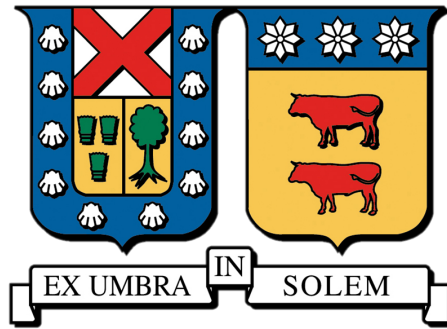




Universidad Técnica Federico Santa María

Ingeniería Civil Telemática

“Plataforma tecnológica para solicitar
servicios de salud a domicilio para el adulto
mayor

Cristóbal Ignacio Mancilla Cayupan

Profesor guía: Marcos Zuñiga

Profesor correferente: Francisco Cabezas

Marzo, 2020

*Para mi familias.
Sin ellos hubiera sido imposible.*

1. Agradecimientos

Se agradece a los profesionales de salud entrevistados, por sus ideas, tiempo y disposición. A las personas mayores por contar su estilo de vida. Y al programa de memoria multidisciplinaria por la organización que ayudo a concluir el proyecto de forma grupal y remota.

2. Resumen

El presente informe documenta el desarrollo de una plataforma tecnológico llamada VIDAT, que tiene como objetivo vincular a profesionales de la salud con personas mayores. El adulto mayor solicita un oferente de servicios a domicilio, por medio de una aplicación móvil, permitiendo comunicación directa y automatización en el agendamiento.

Con lo anterior mencionado, y junto con el panorama mundial de las comunicaciones y masificación de las herramientas tecnológicas, fue necesario un estudio de las partes involucradas. Según esto, un nuevo adulto digital, que nace prontamente, hace viable el proyecto en gran medida, y trae consigo riesgos relevantes que deben ser tratados. Además, en este documento se detallan aspectos técnicos para la viabilidad del software, estudios de mercado, modelos de negocios, diseño y arquitectura, seguridad, entre otros.

Palabras claves: VIDAT, profesional de la salud, adulto mayor, aplicación móvil, backend, frontend.

3. Abstract

This report documents the development of a technological platform called VIDAT, whose mission is to connect health professionals with older people. The elderly adult requests a provider of home services, through a mobile application, allowing direct communication and automation in scheduling.

With the aforementioned, and together with the global panorama of communications and the massification of technological tools, a study of the parties involved was necessary. According to this, a new digital adult, which is born soon, makes the project viable to a large extent, and brings with it relevant risks that must be addressed. In addition, this document details technical aspects for the viability of the software, market studies, business models, design and architecture, security, among others.

Keywords: VIDAT, health professional, elderly, mobile application, backend, frontend.

4. Glosario

Se referirá en ciertos casos al profesional de salud como oferente de servicios, y al adulto mayor como demandante.

- **Backend** : Sistema del lado del servidor.
- **IP** : *Internet Protocol*.
- **SSH** : *Secure Shell*.
- **HTTP** : *Hypertext Transfer Protocol*.
- **HTTPS** : *Hypertext Transfer Protocol Secure*.
- **Load balancer** : Balanceador de carga. Redirige peticiones HTTP.
- **ExpressJS** : *Framework* para desarrollo de servidores web.
- **SQL** : *Structures Query Language*.
- **Frontend** : Sistema del lado de la interfaz usuaria.
- **API REST** : Interfaz de programación de aplicaciones definida con una arquitectura de transferencia de estado representacional.
- **React Native** : Framework para desarrollo Frontend para aplicaciones móviles.
- **Request HTTP** : Métodos de solicitud para indicar la acción que se desea.
- **JavaScript** : Lenguaje de programación.
- **Geolocalización** : Capacidad para obtener la ubicación geográfica real de un objeto.
- **Stack** : Es una lista ordenada o estructura de datos que permite almacenar y recuperar datos. En el contexto del Frontend, se referirá a stack como un conjunto de vistas.

Índice de contenidos

1. Agradecimientos	3
2. Resumen	4
3. Abstract	5
4. Glosario	6
5. Introducción	11
5.1. Descripción del problema	12
5.2. Soluciones Existentes / Estado del Arte	12
6. Desarrollo del tema	15
6.1. Estudio Mercado	15
6.2. Elicitación	19
6.3. Diseño del Backend	19
6.3.1. Modelo de datos	19
6.3.2. Microservicio: Usuarios	21
6.3.3. Microservicio: Servicios	21
6.3.4. Microservicio: Comunicación	22
6.3.5. Microservicio: Pagos	22
6.4. Implementación del Backend	22
6.5. Diseño del Frontend	23
6.6. Implementación del Frontend	32
6.6.1. Descripción de los componentes y funcionalidades	32
6.6.2. Interfaz Usuaría	34
6.7. Conceptos y estrategias de funcionalidades	40
6.8. Integración Backend-Frontend	41
7. Evaluación de Riesgos	44
8. Conclusiones	46
9. Anexos	48
9.1. Historias de usuarios y requisitos funcionales del sistema	48
9.2. Casos de uso	49
9.3. Softgoal	52
9.4. Diagrama de secuencia registro e inicio sesión	54
9.5. Diagrama de secuencia agendamiento	56
9.6. Diagrama de secuencia comunicación	57
9.7. Configuración API Google Geocoding	57

Índice de figuras

1.	Página web oficial Habitissimo.	12
2.	Página web oficial Listoco	13
3.	Página web oficial Doctoralia	14
4.	Grupos etarios. Fuente: Encuesta Casen 2017	15
5.	Proyección proceso envejecimiento. Fuente: Encuesta Casen[7]	15
6.	Uso de internet por población con 60 años o más. Fuente: Encuesta Casen 2017[7]	16
7.	Distribución estado civil. Fuente: Encuesta Casen 2017[7]	17
8.	Fuente: Encuesta Fondef[12] Idea- Sistema de comunicación en salud Adultos mayores y redes de apoyo.	18
9.	Modelo base de datos	20
10.	Arquitectura del Backend. Fuente: Autoría memoristas.	20
11.	Patrón Facade. Fuente: Autoría memoristas.	21
12.	Estructura de modulos de Software y Servicios. Fuente: Autoría memoristas.	23
13.	Flujo principal VIDAT demandante. Fuente: Autoría memoristas.	24
14.	Flujo secundario VIDAT demandante. Fuente: Autoría memoristas.	25
15.	Prototipo Adobe Xd para demandante. Vista Muro y Buscar.	26
16.	Prototipo Adobe Xd para demandante. Vista Contactos.	26
17.	Maqueta Adobe Xd para demandante.	27
18.	Flujo principal VIDAT oferente. Fuente: Autoría memoristas.	28
19.	Flujo secundario VIDAT oferente. Fuente: Autoría memoristas.	29
20.	Prototipo Adobe Xd para oferente. Vista Login y Register.	30
21.	Prototipo Adobe Xd para oferente. Vista Muro y Mis horarios.	30
22.	Prototipo Adobe Xd para oferente. Vista Perfil y Agenda.	31
23.	Prototipo Adobe Xd para oferente. Vista Contactos y Chat.	31
24.	Vistas Inicio de sesión y Menú navegación.	35
25.	Vistas Búsqueda Profesional y Georreferenciación API Google.	35
26.	Vistas Lista de profesionales en cobertura e información del profesional.	36
27.	Vistas de calendario con horarios disponibles y ventana modal para enviar solicitud	36
28.	Vistas Muro con el estado de solicitudes, y el Chat temporal.	37
29.	Vistas Perfil y Configuración.	37
30.	Vistas Inicio Sesión y Registro.	38
31.	Vistas Muro y Mis horarios.	38
32.	Vistas Agenda y Perfil.	39
33.	Stack con vistas de Contactos y Chat.	39
34.	Relación disponibilidad de horarios oferente.	40
35.	Petición Get para leer todos los oferentes.	42
36.	Petición POST actualización atributos demandante.	42

37.	Petición Get para buscar todas las solicitudes realizadas por el “id_oferente”. Este endpoint está configurado para buscar por todos los parámetros que se desee.	43
38.	Autenticación y Acreditación de cuenta. Fuente: Autoría memoristas. .	49
39.	Administración de servicios ofrecidos/solicitados. Fuente: Autoría me- moristas.	50
40.	Administración de relaciones inter-usuarias. Fuente: Autoría memoristas.	51
41.	Mecanismos de Remuneración y Valoración. Fuente: Autoría memoristas.	52
42.	Softgoal. Fuente: Autoría memoristas.	53
43.	Registro. Fuente: Autoría memoristas.	54
44.	Inicio Sesión. Fuente: Autoría memoristas.	55
45.	Agendamiento. Fuente: Autoría memoristas.	56
46.	Comunicación. Fuente: Autoría memoristas.	57
47.	Google Cloud Plataform.	58
48.	Google Cloud Plataform. API Geocoding.	58
49.	Google Cloud Plataform. API Geocoding.	59
50.	Google Cloud Plataform. Credentials.	59
51.	Google Cloud Plataform. API KEY.	60
52.	Función Fetch de JS al endpoint de Google.	60
53.	Response Api Google Geocoding.	61

Índice de tablas

1.	Descripción componente de Gestión de Usuarios.	32
2.	Descripción componente de Agendación.	33
3.	Descripción componente de Comunicación.	34
4.	Riesgos técnicos.	44
5.	Riesgos de valor.	44
6.	Riesgos de calidad.	44
7.	Riesgos de malversación.	45
8.	Riesgos de rechazo.	45
9.	Riesgos de filtración.	45
10.	Requisito Funcional 1, Administración de usuarios.	48
11.	Requisito Funcional 2, Administración de solicitudes de servicio.	48

5. Introducción

Diferentes son las circunstancias por las que un adulto mayor recurre a la prestación de un servicio de manera particular en sus hogares. Acuden a un experto en alguna área, como salud, construcción, jardinería, etc, con una necesidad, sin el inconveniente de que ellos salgan; esto ya sea porque no pueden o porque en esencia facilita su forma de vivir. Una persona de mayor edad debe sobrellevar muchos problemas. Uno de ellos es la movilidad, esto se ramifica en laborales básica del hogar, como cocinar, lavar ropa, aseo y ornato, compra de alimentos, entre muchos otros. Problemas de salud son quizás los más importantes, enfermeras para vacunas o aseo personal, compra de fármacos, trabajo kinésico por quemadura, accidentes o simplemente por la nula actividad física que el adulto mayor realiza.

En la actualidad esta comunicación para la prestación y agendación de servicios se hace por contactos de confianza, adquiridos de boca en boca para tareas más domésticas; contactos adquiridos en centros médicos en aspectos de servicio de salud, o contactos encontrados en anuncios publicitarios. Ahora, las tecnologías pueden estar al servicio de ellos, y mejorar la comunicación entre el adulto mayor y su oferente de servicio. No obstante, una persona de tercera edad no está estrechamente familiarizado, aún, con un aparato electrónico que le ayude. Y tanto adultos mayores como profesionales de salud no tienen acceso a esta red de clientes.

Este proyecto fue abordado por dos memoristas de ingeniería civil telemática. En pos de un proyecto escalable, ordenado y colaborativo, se separó el desarrollo del software en Backend y Frontend, y con ello se diferenció los perfiles específicos del trabajo de equipo. Ignacio Andrés García Peñailillo, como desarrollador de Backend se encargó del sistema por el lado del servidor, cuyo principal objetivo es proveer mecanismos para la administración de datos, dado que se trabaja con información sensible es necesario proveer capas de seguridad. Cristóbal Mancilla como desarrollador Frontend se encargó del sistema con el que los usuarios interactúan, en conjunto con el diseño previo de las vistas y el flujo de navegación, utilizando los servicios que el Backend proporcionaba. Si bien la implementación del Backend y Frontend fue por separado, la planificación previa como lo es el modelo de la base de datos, los requisitos funcionales y no funcionales del sistema, historias de usuario, el estudio de mercado y modelo de negocios fue un trabajo en conjunto en los módulos de diseño correspondientes.

5.1. Descripción del problema

El desafío es mejorar la calidad de vida de los adultos mayores mediante las TIC's. Este problema social en este sector particular, es motivado justamente por brindar ayuda a quienes no saben utilizar las tecnologías. Un software facilita la forma de vivir, de muchas maneras, permite la interacción con el ordenador que finalmente realiza innumerables tareas. Se espera llevar estas herramientas a quienes lo necesitan, entendiendo que la innovación comprende muchos escenarios.

Si bien este desafío apunta a la problemática de solicitud de servicios a domicilio, la resolución a este también engloba alternativas de búsqueda de trabajo por la contraparte. Permitiría que un oferente de servicio pueda realizarlo de manera particular, apoyado por el sistema de gestión de servicio y de comunicación.

5.2. Soluciones Existentes / Estado del Arte

A. Habitissimo[1]: se creó en abril del año 2009 y desde entonces se ha consolidado como la comunidad online de obras y reformas en España, Brasil, Italia, Portugal, Argentina, Chile, México y Colombia. Es una empresa de Internet líder que ayuda a conectar oferta y demanda en el sector de obras, reformas y servicios para el hogar. A los solicitantes les ofrece la posibilidad de publicar un presupuesto para que hasta cuatro profesionales de su zona les puedan contactar. Cuenta con más de 270.000 profesionales registrados y ya ha publicado más de un millón de presupuestos online para reformas y servicios de hogar.

“Habitissimo ayuda en diferentes proyectos de construcción en países en desarrollo mediante Kiva. Kiva permite conectar con y prestar dinero a pequeños negocios, ofrece la posibilidad de realizar donaciones y de presentar proyectos que necesiten financiación. Así, diferentes trabajadores de Habitissimo colaboran realizando donativos a personas que necesitan desarrollar un proyecto de construcción en cualquier parte del mundo” [2].



Figura 1: Página web oficial Habitissimo.

B. Listoco[3]: Es una plataforma que busca automatizar la contratación servicios profesionales para los hogares y empresas. Su aplicación móvil tanto para Android y iOS, te permite buscar un profesional que solucione un problema particular, especificando dónde y cuándo quieres el servicio. Poseen una red de profesionales donde buscas entre 30 categorías y cotizas. Se enfocan en dos tipos de clientes. El solicitante del servicio y el profesional que lo presta. Buscan también hacer crecer el negocio del profesional o empresa en cuestión, permitiéndole publicidad para llegar a nuevos clientes. No cobran comisión por los trabajos realizados, sino un pago mensual por el uso de la plataforma. La plataforma web es de registro e informativa, y la aplicación móvil es donde ocurre la gestión y agendación de los servicios.

Misión: “Ayudar a profesionales independientes, microempresas, emprendedores y a toda persona que quiera una oportunidad para ofrecer sus servicios, a obtener nuevos clientes para hacer crecer su negocio, mejorando la calidad de vida de miles de familias, aportando al desarrollo del país.” [4]

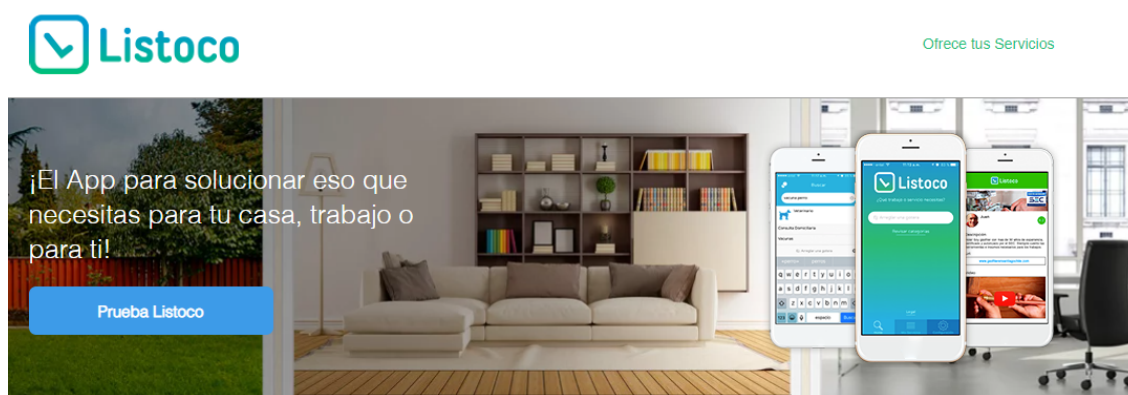


Figura 2: Página web oficial Listoco

C. Doctoralia[5]: Doctoralia es una plataforma española, que forma parte del grupo DocPlanner[6], y es la plataforma líder mundial que conecta pacientes con profesionales de la salud, haciendo la experiencia en salud más humana. Doctoralia ganó el premio AppCircus¹ a la mejor aplicación móvil relacionada con la salud y el bienestar. Para principios del 2020 la constituían 45.000 médicos y centros sanitarios de España y 15 países más. Para esa fecha tenía más de 100.000 descargas. Se encuentra disponible en Android y Apple en siete idiomas distintos.

Misión: “Queremos que los pacientes encuentren al médico perfecto y programen una cita de la manera más fácil. El viaje del paciente debe ser agradable, y por eso siempre estamos a su lado: para ayudarlo a encontrar la mejor atención posible. En cualquier momento, en cualquier lugar.

¹<https://www.europapress.es/portaltic/software/noticia-app-espanola-doctoralia-gana-unanimidad-internacional-premio-appcircus-20121119190631.html>

También ayudamos a los médicos a administrar mejor su práctica y a construir su reputación en línea. Con nuestra solución integral integrada, los médicos pueden no solo mejorar su presencia en línea, sino también dedicar su tiempo a lo que realmente importa: sus pacientes. ”[6]



Figura 3: Página web oficial Doctoralia

6. Desarrollo del tema

6.1. Estudio Mercado

En la evolución de la población chilena, según la encuesta Casen por el Ministerio de Desarrollo Social[7], desde 1990 hasta 2017 ha mostrado un crecimiento notorio de los adultos mayores en la sociedad. Este hecho es mostrado en una segmentación de 5 grupos etarios en un rango de 27 años.

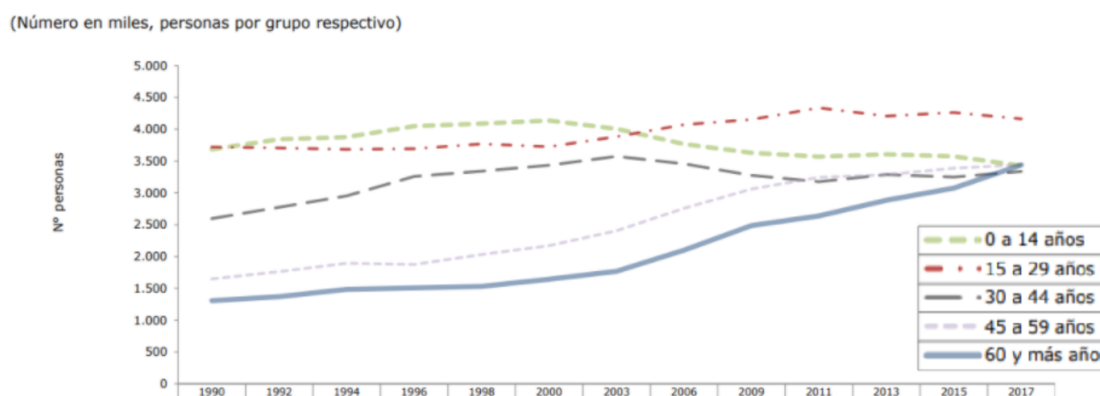


Figura 4: Grupos etarios. Fuente: Encuesta Casen 2017

De acuerdo a las cifras entregadas por el Instituto Nacional de Estadísticas[8], utilizando lo recabado en el Censo 2017, entrega una proyección en la demografía reflejada en la Figura 2.

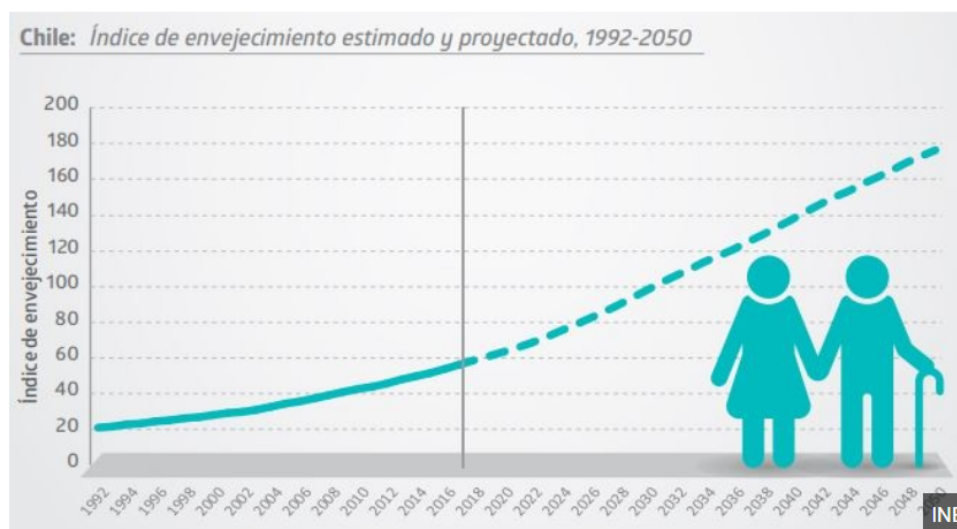


Figura 5: Proyección proceso envejecimiento. Fuente: Encuesta Casen[7]

La esperanza de vida para ambos sexos aumentó y para el 2050 será de 83,2 años para los hombres, y de 87,8 años para las mujeres, lo que se traduce en un menor número de nacimientos y el aumento de la población de adultos mayores. Según el organismo la población de 65 años o más representará el 25 % de la población en 2050.

Ahora, frente al uso de la tecnología, hay que estudiar la relación que existe con el adulto mayor, para poder tomar decisiones de cómo llegar a ellos. La interacción que tiene los adultos mayores con la conexión a Internet se refleja en el siguiente gráfico. Se presenta cuánto porcentaje de los adultos mayores usa internet del 2006 al 2017.

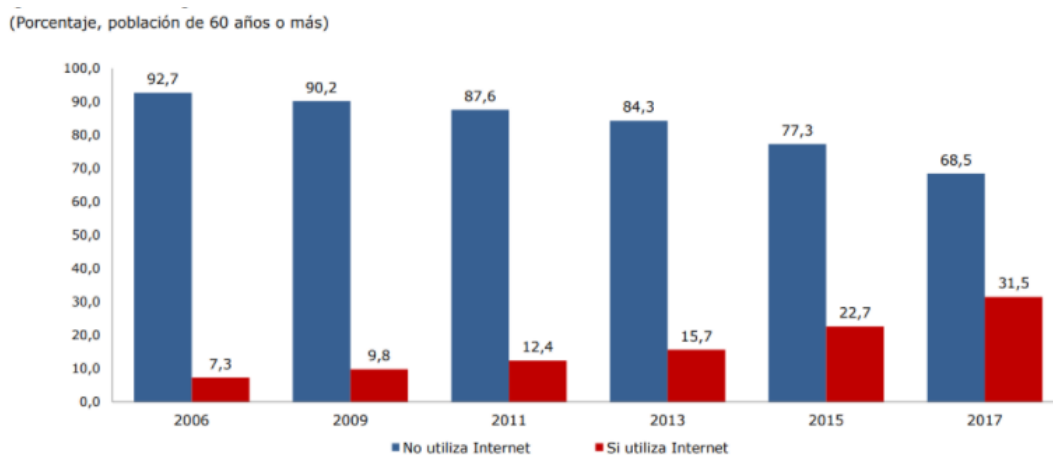


Figura 6: Uso de internet por población con 60 años o más. Fuente: Encuesta Casen 2017[7]

Se aprecia un crecimiento sostenible. Considerar además, que las generaciones van avanzando, y las personas de la actualidad con un fuerte acercamiento a la tecnología, serán finalmente el adulto digital.

Hay que considerar, como usuarios no solo a los adultos mayores. Esto porque un conviviente de ellos, ya sea un tutor o apoderado, familia, vecinos, podrían utilizar la aplicación para la solicitud del servicio. Este aspecto, consiste en ver cómo es la comunicación del adulto mayor en su entorno, y para describir eso se va a separar en características distintas. Con primera característica es su estado civil. Se van a separar en hombres y mujeres, y su estado civil va a poder ser una de las 5 situaciones que se presentan en la siguiente tabla.

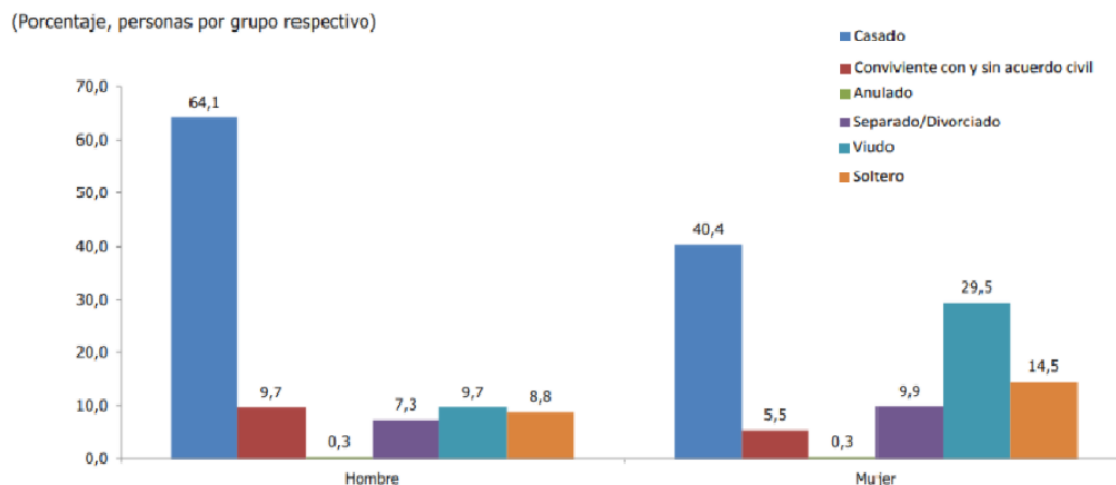


Figura 7: Distribución estado civil. Fuente: Encuesta Casen 2017[7]

Se presenta que en los hombres hay un poco más de convivencia con su pareja. Sin embargo, hay que tener en cuenta que las mujeres tienen una esperanza de vida mayor que los hombres. Aún así, en términos generales la mitad de los adultos mayores conviven con sus parejas. Finalmente según la Comisión Nacional de Investigación Científica y Tecnológica[9], con el programa IDeA[10] y en su artículo “Adultos mayores en Chile: descripción de sus necesidades en comunicación en salud preventiva”[11] describe tres grupos:

- Círculo 1: Personas más cercanas al adulto mayor.
- Círculo 2: Personas cercanas al adulto mayor.
- Círculo 3: Personas más alejada del adulto mayor.

En un contexto de adultos mayores autovalente, entrega la siguientes características.

		Círculo 1	Círculo 2	Círculo 3
Llamada telefónica	Todos los días	42,6	12,9	6,5
	Varias veces por semana	31,4	21,9	20,3
	Alrededor de una vez por semana	13,5	27	28,4
	Menos de una vez por semana	5,7	29,9	31
	Nunca	6,8	8,3	13,8
	Total	100	100	100
		N: 542	N: 411	N: 232
Mensaje de texto	Todos los días	25,9	10,6	3,9
	Varias veces por semana	16,1	13,3	15,2
	Alrededor de una vez por semana	8,1	7,1	7
	Menos de una vez por semana	6,7	12,8	14,8
	Nunca	43,3	56,3	59,1
	Total	100	100	100
		N: 541	N: 407	N: 230
E-mail	Todos los días	3,7	1,2	2,6
	Varias veces por semana	11,7	6,6	7,7
	Alrededor de una vez por semana	14,7	8,8	11,6
	Menos de una vez por semana	18,4	16,6	20,6
	Nunca	51,5	66,7	57,5
	Total	100	100	100
		N: 538	N: 409	N: 233
Presencial	Todos los días	39,5	17,3	9,5
	Varias veces por semana	16	10,9	15,5
	Alrededor de una vez por semana	25,4	30,7	22
	Menos de una vez por semana	15,6	34,8	39,7
	Nunca	3,5	6,3	13,4
	Total	100	100	100
		N: 539	N: 411	N: 232

Figura 8: Fuente: Encuesta Fondef[12] Idea- Sistema de comunicación en salud Adultos mayores y redes de apoyo.

Los resultados entregan que, para los 3 círculo, las llamadas telefónicas resulta ser un medio igual de recurrente a una conversación de forma presencial.

Finalmente, se consideran varios aspectos. Gracias a, el aumento de las herramientas informáticas, la esperanza de vida que va en aumento junto con su calidad de vida, el uso de internet, y la digitalización a medida que las generaciones avanzan, se plantea la solución propuesta: Plataforma tecnológica que permita el contacto directo y la agendación automatizada.

6.2. Elicitación

La elicitación es el acto de tomar requerimientos, su finalidad es comprender el problema para proveer una solución adecuada para las necesidades del cliente. Este ejercicio se lleva a cabo entre los desarrolladores y el cliente.

Una forma de clasificar requerimientos consiste en discriminar si son funcionales o no funcionales. Por un lado los requisitos funcionales son capacidades o comportamientos que debe tener el sistema, por otro lado los requisitos no funcionales son cualidades o características que debe tener el sistema. Otra manera de clasificar requerimientos es distinguir si son obligatorios o deseables. Los obligatorios son aquellos fundamentales para el funcionamiento del sistema, los deseables no son fundamentales para el funcionamiento, sin embargo cubren alguna necesidad complementaria o aportan alguna funcionalidad o atributo extra.

Los requisitos funcionales deben ser comprobables. Un criterio para determinar si se cumple un requisito es mediante las **historias de usuario**(ver apartado anexos 9.1), esta técnica consiste en enunciar una sentencia cuyo valor de verdad indica si el requisito es satisfecho. Otro método consiste en definir los **casos de uso**(ver apartado anexos 9.2) relacionados al requerimiento. Los requisitos no funcionales son complejos de medir, por lo tanto difíciles de comprobar. Una técnica para determinar que decisiones de diseño permiten satisfacer estos requerimientos consiste en elaborar un diagrama **soft-goals**(ver apartado anexos 9.3) para determinar que medidas contribuyen a satisfacer estas necesidades.

6.3. Diseño del Backend

6.3.1. Modelo de datos

El modelo de datos se define con una tabla para Demandantes y Oferentes, con sus atributos correspondientes de registro y contacto: email, password, name, dirección, Rut, fotos etc. Por un lado para el demandante se definen atributos para su perfil médico, en cambio para el oferente se definen atributos tales como sus honorarios, valoración, previsión, profesión, entre otros aspectos relevantes de trabajo. Una tabla de Solicitud, que maneja los estados del agendamiento de VIDAT, además de sus coordenadas geográficas y dirección física, la fecha de registro y los identificadores de las dos partes involucradas. Como un oferente de salud puede tener más de un horario disponible por día, la relación muchos a muchos es manejada con una tabla horas_has_fecha, que relacionan la Fecha con el bloque de horario. Finalmente el oferente tiene un carrusel de imágenes en su perfil de presentación de servicio, más detalles en el Frontend

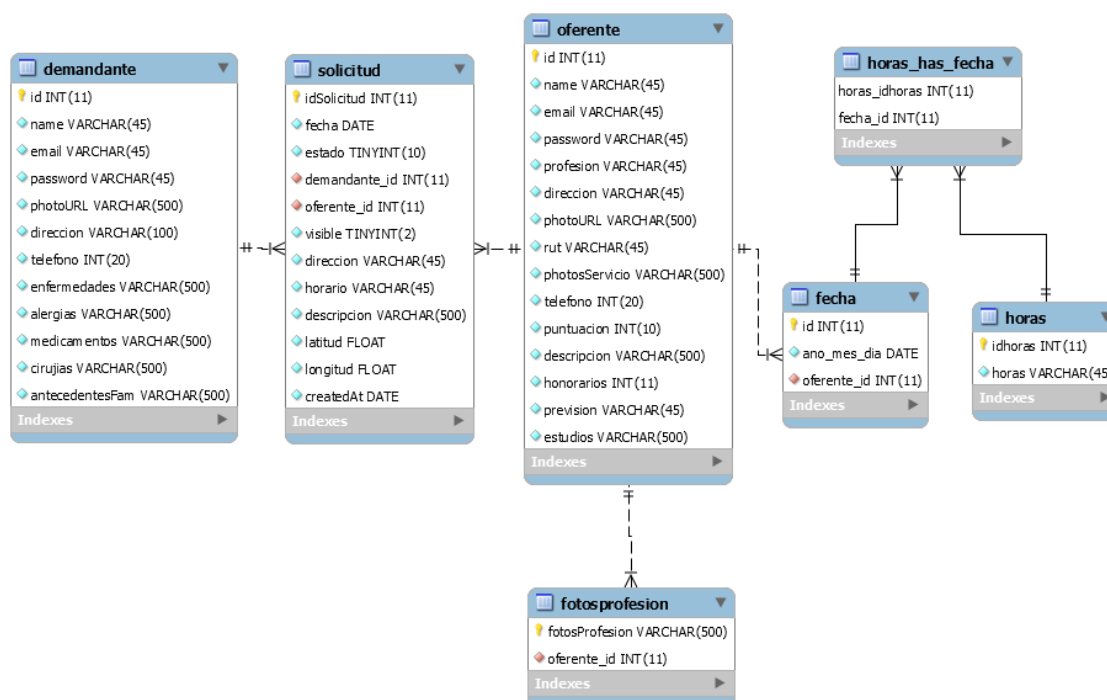


Figura 9: Modelo base de datos

Debido al flujo de la información en la red se pueden generar cuellos de botella la aplicación web, además en miras de buscar escalabilidad es conveniente considerar un esquema de microservicios. Esta arquitectura de software puede llegar a contribuir con la resiliencia del sistema, capacidad de respuesta, optimización de recursos, adaptabilidad, modularidad (ver figura 10).

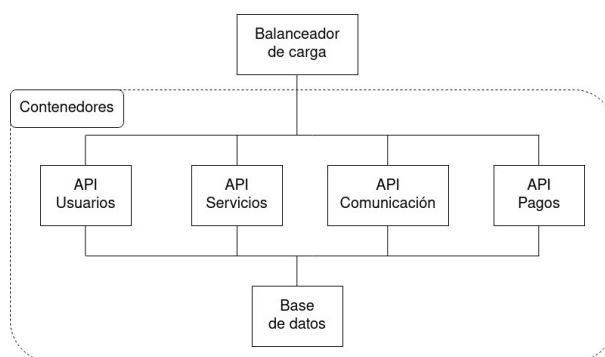


Figura 10: Arquitectura del Backend. Fuente: Autoría memoristas.

Ahora, es necesario establecer el propósito de cada microservicios, una estrategia es relacionarlos a los requisitos funcionales, luego en caso de algún problema, algunas funcionalidades fundamentales pueden permanecer operativas. Se muestra a continuación

el patrón de Diseño Facade. Este patrón estructural permitirá simplificar el sistema desde el punto de vista del cliente. Proporcionar una interfaz unificada para un subconjunto de sistemas. Y para cada subsistema se tendrá una API. Los componentes del sistema, permitirán hacer fetch a distintas APIs, de esa manera mejora la resiliencia del sistema, y por consiguiente mayor survivability².

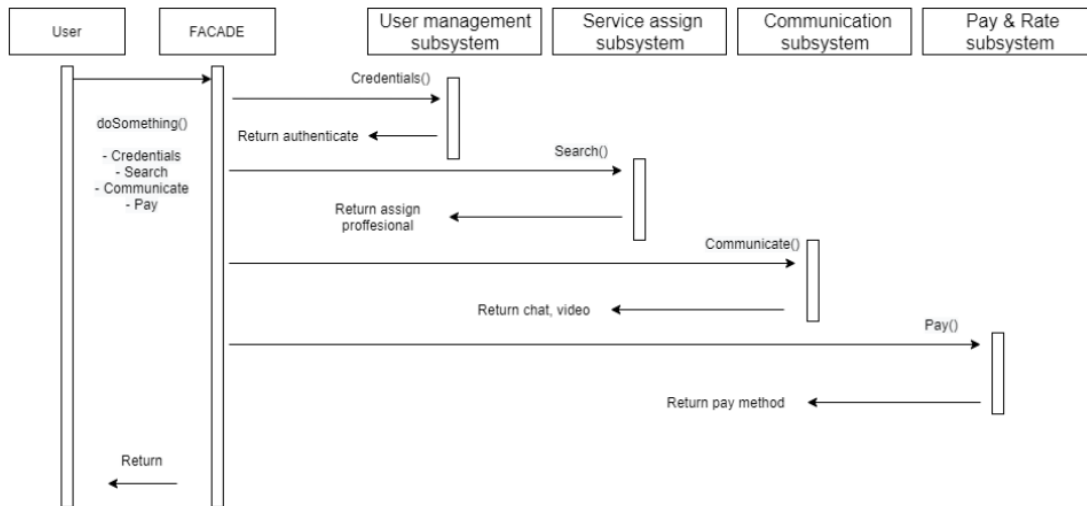


Figura 11: Patrón Facade. Fuente: Autoría memoristas.

6.3.2. Microservicio: Usuarios

Este módulo se encarga de administrar la información de los usuarios que utilizan el sistema y asegurar su seguridad. Algo propio del problema es que por la seguridad del demandantes de servicios se debe asegurar que los oferentes de servicios están certificados para brindar atención. Para el caso de las prestaciones de salud, los oferentes de servicio son profesionales de la salud y la certificación es pertenecer al Registro Nacional de profesionales individuales de salud” (Se adjunta diagrama de secuencia en anexos 9.4)

6.3.3. Microservicio: Servicios

Este módulo se encarga de administrar la información asociada a las prestaciones de servicios, es importante recalcar que la lógica del negocio con la que se ejecuta el servicio es parte de la experiencia usaria que brinda el Frontend. (Se adjunta diagrama de secuencia en anexos 9.5)

²Capacidad de un sistema para continuar prestando un servicio activamente bajo condiciones adversas. Se consideran tanto desastres naturales, como accidentes y ataques deliberados

6.3.4. Microservicio: Comunicación

Este módulo se encarga de administrar las canales de comunicación que existen entre los usuarios. Para beneficio del software como producto comercial es necesario que la mayor cantidad de personas se mantengan utilizando el servicio, por lo que es necesario que demandantes y oferentes de salud gestionen la hora de atención mediante la aplicación móvil (Se adjunta diagrama de secuencia en anexos 9.6)

6.3.5. Microservicio: Pagos

Este módulo es necesario para hacer del sistema económicamente rentable, pues la ganancia de dinero ocurre al crear un lazo entre un demandante y un oferente de servicios.

6.4. Implementación del Backend

Al momento de implementar es necesario escoger que herramientas utilizar. Para gestionar contenedores se ha escogido Docker por su basta documentación y soporte, mientras que para efectuar configuraciones en la red y balancear la carga de las peticiones en los contenedores se ha escogido HAProxy³ como balanceador de cargas. La implementación del Backend consta de dos fases: Desarrollo de la aplicación web y despliegue en la red. Para el desarrollo de la aplicación web se han utilizado las siguientes herramientas: MariaDB, ExpressJS, Sequelize, Nodemon. Para el despliegue en la red se han escogido Docker para implementar microservicios y HAProxy para configurar la comunicación de los contenedores.

La estructura de los módulos de software y servicios utilizados en cada microservicio se ilustra en la figura 12. El circuito que conecta los módulos representa el flujo de datos desde internet hacia la base de datos. Por un lado, los servicios son módulos disponibles en una comunidad de internet para propósitos generales, por otro lado, los software son módulos desarrollados para fines específicos.

El balanceador de cargas se ubica en la frontera del sistema operativo (ver figura 12), pues representa la puerta de enlace hacia internet. El balanceador de cargas es responsable de redireccionar las peticiones HTTP/HTTPS a los distintos contenedores Docker. Entre los servicios disponibles para balancear cargas se encuentran NginX, Apache, HAProxy, siendo el último el elegido.

Docker es un servicio que permite reservar recursos para proveer un ambiente de ejecución para programas específicos, de esta manera los recursos de un servidor pueden ser utilizados de manera eficiente, además facilita el despliegue en producción del microservicio. (ver figura 12)

³<https://www.haproxy.com/>

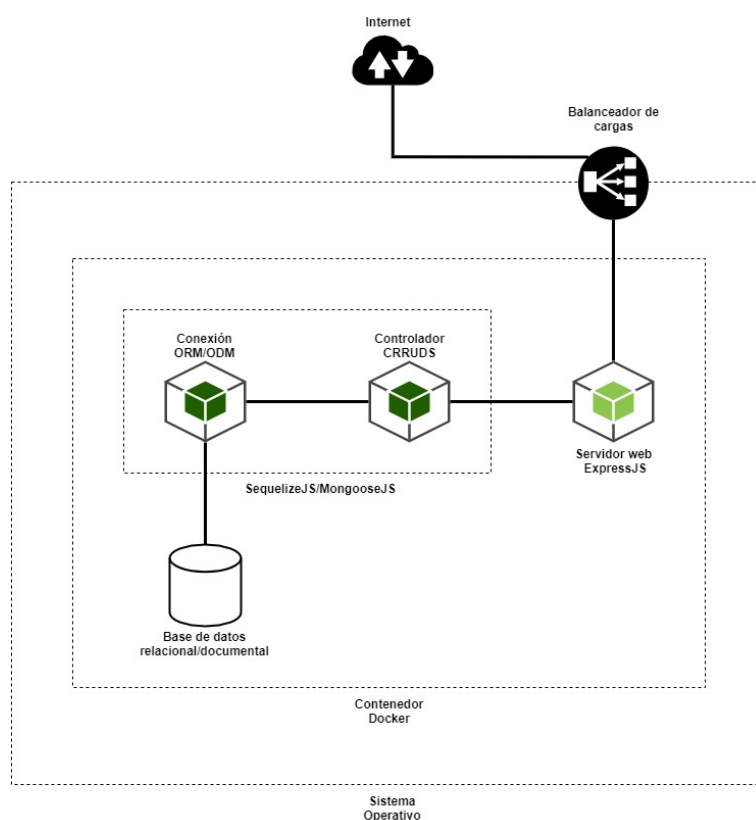


Figura 12: Estructura de módulos de Software y Servicios. Fuente: Autoría memoristas.

6.5. Diseño del Frontend

Antes de la implementación a nivel de códigos del Frontend, se realizaron dos hitos claves: flujo de navegación y la maqueta de las vistas.

Según los sistemas descritos en los apartados anteriores: Registro e inicio sesión, Agendación, Comunicación y Pago y Valoración; que tuvo también protagonismo en el Backend, se definieron las 4 fases para el flujo de navegación. Tanto para la aplicación del demandante como la del oferente. En la práctica se definió primero el flujo de navegación del demandante y su maqueta para las vistas. Una vez diseñada dicha aplicación con éxito, se continuo con la aplicación del oferente. Así luego, se tuvo dos prototipos funcionales y se fueron mejorando en iteraciones cortas en conjunto. Se orienta la explicación en el informe de la misma forma. El flujo de navegación es el siguiente.

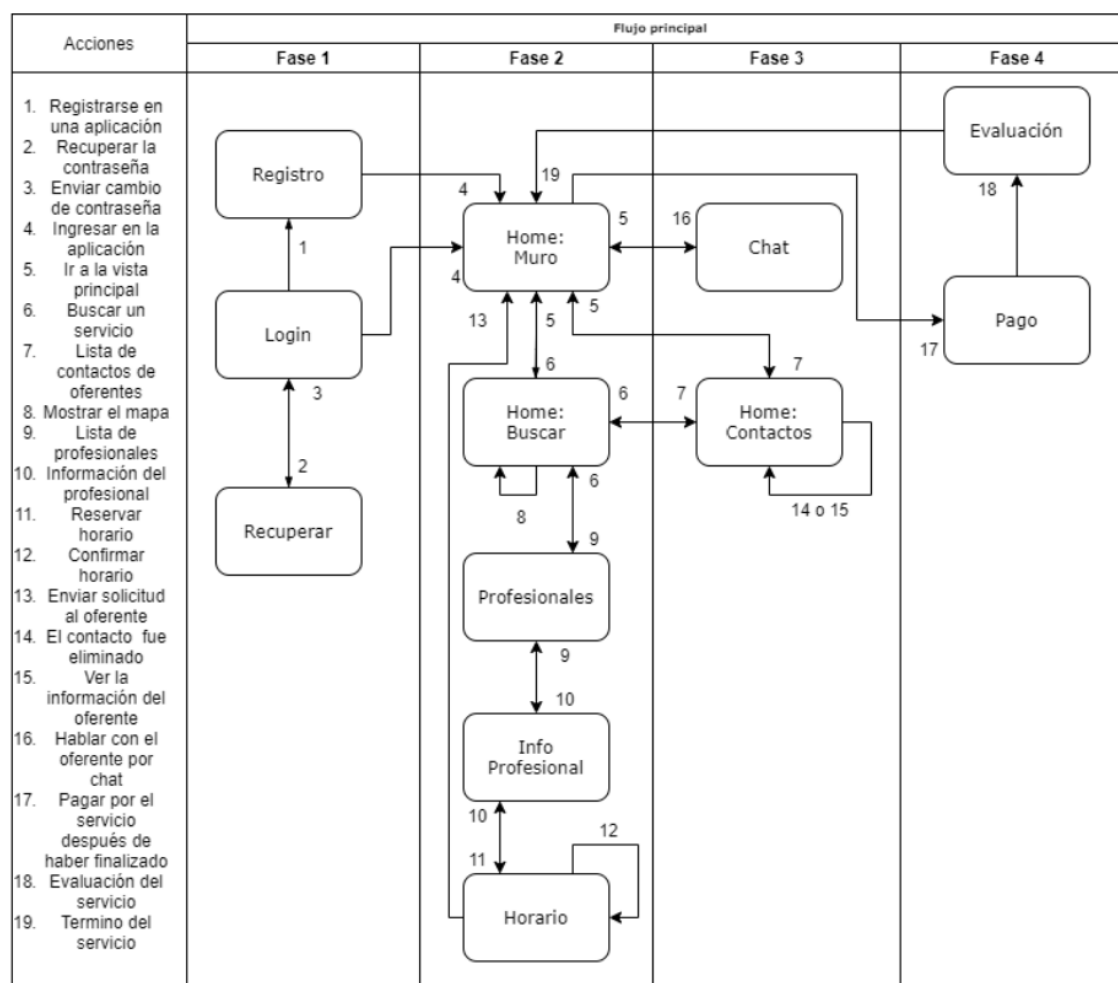


Figura 13: Flujo principal VIDAT demandante. Fuente: Autoría memoristas.

A modo explicativo, el inicio del flujo comienza con el registro y el inicio de sesión. Una vez autenticado la redirección inicial a la fase 2 será hacia stack principal HOME. Este contiene las vistas de MURO, BUSCAR Y CONTACTOS. Todas las funcionalidades se encuentran detalladas en el flujo y en el apartado anterior con los casos de uso. Por otra parte, la fase 3 y 4, que son el subsistema de comunicación y pago respectivamente, se accede desde el Muro, según los eventos correspondientes. Además, en la aplicación existe un menú. Allí se puede navegar en los stack principales, que son HOME, PERFIL, HISTORIAL, ESTADÍSTICAS, PAGO Y CONFIGURACIONES, ver flujo secundario (ver Figura 14). Considera las actualizaciones del perfil, configuraciones del método de pago, configuraciones generales, entre otros. Estas funcionalidades no corresponden a los subsistemas ni a los requisitos funcionales, aún así son características claves de la experiencia del usuario.

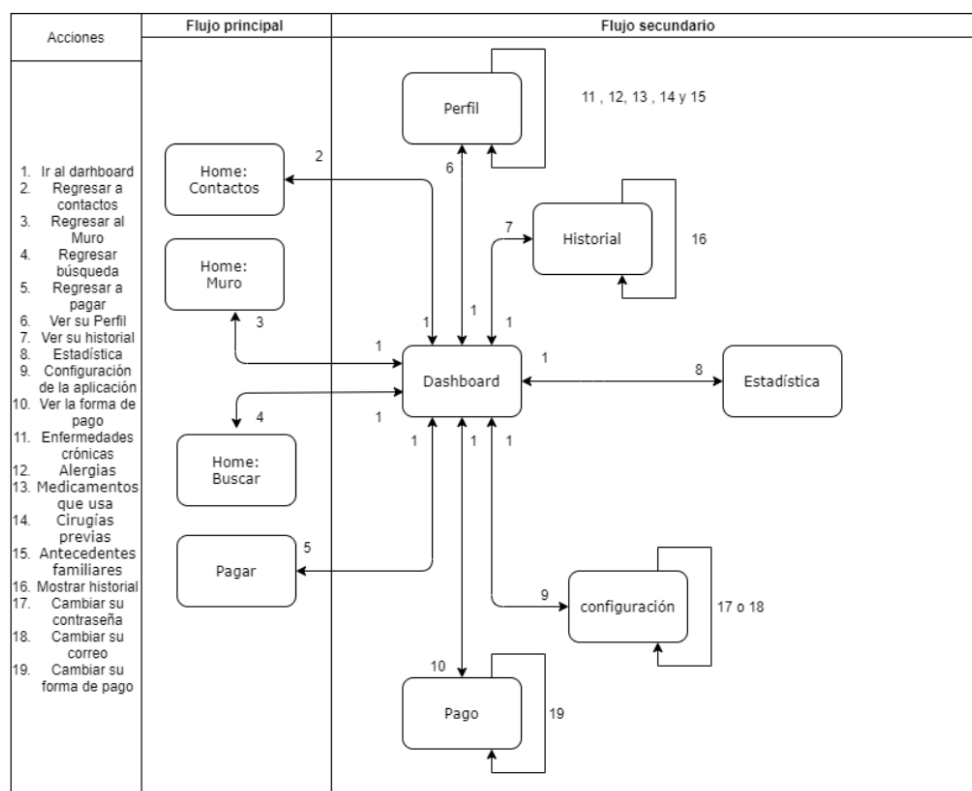


Figura 14: Flujo secundario VIDAT demandante. Fuente: Autoría memoristas.

Para los prototipos iniciales se utilizó el programa Adobe Xd[14], esto porque en primeras instancias se debe diseñar prototipos baratos que pudieran ser desechados rápidamente. Se muestra a continuación las imágenes de la maqueta de la aplicación VIDAT para el demandante.

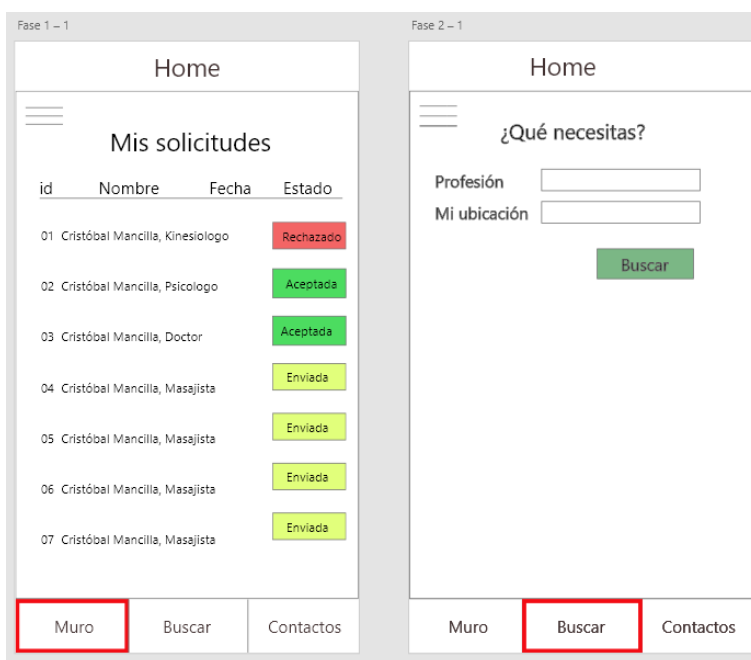


Figura 15: Prototipo Adobe Xd para demandante. Vista Muro y Buscar.

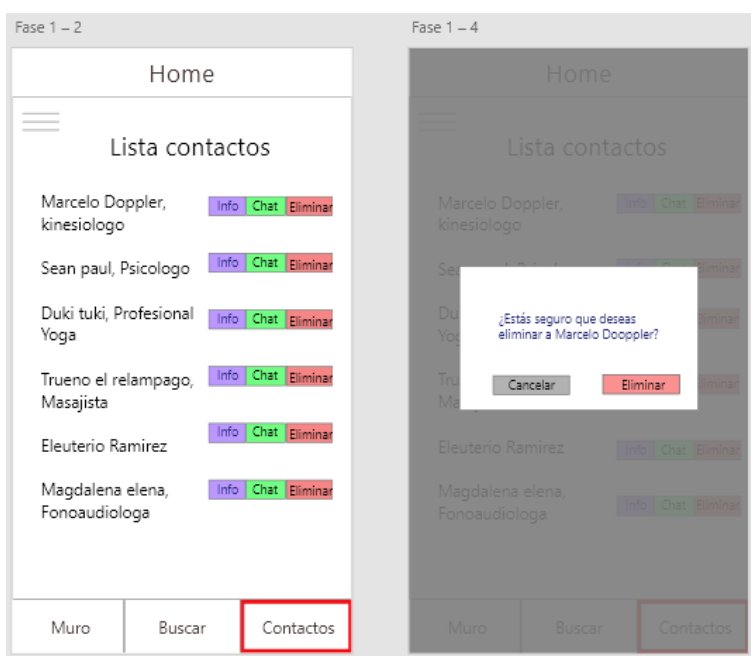


Figura 16: Prototipo Adobe Xd para demandante. Vista Contactos.

Estas vistas permitirán solicitar un profesional de la salud a domicilio. Y una vez aceptada y agendada la solicitud, permitirá comunicarse con oferente. Se muestra todas

las vistas a continuación. Para mayor detalle se adjunta los fuentes en el entorno de trabajo[13]

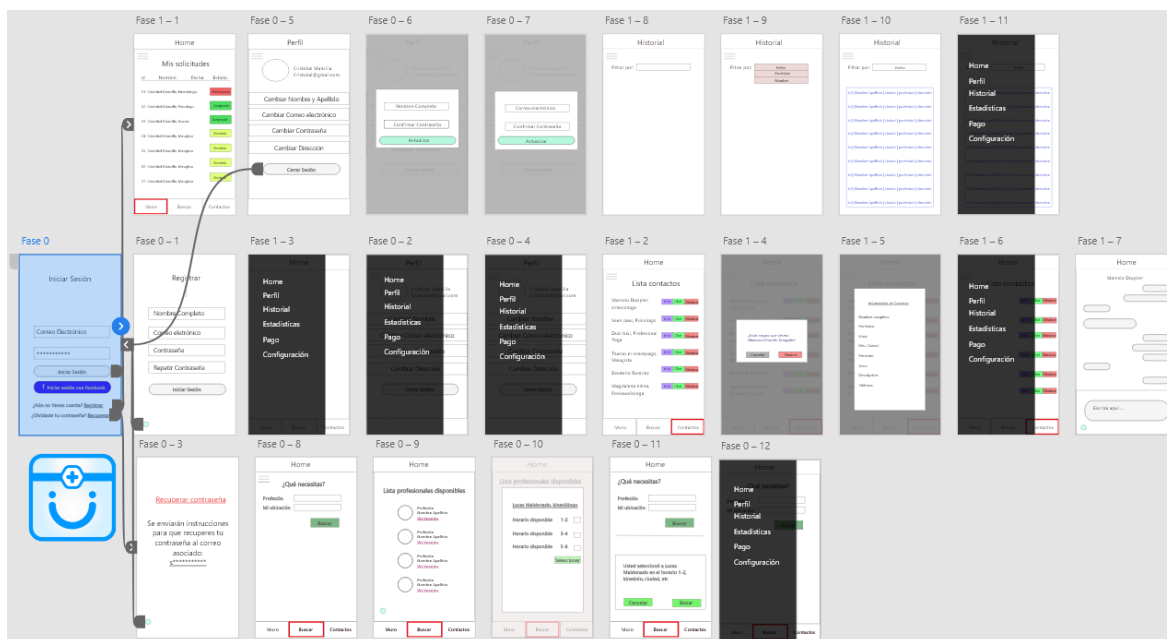


Figura 17: Maqueta Adobe Xd para demandante.

Por consiguiente, la aplicación del oferente sigue la misma idea, con algunas funciones distintas. Se adjunta el flujo principal de navegación.

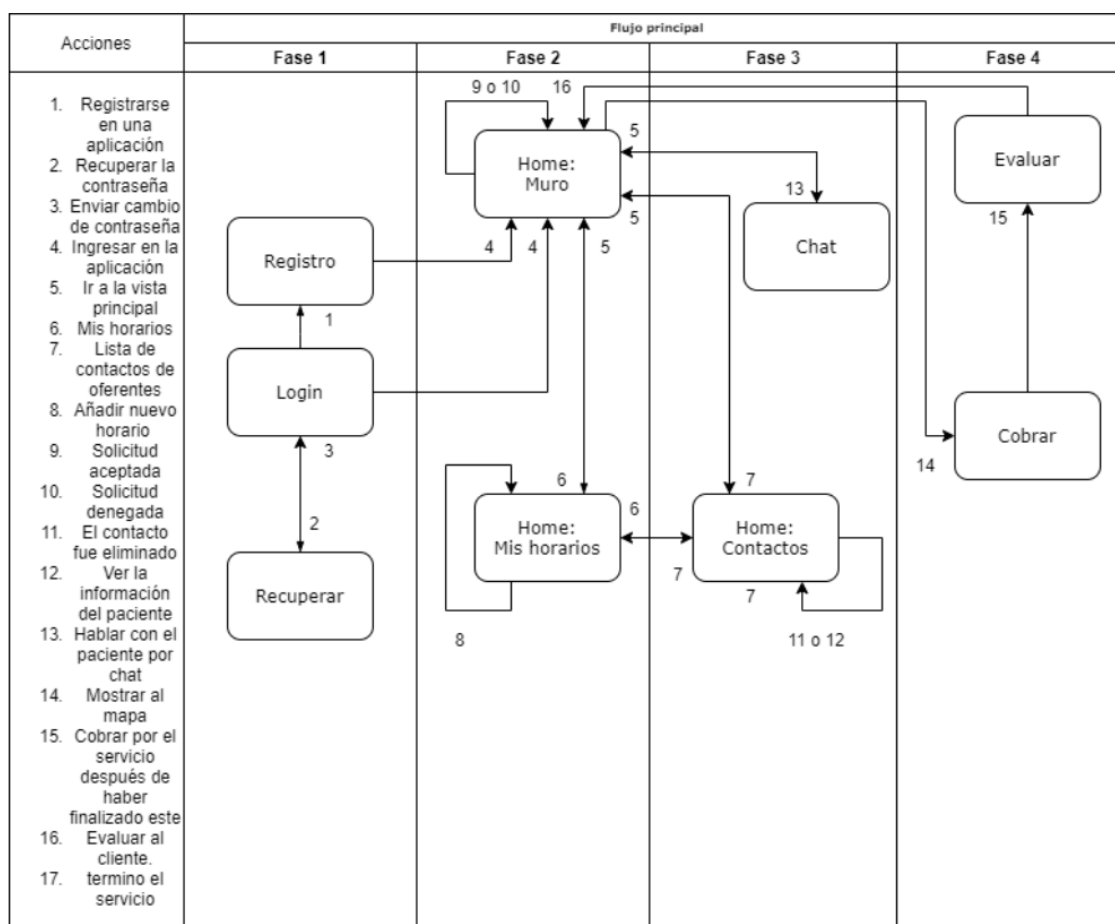


Figura 18: Flujo principal VIDAT oferente. Fuente: Autoría memoristas.

De igual manera, de la fase 1 de autenticación, se redirecciona a la fase 2, y se encuentra el stack del HOME, que contiene las vistas de MURO, MIS HORARIOS, CONTACTOS. A la fase 3 y 4, de Comunicación y Pago, respectivamente, se puede acceder desde el Muro. Considerar que el flujo es interactivo, esto significa que teniendo el menú desplegable con distintos stack, se puede navegar de vista en vista.

Existe también un flujo secundario, con las configuraciones generales y de perfil. El profesional dispone de un carrusel de imágenes, que le permite mostrar su trabajo como carta de presentación hacia los usuarios. Estas funcionalidades afectan directamente a la experiencia de usuario, característica relevante en el desarrollo de aplicaciones móviles. Además, en este flujo se define la vista de Agenda, que administra las solicitudes aceptadas y confirmadas en un calendario semanal, con la información correspondiente del paciente y domicilio.

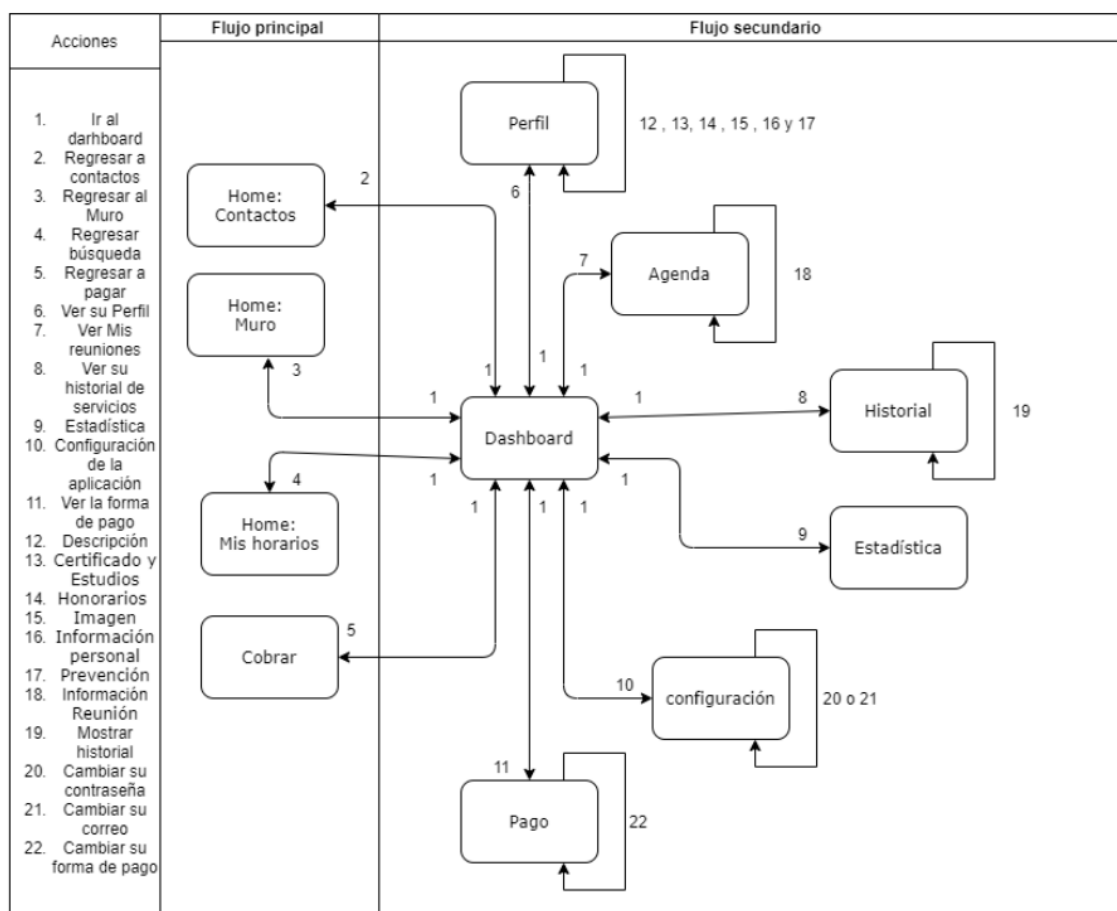


Figura 19: Flujo secundario VIDAT oferente. Fuente: Autoría memoristas.

Se utilizó Adobe Xd nuevamente para diseñar el prototipo y darle visibilidad al flujo de navegación para la aplicación del oferente. Se adjunta la dirección del entorno de trabajo de Adobe Xd[14]. Estas maquetas simulan las funcionalidades de las aplicaciones móviles, desde el flujo de navegación hasta el comportamiento de la agendación entre las dos partes.

The image shows two side-by-side mobile app screens for a service provider. Both screens feature a blue icon of a medical bag with a smiley face.

Left Screen (Login):

- Greeting: ¡Hola, soy Vidat!
- Input fields: Correo Electrónico, Password (masked with *).
- Buttons: Iniciar Sesión, Iniciar sesión con facebook.
- Links: ¿Aún no tienes cuenta? [Registrar](#), ¿Olvidaste tu contraseña? [Recuperar](#).

Right Screen (Register):

- Header: Registrar
- Input fields: Nombre Completo, Correo electrónico, Contraseña, Repetir Contraseña.
- Buttons: Iniciar Sesión.

Figura 20: Prototipo Adobe Xd para oferente. Vista Login y Register.

The image shows two side-by-side mobile app screens for a service provider. Both screens have a blue header with a hamburger menu icon and the word 'Home'.

Left Screen (Mis solicitudes):

Header: Mis solicitudes

Nº	Paciente	Hora	Confirmar
01	Sebastián Ramírez	14:00 - Vi 04	Si No
02	Cristian Ocampo	15:00 - Sa 05	Si No
03	Luis Pacheco	12:00 - Do 06	Si No
04	Miguel Águila	12:00 - Ju 03	Aceptada
05	Nicol Quintero	18:00 - Ju 03	Aceptada
06	Andrea Molina	20:00 - Ju 03	Denegada

Bottom navigation: Muro, Mis Horarios, Contactos.

Right Screen (Mis horarios disponibles):

Header: Mis horarios disponibles

Date range: 5 Octubre - 7 Octubre

	Lu	Ma	Mi	Ju	Vi
08:00					
09:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					

Bottom navigation: Muro, Mis Horarios, Contactos.

Figura 21: Prototipo Adobe Xd para oferente. Vista Muro y Mis horarios.

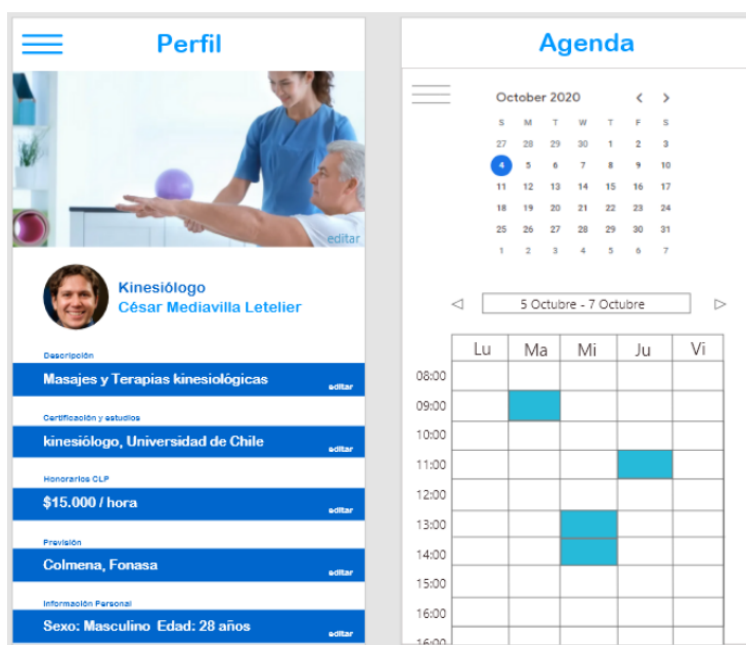


Figura 22: Prototipo Adobe Xd para oferente. Vista Perfil y Agenda.



Figura 23: Prototipo Adobe Xd para oferente. Vista Contactos y Chat.

6.6. Implementación del Frontend

Si bien la arquitectura, junto con las bases de datos engloba un solo sistema general, tanto oferentes como demandantes tienen sus propias funciones. Es por esto que se realizarán dos aplicaciones móviles. Las aplicaciones que se detallarán a continuación fueron diseñados e implementados para dispositivos con sistema operativo Android y iOS. Como la aplicación contempla usuarios en 5 a 7 años más, la implementación se sugiere en una pantalla de 5.5 pulgadas. Esto es relevante para el diseño de la interfaz hacia los adultos mayores, considera tamaño de letras, botones, vistas, etc. Las consultas a la API proporcionada por el Backend se detallarán en el apartado de Integración.

6.6.1. Descripción de los componentes y funcionalidades

Gestión usuarios	
Descripción General	Este componente cumple con el registro y validación de las credenciales de los usuarios. Tanto para los oferentes como los demandantes. En el caso de los oferentes, se debe comprobar que se encuentre en los registros de la superintendencia como profesionales de salud certificados.
Interacción	Este componente es el primero con el que el usuario interactúa. Da lugar a todas las funcionalidades de la plataforma tecnológica. Interactúa con las bases de datos respectivas de cada usuario.
Marco teórico de implementación	La interfaz se implementa como un stack con 3 vistas: Inicio sesión, registrarse y recuperar contraseña.
Elementos tecnológicos	Cuando el sistema registra o loguea, realiza una petición http a la API; esta fue configurada con express, mysql, socket, etc. Es una API encargada de administrar la información relacionada a los usuarios. Efectúa operaciones CRUD sobre una base de datos para la administración de los usuarios. Las limitación de cómputo están relacionadas con los recursos físicos reservados. Utiliza componentes de React Native para cumplir con el flujo de navegación.
Funciones	La ejecución del login recibe como parámetros las credenciales (correo, pass, o social login), autentifica y permite acceso al sistema. Para la función del registro, recibe como parámetros input relevantes para poder realizar los demás subsistemas (nombre, correo, contraseña, dirección). Esto es crucial, porque si no se gestiona bien, la agendación de servicio no será posible. Además se utiliza un servicio externo, que son los bancos de la superintendencia, para la acreditación de los profesionales de salud certificados.

Tabla 1: Descripción componente de Gestión de Usuarios.

Agendación	
Descripción General	Solicitar un servicio de salud a domicilio, según horarios disponibles del oferente.
Interacción	En este componente, hay interacción indirecta entre el profesional de la salud y el demandante. Cada usuario interactúa con su interfaz y con sus respectivas funciones. Hacen peticiones a la misma API que gestiona la asignación de servicios.
Marco teórico de implementación	Para el adulto mayor comprende 2 stacks. El primer stack es “Buscar”, con 4 vistas las cuales permiten: identificar la dirección y el servicios que necesita, mostrar el listado de profesionales, el perfil del profesional, y finalmente los horarios disponibles. El segundo stack es el “Muro” donde refleja si el estado de las solicitudes enviadas. Para el oferente de salud. También corresponden 2 stack. El primero es para configurar los horarios disponibles semanalmente. Y el segundo se encuentra en su “Muro”, donde acepta o rechaza solicitudes. Las solicitudes tienen estados: Enviado, Aceptada, Rechazada, Confirmada, Cancelada, Finalizada.
Elementos tecnológicos	Algunas tecnologías para el prototipo son el componente “Maps” de React Native, para seleccionar la dirección del domicilio. La Api de Google para convertir las coordenadas geográficas en dirección física (se Adjunta la configuración a la API Geocoding en el apartado anexos 9.9). Se utiliza “Picker” para que el usuario ingrese input predeterminados. Como herramienta tecnológica se utiliza el “Calendar” del framework, para seleccionar los horarios disponibles del oferente. Todo esto en conjunto a la api del subsistema de la Agendación. Otros componentes relevantes son: carrousel, rating, expo-permissions, FlatList, ventanas modales, etc.
Funciones	Permitir al demandante agendar una hora con el profesional de la salud que necesite, respecto a un banco de registros con profesionales acreditados en su ciudad correspondiente. Donde el oferente, por la contraparte, acepta o rechaza las solicitudes, en base a sus consideraciones. La solicitud aceptada estará vigente en las vistas de la Agenda del oferente y en el Muro del solicitante, aún así existirá un ultimo estado de la solicitud donde confirma la hora, esto debido a que se da la opción de cancelar la hora eventualmente.

Tabla 2: Descripción componente de Agendación.

Comunicación	
Descripción General	Comunicación directa en tiempo real, de las dos partes vinculadas. Implica un chat y la información de contacto de la contraparte.
Interacción	Interactúa el adulto mayor con el profesional de la salud, mediante sus identificadores y servicios externos.
Marco teórico de implementación	Se realiza un chat temporal (en la vista del muro) una vez que el vínculo es creado. Esto significa, desde que el oferente aceptó la solicitud de un demandante, hasta que el trabajo es realizado con éxito. Cabe mencionar que existe un plazo después del trabajo realizado, para considerar comunicación en trabajos no completados o reclamos del servicio en cuestión.
Elementos tecnológicos	Para la realización del chat, se considera un servicio externo (Firebase) para el almacenamiento de una gran cantidad de data, como los son las colecciones de textos de cada vínculo creado entre los usuarios. Se utiliza el componente gifted-chat de React Native. Además conecta a la API encargada del subsistema de comunicación.
Funciones	Permite, durante un tiempo, que el oferente y demandante, se comuniquen, e intercambien información relevante para la realización del servicio. Eventualmente el chat temporal, desaparece pero la información de contacto, como el correo por ejemplo, queda a la disposición de los usuarios.

Tabla 3: Descripción componente de Comunicación.

6.6.2. Interfaz Usuaría

Según lo diseñado se muestra a continuación la implementación, con los framework de React⁴ y React Native⁵, de las dos aplicaciones móviles de VIDAT, utilizando un iPhone 8 plus, y con la herramienta EXPO⁶ para desarrolladores de aplicaciones móviles.

⁴<https://es.reactjs.org/>

⁵<https://reactnative.dev/>

⁶<https://docs.expo.io/workflow/development-mode/>

A. Aplicación Demandante

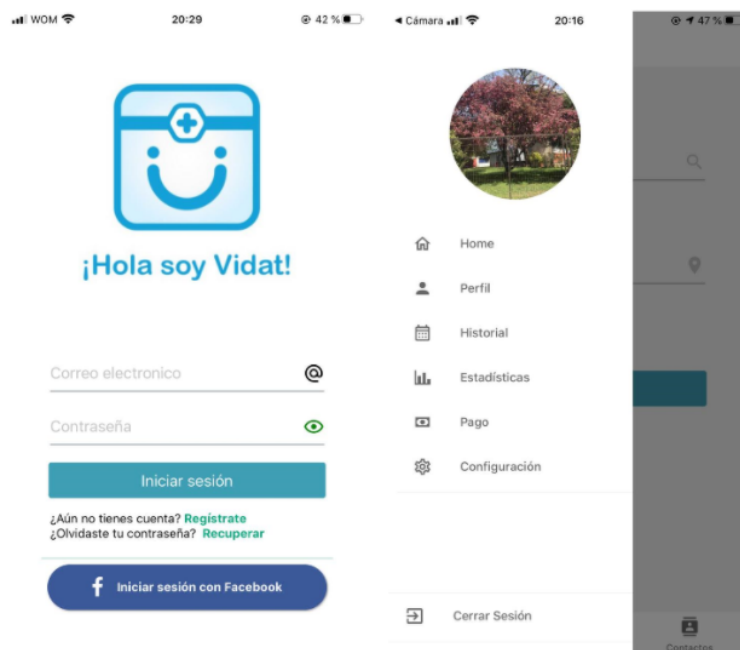


Figura 24: Vistas Inicio de sesión y Menú navegación.

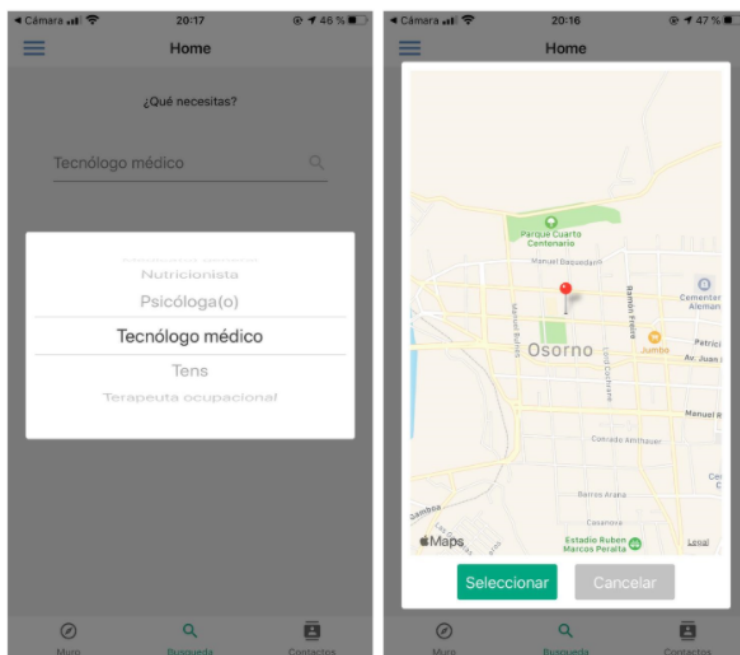


Figura 25: Vistas Búsqueda Profesional y Georreferenciación API Google.

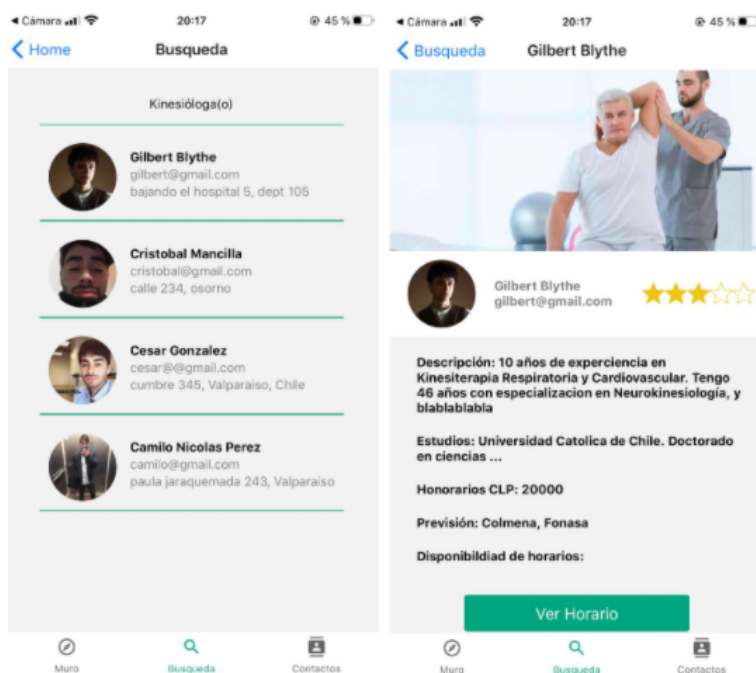


Figura 26: Vistas Lista de profesionales en cobertura e información del profesional.

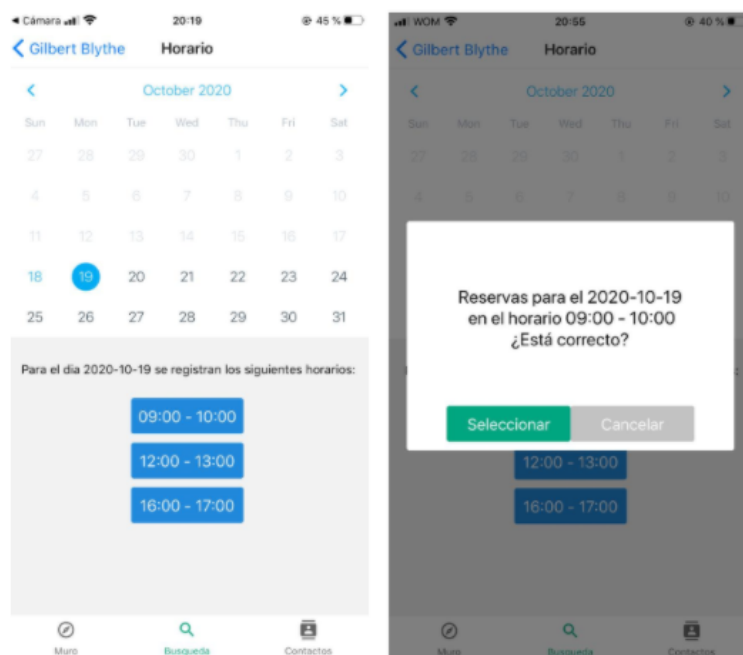


Figura 27: Vistas de calendario con horarios disponibles y ventana modal para enviar solicitud

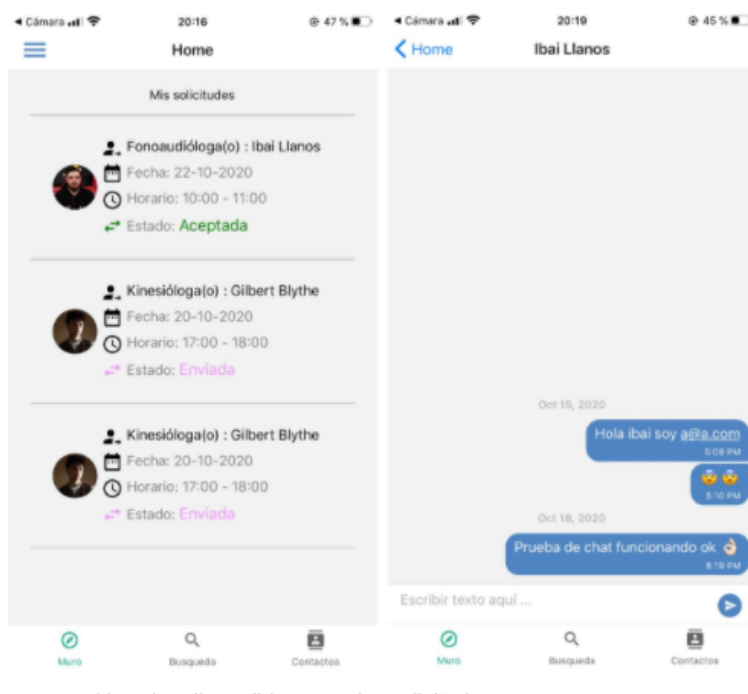


Figura 28: Vistas Muro con el estado de solicitudes, y el Chat temporal.

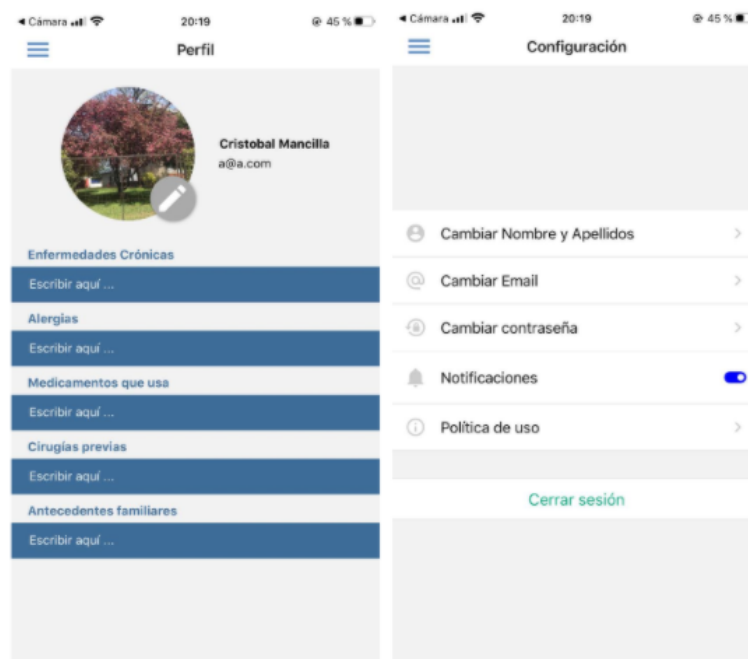


Figura 29: Vistas Perfil y Configuración.

B. Aplicación oferente

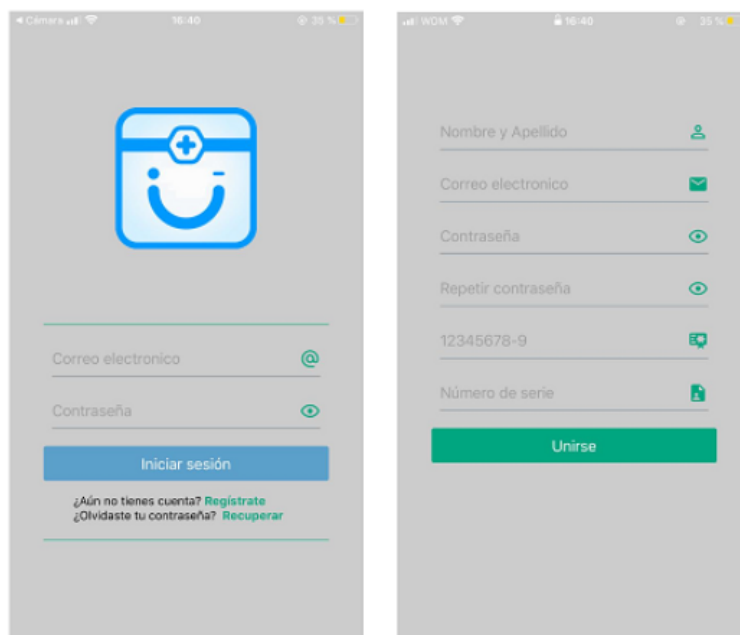


Figura 30: Vistas Inicio Sesión y Registro.

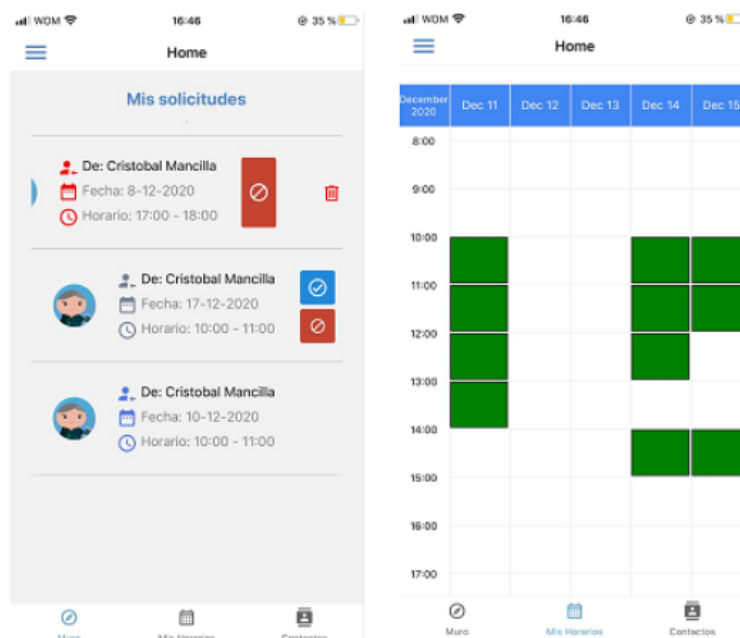


Figura 31: Vistas Muro y Mis horarios.

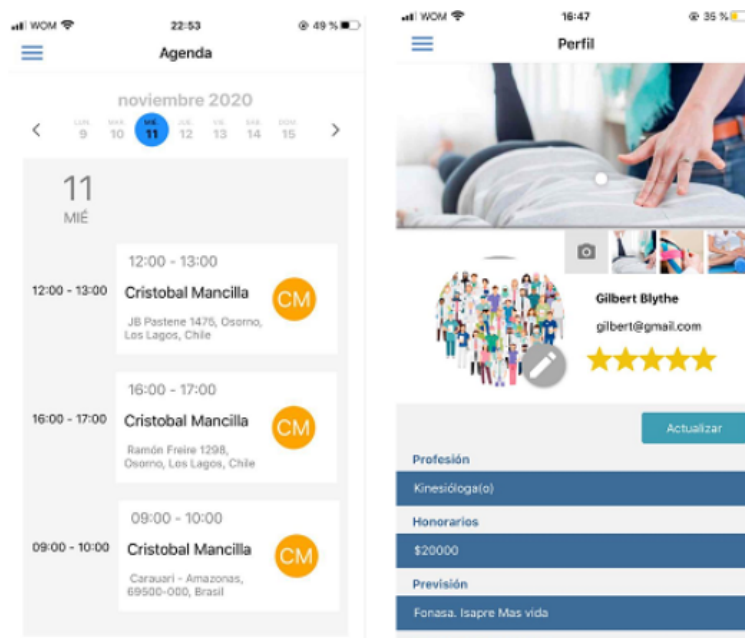


Figura 32: Vistas Agenda y Perfil.

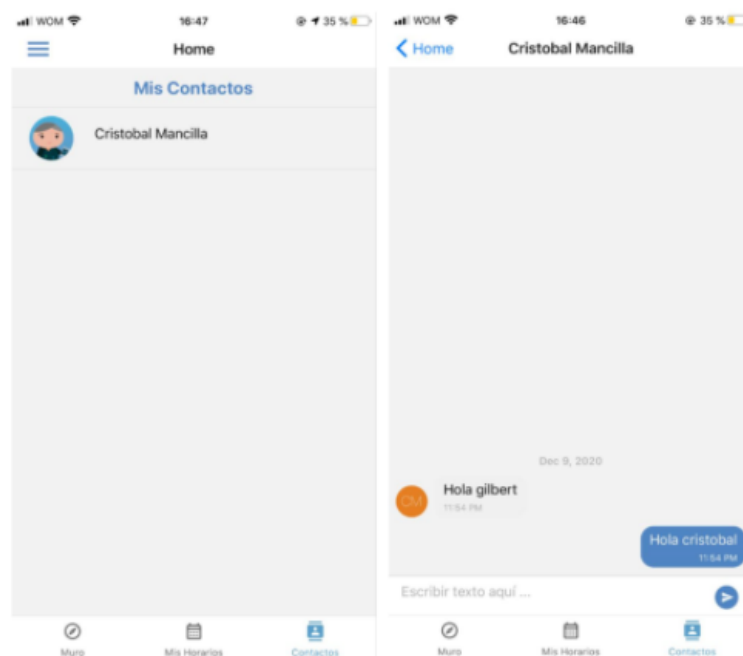


Figura 33: Stack con vistas de Contactos y Chat.

6.7. Conceptos y estrategias de funcionalidades

El núcleo de VIDAT es el agendamiento. La idea clave es, el sistema debe permitir que el demandante solicite los servicios del oferente. De allí se recogen las 3 clases potenciales que se modelaron en la base de datos: demandante, oferente y solicitud. La estrategia que se utilizó para implementar esta funcionalidad, fue diseñar un sistemas de estados. Esta solicitud tienen 6 estados posibles: Enviada, Aceptada, Rechazada, Confirmada, Cancelada y Terminada. Si el oferente tiene disponibilidad, significa que su horario no ha sido reservado, entonces el demandante tiene la posibilidad de reservar ese bloque y crear una solicitud con estado inicial “Enviado”. El profesional de la salud recibirá estos pool de solicitudes, y podrá aceptarlas o rechazarlas, cambiando el atributo en cuestión.

Ahora, para manejar la disponibilidad de horarios hay que considerar que todos los oferentes pueden tener más de un horarios disponible por día, significa una relación de “muchos es a muchos”; teniendo que crear una tabla intermedia que registre la fecha (04-03-2020) junto con el horario (08:00 - 09:00).

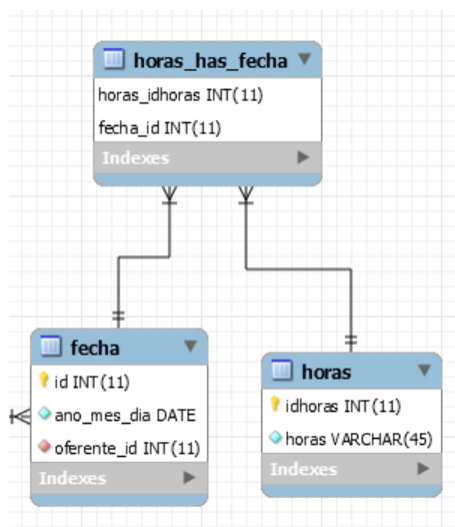


Figura 34: Relación disponibilidad de horarios oferente.

Cuando una solicitud es aceptada, el sistema realiza una petición al microservicio de la Agendación, actualizando el estado de la solicitud, reservando el horario y por ende quitando la disponibilidad. El oferente es el encargado de actualizar y mantener sus bloques de disponibilidad por semana, como se aprecia en el apartado anterior, ver figura 31 vista “Mis horarios”. De momento, el límite para reservar horarios futuros es hasta un año.

Existe la posibilidad de que una vez aceptada una solicitud, tanto el oferente como el demandante pueden cancelar dicha solicitud por algún motivo. Por consiguiente, lle-

gará un mensaje un día antes, donde el demandante tendrá que confirmar la reserva. Posterior a la realización del servicio, si este es finalizado con éxito, el estado de la solicitud cambia a “Finalizado”, y VIDAT continua con el microservicio de Valoración y Pago.

Este sistema de estados, permite no solo organizar el agendamiento, sino que también añade experiencia al usuario, al informándole a todos los participantes la situación en la que se encuentra la solicitud. Además, como proyección futura de VIDAT, puede dar paso a estadísticas y registros que pudieran interesarle a los oferentes, como por ejemplo, porcentaje de solicitudes canceladas, o cobertura donde existe mayor solicitudes aceptadas, etc.

Ahora respecto a la cobertura. Los posibles oferentes de servicio a los que el demandante puede solicitar, deben estar sujeto a ciertas restricciones. Por ejemplo, un oferente no puede aceptar una solicitud a la que no alcanzará a llegar; es decir, no le puede llegar una solicitud continua en horario, a otra ya aprobada, cuando la distancia de un punto al otro no es el suficiente. Esto podría verse afectado en ciudades grandes, como Santiago. La estrategia para abordar estos problemas de trayectoria son utilizar nuevamente los servicios de Google. La API Distance Matrix de Google Maps⁷, proporciona la distancia y el tiempo de viaje para una matriz de orígenes y destinos. La API devuelve información basada en la ruta recomendada entre los puntos inicial y final, calculada por la herramienta de Google Maps. Con esto, el Backend puede desarrollador logística y coordinar con el Frontend, para que al demandante solo le aparezcan posibles oferentes que si puedan, según el criterio que describe VIDAT, realizar sus servicios.

6.8. Integración Backend-Frontend

Respecto a la integración del Backend con el Frontend, los servicios se encuentran en la nube con su dirección IP y los endpoint necesarios para hacer peticiones a la Api. Se adjunta a continuación unas imágenes de ejemplo con requests HTTP⁸ en POSTMAN⁹, como validación.

⁷<https://developers.google.com/maps/documentation/distance-matrix/overview>

⁸https://www.w3schools.com/tags/ref_httpmethods.asp

⁹<https://www.postman.com/downloads/>

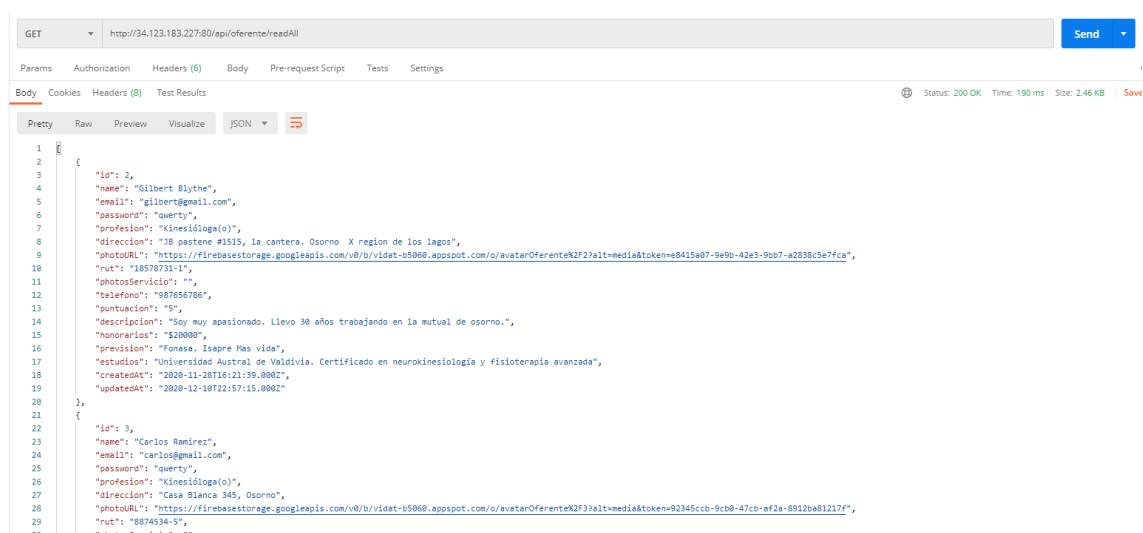


Figura 35: Petición Get para leer todos los oferentes.

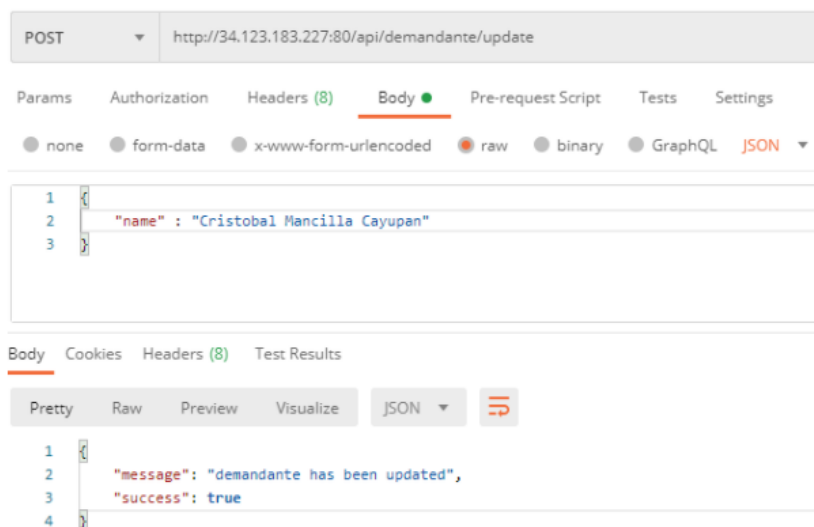
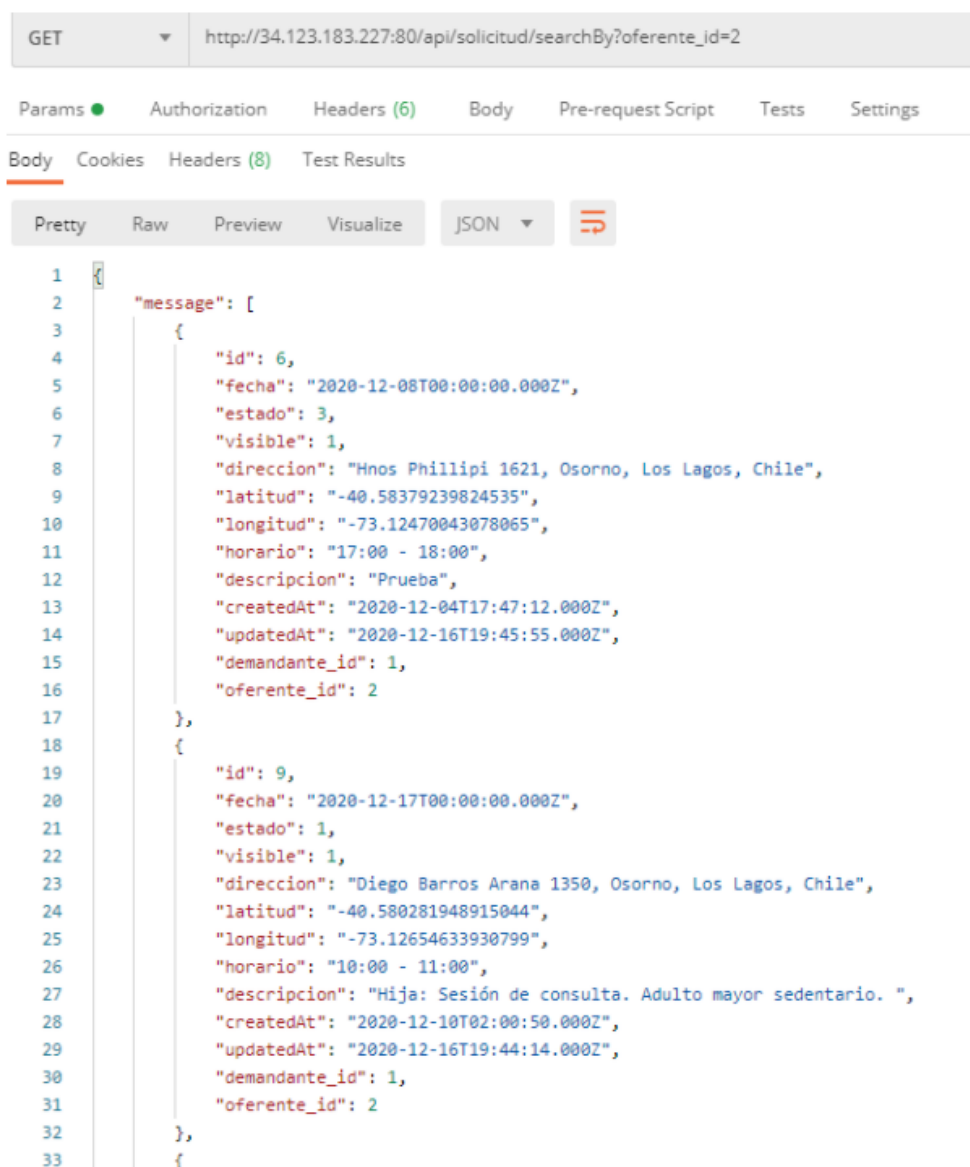


Figura 36: Petición POST actualización atributos demandante.



```

1  {
2    "message": [
3      {
4        "id": 6,
5        "fecha": "2020-12-08T00:00:00.000Z",
6        "estado": 3,
7        "visible": 1,
8        "direccion": "Hnos Phillipi 1621, Osorno, Los Lagos, Chile",
9        "latitud": "-40.58379239824535",
10       "longitud": "-73.12470043078065",
11       "horario": "17:00 - 18:00",
12       "descripcion": "Prueba",
13       "createdAt": "2020-12-04T17:47:12.000Z",
14       "updatedAt": "2020-12-16T19:45:55.000Z",
15       "demandante_id": 1,
16       "oferente_id": 2
17     },
18     {
19       "id": 9,
20       "fecha": "2020-12-17T00:00:00.000Z",
21       "estado": 1,
22       "visible": 1,
23       "direccion": "Diego Barros Arana 1350, Osorno, Los Lagos, Chile",
24       "latitud": "-40.580281948915044",
25       "longitud": "-73.12654633930799",
26       "horario": "10:00 - 11:00",
27       "descripcion": "Hija: Sesión de consulta. Adulto mayor sedentario. ",
28       "createdAt": "2020-12-10T02:00:50.000Z",
29       "updatedAt": "2020-12-16T19:44:14.000Z",
30       "demandante_id": 1,
31       "oferente_id": 2
32     },
33   ]

```

Figura 37: Petición Get para buscar todas las solicitudes realizadas por el “id_oferente”. Este endpoint está configurado para buscar por todos los parámetros que se desee.

Las peticiones siguen la sintaxis:

`http://ipserver:port/api/model/endpoint?<params>`

- “model” son las tablas definidas en el la base de datos.
- “endpoint” es la acción de la petición: readAll, readOne, SearchBy, update, delete, create.
- “params” son los atributos definidos en el diseño de la base de datos, y se anidan con “&”.

7. Evaluación de Riesgos

Basándose en la elicitación, investigación y estudio de mercado, existen ciertos riesgos con distinta relevancia, algunos más críticos que otros, ya sea en el ámbito técnico como en la seguridad de la persona.

Riesgo	Falla Técnica
Descripción	Es el escenario en que los servicios dejan de funcionar y los datos de la aplicación se pueden pierden.
Evaluación	• Posibilidad de ocurrencia: Medianamente posible. • Impacto sobre el sistema: Afecta operación. • Gobernabilidad: Control relativo.

Tabla 4: Riesgos técnicos.

Riesgo	Pérdida de valor
Descripción	El oferente de servicio y el adulto mayor se contactan y agenda fuera de la aplicación para realizar el servicio.
Evaluación	• Posibilidad de ocurrencia: Medianamente posible. • Impacto sobre el sistema: Afecta costos y usabilidad. • Gobernabilidad: Influencia relativa.

Tabla 5: Riesgos de valor.

Riesgo	Calidad de servicio
Descripción	El oferente de servicio realiza una mala atención, lo que afecta directamente a la valoración del cliente hacia la aplicación, y por consiguiente, poca recurrencia a la utilización de VIDAT.
Evaluación	• Posibilidad de ocurrencia: Posible. • Impacto sobre el sistema: Afecta costos y usabilidad. • Gobernabilidad: Influencia relativa.

Tabla 6: Riesgos de calidad.

Riesgo	Malversación de la aplicación para delinquir
Descripción	Existe el riesgo de que se solicite a un oferente a un domicilio y el fin sea la delincuencia. O por el otro lado, se suplante las credenciales de un oferente, para recurrir al hogar de un adulto mayor para fines delictuales.
Evaluación	• Posibilidad de ocurrencia: Baja. • Impacto sobre el sistema: Afecta seguridad clientes. • Gobernabilidad: Control Relativo.

Tabla 7: Riesgos de malversación.

Riesgo	Rechazo aplicación
Descripción	Situación en donde el adulto mayor no utiliza la aplicación porque para su perspectiva no le entrega ningún valor. O que exista poca participación de profesionales certificados, lo que implica pocas opciones o una baja disponibilidad de horarios.
Evaluación	• Posibilidad de ocurrencia: Posible. • Impacto sobre el sistema: Inviabile. • Gobernabilidad: Control relativo.

Tabla 8: Riesgos de rechazo.

Riesgo	Filtración de datos personales
Descripción	Los datos de los clientes son descubiertos de forma ilícita. Este tipo de riesgo puede generar inseguridad en los clientes y provocar que dejen la aplicación.
Evaluación	• Posibilidad de ocurrencia: Baja. • Impacto sobre el sistema: Baja rentabilidad. • Gobernabilidad: Control relativo.

Tabla 9: Riesgos de filtración.

8. Conclusiones

La integridad de los datos siempre es necesaria para el correcto funcionamiento del sistema. Las fallas de integridad se pueden deber a daños accidentales o maliciosos. En este contexto se necesita protección frente a aquellos casos intencionados, ya que para disminuir la probabilidad de un daño accidental es necesario mejorar hardware. Para disminuir la probabilidad de daño malicioso es necesario una correcta configuración de la máquina por parte del sysadmin (Base de datos, Puertos IP, llaves SSH, etc). Ahora, la credibilidad es otro aspecto importante para el funcionamiento de estas aplicaciones. Como se busca que solicitantes y oferentes de servicio logren acuerdo, es necesario que la información que se maneja sea fidedigna. Por consiguiente, es imperativo operar con confidencialidad, porque se manejan datos personales; y es por esto, que la aplicación considera mecanismos para autorizar y validar el acceso de información.

Los desafíos de VIDAT hacia el adulto mayor radican en la integración de éste al mundo digital. Se necesita explicar de manera didáctica el funcionamiento de los servicios que entrega VIDAT, y también cuáles son los riesgos de seguridad. La cercanía de una aparato inteligente hacia este segmento ya existe, e irá en aumento a medida que las generaciones avanzan. Los datos del Censo 2017 indican que un 15,7 % de la población tiene 60 años o más, cifra que va en aumento a una tasa de crecimiento anual estimada en un 0.5 %

Con VIDAT se disminuirán los tiempos de respuestas. El cliente podrá tener comunicación directa con el oferente sin necesidad de reservar horas disponibles en los centros médicos. Por consiguiente, permitirá aumentar la red de contacto de la salud al que profesionales y personas de la tercera edad tienen alcance. Ahora, respecto al profesional de la salud, en este contexto de la salud a domicilio, la ventaja competitiva más relevante que entrega VIDAT es, la gama de clientes, el agendamiento automatizado y sistema de disponibilidad, y la carta de presentación hacia el adulto mayor. A su vez, los servicios de VIDAT permiten que las solicitudes registradas estén ordenadas y sean viables; esto debido a la logística y al sistema de estados que mantiene la organización de las solicitudes.

Respecto a los riesgos a los que esta sujeto la aplicación y los usuarios, en vista de la investigación y del diseño se tienen distintas estrategias. Respecto al riesgo de pérdida de valor, la información de contacto, como teléfono o correo electrónico, será información segura y no visible, considerando un chat temporal mientras dure el vínculo y la solicitud este vigente. Para el riesgo que considera una mala realización del servicio, existirán casillas de reclamos con posibilidad de reembolso, y desvinculación del oferente de la plataforma, todo sujeto a evaluación. Respecto a la seguridad de los usuarios, la acreditación de los profesionales será contra la superintendencia de salud y con la serie del documento nacional, y así, validar y conocer a los clientes suscritos. VIDAT también tiene en cuenta un botón de pánico que contacte a un familiar,

en casos extremos que la situación amerite. Finalmente, frente a las fallas técnicas, se tendrán respaldos de las bases de datos, y balanceadores de carga que controlen y mitiguen las sobrecargas de solicitudes, apoyando a la usabilidad y experiencia del usuario.

Para este proyecto, hubo ciertos elementos que si bien fueron diseñados, no fueron posible su implementación por un tema de recursos y tiempo. Uno de ellos es el trabajo de la cobertura y la lógica de las solicitudes del oferente en una zona geográfica. Por otro lado, si se logró el sistema de valoración por parte de los usuarios (rating), sin embargo no se concluyó el sistema de galardones que cada cuenta de VIDAT tendría, a modo de entregar una mejor referencia al ver el perfil público del oferente. A su vez, tampoco se implementó el microservicio del Pago, por tema de tiempo hacia este MVP (producto mínimo viable). Como trabajos futuros, además de lo ya mencionado, se espera utilizar la data obtenida de los múltiples servicios realizados, con éxito o no, y otorgarle a los clientes, tablas estadísticas y/o números relevante que sean de interés.

La plataforma de VIDAT fue diseñada apuntando al segmento de oferentes de la salud. Sin embargo, puede ser adaptado hacia otros rubros que faciliten la vida del adulto mayor, de los que no se tienen contactos y se desconocen las ofertas. O bien, si se sigue por la línea de la salud, es perfectamente escalable a telemedicina. Algunas ideas podrían ser, implementar un subsistema de supervisión remota, por videoconferencia, para tratar pacientes con dolores muscular, molestias urinarias, problemas cardiovasculares, o realizar telepatología, etc. Otra alternativa, sería incorporar nuevas herramientas complementarias, como juegos que ayuden a la memoria o al estrés, asistentes de voz, o incluso control sobre otros aparatos tecnológicos apoyado con IoT.

9. Anexos

9.1. Historias de usuarios y requisitos funcionales del sistema

Requisito funcional	Historias de usuarios
El sistema debe ofrecer mecanismos para administrar datos de usuarios	Como usuario quiero registrarme como solicitante para solicitar un servicio
	Como usuario quiero iniciar sesión como solicitante de servicios
	Como usuario quiero registrarme como oferente para ofrecer un servicio.
	Como usuario quiero iniciar sesión como oferente de servicios.

Tabla 10: Requisito Funcional 1, Administración de usuarios.

Requisito funcional	Historias de usuarios
El sistema debe ofrecer mecanismos para administrar solicitudes de servicios	Como solicitante quiero publicar una solicitud de servicio
	Como solicitante quiero buscar servicios ofrecidos
	Como solicitante quiero editar la publicación de un servicio solicitado
	Como solicitante quiero borrar una publicación de un servicio solicitado
	Como oferente quiero publicar un servicio ofrecido
	Como oferente quiero buscar servicios necesarios
	Como oferente quiero editar un servicio ofrecido
	Como oferente quiero borrar un servicio ofrecido

Tabla 11: Requisito Funcional 2, Administración de solicitudes de servicio.

9.2. Casos de uso

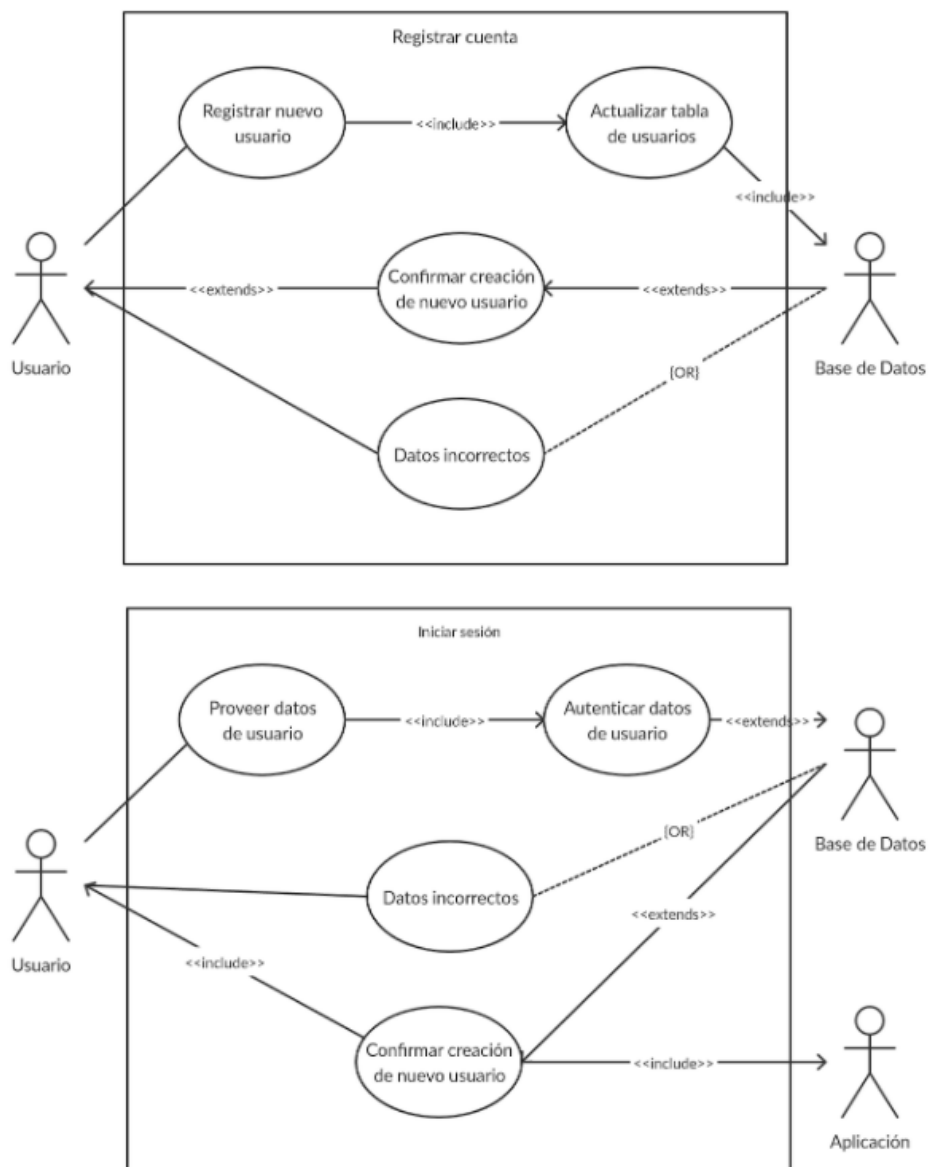


Figura 38: Autenticación y Acreditación de cuenta. Fuente: Autoría memoristas.

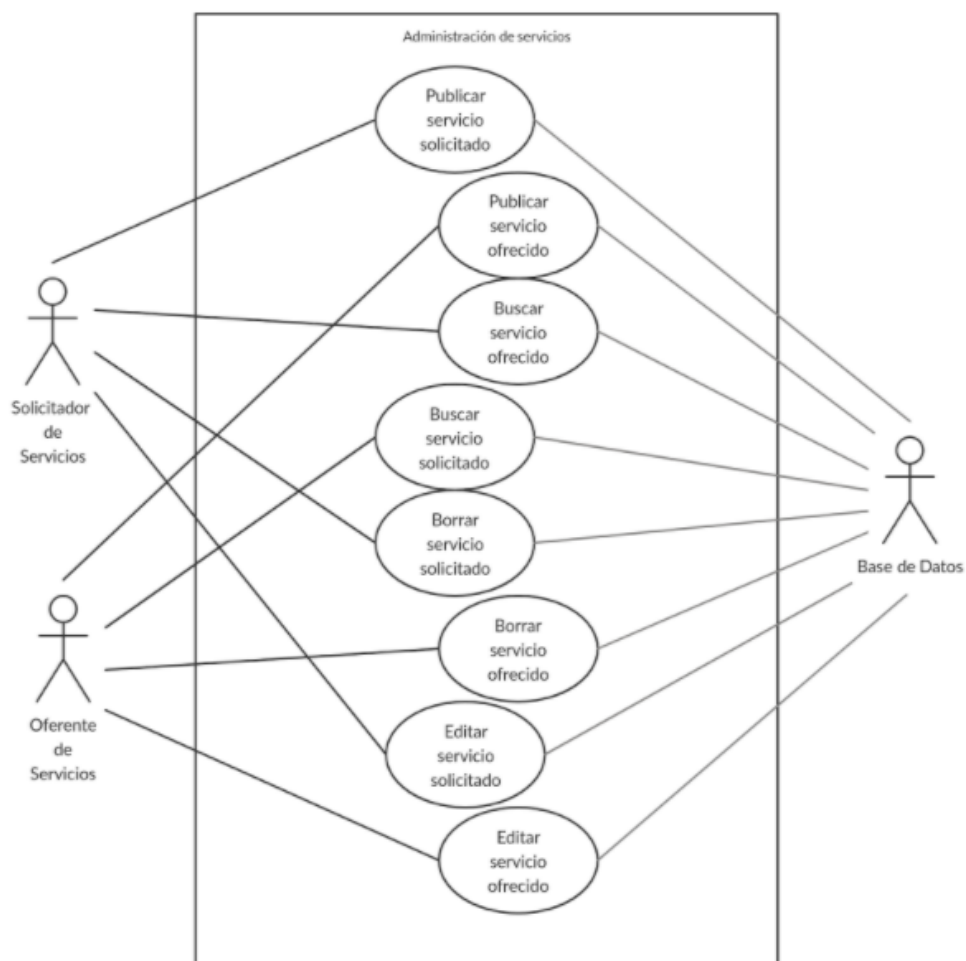


Figura 39: Administración de servicios ofrecidos/solicitados. Fuente: Autoría memoris-tas.

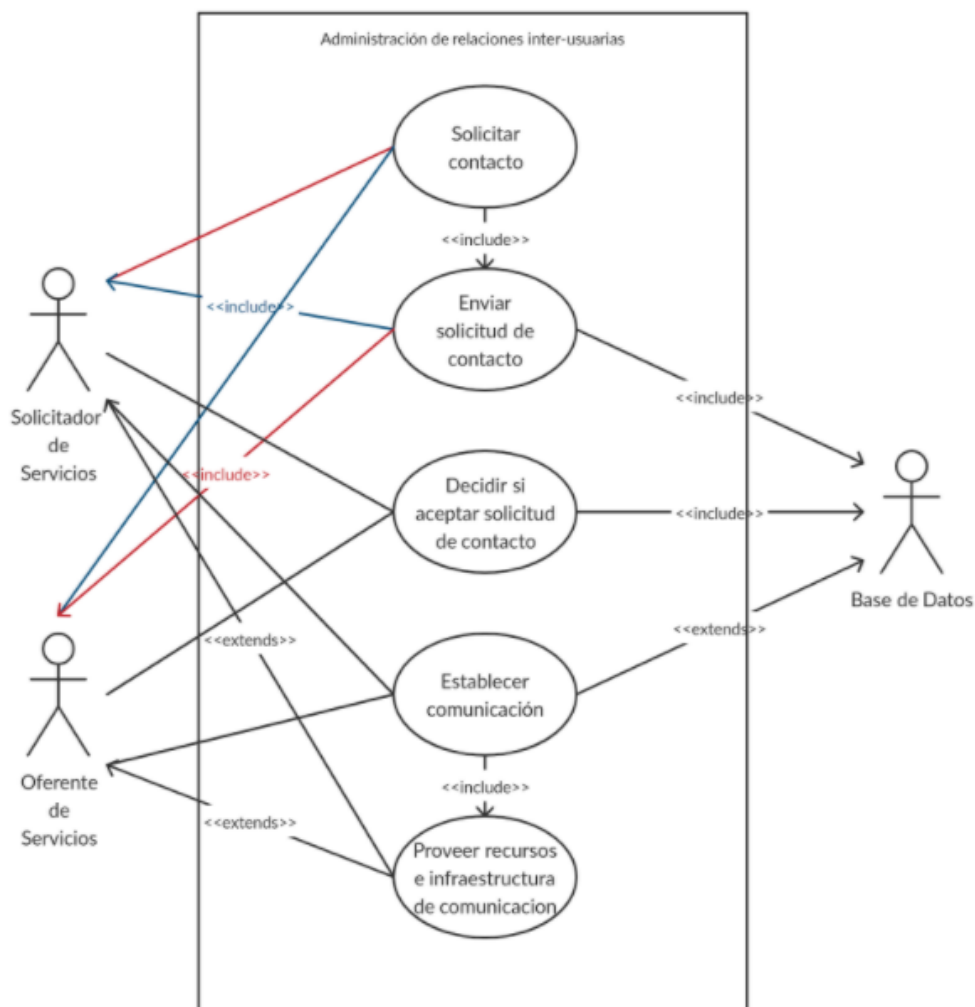


Figura 40: Administración de relaciones inter-usuarias. Fuente: Autoría memoristas.

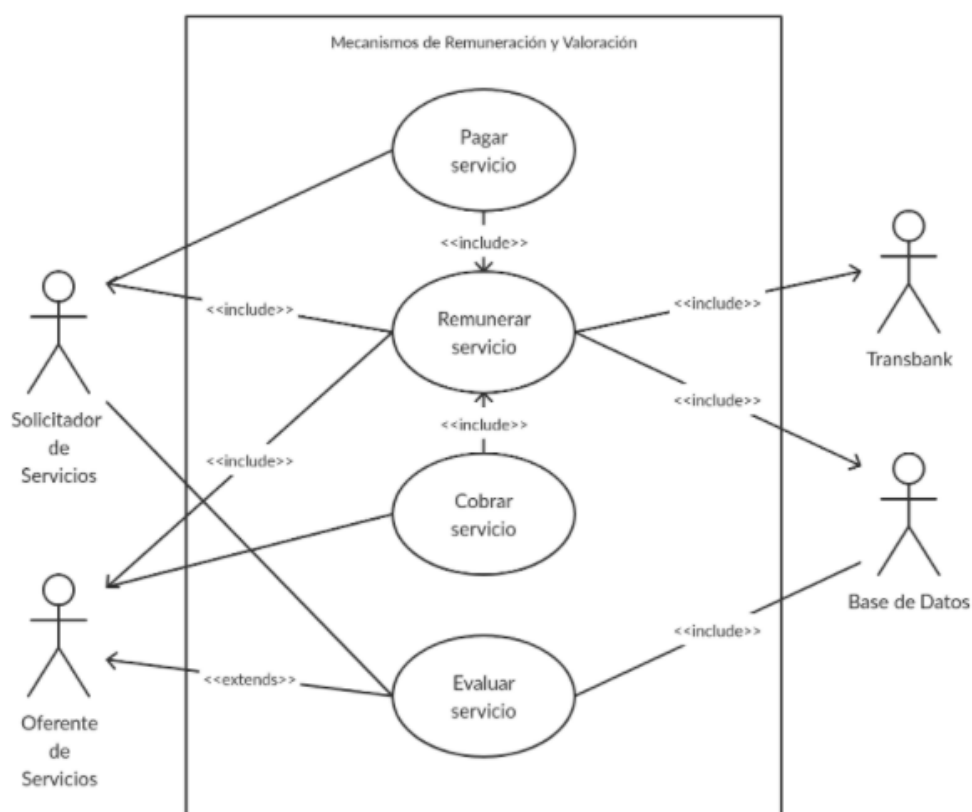


Figura 41: Mecanismos de Remuneración y Valoración. Fuente: Autoría memoristas.

9.3. Softgoal

El siguiente diagrama Softgoal desagrega los atributos de calidad: Performance, Usabilidad y Seguridad, en objetos o acciones para ilustrar las relaciones entre éstas.

- **Performance:** No existen requerimientos específicos en cuanto a performance, pero se busca un tiempo de respuesta adecuado para uso humano.
- **Usabilidad:** Este atributo de calidad es muy importante, ya que la aplicación está enfocada para ser usadas por personas de tercera edad. La usabilidad mide cuán fácil es para un usuario aprender a utilizar el sistema, o, interpretar entradas y salidas de la aplicación. De esta manera es fundamental que las interfaces usuarias (UI's) sean diseñadas para esperar un uso intuitivo de éstas, para conseguirlo es fundamental hacer uso de patrón de diseño de frontend adecuado. Por otro lado, si bien es preferible evitar que el usuario incurra en leer documentación para utilizar el sistema, elaborar este documento puede ayudar a desarrollar un mecanismo de soporte para el usuario. Se debe tener en consideración que esta decisión de diseño impactará negativamente en el tiempo de respuesta, ya que se sobrecarga el sistema.

- **Seguridad:** Este atributo de calidad es muy importante. La seguridad asegura que no se permiten accesos no autorizados al sistema y los datos, y, que se mantiene la integridad de la información.

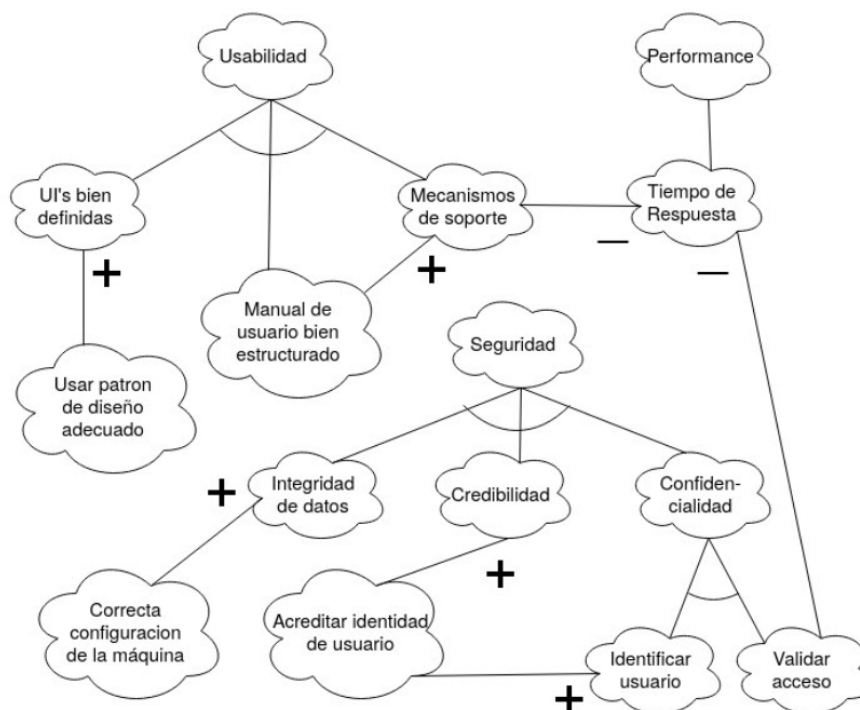


Figura 42: Softgoal. Fuente: Autoría memoristas.

9.4. Diagrama de secuencia registro e inicio sesión

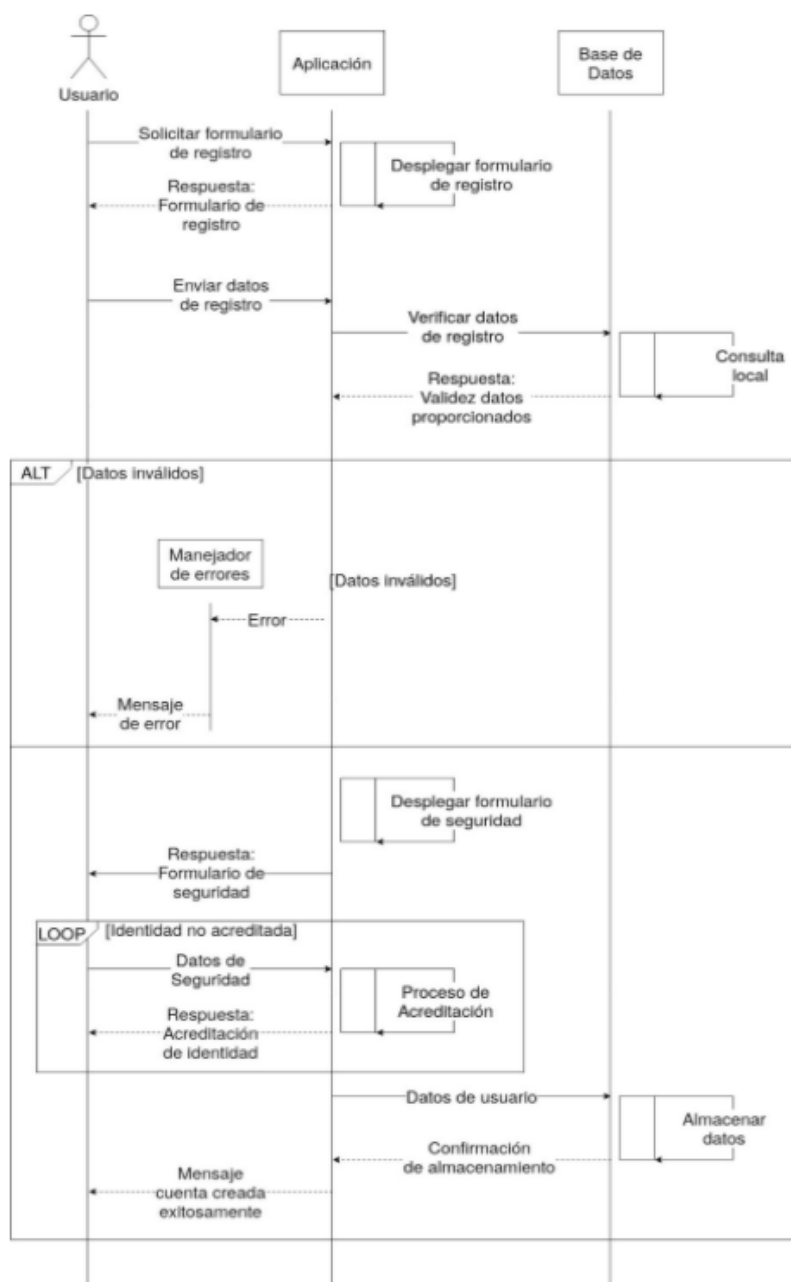


Figura 43: Registro. Fuente: Autoría memoristas.

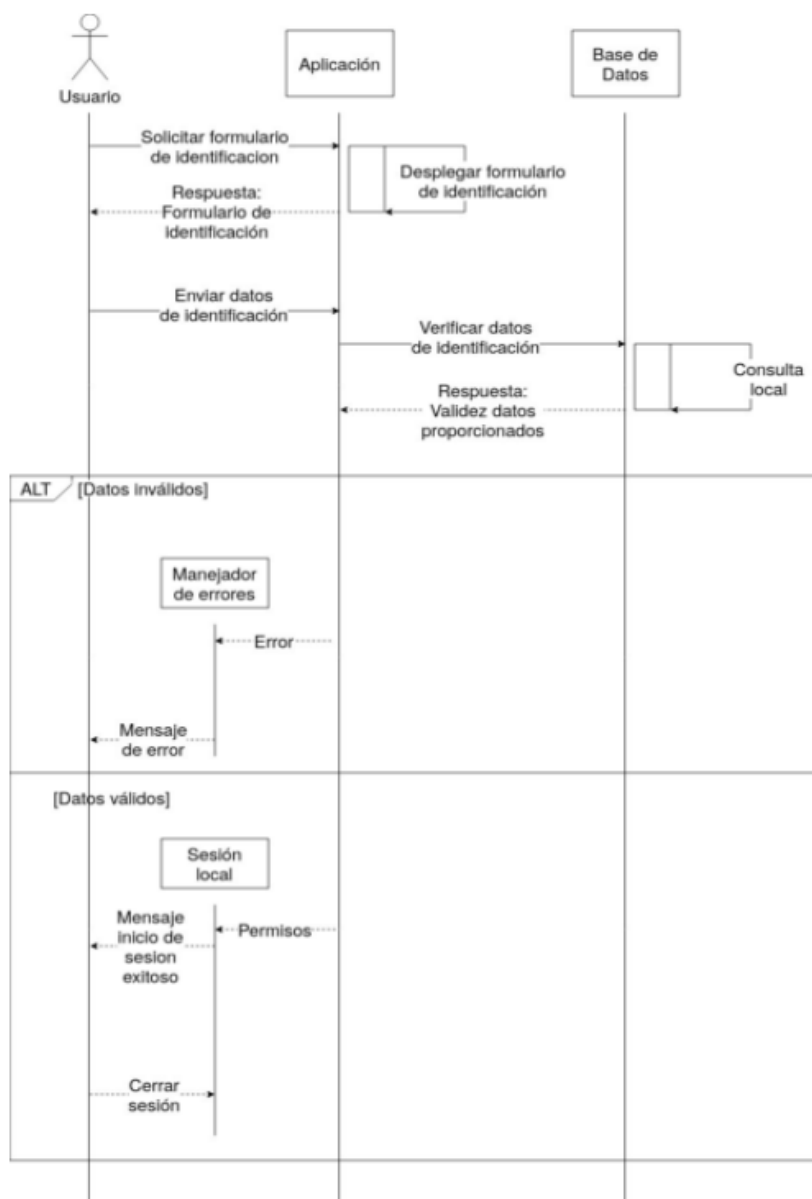


Figura 44: Inicio Sesión. Fuente: Autoría memoristas.

9.5. Diagrama de secuencia agendamiento

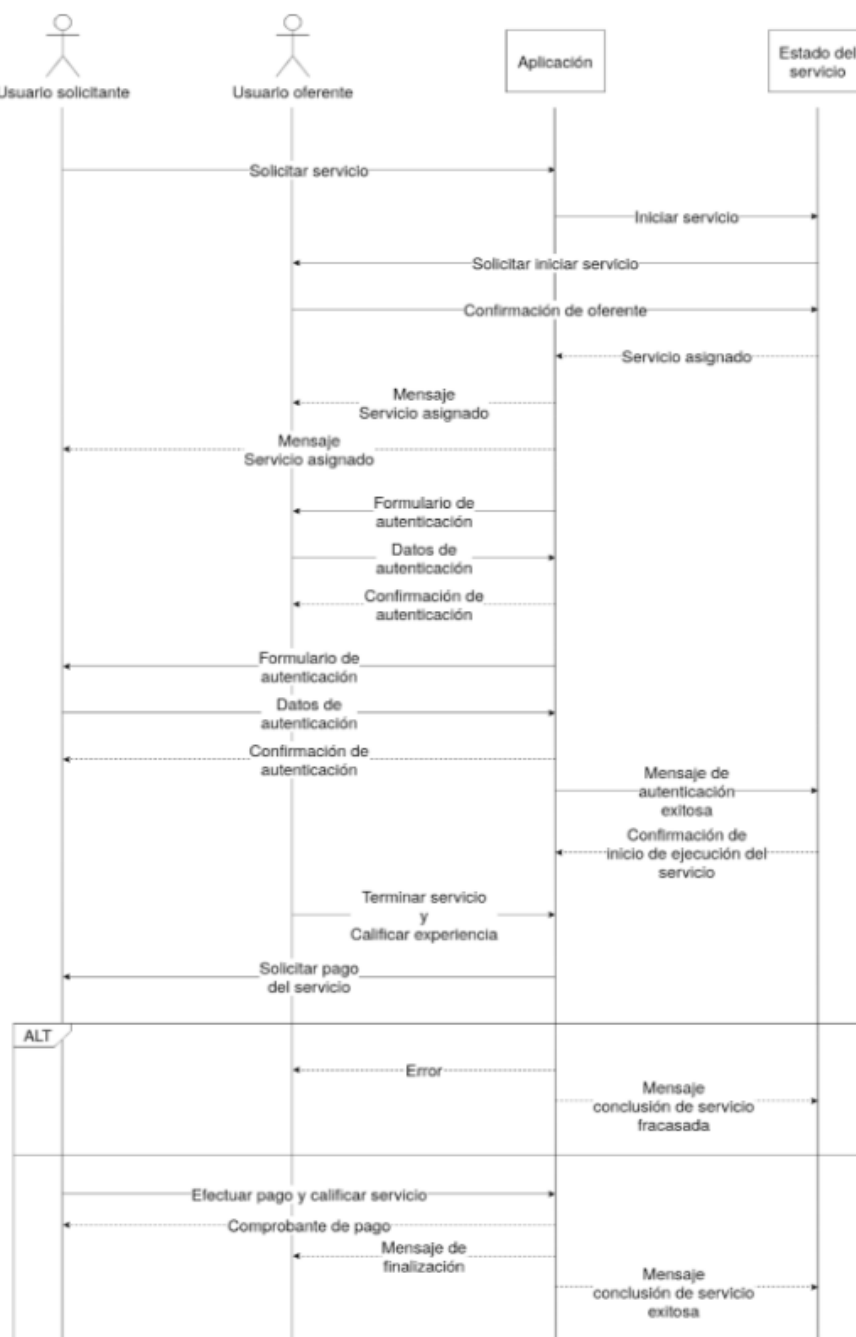


Figura 45: Agendamiento. Fuente: Autoría memoristas.

9.6. Diagrama de secuencia comunicación

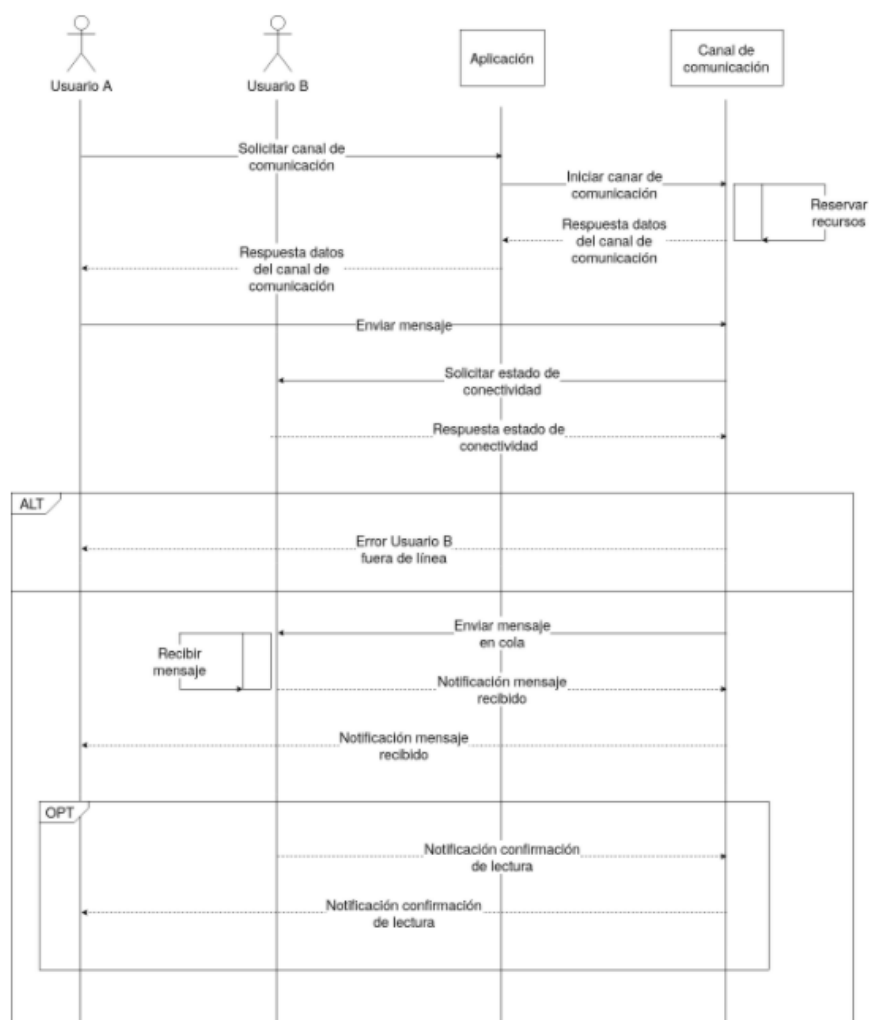


Figura 46: Comunicación. Fuente: Autoría memoristas.

9.7. Configuración API Google Geocoding

Para utilizar la Api de Google que permite obtener la dirección física a partir de coordenadas geográficas, primero debe crearse un proyecto en la plataforma de Google Cloud[15](GCP). Una vez creado el proyecto, llamado en este ejemplo “Geolocalización”, al que google facturará por cada request, se habilita la API Geocoding. Consultar la documentación oficial del web service[16]. Se visualiza el procedimiento en las siguientes imágenes

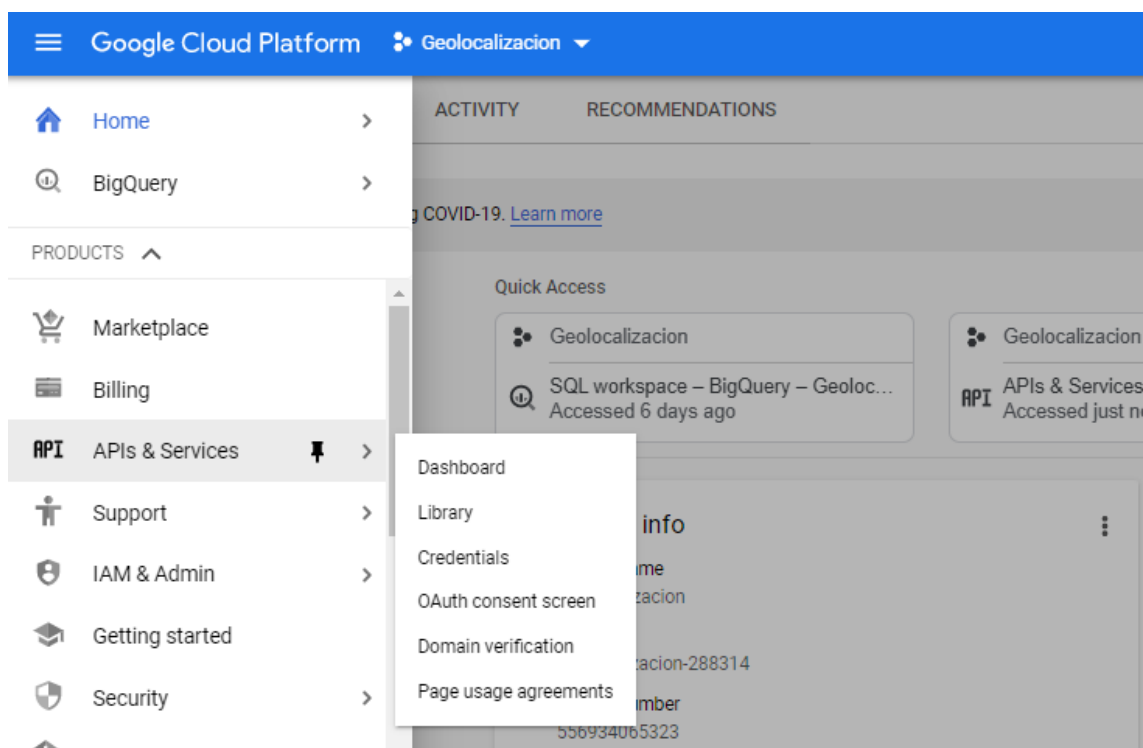


Figura 47: Google Cloud Plataform.

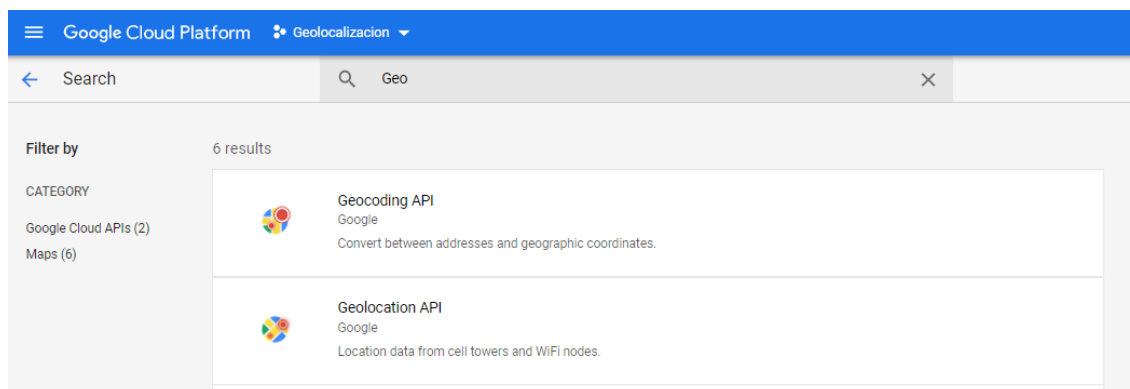


Figura 48: Google Cloud Plataform. API Geocoding.

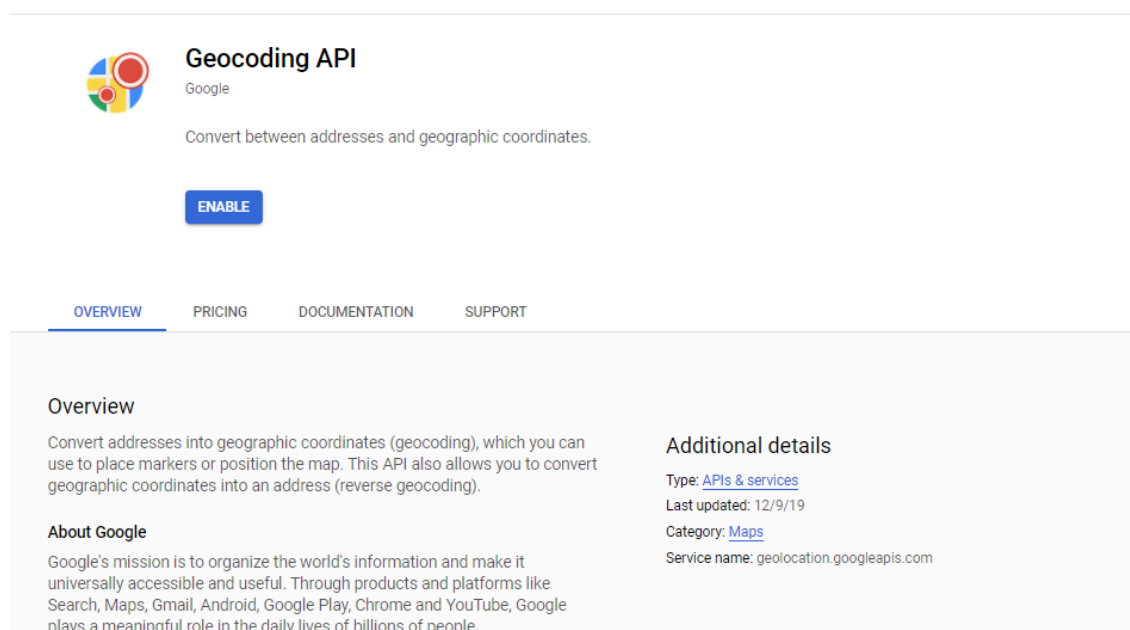


Figura 49: Google Cloud Plataform. API Geocoding.

En el mismo ítem de APIs & Services debe ir al apartado de Credentials y crear una nueva API key; puede configurar y restringir la clave si lo desea. Esta será necesaria a la hora de programar la petición a los servicios de Google para mapear las coordenadas geográficas. El endpoint se describe a continuación, en una función fetch de Javascript.

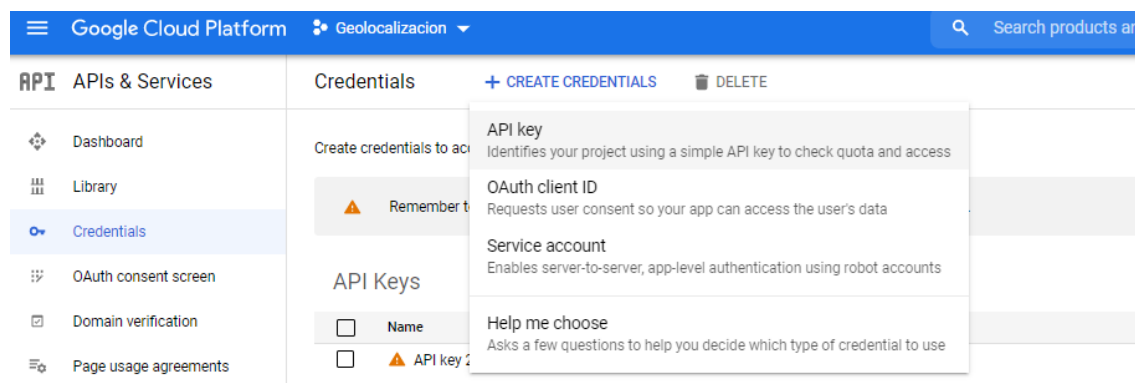


Figura 50: Google Cloud Plataform. Credentials.

API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key
AIzaSyAojriGKZbF4oUyRd11cFZEmkkWQr7u8Fw

⚠ Restrict your key to prevent unauthorized use in production.

[CLOSE](#) [RESTRICT KEY](#)

Figura 51: Google Cloud Plataform. API KEY.

```
const confirmLocation = () =>{
  toastRef.current.show("Localizacion guardada correctamente");
  setLatitud(location.latitude);
  setLongitud(location.longitude);
  setIsVisibleMap(false); //cerramos el mapa
  fetch("https://maps.googleapis.com/maps/api/geocode/json?latlng="+location.latitude+","+location.longitude+"&key=AIzaSyDmj3i5g5x7yx6P5o9TmuLe8vHow_INKD0")
    .then((response) => response.json())
    .then((json) => {
      setLocationDemandante(json.results[1].formatted_address);
    })
}
```

Figura 52: Función Fetch de JS al endpoint de Google.

La repuesta es un JSON con los componentes de la dirección, con distintos formatos, dependiendo de la exactitud del paralelo y longitud descrito. Algo de la forma

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "1600",
          "short_name" : "1600",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Amphitheatre Parkway",
          "short_name" : "Amphitheatre Pkwy",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Mountain View",
          "short_name" : "Mountain View",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "Santa Clara County",
          "short_name" : "Santa Clara County",
          "types" : [ "administrative_area_level_2", "political" ]
        },
        {
          "long_name" : "California",
          "short_name" : "CA",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "United States",
          "short_name" : "US",
          "types" : [ "country", "political" ]
        },
        {
          "long_name" : "94043",
          "short_name" : "94043",
          "types" : [ "postal_code" ]
        }
      ],
      "formatted_address" : "1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA",
    }
  ]
}
```

Figura 53: Response Api Google Geocoding.

Referencias

- [1] Habitissimo. <https://www.habitissimo.cl/>
- [2] Misión Habitissimo <https://www.habitissimo.cl/acercade>
- [3] Listoco. <https://www.listoco.cl/>
- [4] Misión Listoco. <https://www.listoco.cl/info#:~:text=Nuestra%20Misi%C3%B3n,aportando%20al%20desarrollo%20del%20pa%C3%ADs.>
- [5] Doctoralia. <https://www.doctoralia.es/>
- [6] Sobre Doctoralia. <https://www.doctoralia.es/quienes-somos>
- [7] Ministerio Desarrollo Social, Observatorio Social. http://www.observatorio.ministeriodesarrollosocial.gob.cl/casen-multidimensional/casen/docs/Resultados_Adulto_Mayores_casen_2017.pdf
- [8] Instituto Nacional de Estadísticas. <https://www.ine.cl/estadisticas/sociales/demografia-y-vitales/proyecciones-de-poblacion>
- [9] Comisión Nacional de Investigación Científica y Tecnológica. <https://www.conicyt.cl/sobre-conicyt/que-es-conicyt/>
- [10] Investigación y Desarrollo en Acción. <https://www.conicyt.cl/fondef/lineas-de-programa/instrumentos-vigentes/programa-idea/>
- [11] Artículo Conicyt. <https://scielo.conicyt.cl/pdf/cinfo/n38/art06.pdf>
- [12] Fondo de Fomento al Desarrollo Científico y Tecnológico. <https://www.conicyt.cl/fondef/sobre-fondef/que-es-fondef/>
- [13] Adobe Xd. Entorno trabajo maqueta demandante <https://xd.adobe.com/view/2f3662b6-e598-4d77-a4f3-36b28c80b351-7a65/>
- [14] Adobe Xd. Entorno trabajo maqueta oferente <https://xd.adobe.com/view/a2e9a091-6a78-4fc3-86bc-dd15511b2c3d-f7ba/>
- [15] Google Cloud Plataform. <https://console.cloud.google.com/>
- [16] API Geocoding. <https://developers.google.com/maps/documentation/geocoding/overview?hl=es-419>