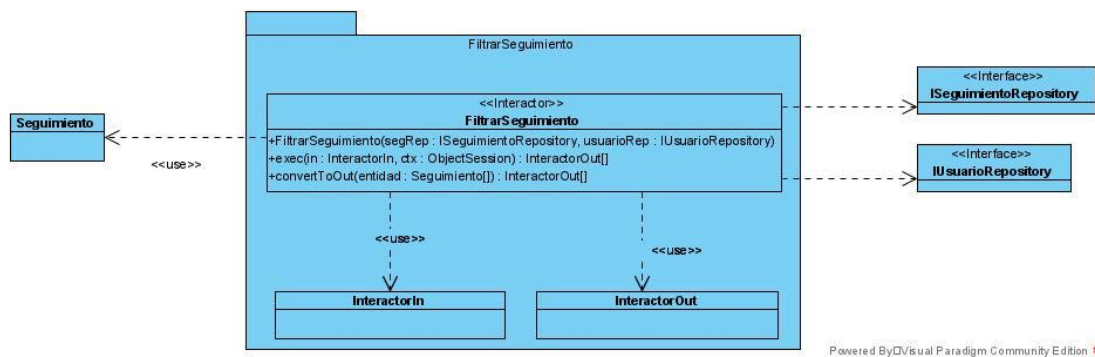


Powered By DV/Visual Paradigm Community Edition

- Como paciente, doctor, administrador, director, quiero filtrar seguimientos para listar cuales fueron atendidos, agendados o están solicitados.

Figura 90

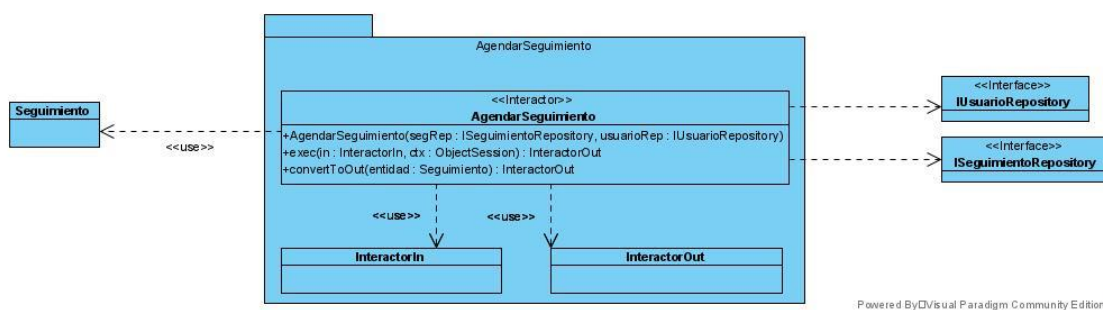
Sprint 3 – Interactor Filtrar seguimiento



- Como doctor quiero agendar un seguimiento para que no existan conflictos con el seguimiento respecto a un doctor con otro, y evitar la saturación de llamadas al mismo tiempo para un doctor.

Figura 91

Sprint 3 – Interactor Agendar seguimiento

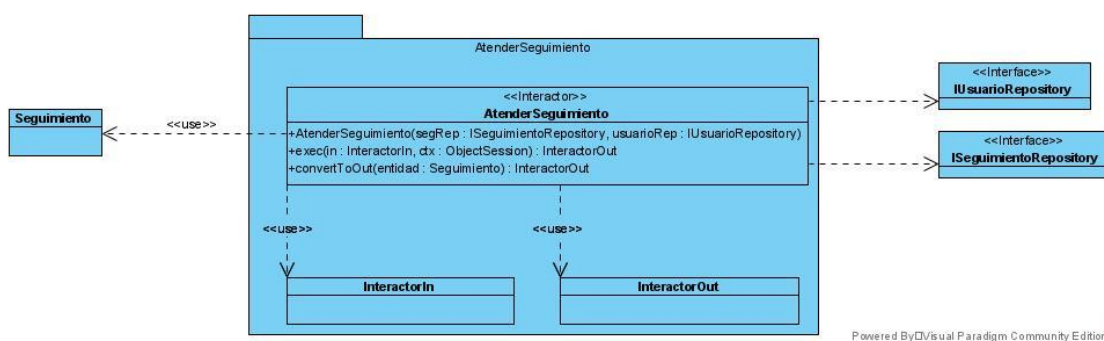


Sprint 4: Interactors

1. Como doctor quiero atender una solicitud de seguimiento, para realizar una video llamada con el paciente y realizar la respectiva toma de datos para el registro del seguimiento.

Figura 92

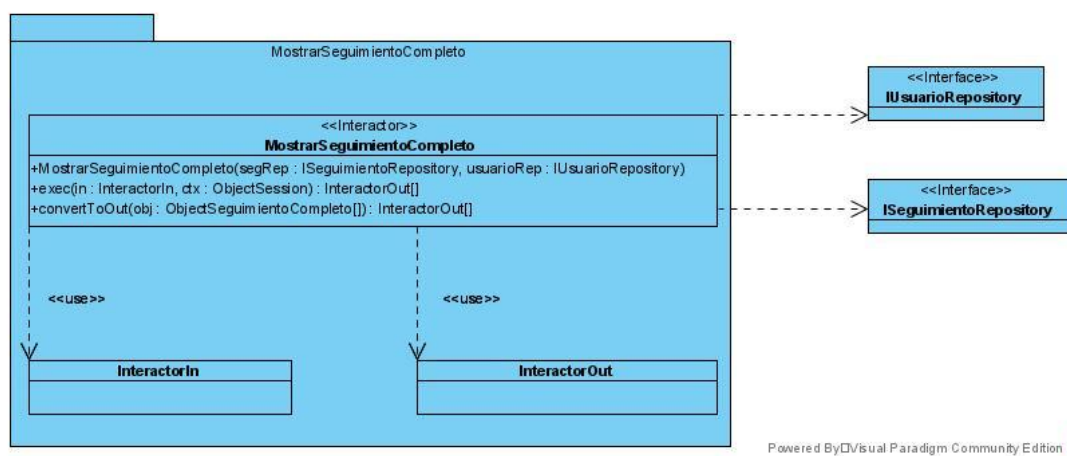
Sprint 4 – Interactor Atender seguimiento



2. Como paciente, doctor, director quiero ver el total de seguimientos de un solo paciente para ver el historial de seguimientos tanto solicitados como atendidos, agrupados por días.

Figura 93

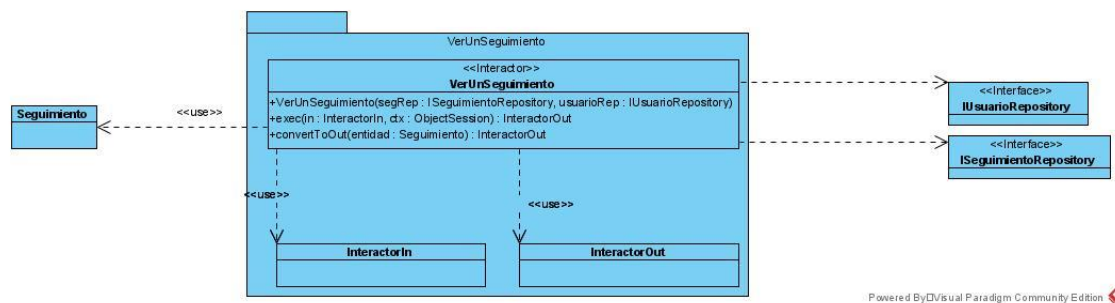
Sprint 4 – Interactor Mostrar seguimiento completo



- Como doctor, paciente quiero consultar un seguimiento para conocer el estado del paciente ese día en ese seguimiento.

Figura 94

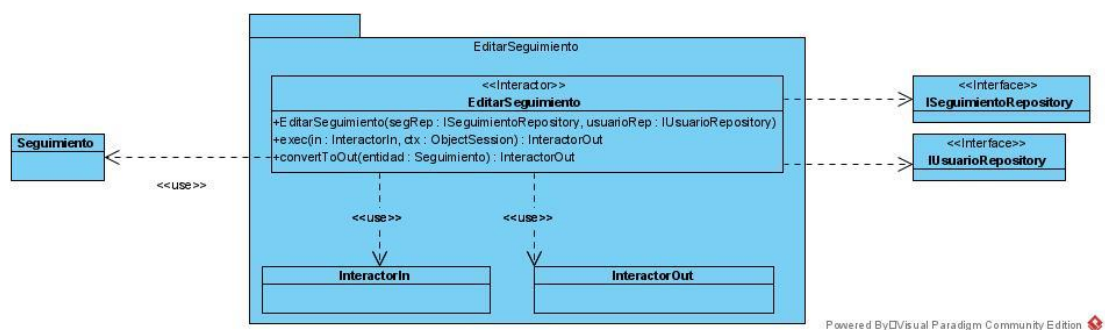
Sprint 4 - Interactor Ver un seguimiento



- Como doctor quiero editar un seguimiento para agregar mis observaciones en los respectivos campos, de acuerdo a la información obtenida en una llamada.

Figura 95

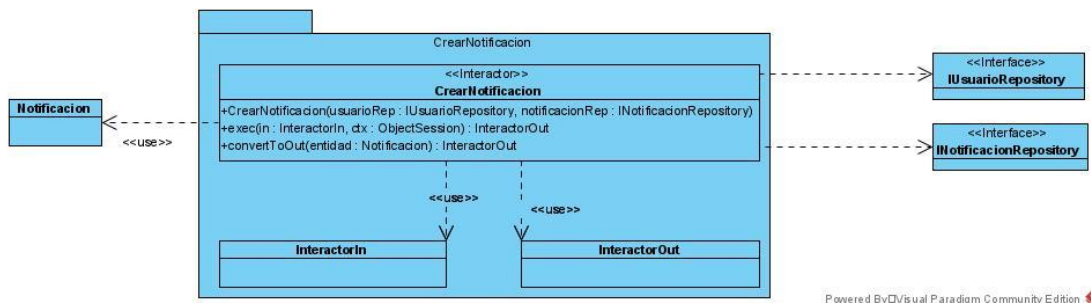
Sprint 4 – Interactor Editar un seguimiento



- Como root, administrador, doctor, quiero crear una notificación para informar al paciente de que se va a realizar una video llamada.

Figura 96

Sprint 4 – Interactor Crear una notificación

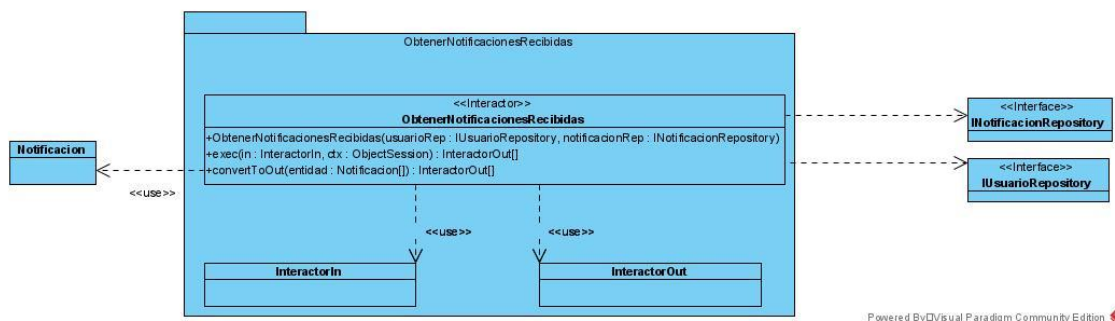


Sprint 5: Interactors

- Como root, administrador, doctor, paciente quiero obtener/Listar mis notificaciones recibidas, enviadas por otro usuario para estar informado respecto a lo que me envían otros usuarios.

Figura 97

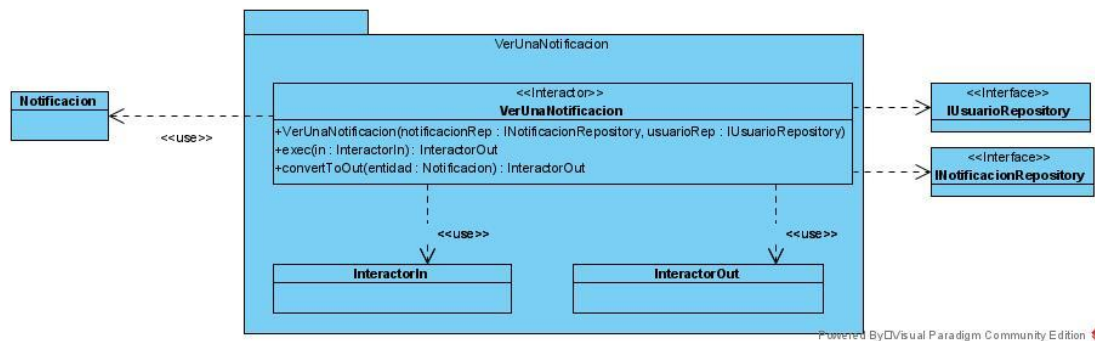
Sprint 5 – Interactor Obtener notificaciones recibidas



- Como root, administrador, doctor, paciente quiero consultar una notificación para acceder al enlace de la llamada o a su contenido.

Figura 98

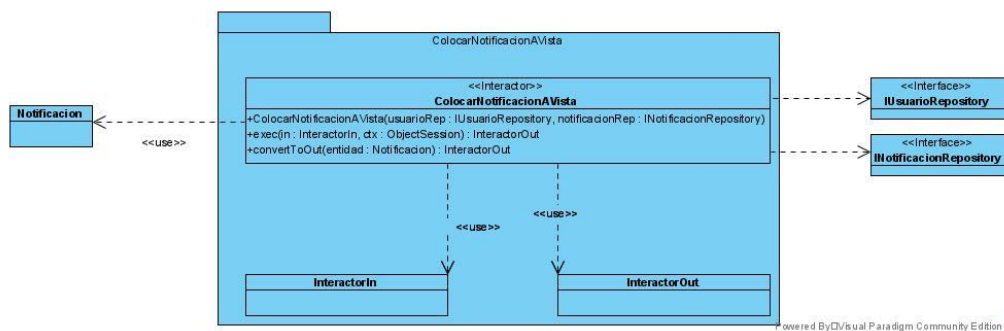
Sprint 5 – Interactor Ver una notificación



- Como doctor quiero comunicarme con el paciente a través de una video llamada con el fin de obtener los datos y realizar el seguimiento.
La video llamada es posible a través del API de Agora.io, la cual permite integrar este servicio a la aplicación web por medio de llamadas REST y sockets en tiempo real.
- Como root, doctor, paciente, administrador, quiero que las notificaciones se coloquen como vistas para cuando sean leídas estas cambien a ese estado. (Esto es interno - servidor - y se muestra a través del API).

Figura 99

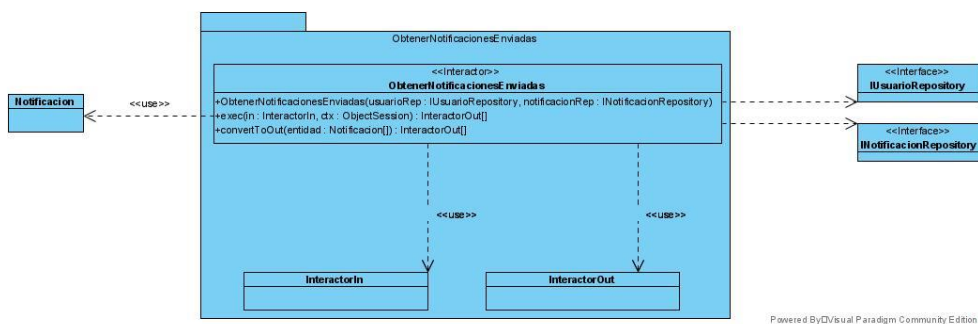
Sprint 5 – Interactor Colocar notificación como vista(leída)



5. Como root, doctor, paciente, administrador quiero obtener/Listar notificaciones enviadas para mostrarlo en el api graphql (Esto es interno - servidor - y se muestra a través del API).

Figura 100

Sprint 5 – Interactor Obtener notificaciones enviadas



6. Como root y director quiero obtener los datos de los pacientes por role y hospital, para visualizar sus coordenadas en un mapa.
Es posible visualizar las coordenadas en un mapa, gracias a los datos devueltos por el el interactor “VerMapaDatos” que se puede visualizar en la historia de usuario número 2 del sprint 6.

Sprint 6: Interactors

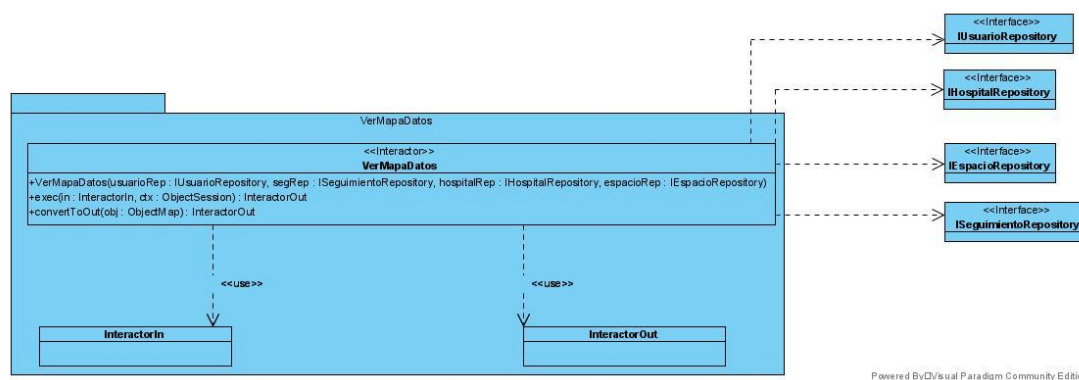
1. Como root y director quiero obtener los datos de los pacientes por día y diagnóstico, para visualizar sus coordenadas en un mapa.

Es posible visualizar las coordenadas en un mapa, gracias a los datos devueltos por el el interactor “VerMapaDatos” que se puede visualizar en la historia de usuario número 2 de este sprint.

2. Como root y director quiero obtener los datos de los pacientes por diagnóstico, para visualizar sus coordenadas en un mapa.

Figura 101

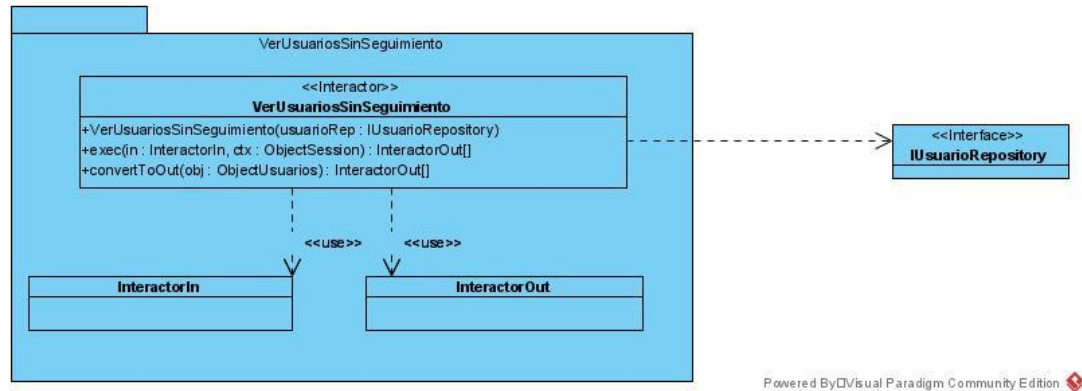
Sprint 6 – Interactor Ver mapa



3. Como director quiero un listado de los usuarios sin seguimientos por día para tener conocimiento de pacientes que ya no están registrando su información.

Figura 102

Sprint 6 – Ver usuarios sin seguimientos



4. Como root y director quiero un contador de pacientes por rol y hospital para visualizarlo en la interfaz.

El interactor se puede visualizar en el sprint 7 en la historia de usuario número 1.

5. Como root y director quiero un contador de pacientes por día y diagnóstico para visualizarlo en la interfaz y como gráfica.

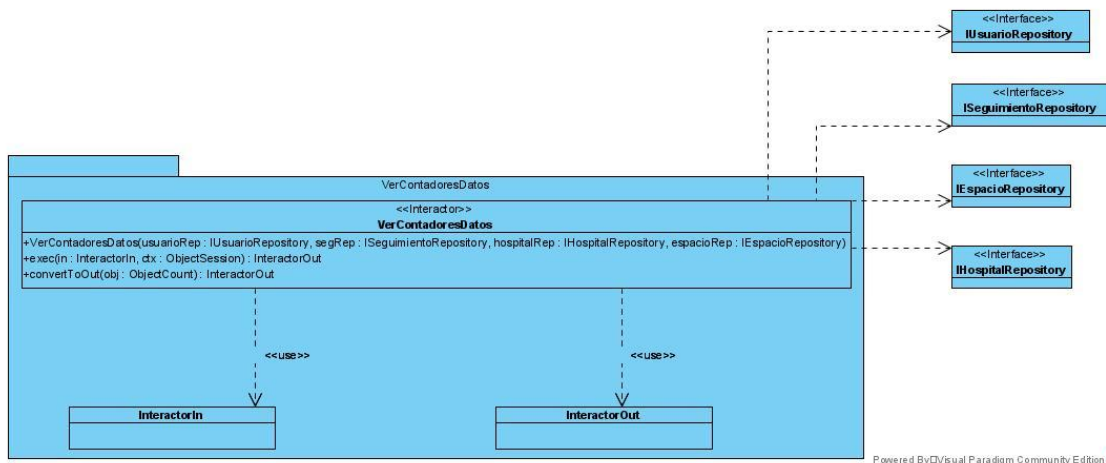
El interactor se puede visualizar en el sprint 7 en la historia de usuario número 1,

Sprint 7: Interactors

1. Como root y director quiero un contador de pacientes por diagnóstico para visualizarlo en la interfaz y como gráfica.

Figura 103

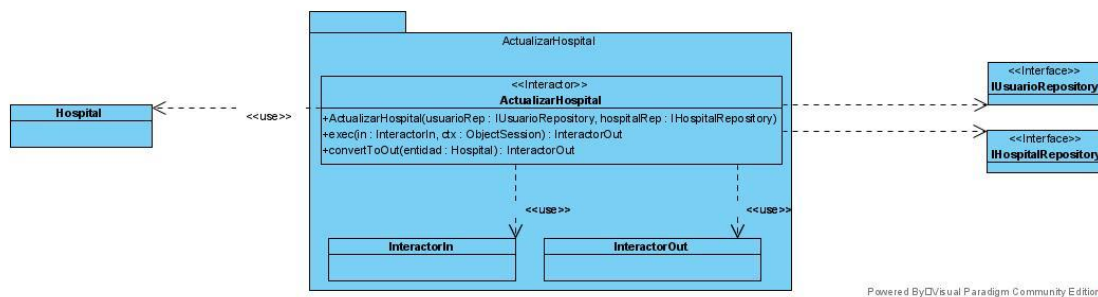
Sprint 7 – Interactor Ver contadores



2. Como root quiero actualizar un hospital para realizar cambios en los campos.

Figura 104

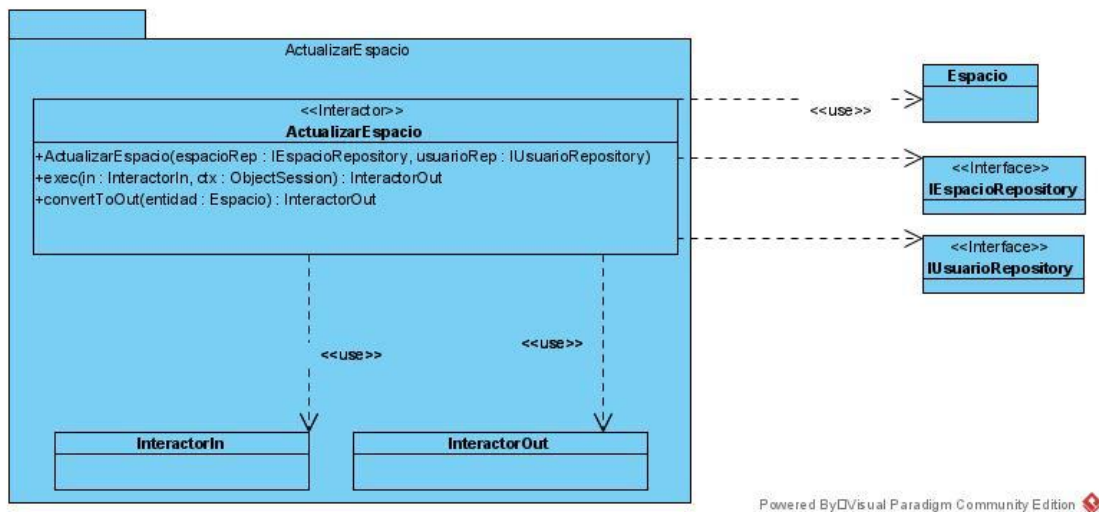
Sprint 7 – Interactor Actualizar hospital



3. Como root quiero actualizar un espacio, para cambiar sus datos dentro de la aplicación.

Figura 105

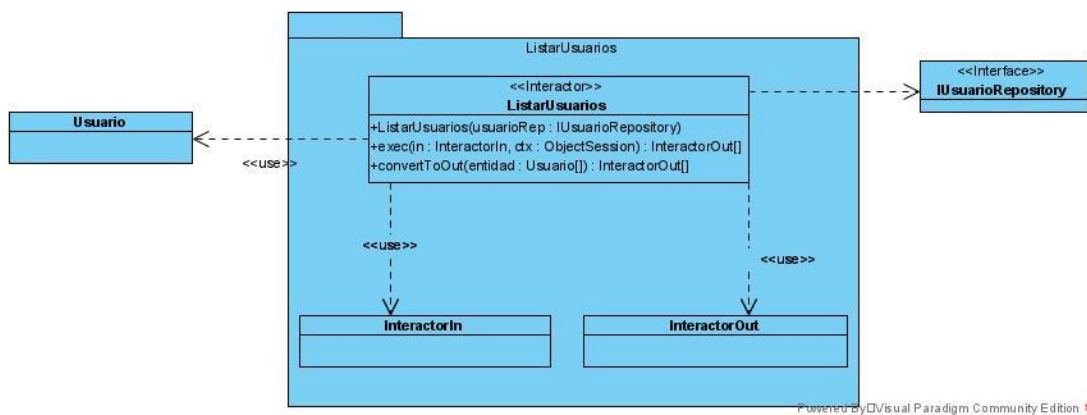
Sprint 7 – Interactor Actualizar espacio



4. Como root, administrador, doctor, quiero listar usuarios para ver los usuarios actualmente registrados en la aplicación.

Figura 106

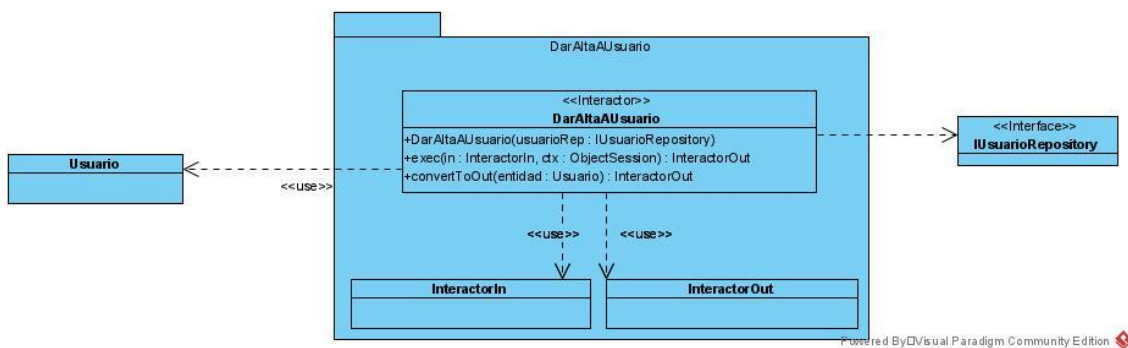
Sprint 7 – Interactor Listar usuarios



- Como usuario doctor, paciente quiero un formulario o campo de motivo de alta, para colocar el motivo de alta de un paciente.

Figura 107

Sprint 7 – Interactor Dar de alta a un usuario

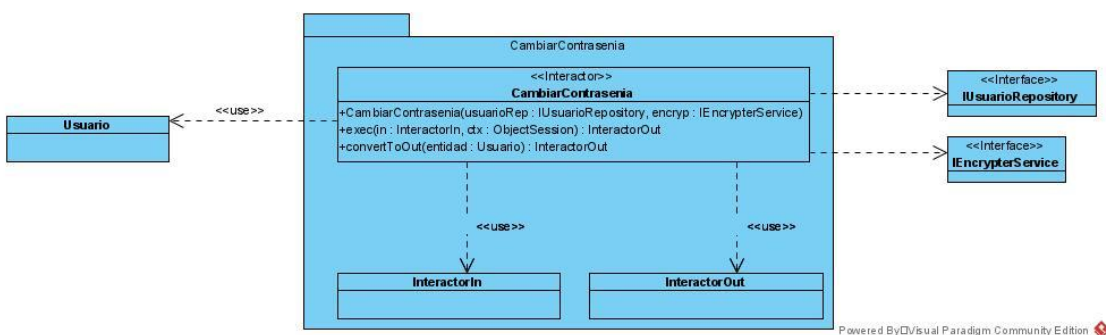


Sprint 8: Interactors

- Como usuario quiero un formulario para cambiar mi contraseña en la aplicación.

Figura 108

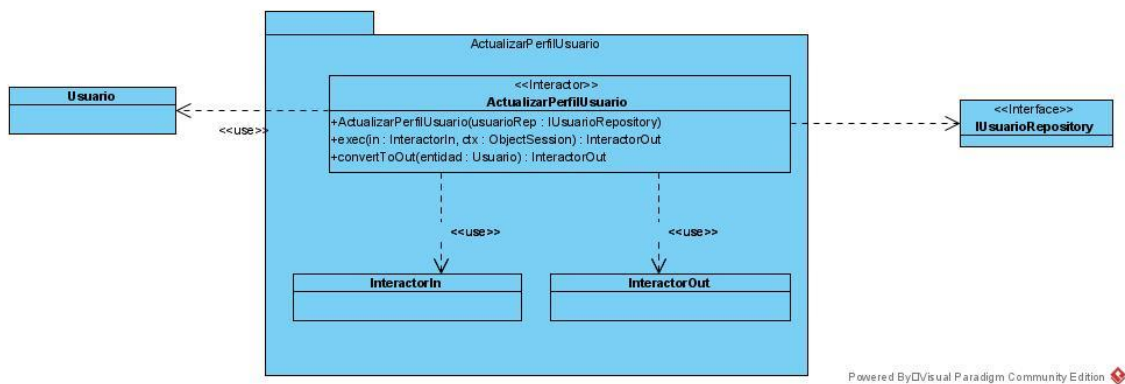
Sprint 8 – Interactor Cambiar contraseña



- Como usuario quiero actualizar mis datos para que los cambios sean reflejados en la vista de perfil.

Figura 109

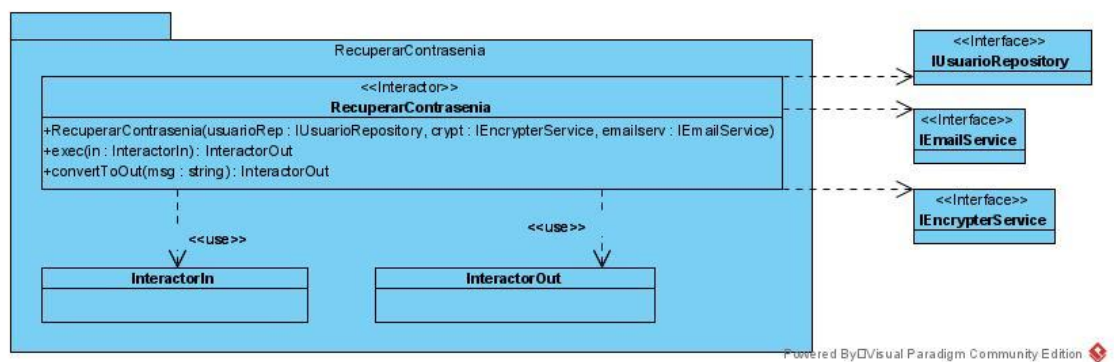
Sprint 8 – Interactor Actualizar perfil de usuario



- Como usuario quiero un botón de olvide mi contraseña o similar para recuperar mi contraseña.

Figura 110

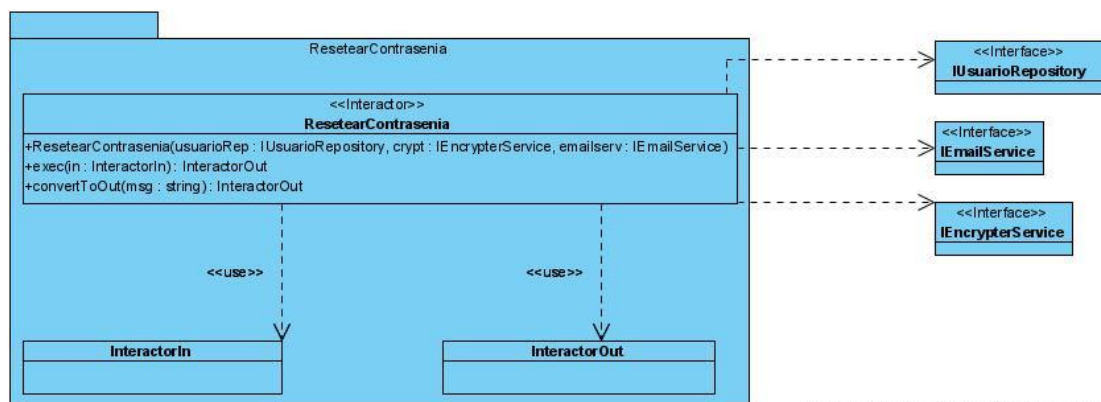
Sprint 8 – Interactor Recuperar contraseña



- Como usuario quiero recibir un enlace a mi correo en caso de recuperar mi contraseña para ser redirigido a una vista y colocar mi nueva contraseña.

Figura 111

Sprint 8 – Interactor Resetear contraseña



3.4.2. Capa de detalle

La capa de detalle comprende a clases que instancian a los interactores y clases concretas que implementan interfaces o clases abstractas que expone el dominio y a su vez serán inyectadas al constructor de los interactores.

Servicios

A continuación, se muestra el diagrama de clases de los servicios concretos implementando su respectiva interfaz.

Figura 112

Detalle – Clase concreta del servicio de encriptado

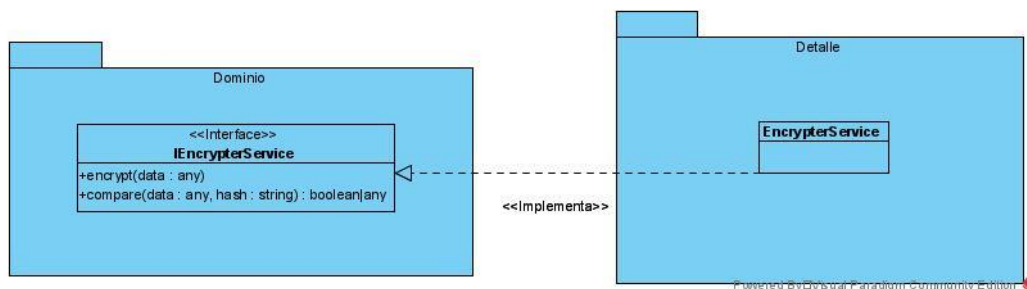
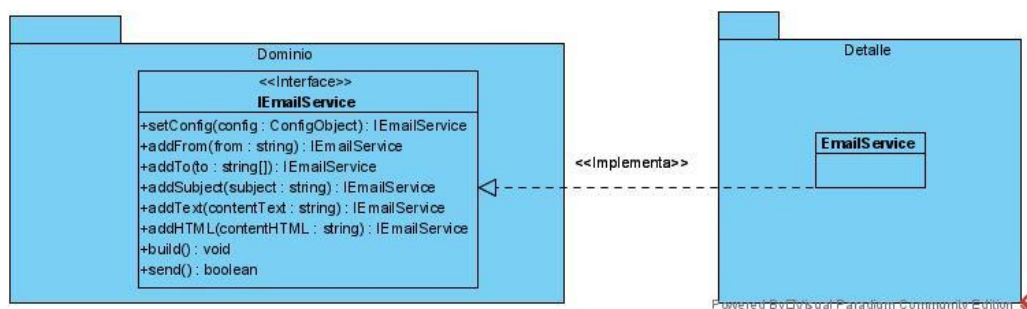


Figura 113

Detalle – Clase concreta del servicio de correo

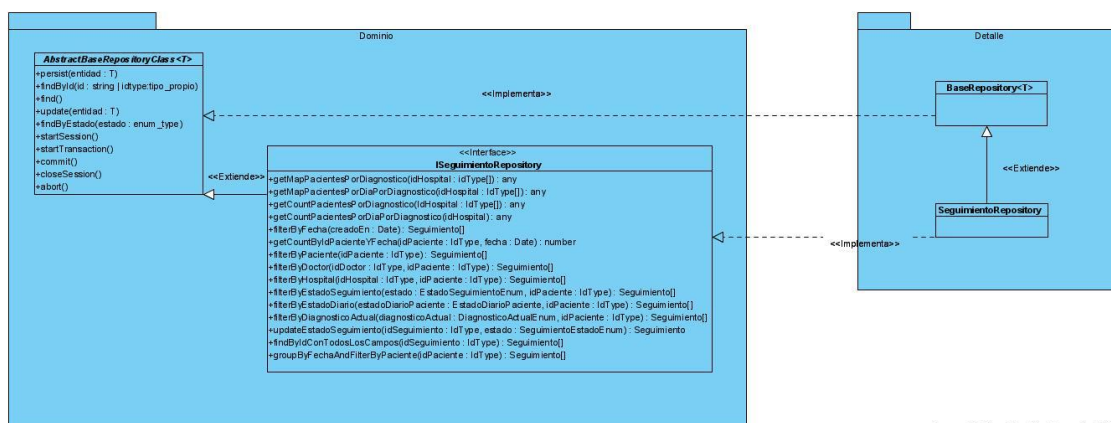


Persistencia

A continuación, se muestra el diagrama de clases de las clases de persistencia concretas implementando su respectiva interfaz y extendiendo de una clase base.

Figura 114

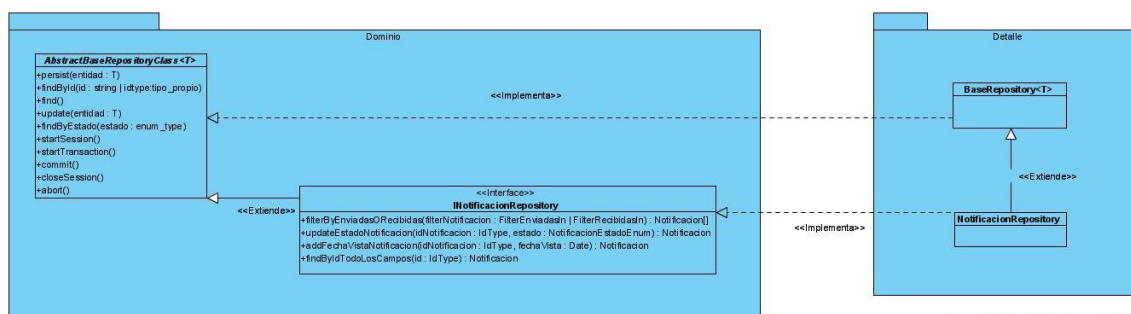
Detalle – Clase concreta del repositorio de seguimiento



Powered By/Visual Paradigm Community Edition

Figura 115

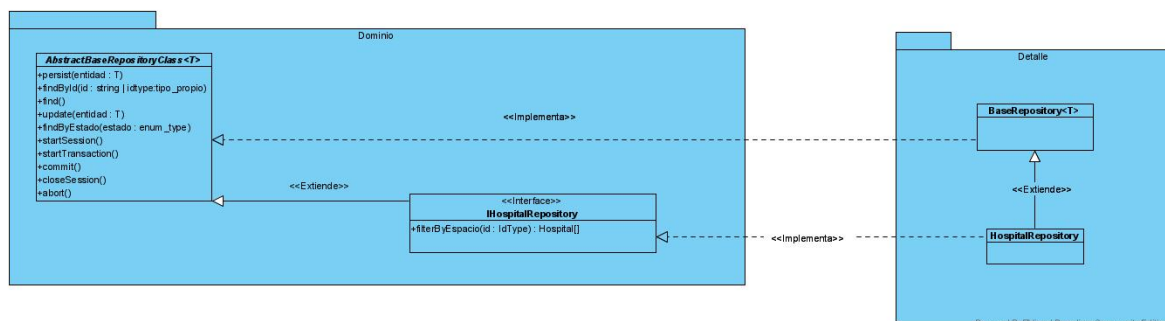
Detalle – Clase concreta del repositorio de notificación



Powered By/Visual Paradigm Community Edition

Figura 116

Detalle – Clase concreta del repositorio de hospital



Powered By/Visual Paradigm Community Edition

Figura 117

Detalle - Clase concreta del repositorio de espacio

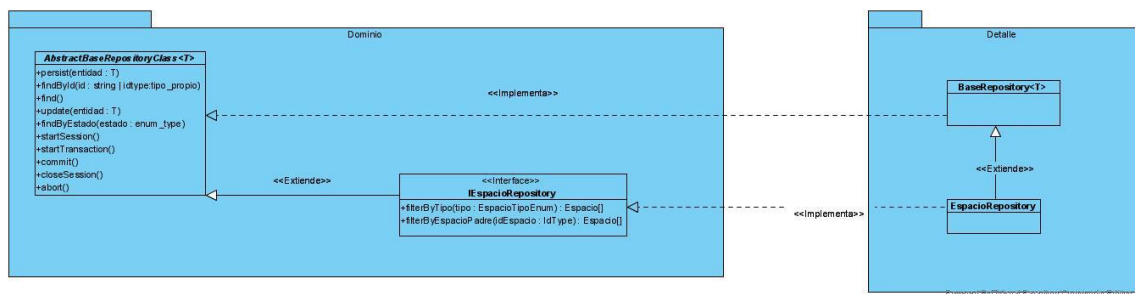
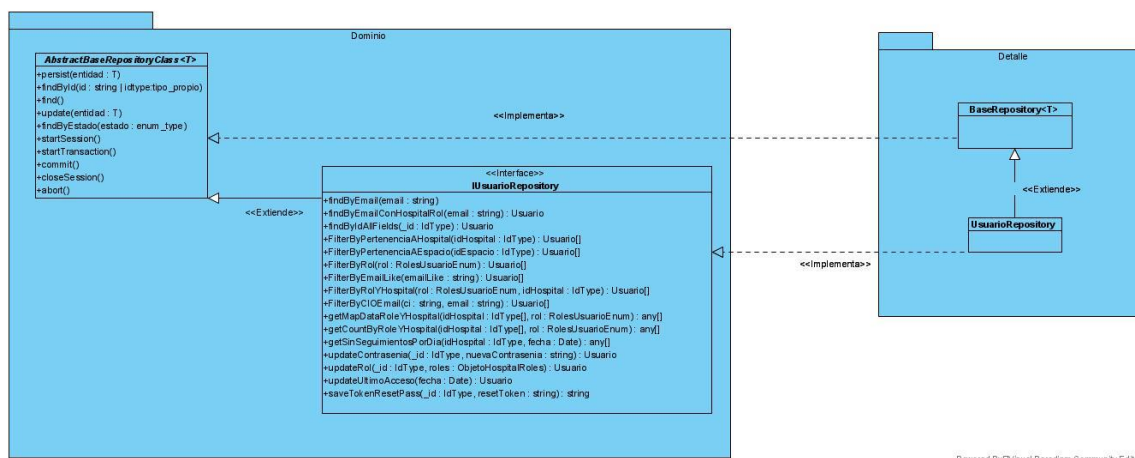


Figura 118

Detalle - Clase concreta del repositorio de usuario



API resolvers

Los resolvers son clases encargadas de resolver la petición que viene del lado del cliente, lo hace a través de los métodos query en el caso de una consulta o mutation en el caso de una actualización. Dentro de estas clases se instancian las clases concretas de servicios y persistencia para inyectarse en el constructor de las instancias de los interactores.

Figura 119

Detalle – Resolver usuario

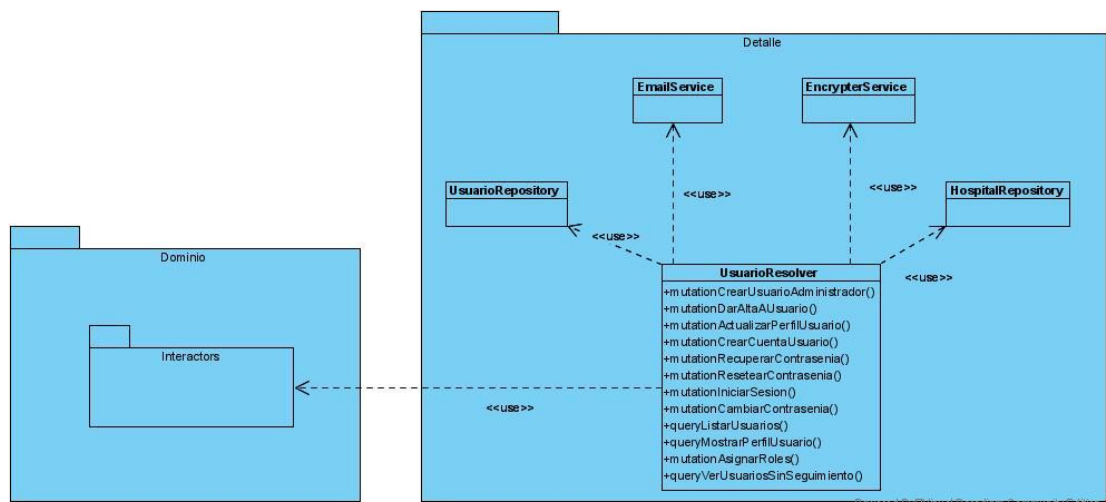


Figura 120

Detalle – Resolver seguimiento

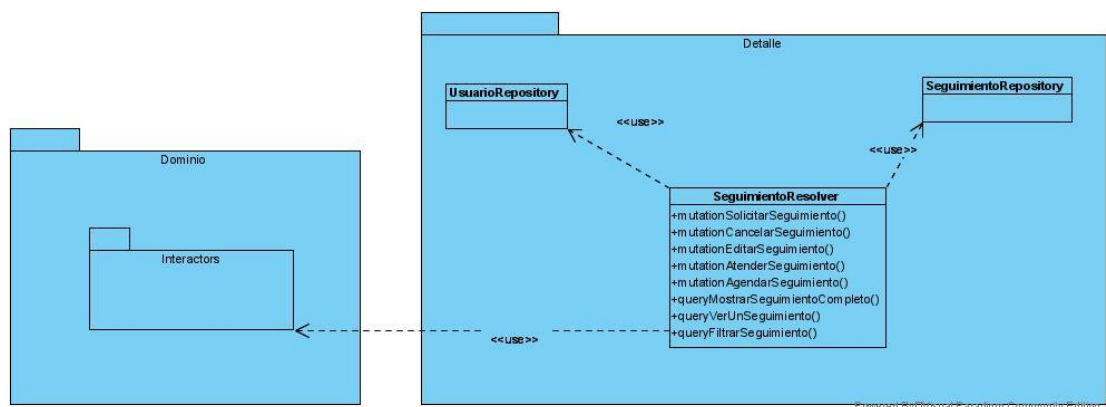


Figura 121

Detalle – Resolver notificación

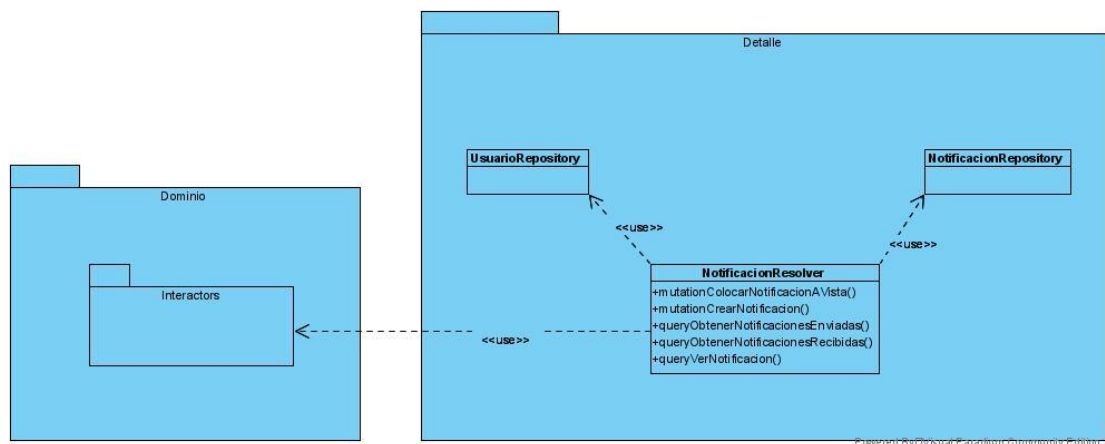


Figura 122

Detalle – Resolver hospital

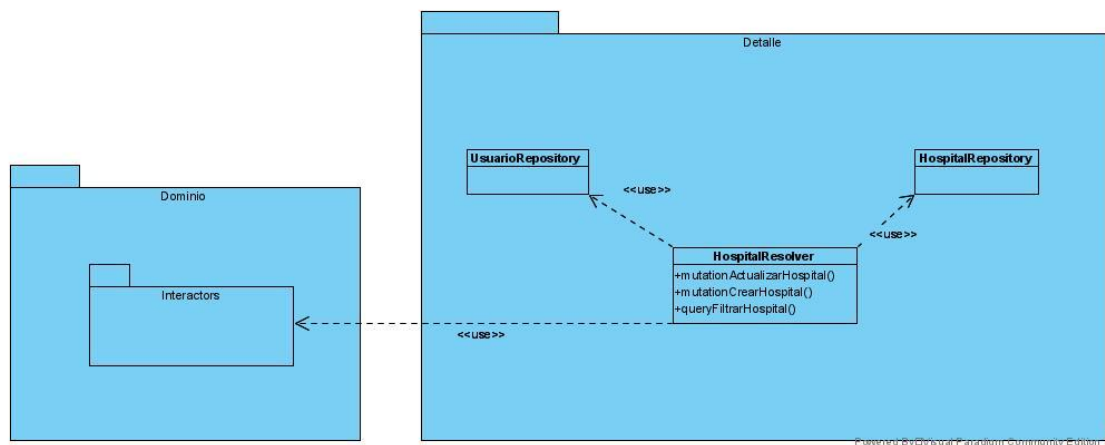


Figura 123

Detalle – Resolver espacio

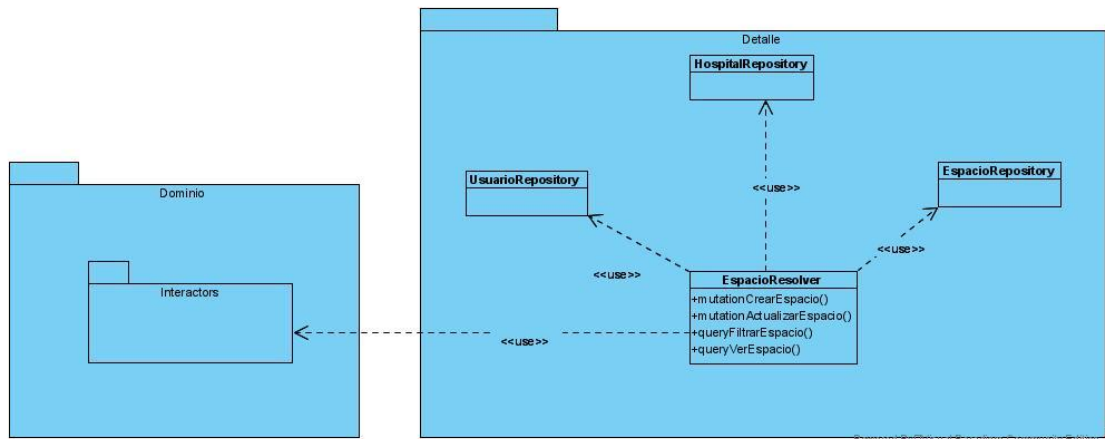


Figura 124

Detalle – Resolver Data(Para interacto de mapa y contadores)

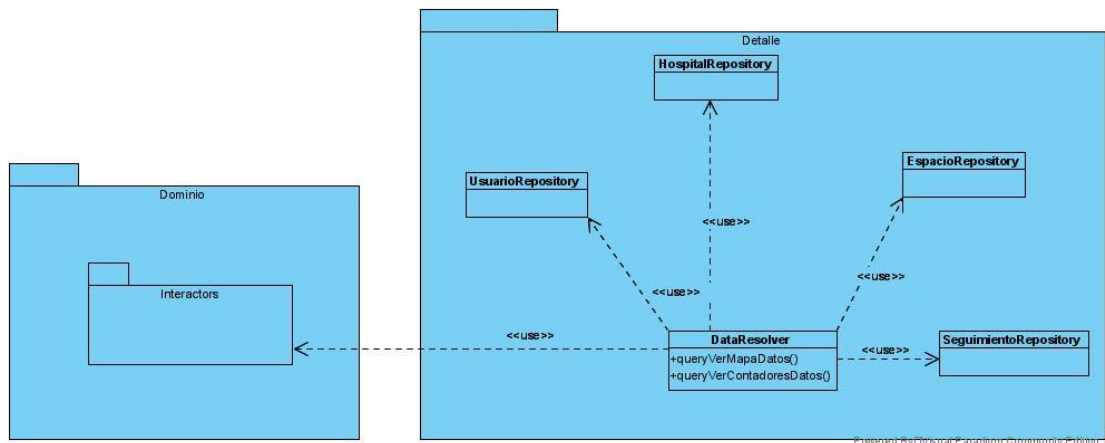
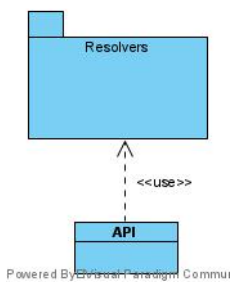


Figura 125

Detalle - API



3.4.3. Cliente: Front end

A continuación, se presentan las vistas principales de la aplicación web, las cuales fueron desarrolladas con el framework de Angular js y se basaron en las maquetas presentadas en el apartado de diseño del presente documento.

Figura 126

Cliente – Vista de inicio de sesión

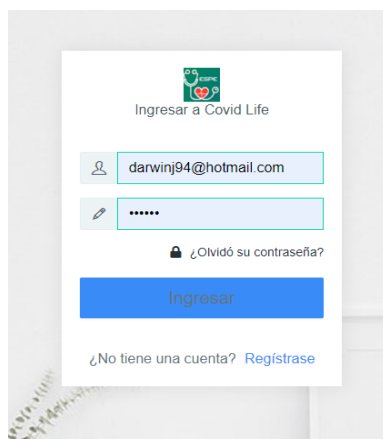


Figura 127

Cliente – Vista de registro de usuario

Por favor ingrese sus datos de contacto.

Nombres	Apellidos
<input type="text" value="Nombres"/>	<input type="text" value="Apellidos"/>
Email	Contraseña
<input type="text" value="Correo"/>	<input type="text" value="Contraseña"/>
Cédula	Fecha de nacimiento
<input type="text" value="Cédula o pasaporte"/>	<input type="text" value="dd/mm/aaaa"/>
Género	Teléfono
<input type="radio"/> Masculino <input type="radio"/> Femenino <input type="radio"/> Preferlo no decirlo	<input type="text" value="Teléfono"/>
Dirección	
<input type="text" value="Dirección"/>	

Siguiete

Figura 128

Cliente – Vista de tabla (pacientes)

Pacientes

Pacientes actualmente registrados

[Agregar usuarios](#)

#	Nombre	Apellido	Correo	cédula	Teléfono	Acciones
1	Darwin	Jair	darwinj94@hotmail.com	1600666026	0993066810	
2	Fernanda	Paulina	fernanda@espe.edu.ec	1100444026	123123123	
3	Glenda	Lisbeth	glenda@espe.edu.ec	12121212312	1234567890	
4	Jeovana	Alba	jeovana@espe.edu.ec	1200334567	1231231234	
5	Susana	Lucia	susana@espe.edu.ec	1099777026	9897897896	
6	David	Collaguazo	dcollaguazo@hotmail.com	1709288441	09799999898	

Figura 129

Cliente – Vista de seguimientos

COVID LIFE

ESPE Sede Latacunga

Darwin Jair18

Doctor 2

Administrador 3

Estadísticas 3

Seguimientos

Seleccione cada tarjeta para mover al estado correspondiente

Seguimiento Solicitud de Video Llamada Agendadas Atendidas

Glenda Lisbeth

🕒 | 4:05 PM
IGUAL_EVOLUCION
📅 | Primer seguimiento del día

Glenda Lisbeth

🕒 | 4:06 PM
EMPEORANDO
📅 | Segundo seguimiento del día

Susana Lucia

🕒 | 4:03 PM
EMPEORANDO
📅 | Me siento fatigada

Susana Lucia

🕒 | 4:03 PM
EMPEORANDO
📅 | Me siento fatigada

Susana Lucia

🕒 | 4:04 PM
MEJOR_EVOLUCION
📅 | Me siento mejor

Figura 130*Cliente – Vista de resumen de seguimientos*

Datos personales Resumen

Día y Fecha	Día 1 9/6/20	Día 2 9/7/20
Evolución	4:39 PM - (EMPEORANDO)	5:48 AM - (EMPEORANDO)
	5:04 PM - (EMPEORANDO)	5:47 AM - (EMPEORANDO)
	- ()	5:47 AM - (IGUAL_EVOLUCION)
Temperatura	4:39 PM - (37.2)	5:48 AM - (37.2)5:47 AM - (37.3)
	5:04 PM - (37.3)- ()	5:47 AM - (37.3)

Figura 131*Cliente - Vista de seguimientos*

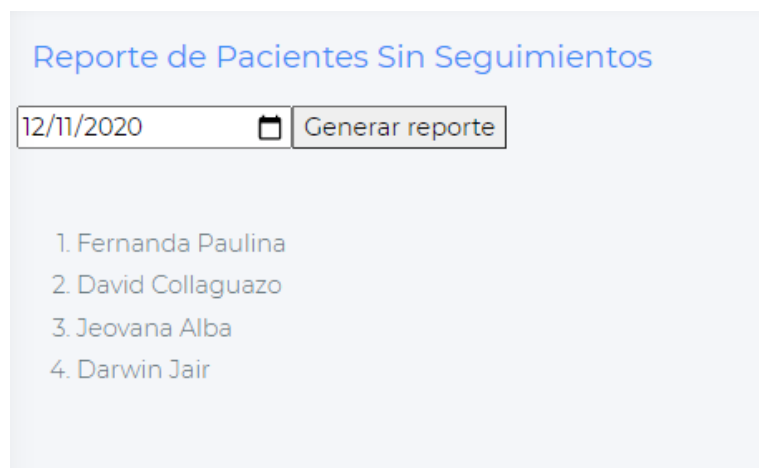
Atender un seguimiento

Atender Datos personales Resumen Temperatura

Día y Fecha	Día 1 8/11/2020	Día 2 9/11/2020	Día 3 10/11/2020	Día 4 12/11/2020
Oxígeno	9:09 PM - (0) - (0)	11:01 AM - (79)	9:47 AM - (80)	4:03 PM
Dificultad Respirar	8:30 PM - (SI)	11:01 AM - (SI)	9:48 AM - (NO)	4:03 PM
	9:09 PM - (NO)	11:01 AM - (NO)	9:47 AM - (SI)	4:03 PM
	8:28 PM - (SI)			
Diagnóstico Actual	8:30 PM - (PROBABLE)			
	9:09 PM - (PROBABLE)	11:01 AM - () 11:01 AM - ()	9:48 AM - () 9:47 AM - ()	4:03 PM 4:03 PM
	8:28 PM - (PROBABLE)			
¿Está en Cuarentena?	NO	NO	NO	NO

Figura 132

Cliente – Vista de pacientes sin seguimientos

**Figura 133**

Cliente – Vista de atención a través de una video llamada

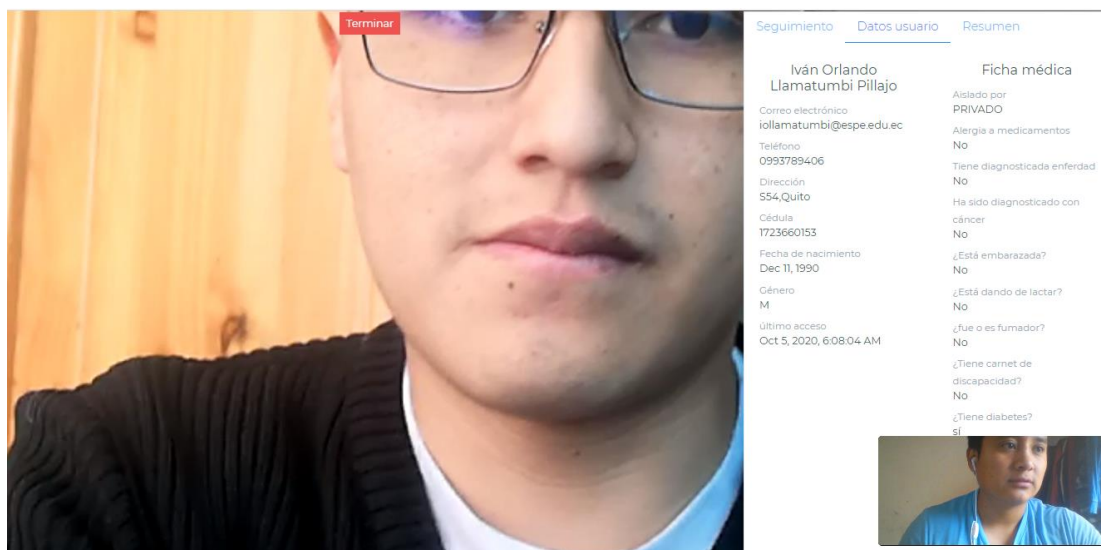


Figura 134

Cliente – Vista de atención de un seguimiento formulario

Atender un seguimiento

Atender Datos personales Resumen Temperatura

Evolución (No editable)
MEJOR_EVOLUCION

Ritmo cardíaco (Opcional)
130

Dificultad para respirar (Opcional)
 Sí
 No

Nota enviada por el paciente (No editable)
Me siento mejor

Aislado desde (opcional)
dd/mm/aaaa

Diagnóstico actual (Obligatorio)
PROBABLE

Temperatura (No editable)
39

Saturación de oxígeno (Opcional)
80

Exámen (Opcional)
[Dropdown menu]

Escriba una breve observación (Opcional)
[Text area]

Aislado hasta (opcional)
dd/mm/aaaa

Atendido por:
SEGUIMIENTO SIN ATENDER

Atender

Cerrar

Figura 135

Cliente – Vista de ficha médica de un paciente

Atender Datos personales Resumen Temperatura

Susana Lucia

Correo electrónico
susana@espe.edu.ec

Teléfono
9897897896

Dirección
Calle las Hortencias

Cédula
1099777026

Edad
50 años

Género
F

último acceso
11/12/20, 4:02 PM

Ficha médica

Aislado por
ISSFA

Alergia a medicamentos
no

Tiene diagnosticada enfermedad
no

Ha sido diagnosticado con cáncer
No

¿Está embarazada?
No

¿Está dando de lactar?
No

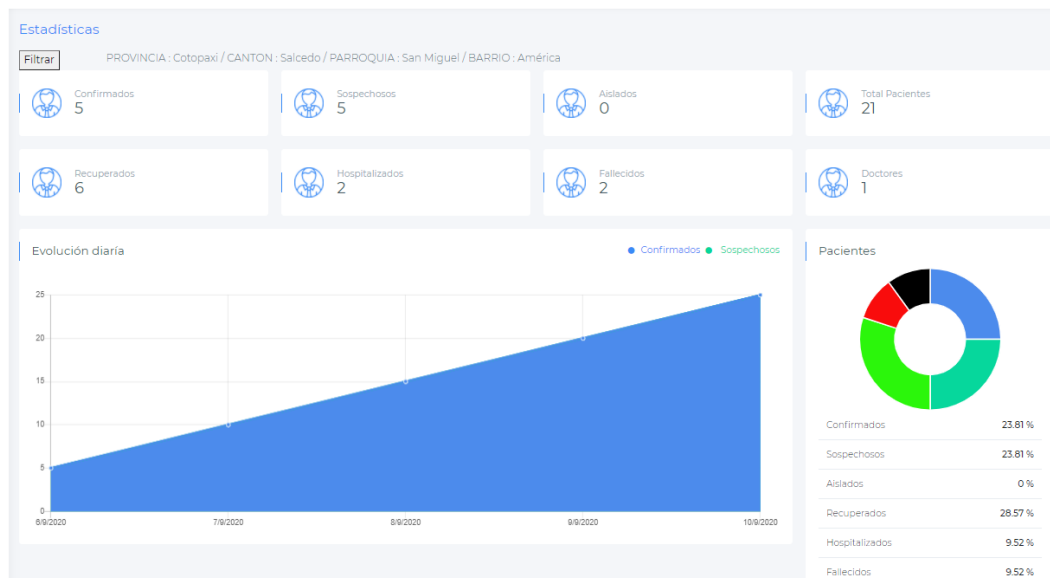
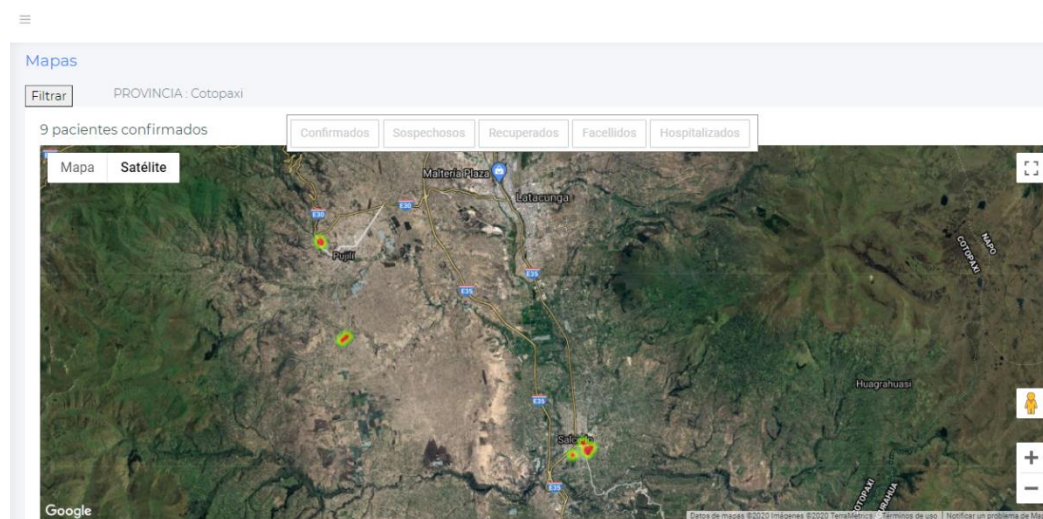
¿fue o es fumador?
No

¿Tiene carnet de discapacidad?
No

¿Tiene diabetes?
No

¿Tiene la presión alta??
No

Familiares en cerco epidemiológico
3

Figura 136*Cliente – Vista de gráficas***Figura 137***Cliente – Vista de mapa*

3.5. Pruebas e Implementación

3.5.1. Pruebas

Las pruebas se realizan en base a los criterios de aceptación de cada una de las historias de usuario mediante una lista de chequeo.

Tabla 28

Lista de chequeo N° 1

Lista de chequeo N° 1			
Historia de usuario: 1. Como usuario root, quiero tener el rol de super administrador para tener control sobre la aplicación web y asignar roles de administrador.			
Criterio de aceptación	Si	No	Observación
El root es el único que asigna el rol de administrador.	X		
La aplicación tiene un único rol de root.	X		

Tabla 29

Lista de chequeo N° 2

Lista de chequeo N° 2			
Historia de usuario: Como root quiero crear espacios para sectorizar los hospitales en espacios físicos.			
Criterio de aceptación	Si	No	Observación
Se permite la creación de espacio solo al usuario root.	X		
Un espacio contiene un campo de espacio que representa a un espacio padre.	X		
Un espacio contiene un campo tipo (barrio, parroquia, cantón, provincia).	X		

Tabla 30*Lista de chequeo N° 3*

Lista de chequeo N° 3			
Historia de usuario: Como root, administrador quiero filtrar los espacios para listarlos por tipo y espacio padre.			
Criterio de aceptación	Si	No	Observación
Se visualiza el listado de los espacios según el filtro aplicado.	X		

Tabla 31*Lista de chequeo N° 4*

Lista de chequeo N° 4			
Historia de usuario: Como root, director, doctor quiero ver un espacio para conocer sus campos incluyendo latitud y longitud.			
Criterio de aceptación	Si	No	Observación
Se visualiza los campos de un espacio.	X		

Tabla 32*Lista de chequeo N° 5*

Lista de chequeo N° 5			
Historia de usuario: Como root quiero crear hospitales para agrupar un administrador, pacientes y doctores.			
Criterio de aceptación	Si	No	Observación
Se permite la creación de un hospital solo al usuario root.	X		

Tabla 33*Lista de chequeo N° 6*

Lista de chequeo N° 6			
Historia de usuario: Como usuario registrado en la aplicación quiero filtrar hospitales por campo espacio para obtener los hospitales que pertenezcan a un espacio en específico.			
Criterio de aceptación	Si	No	Observación
Se obtiene un listado de los hospitales que pertenecen a un espacio en específico.	X		

Tabla 34*Lista de chequeo N° 7*

Lista de chequeo N° 7			
Historia de usuario: Como usuario quiero crear una cuenta para poder ser asignado a un hospital con mi rol.			
Criterio de aceptación	Si	No	Observación
El usuario crea una cuenta dentro de la aplicación.	X		
Solo en el caso de paciente, selecciona un hospital donde ser atendido.	X		
El usuario crea una cuenta dentro de la aplicación.	X		

Tabla 35*Lista de chequeo N° 8*

Lista de chequeo N° 8			
Historia de usuario: Como usuario quiero iniciar sesión para acceder a mi cuenta dentro de la aplicación.			
Criterio de aceptación	Si	No	Observación
El usuario puede iniciar sesión y visualizar la ventana de inicio y seleccionar un hospital donde actuar con su rol asignado por un administrador, en el caso de paciente no es necesario que se asigne un rol.	X		

Tabla 36*Lista de chequeo N° 9*

Lista de chequeo N° 9			
Historia de usuario: Como administrador quiero agregar usuarios a un hospital y asignar un rol dentro del hospital.			
Criterio de aceptación	Si	No	Observación
El administrador pertenece a un hospital y está permitido agregar a otros usuarios y asignarles un rol dentro del hospital.	X		

Tabla 37*Lista de chequeo N° 10*

Lista de chequeo N° 10			
Historia de usuario: Como usuario root, administrador, doctor, paciente quiero una vista de perfil de usuario, para ver los datos de mi perfil y de otros usuarios, excepto en el caso de paciente, solo ve de su perfil.			
Criterio de aceptación	Si	No	Observación
Se visualiza el perfil de un usuario.	X		
Solo el paciente y el director no pueden ver el perfil de otro usuario que no sea su propio perfil.	X		

Tabla 38*Lista de chequeo N° 11*

Lista de chequeo N° 11			
Historia de usuario: Como paciente quiero solicitar un seguimiento para enviar sus signos y obtener una valoración de un doctor.			
Criterio de aceptación	Si	No	Observación
El paciente está permitido solicitar 3 seguimientos por día.	X		

Tabla 39*Lista de chequeo N° 12*

Lista de chequeo N° 12			
Historia de usuario: Como doctor, paciente quiero cancelar una solicitud de seguimiento para evitar ser atendido o para no agendar a un paciente.			
Criterio de aceptación	Si	No	Observación
El doctor y el paciente están permitidos cancelar una solicitud.	X		

Tabla 40*Lista de chequeo N° 13*

Lista de chequeo N° 13			
Historia de usuario: Como paciente, doctor, administrador, director, quiero filtrar seguimientos para listar cuales fueron atendidos, agendados o están solicitados.			
Criterio de aceptación	Si	No	Observación
Se visualiza los seguimientos distribuidos en atendidos, agendados o que están solicitados.	X		

Tabla 41*Lista de chequeo N° 14*

Lista de chequeo N° 14			
Historia de usuario: Como doctor quiero agendar un seguimiento para que no existan conflictos con el seguimiento respecto a un doctor con otro, y evitar la saturación de llamadas al mismo tiempo para un doctor.			
Criterio de aceptación	Si	No	Observación
El doctor puede agendar un seguimiento, si este seguimiento es solicitado con llamada.	X		

Tabla 42*Lista de chequeo N° 15*

Lista de chequeo N° 15			
Historia de usuario: Como doctor quiero atender una solicitud de seguimiento, para realizar una video llamada con el paciente y realizar la respectiva toma de datos para el registro del seguimiento.			
Criterio de aceptación	Si	No	Observación
El doctor puede atender un seguimiento, si este está agendado.	X		

Tabla 43*Lista de chequeo N° 16*

Lista de chequeo N° 16			
Historia de usuario: Como paciente, doctor, director quiero ver el total de seguimientos de un solo paciente para ver el historial de seguimientos tanto solicitados como atendidos, agrupados por días.			
Criterio de aceptación	Si	No	Observación
Se visualiza en una tabla la clasificación por día de los seguimientos realizados por un solo paciente.	X		

Tabla 44*Lista de chequeo N° 17*

Lista de chequeo N° 17			
Historia de usuario: Como doctor, paciente quiero consultar un seguimiento para conocer el estado del paciente ese día en ese seguimiento.			
Criterio de aceptación	Si	No	Observación
Se visualiza un seguimiento específico de un día específico.	X		

Tabla 45*Lista de chequeo N° 18*

Lista de chequeo N° 18			
Historia de usuario: Como doctor quiero editar un seguimiento para agregar mis observaciones en los respectivos campos, de acuerdo a la información obtenida en una llamada.			
Criterio de aceptación	Si	No	Observación
El seguimiento permite la actualización y se ven reflejados estos cambios en las vistas donde es consultado.	X		
El seguimiento no puede ser editado si todavía no es atendido.	X		
El seguimiento no puede ser editado si no ha sido atendido por el mismo doctor.	X		
El seguimiento no puede ser editado si no pertenece al día actual.	X		

Tabla 46*Lista de chequeo N° 19*

Lista de chequeo N° 19			
Historia de usuario: Como root, administrador, doctor, quiero crear una notificación para informar al paciente de que se va a realizar una video llamada.			
Criterio de aceptación	Si	No	Observación
La notificación llega al paciente, luego de ser enviada.	X		

Tabla 47*Lista de chequeo N° 20*

Lista de chequeo N° 20			
Historia de usuario: Como root, administrador, doctor, paciente quiero obtener/Listar mis notificaciones recibidas, enviadas por otro usuario para estar informado respecto a lo que me envían otros usuarios.			
Criterio de aceptación	Si	No	Observación
Las notificaciones que recibe un usuario y que son enviadas por otro, pueden ser visualizadas.	X		

Tabla 48*Lista de chequeo N° 21*

Lista de chequeo N° 21			
Historia de usuario: Como root, administrador, doctor, paciente quiero consultar una notificación para acceder al enlace de la llamada o a su contenido.			
Criterio de aceptación	Si	No	Observación
Al seleccionar la notificación me indica su contenido o me redirige a la vista de video llamada.	X		

Tabla 49*Lista de chequeo N° 22*

Lista de chequeo N° 22			
Historia de usuario: Como doctor quiero comunicarme con el paciente a través de una video llamada con el fin de obtener los datos y realizar el seguimiento.			
Criterio de aceptación	Si	No	Observación
La video llamada entre el doctor y el paciente permite la comunicación entre ambos usuarios. Y permite que el doctor tome datos sobre la salud del paciente.	X		

Tabla 50*Lista de chequeo N° 23*

Lista de chequeo N° 23			
Historia de usuario: Como root, doctor, paciente, administrador, quiero que las notificaciones se coloquen como vistas para cuando sean leídas estas cambien a ese estado. (Esto es interno - servidor - y se muestra a través del API).			
Criterio de aceptación	Si	No	Observación
El API GraphQL permite el cambio de estado de una notificación ha vista (leída).	X		

Tabla 51*Lista de chequeo N° 24*

Lista de chequeo N° 24			
Historia de usuario: Como root, doctor, paciente, administrador quiero obtener/Listar notificaciones enviadas para mostrarlo en el api graphql (Esto es interno - servidor - y se muestra a través del API).			
Criterio de aceptación	Si	No	Observación
Al consultarse sobre notificaciones enviadas, el API GraphQL nos responde con la información de las mismas.	X		

Tabla 52*Lista de chequeo N° 25*

Lista de chequeo N° 25			
Historia de usuario: Como root y director quiero obtener los datos de los pacientes por rol y hospital, para visualizar sus coordenadas en un mapa.			
Criterio de aceptación	Si	No	Observación
En el mapa se visualizan las coordenadas de los pacientes agrupados por rol y hospital.	X		

Tabla 53*Lista de chequeo N° 26*

Lista de chequeo N° 26			
Historia de usuario: Como root y director quiero obtener los datos de los pacientes por día y diagnóstico, para visualizar sus coordenadas en un mapa.			
Criterio de aceptación	Si	No	Observación
En el mapa se visualizan las coordenadas de los pacientes agrupados por día y diagnóstico.	X		

Tabla 54*Lista de chequeo N° 27*

Lista de chequeo N° 27			
Historia de usuario: Como root y director quiero obtener los datos de los pacientes por diagnóstico, para visualizar sus coordenadas en un mapa.			
Criterio de aceptación	Si	No	Observación
En el mapa se visualizan las coordenadas de los pacientes agrupados por diagnóstico.	X		

Tabla 55*Lista de chequeo N° 28*

Lista de chequeo N° 28			
Historia de usuario: Como director quiero un listado de los usuarios sin seguimientos por día para tener conocimiento de pacientes que ya no están registrando su información.			
Criterio de aceptación	Si	No	Observación
Se visualiza un listado con los pacientes que no han solicitado seguimientos el día actual.	X		

Tabla 56*Lista de chequeo N° 29*

Lista de chequeo N° 29			
Historia de usuario: Como root y director quiero un contador de pacientes por rol y hospital para visualizarlo en la interfaz.			
Criterio de aceptación	Si	No	Observación
Se visualiza en la interfaz un contador con el número de pacientes por rol y hospital.	X		

Tabla 57*Lista de chequeo N° 30*

Lista de chequeo N° 30			
Historia de usuario: Como root y director quiero un contador de pacientes por día y diagnóstico para visualizarlo en la interfaz y como gráfica.			
Criterio de aceptación	Si	No	Observación
Se visualiza en la interfaz una gráfica con el número de pacientes por día y por diagnóstico.	X		

Tabla 58*Lista de chequeo N° 31*

Lista de chequeo N° 31			
Historia de usuario: Como root y director quiero un contador de pacientes por diagnóstico para visualizarlo en la interfaz y como gráfica.			
Criterio de aceptación	Si	No	Observación
Se visualiza en la interfaz un contador y una gráfica con el número de pacientes por diagnóstico.	X		

Tabla 59*Lista de chequeo N° 32*

Lista de chequeo N° 32			
Historia de usuario: Como root quiero actualizar un hospital para realizar cambios en los campos.			
Criterio de aceptación	Si	No	Observación
Se permite la actualización de un hospital solo al usuario root.	X		

Tabla 60*Lista de chequeo N° 33*

Lista de chequeo N° 33			
Historia de usuario: Como root quiero actualizar un espacio, para cambiar sus datos dentro de la aplicación.			
Criterio de aceptación	Si	No	Observación
Se permite la actualización de un espacio solo al usuario root.	X		

Tabla 61*Lista de chequeo N° 34*

Lista de chequeo N° 34			
Historia de usuario: Como root, administrador, doctor, quiero listar usuarios para ver los usuarios actualmente registrados en la aplicación.			
Criterio de aceptación	Si	No	Observación
Se lista los usuarios que están registrados en la aplicación web.	X		

Tabla 62*Lista de chequeo N° 35*

Lista de chequeo N° 35			
Historia de usuario: Como usuario doctor, paciente quiero un formulario o campo de motivo de alta, para colocar el motivo de alta de un paciente.			
Criterio de aceptación	Si	No	Observación
El doctor puede dar de alta a un paciente, el paciente también puede darse de alta de la aplicación web.	X		

Tabla 63*Lista de chequeo N° 36*

Lista de chequeo N° 36			
Historia de usuario: Como usuario quiero un formulario para cambiar mi contraseña en la aplicación.			
Criterio de aceptación	Si	No	Observación
El usuario puede cambiar su contraseña por una nueva.	X		

Tabla 64*Lista de chequeo N° 37*

Lista de chequeo N° 37			
Historia de usuario: Como usuario quiero actualizar mis datos para que los cambios sean reflejados en la vista de perfil.			
Criterio de aceptación	Si	No	Observación
Se visualiza los cambios realizados por el usuario, en la vista de perfil.	X		

Tabla 65*Lista de chequeo N° 38*

Lista de chequeo N° 38			
Historia de usuario: Como usuario quiero un botón de olvide mi contraseña o similar para recuperar mi contraseña.			
Criterio de aceptación	Si	No	Observación
Al dar clic en el botón olvide mi contraseña, el usuario es redirigido a un formulario para que ingrese su correo.	X		

Tabla 66*Lista de chequeo N° 39*

Lista de chequeo N° 39			
Historia de usuario: Como usuario quiero recibir un enlace a mi correo en caso de recuperar mi contraseña para ser redirigido a una vista y colocar mi nueva contraseña.			
Criterio de aceptación	Si	No	Observación
El usuario recibe en su correo un enlace que lo redirigirá a un formulario para actualizar su contraseña.	X		
Este enlace es válido por 45min.	X		

Del resultado de las listas de chequeo se concluye que el 100% de las historias de usuario fueron realizadas y cumplen con los criterios de aceptación que se establecieron en el apartado de análisis junto con el experto, por lo que el desarrollo de la aplicación web se concluyó con éxito.

3.5.2. Implementación

La implementación se llevó a cabo con lo siguiente:

- Se utilizó un repositorio en Github para almacenar el código y controlar las versiones.
- La aplicación web tanto backend como frontend se subieron a un servidor Linux Centos 8 perteneciente a la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.
- El servidor node.js y la base de datos mongodb están instalados en el mismo servidor.
- El servidor node.js se encuentra en ejecución continua gracias que está ejecutándose como un servicio. Al igual que la base de datos.

Capítulo IV

Validación

4.1. Introducción

Para llevar a cabo el proceso de validación el sistema fue implementado en los servidores de la ESPE Sede Latacunga. En la adquisición de datos participó el personal militar que asiste presencialmente a la Universidad De las Fuerzas Armadas ESPE Sede Latacunga a partir del 19 de mayo del 2020, cuya población fue de 96 militares, con el fin de hacer un análisis más eficiente y dada la situación emocional del paciente que no quiere ser fuente de información por temor o miedo a perder su intimidad, se realizó un muestreo no probabilístico por conveniencia a 10 sujetos que enviaron la información más completa posible, cabe recalcar que este tipo de muestreo permite seleccionar a los individuos que acepten ser incluidos. Se seleccionó a las personas que enviaron por 14 días o más, como mínimo dos y máximo tres veces al día su temperatura, ritmo cardíaco, saturación de oxígeno (opcional) y una breve descripción de sus síntomas y evolución. En la validación participó el Doctor Bolívar Llamuca, médico de la institución, el cual tenía como principal labor evaluar diariamente la evolución de los pacientes con el fin de detectar posibles casos y aislarlos para así detener la cadena de contagio. Hay que tener en cuenta que los fines de semana al no haber personal médico a cargo fue opcional el envío de seguimientos. Para llevar a cabo este proceso de validación sin inconvenientes se capacitó convenientemente al médico encargado de hacer el seguimiento y a los militares que participaron en el envío de sus datos.

4.2. Análisis de resultados

A continuación, se analizan resultados obtenidos en función de los siguientes indicadores: porcentaje de personas atendidas (asesoradas) vía remota, Porcentaje de seguimientos realizados, e Índice de satisfacción del usuario.

En la siguiente tabla se puede un observar un resumen general de la información de los pacientes, cabe señalar que todo paciente una vez es ingresado al sistema ingresa como sospechoso y que para que un doctor revise un seguimiento, él debe observar alguna anomalía en los datos enviados por el paciente esto con el fin de optimizar al máximo los recursos humanos disponibles.

Tabla 67

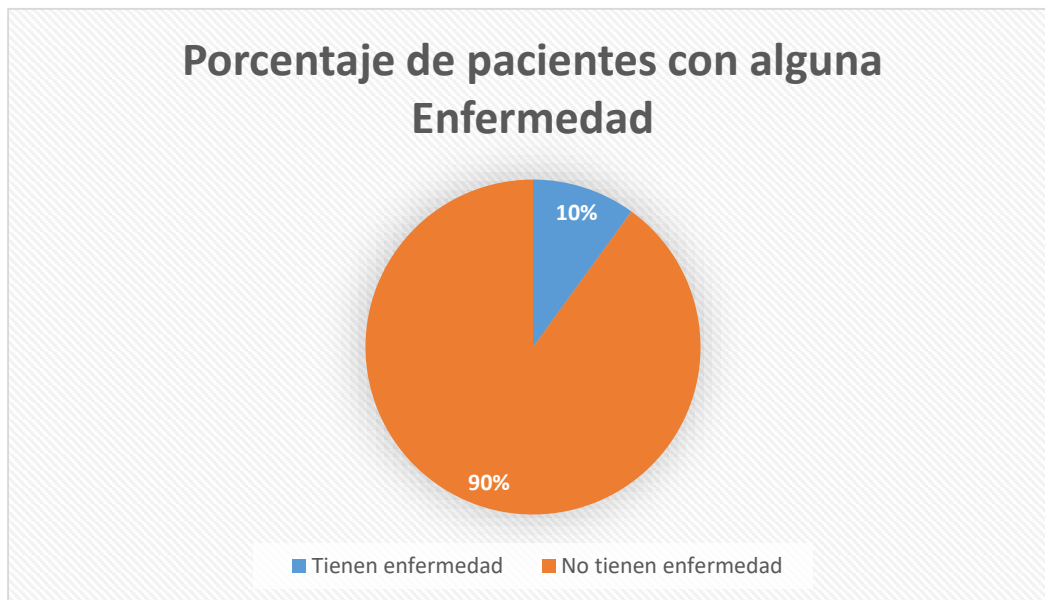
Resumen general de la información de los pacientes

Paciente	Tiene alguna enfermedad	¿Es o fue fumador?	Mayor Temperatura registrada	Temperatura promedio	Diagnostico	Atendido
Paciente 1	No	No	36,7	36,48	Sospechoso	NO
Paciente 2	No	No	36,3	36,08	Sospechoso	SI
Paciente 3	No	No	36,5	36,21	Confirmado	SI
Paciente 4	No	Sí	36,9	36,36	Sospechoso	SI
Paciente 5	No	No	37	36,03	Sospechoso	NO
Paciente 6	Rinitis alérgica	No	35,1	36,00	Sospechoso	SI
Paciente 7	No	No	35	34,34	Sospechoso	SI
Paciente 8	No	No	36,5	36,17	Sospechoso	SI
Paciente 9	No	No	37,2	36,41	Sospechoso	SI
Paciente 10	No	No	36,6	35,93	Sospechoso	NO

De la tabla anterior se extraen las siguientes gráficas mostradas a continuación:

Figura 138

Porcentaje de pacientes con alguna enfermedad



Análisis

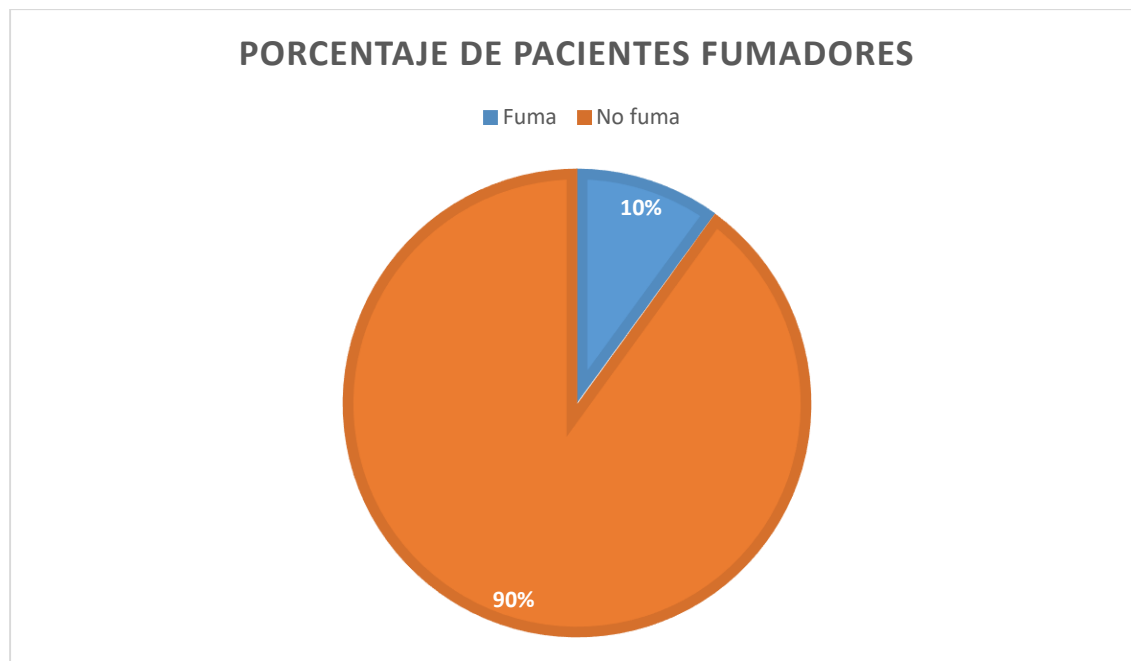
Como se puede observar en la Figura 138 anterior solo el 10% de los pacientes tienen una enfermedad que podría influir en la gravedad de sus síntomas en caso de contagio.

Interpretación

La gran mayoría de pacientes no presenta enfermedades existentes que pudieran agravar sus síntomas en caso de contagio.

Figura 139

Porcentaje de pacientes fumadores

**Análisis**

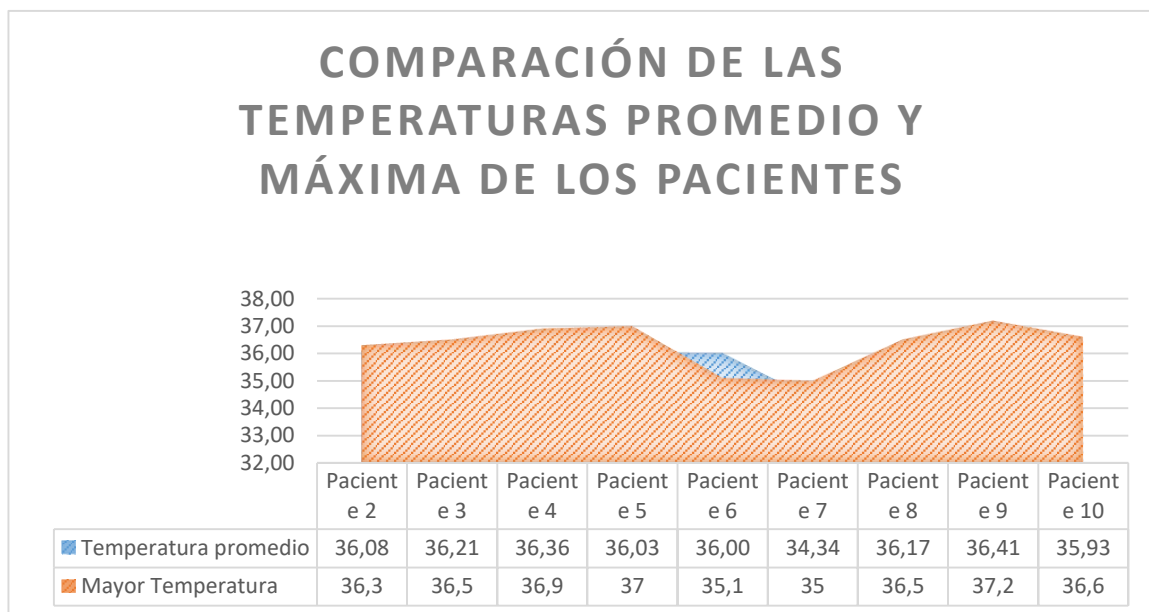
Como se puede apreciar en la Figura 139, solo el 10% de los pacientes son fumadores, esta condición es importante ya que puede influir en la gravedad de los síntomas de un paciente.

Interpretación

Se concluye que mayoría de pacientes no es fumador.

Figura 140

Comparación de las temperaturas de los pacientes



Análisis

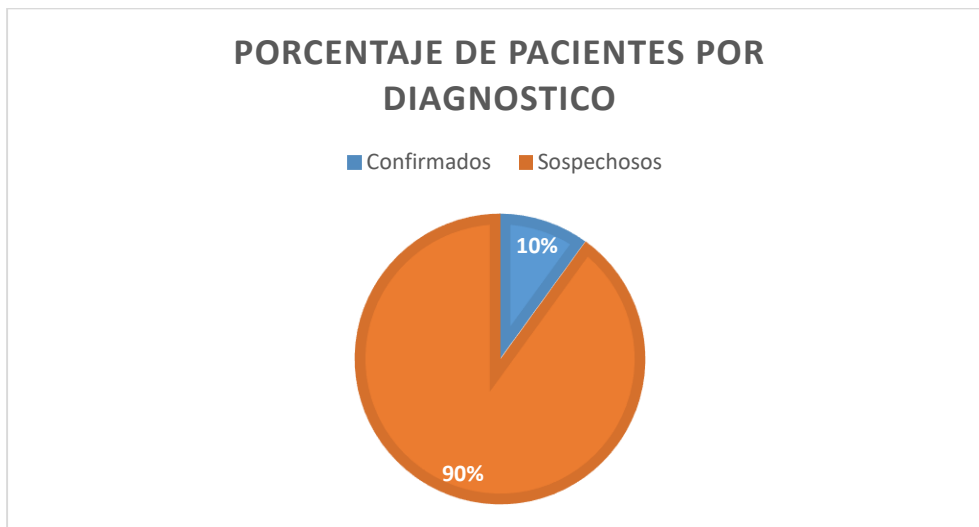
En la Figura 140 se puede observar que no existen cambios bruscos de temperatura si comparamos la temperatura promedio que el paciente tuvo y su máxima temperatura alcanzada durante el periodo de prueba.

Interpretación

Se puede concluir que todos los pacientes tuvieron durante el periodo de prueba una temperatura normal.

Figura 141

Porcentaje de pacientes por diagnóstico

**Análisis**

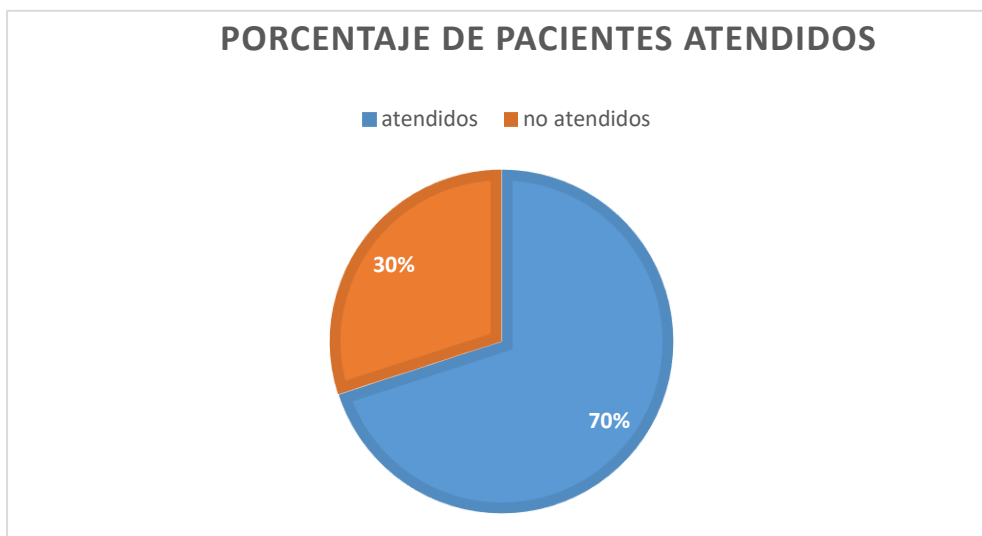
En la Figura 141 se muestran los porcentajes de pacientes que han sido diagnosticados como confirmados y sospechosos por el personal de salud encargado.

Interpretación

Se concluye que solo un porcentaje significativamente inferior fue contagiado con el virus.

Figura 142

Porcentaje de pacientes atendidos

**Análisis**

El gráfico muestra que el 70% de los pacientes que registraron un seguimiento en el aplicativo fueron atendidos que representa un porcentaje mayor respecto a los no atendidos.

Interpretación

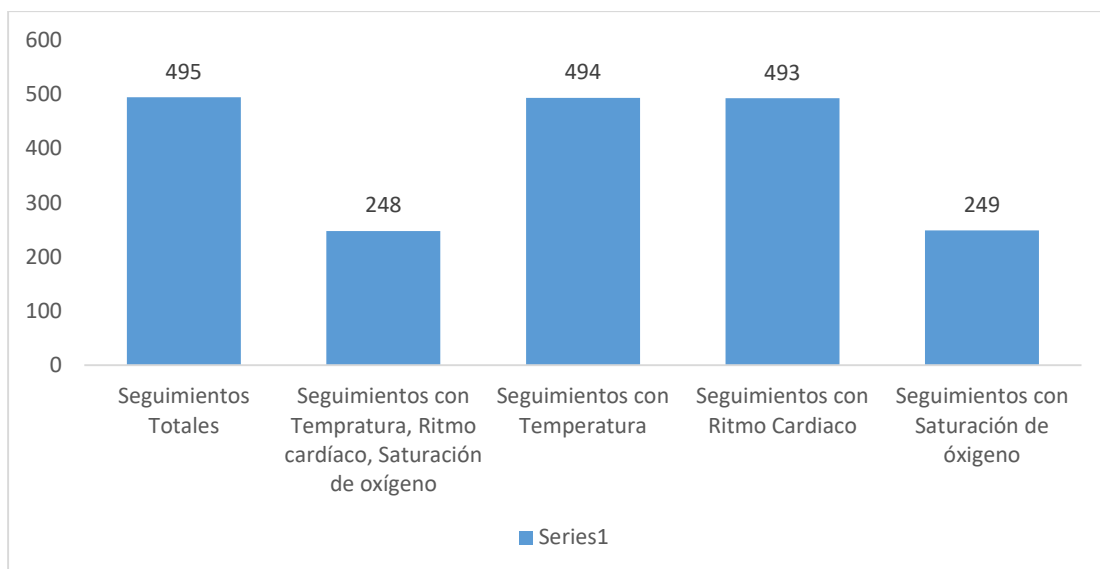
Se concluye en base a los resultados obtenidos que la mayoría de los pacientes fueron atendidos. Lo cual demuestra que el indicador de porcentaje de pacientes atendidos se cumple.

A continuación, en la siguiente tabla se muestra el porcentaje de seguimientos realizados por los pacientes y se analiza si tienen los campos temperatura, ritmo cardíaco y saturación de oxígeno.

Tabla 68

Porcentaje de seguimientos realizados por los pacientes

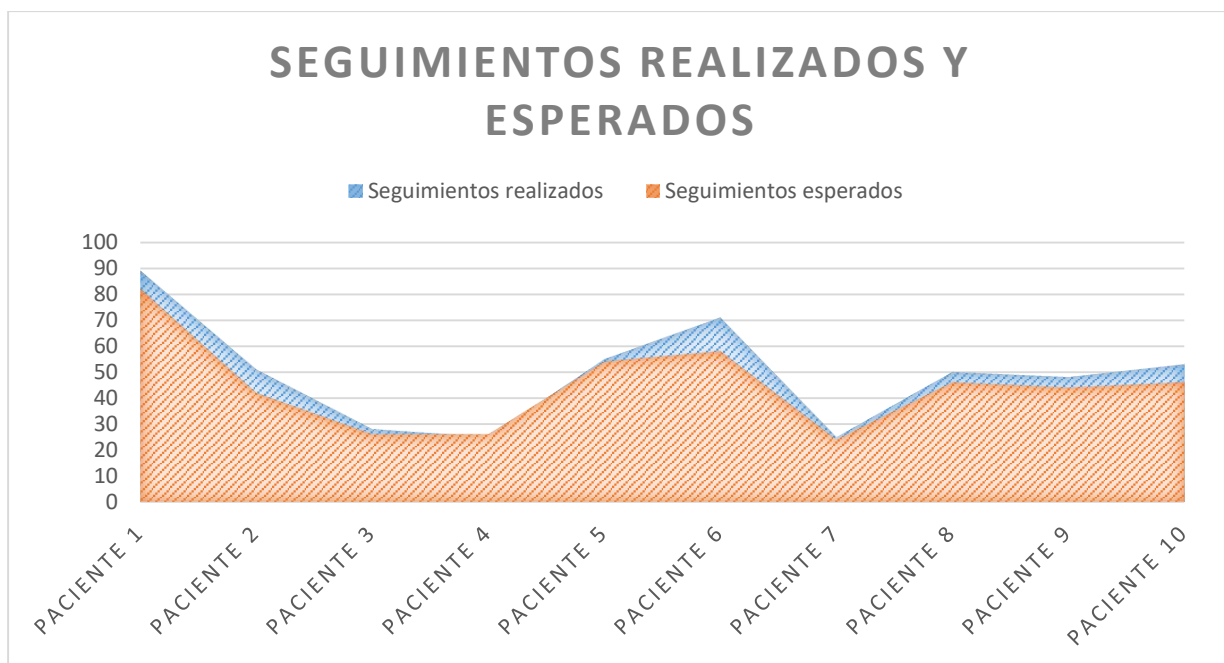
Paciente	Días Totales de seguimiento	Número de seguimientos realizados	número de seguimientos esperados	Nº Seguimientos Promedio diario
paciente 1	41	89	82	2,17
paciente 2	21	51	42	2,43
paciente 3	13	28	26	2,15
paciente 4	13	25	26	1,92
paciente 5	27	55	54	2,04
paciente 6	29	71	58	2,45
paciente 7	12	25	24	2,08
paciente 8	23	50	46	2,17
paciente 9	22	48	44	2,18
paciente 10	23	53	46	2,30

Figura 143*Gráfica de seguimientos***Análisis**

En la Figura 143 se contabilizan los seguimientos de acuerdo a los signos vitales enviados, como se puede apreciar el 99,79% de los seguimientos fueron enviados con el campo temperatura, el 99,59% se enviaron con el campo ritmo cardíaco, el 50,30% se envió con el campo saturación de oxígeno y solo el 50,10% de los seguimientos enviados tenían en los tres campos (Temperatura, ritmo cardíaco, saturación de oxígeno). Cabe recalcar que este análisis es importante ya que el doctor puede realizar una valoración más precisa de acuerdo al nivel de detalle con el que se envíe el seguimiento.

Interpretación

Se concluye que los pacientes en su mayoría enviaron los seguimientos la mayoría de veces sólo con los campos temperatura y ritmo cardíaco, el campo saturación de oxígeno a menudo lo dejan en blanco.

Figura 144*Seguimientos realizados y esperados***Análisis**

Como se puede apreciar en la Figura 144 el número de seguimientos realizados se asemeja al número de seguimientos esperados, es decir que en promedio los pacientes enviaron 2 seguimientos diariamente.

Interpretación

Se concluye que la mayoría de pacientes enviaron al menos dos seguimientos diarios lo cual confirma que se cumple el indicador porcentaje de seguimientos realizados.

4.3. Análisis de satisfacción respecto a la aplicación web

Por último, para el indicador satisfacción de los usuarios se realizaron dos encuestas, a los pacientes y al doctor encargado, con el fin de medir el grado de satisfacción de los usuarios con la aplicación. Con el fin de medir las opiniones y puntos de vista de los pacientes con un mayor grado de especificidad se utilizó una escala de Likert. A continuación, se muestran los resultados ya tabulados:

Tabla 69

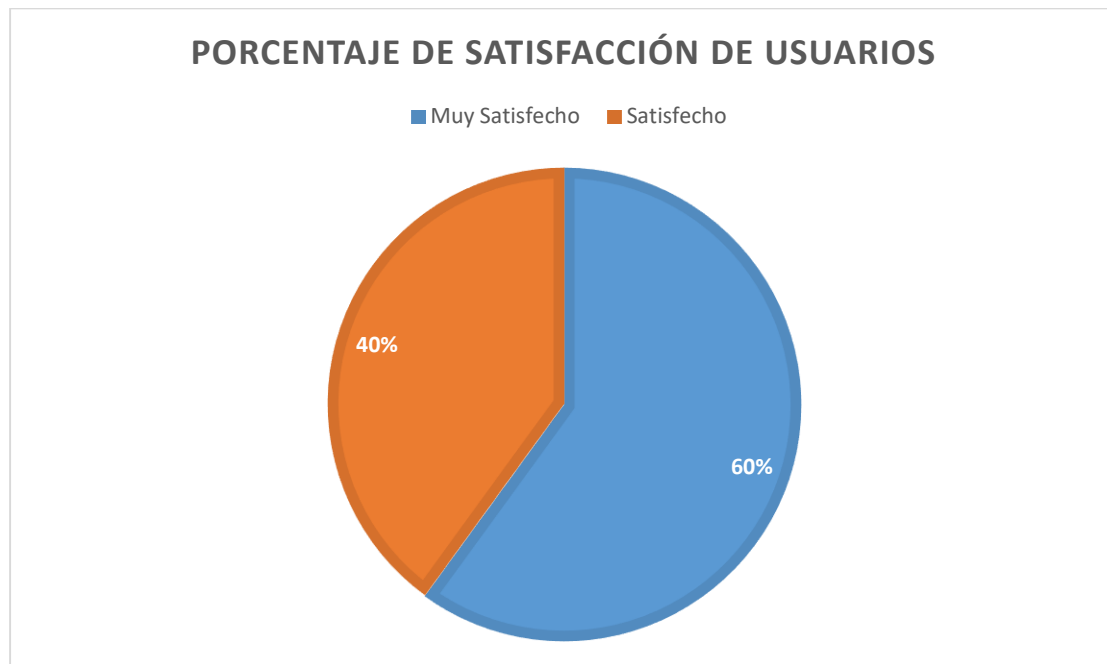
Resultados tabulados de las encuestas de satisfacción

Nº Paciente	¿Rango de edad?	¿Fue fácil registrarse en la aplicación?	¿El aplicativo le permitió enviar sus datos con facilidad?	¿Le pareció agradable el diseño del aplicativo?	¿Considera que el aplicativo fue de utilidad para informar su estado de salud?	Total	Valoración
paciente 1	18-30	5	5	5	5	20	Muy satisfecho
paciente 2	18-30	5	4	4	4	17	Muy satisfecho
paciente 3	45-60	4	4	5	4	17	Muy satisfecho
paciente 4	45-60	4	4	4	4	16	Satisfecho
paciente 5	18-30	4	5	4	4	17	Muy satisfecho
paciente 6	30-45	5	5	5	5	20	Muy satisfecho
paciente 7	18-30	3	4	3	4	14	Satisfecho
paciente 8	18-30	4	4	4	4	16	Satisfecho
paciente 9	30-45	4	4	4	4	16	Satisfecho
paciente 10	30-45	5	5	4	4	18	Muy satisfecho

De la tabla se obtiene la siguiente gráfica:

Figura 145

Porcentaje de satisfacción de usuarios



Análisis

De la encuesta realizada a los pacientes el 60% de los pacientes se consideran muy satisfechos con el uso de la aplicación y un 40% consideran satisfechos con la aplicación.

Interpretación

Se concluye que la mayoría de los pacientes están satisfechos con el funcionamiento del aplicativo. Con lo cual se demuestra que el indicador de satisfacción de usuario se cumple.

Al experto médico a cargo del seguimiento de pacientes también se le realizó unas preguntas para validar su nivel de satisfacción con el uso de la aplicación, esas preguntas se muestran en la siguiente tabla.

Tabla 70

Preguntas sobre el nivel de satisfacción para el experto

Pregunta	Respuesta	puntaje
¿El aplicativo le permitió ver y realizar seguimientos a pacientes con facilidad?	De acuerdo	4
¿Le pareció agradable el diseño del aplicación?	De acuerdo	4
¿Considera que se cumplió el objetivo de dar seguimiento médico a personas expuestas a Covid 19 ?	Totalmente de acuerdo	5
¿Considera que el aplicativo es una herramienta útil que apoya en el seguimiento de personas expuestas al COVID-19 en la ESPE Sede Latacunga?	Totalmente de acuerdo	5

Interpretación

Sumado los puntajes de cada pregunta se obtiene 18 de 20 puntos posibles lo que significa que el doctor está muy satisfecho con el aplicativo.

Ante la hipótesis de la presente investigación: ¿Si se desarrolla una aplicación web se apoya el seguimiento del estado de salud de personas expuestas al contagio de COVID-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas

ESPE Sede Latacunga?, y en base al análisis del experto médico con los datos obtenidos con el aplicativo web y una vez comprobado que los indicadores se cumplen se puede afirmar que se acepta la hipótesis de esta investigación.

Capítulo V

Conclusiones y recomendaciones

5.1. Conclusiones

De acuerdo a la validación realizada por el experto médico en base a los datos obtenidos con el aplicativo se puede concluir que el aplicativo web podría detectar tempranamente a personas sospechosas de Covid-19 y actuar de una forma rápida y oportuna para así evitar la cadena de contagios precautelando así la salud del personal que asiste presencialmente a la ESPE Sede Latacunga.

Se elaboró un marco teórico sobre el estado del arte del COVID-19 en el que se abordó todos los aspectos principales de esta enfermedad lo cual permitió desarrollar una aplicación web que apoye al seguimiento del estado de salud de personas expuestas al contagio de COVID-19 mediante tele asesoría médica en la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.

Se elaboró un marco teórico sobre los principios fundamentales de las arquitecturas limpias lo que permitió diseñar e implementar satisfactoriamente una arquitectura acorde a estos principios y obtener todas las bondades de este tipo de arquitecturas.

Se desarrolló una aplicación web haciendo uso de tecnologías de software modernas implementando una arquitectura limpia y buenas prácticas de desarrollo de software lo cual permitió tener un software altamente testeable, escalable y mantenible de acuerdo a la conceptualización misma de este tipo de arquitecturas.

5.2. Recomendaciones

Para que el aplicativo funcione correctamente se recomienda a los pacientes a los cuales se les está haciendo el seguimiento envíen diariamente toda la información solicitada con la finalidad de hacer un análisis más exacto y para que el personal médico tome las medidas adecuadas.

Se recomienda que la institución dé las facilidades de acceso para la toma de temperatura, ritmo cardiaco y saturación de oxígeno o su vez brinde capacitaciones sobre la toma de estos signos ya que estos son los datos que alimentan al sistema y son necesarios para un mejor análisis.

Se recomienda desarrollar una aplicación móvil que permita realizar un test virtual para la detección de personas con sospecha de Covid-19.

Se recomienda motivar a las personas para que tomen conciencia y envíen el seguimiento diario.

Bibliografía

- Agora.io. (01 de 01 de 2021). *Agora.io*. Recuperado el 16 de 01 de 2021, de <https://www.agora.io/en/products/video-call/>
- Alicante, U. d. (2007). *Universidad de Alicante*. Recuperado el 26 de Septiembre de 2020, de <https://rua.ua.es/dspace/bitstream/10045/4412/5/03c-AplicacionesWeb.pdf>
- Atencio, L. (2016). *Functional Programming in Javascript*. New York: Manning Publications Co.
- Bashshur, R., Doarn, C., Frenk, J., Kvedar, J., & Woolliscroft, J. (6 de Mayo de 2020). Telemedicine and the COVID-19 Pandemic, Lessons for the Future. *Telemedicine and e-Health*, 26. doi:10.1089/tmj.2020.29040.rb
- Cuevas Valencia, R. E., & Adame Añorve, A. L. (2013). HERRAMIENTAS DE VIDEOCONFERENCIA APLICADAS EN LA EDUCACIÓN EN NIVEL SUPERIOR. *researchgate*. Recuperado el 07 de 10 de 2020, de https://www.researchgate.net/publication/280734306_HERRAMIENTAS_DE_VIDEOCONFERENCIA_APLICADAS_EN_LA_EDUCACION_EN_NIVEL_SUPERIOR
- DesarrolloWeb. (2020). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/home/angular>
- El Comercio. (18 de 02 de 2021). *El Comercio*. Obtenido de El Comercio: <https://www.elcomercio.com/actualidad/ecuador-vacuna-covid-proveedores.html>
- FAO. (2016). *La importancia de la agricultura en la actualidad*. [En línea]. Disponible desde: <http://www.fao.org/3/a0015s/a0015s04.htm>. Recuperado el 21 de febrero de 2020

- Filipova, O., & Vilao, R. (2018). *Software Development From A to Z*. Berlin, Alemania: Apress. doi:10.1007/978-1-4842-3945-2
- Firebase. (12 de 12 de 2020). *Firestore Blog*. Recuperado el 16 de 01 de 2021, de <https://firebase.googleblog.com/2017/10/introducing-cloud-firestore.html>
- García Peñalvo, F. J., García Holgado, A., & Vázquez Ingelmo, A. (2020). *Research Group in InterAction and eLearning of the University of Salamanca*. Recuperado el 11 de 01 de 2021, de METODOLOGÍAS DE INGENIERÍA DE SOFTWARE: <https://repositorio.grial.eu/bitstream/grial/1942/1/4.%20Metodologi%CC%81a-2020.pdf>
- GraphQL. (2020). *graphql.org*. Recuperado el 12 de 12 de 2020, de <https://graphql.org/>
- Hall, G. M. (2017). *Adaptive Code Agile Coding with Design Patterns and SOLID Principles*. United States of America: Microsoft Press.
- Hu, B., Guo, H., Zhou, P., & Shi, Z.-L. (2020). Characteristics of SARS-CoV-2 and COVID-19. *Nature Reviews Microbiology*. doi:<https://doi.org/10.1038/s41579-020-00459-7>
- Jaiswal, M. (Noviembre de 2019). Software Architecture and Software Design. *International Research Journal of Engineering and Technology (IRJET)*, 6. Recuperado el 07 de Octubre de 2020
- James O. Coplien, Gertrud, & Cope. (01 de Enero de 2012). *fulloo.info*. Recuperado el 14 de Octubre de 2020, de [fulloo.info](http://fulloo.info/Documents/CoplienReenskaugASA2012.pdf): <http://fulloo.info/Documents/CoplienReenskaugASA2012.pdf>
- Jin, B., Sahni, S., & Shevat, A. (2018). *Designing Web APIs - Building APIs That Developers Love* (Primera ed.). United States of America: O'Reilly.
- Martin, R. C. (2018). *Clean Architecture A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN*. Pearson Education.

- Mayo Clinic. (03 de 03 de 2020). *Mayo Clinic*. Recuperado el 01 de 01 de 2021, de https://www.mayoclinic.org/es-es/diseases-conditions/sars/symptoms-causes/syc-20351765?utm_source=Google&utm_medium=abstract&utm_content=Severe-acute-respiratory-syndrome-%28SARS%29&utm_campaign=Knowledge-panel
- Ministerio de Salud Pública. (2020). *Actualización de casos de coronavirus en Ecuador*. Recuperado el 18 de 9 de 2020, de Ministerio de Salud Pública: <https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>
- Ministerio de Salud Pública. (9 de 12 de 2020). *Actualización de casos de coronavirus en Ecuador*. Recuperado el 18 de 9 de 2020, de Ministerio de Salud Pública: https://www.salud.gob.ec/wp-content/uploads/2020/09/Boletin-196_Nacional_MSP.pdf
- MongoDB. (Octubre de 2020). *MongoDB*. Obtenido de <https://www.mongodb.com/es>
- Mongoose. (12 de 12 de 2020). *Mongoose ODM*. Recuperado el 16 de 01 de 2021, de <https://mongoosejs.com/>
- Mozilla. (12 de Octubre de 2020). *mozilla*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- OMS. (16 de 09 de 2020). *WHO*. Obtenido de WHO: <https://www.who.int/es/health-topics/coronavirus>
- OMS. (16 de 09 de 2020). *World Health Organization*. Obtenido de WHO Coronavirus Disease (COVID-19) Dashboard: <https://covid19.who.int/>
- OMS. (17 de 03 de 2021). *OMS*. Obtenido de OMS: <https://covid19.who.int/>
- Organización mundial de la Salud. (24 de 02 de 2010). *Alerta y respuesta mundiales*. Obtenido de Organización mundial de la Salud: https://www.who.int/csr/disease/swineflu/frequently_asked_questions/pandemic/es/

- Organización Mundial de la Salud. (17 de 09 de 2020). *WHO Coronavirus Disease (COVID-19) Dashboard*. Obtenido de Organización Mundial de la Salud: <https://covid19.who.int/>
- Organización Mundial de la Salud[OMS]. (2020). *Preguntas y respuestas sobre la enfermedad por coronavirus (COVID-19)*. Obtenido de Organización Mundial de la Salud: <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>
- Oxford Languages. (2020). *Oxford Languajes Léxico*. Recuperado el 12 de 01 de 2021, de <https://www.lexico.com/es/definicion/metodo>
- Palermo, J. (29 de Julio de 2008). *jeffreypalermo.com*. Obtenido de <https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/>
- Pearce, J. (s.f.). *Department Computer Science - San Jose State University*. Obtenido de <http://www.cs.sjsu.edu/~pearce/modules/patterns/enterprise/ecb/ecb.htm>
- Portnoy, J., Waller, M., & Elliott, T. (24 de Marzo de 2020). Telemedicine in the Era of COVID-19. *The Journal of Allergy and Clinical Immunology: In Practice*, 8, 1-3. doi:10.1016/j.jaip.2020.03.008
- Pressman, R. (2005). *Software engineering: a practitioner's approach*.
- Prieto-Reinoso, E. S., & Cabrera-Mejía, J. B. (2020). Chatbot en Telegram para consultas de casos COVID-19 en el Ecuador. *Dominio de las ciencias*, 6(4), 987-1000. doi:<http://dx.doi.org/10.23857/dc.v6i4.1518>
- Primicias. (08 de 08 de 2020). *Primicias*. Obtenido de <https://www.primicias.ec/>: <https://www.primicias.ec/noticias/tecnologia/asi-ecuador-aplicacion-rastrear-covid/>
- Raffino, M. E. (30 de Junio de 2020). *Concepto de*. Recuperado el 07 de Octubre de 2020, de Concepto.de: <https://concepto.de/software-de-aplicacion/>

- RedHat. (2020). *RedHat*. Recuperado el 16 de 01 de 2021, de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- Salguero, E. (06 de Junio de 2018). *Medium.com*. Recuperado el 14 de Octubre de 2020, de Medium: <https://medium.com/@edusalguero/arquitectura-hexagonal-59834bb44b7f>
- Scrum. (2020). *Scrum.org*. Recuperado el 12 de 01 de 2021, de <https://www.scrum.org/resources/what-is-scrum>
- SENESCYT. (19 de 3 de 2020). *Universidades continúan sus actividades mediante aulas virtuales*. Obtenido de Secretaria de educación superior ciencia y tecnología : <https://www.educacionsuperior.gob.ec/universidades-continuan-sus-actividades-mediante-aulas-virtuales/>
- Sommerville, I. (2005). *Software engineering* (7 ed.).
- Stadler, K., Masignani, V., Eickmann, M., Becker, S., Abrignani, S., Klenk, H.-D., & Rappuoli, R. (2003). SARS — beginning to understand a new virus. *Nature Reviews Microbiology*, 209–218. doi: <https://doi.org/10.1038/nrmicro775>
- Stellman, A., & Greene, J. (2015). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban* (Primera ed.). Gravenstein Highway North, Sebastopol: O'Reilly.
- Teleamazonas. (27 de 03 de 2020). *www.teleamazonas.com*. Obtenido de Teleamazonas: <https://www.teleamazonas.com/andres-michelena-explica-el-uso-de-la-app-covidec/>
- Typegoose. (2020). *Typegoose*. Recuperado el 16 de 01 de 2021, de <https://typegoose.github.io/typegoose/>
- TypeGraphQL. (2020). *TypeGraphQL*. Recuperado el 16 de 01 de 2021, de <https://typegraphql.com>
- TypeScript. (12 de 12 de 2020). *TypeScript*. Recuperado el 16 de 01 de 2021, de <https://www.typescriptlang.org/>

Universidad De Las Fuerzas Armadas ESPE. (2019). *Ley de transparencia año 2019*.

Obtenido de Ley de transparencia: <http://transparencia.espe.edu.ec/ley-de-transparencia-ano-2019/>

Verhoeven, F., Tanja-Dijkstra, K., Nijland, N., Eysenbach, G., & Van Gemert-Pijnen, L.

(May de 2010). Asynchronous and Synchronous Teleconsultation for Diabetes

Care:A Systematic Literature Review. *Journal of Diabetes Science and*

Technology, 4. doi:10.1177/193229681000400323

Wallace, A. (26 de Mayo de 2020). *BBC News Mundo*. Obtenido de BBC News Mundo:

<https://www.bbc.com/mundo/noticias-52797076>

World Wide Web Consortium. (12 de 12 de 2020). *W3C*. Recuperado el 16 de 01 de

2021, de <https://www.w3.org/Style/CSS/>

Yeeply. (01 de Enero de 2018). *yeeply*. Recuperado el 07 de Octubre de 2020, de

[yeeply.com: https://www.yeeply.com/blog/6-tipos-desarrollo-de-aplicaciones-web/](https://www.yeeply.com/blog/6-tipos-desarrollo-de-aplicaciones-web/)

Yuen, K.-S., Ye, Z.-W., Fung, S.-Y., Chan, C.-P., & Jin, D.-Y. (2020). SARS-CoV-2 and

COVID-19: The most important research questions. *Cell Biosci*, 10-40.

doi:10.1186/s13578-020-00404-4

Anexos