



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**ALINEANDO CANALES DE INTERACCIÓN SOCIAL PARA SOPORTAR
COMUNICACIÓN INTER-GENERACIONAL**

**TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN**

FELIPE ALONSO RODRÍGUEZ ARANDA

PROFESOR GUÍA:

SERGIO OCHOA DE LORENZI

MIEMBROS DE LA COMISIÓN:

JEREMY BARBAY

FELIPE BRAVO MARQUEZ

PEDRO ROSSEL CID

SANTIAGO DE CHILE

2019

Resumen

La gran cantidad de medios de comunicación existentes, junto a la velocidad con que estos se vuelven populares o dejan de serlo, genera una nube de canales sociales altamente cambiante. Por lo tanto, cualquier persona o sistema que no sea capaz de adaptarse fácilmente a estos cambios, limitará cada vez más su capacidad para comunicarse digitalmente con otros, hasta quedar aislado. Un ejemplo de esto son los adultos mayores, que se enfrentan regularmente a este desafío al momento de intentar interactuar con generaciones más jóvenes de su familia usando tecnología.

Para intentar reducir el impacto de este problema, particularmente sobre la población de adultos mayores, se han desarrollado diversas aplicaciones ad hoc de software, como por ejemplo, Tlatoque, BuildingBridges y SocialConnector. Aunque dichas aplicaciones han mostrado ser útiles, éstas se encuentran vinculadas directamente a algún medio de comunicación digital específico (es decir, un canal/herramienta/servicio de interacción social). Por ende, no consideran la asimetría entre los canales de comunicación preferidos por los adultos mayores y aquellos usados por los miembros de generaciones más jóvenes de la comunidad familiar.

Para dar solución a este problema de asimetría comunicacional, en este trabajo de tesis se diseñó e implementó un sistema intermediario, llamado Traductor de Mensajes Sociales (SMT en inglés), que traduce los mensajes automáticamente en ambos sentidos del canal, sin que los participantes se den realmente cuenta de que están utilizando herramientas distintas para interactuar entre ellos. En otras palabras, permite enviar mensajes a distintas redes sociales, y recibirlos desde éstas, utilizando un formato común y genérico, permitiendo a los interactuantes utilizar sus redes sociales de preferencia.

De esa forma, las personas que utilizan aplicaciones ad hoc (por ejemplo, los adultos mayores), pueden seguir usando sus aplicaciones, sin importar la evolución de los sistemas de interacción social comerciales con los que ellos están interactuando. Debido a la participación del traductor de mensajes, tampoco importa demasiado la preferencia de canal o herramienta de comunicación que tengan los miembros de la comunidad familiar del adulto mayor; es decir, sus hijos y nietos principalmente.

Mediante la traducción de mensajes entre aplicaciones sociales, SMT aborda las asimetrías mencionadas, y así reduce la posibilidad de que los adultos mayores que lograron adoptar tecnología, se vuelvan usuarios huérfanos de las redes sociales en el corto tiempo. La correctitud y performance del sistema de traducción de mensajes fue evaluado con casos de laboratorio, y los resultados obtenidos fueron altamente positivos.

Además, se evaluó la seguridad del traductor y la capacidad de ser utilizado y extendido por parte de desarrolladores de software. Los resultados en estos dos últimos ámbitos fueron también altamente positivos. Esto llama a utilizar los servicios de SMT en múltiples sistemas de apoyo a interacciones sociales, dado que permiten mejorar la interoperabilidad con otras aplicaciones, extendiendo el tiempo de vida de una aplicación ad hoc. Consecuentemente, esto reduce el impacto de las interacciones asimétricas entre adultos mayores y los nativos digitales.

Agradecimientos

Para partir creo que es importante destacar la gran ayuda entregada por parte de mi profesor guía, Sergio Ochoa. El cual me ha acompañado y guiado en mis labores académicas desde mi trabajo de titulación para la carrera de ingeniería hasta hoy, donde además considero que sin su guía no habría logrado realizar el presente trabajo con la misma calidad y de la misma forma que con la que se presenta actualmente. Muchas gracias por su tiempo, dirección, dedicación y apoyo.

Además, debo destacar el apoyo del profesor Francisco Gutiérrez, el cual a pesar de no haber sido mi guía propiamente tal, significó un gran apoyo durante el desarrollo tanto de este trabajo como de las publicaciones asociadas, siempre buscando lograr la mejor calidad posible en el trabajo realizado.

También agradezco enormemente el apoyo de mi familia, ya que sin ella nada de esto habría sido posible. Gracias por darme tanto, a pesar de yo darles tan poco de vuelta. Gracias por soportarme durante todo este tiempo. Gracias por estar ahí siempre que lo he necesitado. Gracias por todo.

Por otra parte, cabe destacar la presencia de mis compañeros del DCC, los cuales fueron un gran apoyo mi vuelta al departamento y a estudiar nuevamente en la facultad. En muchas ocasiones ayudaron a hacer más ameno y grato todo este período, recordando que la facultad puede ser mucho más que un lugar para aprender ciencias e ingeniería. Gracias por todas las risas y alegría que me causaron.

A mis amigos y amigas de la vida, gracias por estar ahí, por acompañarme a salir, a conversar, a tocar música, a conocer lugares y hacer actividades. Gracias por distraerme y hacerme ocupar mi tiempo y mi mente en otras cosas enriquecedoras y que alegran la vida. Siempre destacaré su presencia y a pesar de que quizás nunca entenderé muy bien porque están ahí (no creo ser una persona tan genial), gracias por acompañarme y no duden de mi ayuda cuando sea que la necesiten.

Finalmente, gracias a todos los que estuvieron durante este proceso, sin importar si lo hicieron en mayor o en menor medida. Si compartimos algún momento en donde conversamos o discutimos algo interesante, o simplemente nos reímos juntos, créeme que te lo agradezco.

Este trabajo de tesis ha sido parcialmente financiado por el Proyecto Fondecyt Enlace ENL 02/18, VID, Universidad de Chile, y por el Fondo Nacional de Desarrollo Científico y Tecnológico (Fondecyt), proyecto Nro: 1191516, Chile.

Publicaciones

Los artículos científicos surgidos de este trabajo de tesis son los siguientes:

- Rodríguez, F., Ochoa, S.F., Gutierrez, F.J. “Seamlessly Mediation of Social Interaction Services Respecting Communication Preferences”. *Proc. 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI’18)*, Punta Cana, Dominican Republic. MDPI Proceedings 2(19), article 1249, Dec. 2018.
- Rodríguez, F., Ochoa, S.F., Gutierrez, F.J. “Supporting Asymmetric Interaction in the Age of Social Media”. Under review in the *Journal of Ambient Intelligence and Humanized Computing* – Springer, 2019.

Tabla de contenido

1	Introducción.....	1
1.1	Desafío Computacional Abordado.....	3
1.2	Objetivos de la Tesis.....	4
1.3	Metodología.....	6
1.4	Estructura del documento de tesis.....	7
2	Marco Teórico.....	8
2.1	Desafíos de Socialización para los Adultos Mayores.....	8
2.2	Herramientas de Apoyo a la Interacción Social con Adultos Mayores.....	11
2.3	Alternativas Tecnológicas para Alinear Canales de Comunicación.....	11
2.3.1	Protocolo XMPP (Extensible Messaging and Presence Protocol).....	12
2.3.2	Restricciones al funcionamiento de la solución.....	13
2.4	Aspectos involucrados en el problema a abordar.....	14
3	Determinación de la Necesidad de Alinear Canales.....	16
3.1	Descripción de los participantes.....	16
3.2	Proceso de recopilación de información.....	16
3.3	Resultados obtenidos.....	17
3.4	Implicancias de los resultados.....	18
4	Diseño de la solución.....	19
4.1	Comunicación con el Traductor.....	19
4.1.1	Protocolo de comunicación.....	20
4.1.2	Endpoints de autenticación.....	21
4.1.3	Endpoints de comunidad.....	22
4.1.4	Endpoints de traducción directa.....	23
4.2	Traducción de mensajes.....	23
4.3	Preferencias de redes sociales de los usuarios.....	28
5	Implementación de la Solución.....	32
5.1	Descripción de la Aplicación Cliente: SocialConnector.....	32
5.1.1	Autenticación usando la API desarrollada.....	32
5.1.2	Configuración de las Redes Sociales.....	32
5.1.3	Configuración de la comunidad del usuario.....	33
5.1.4	Verificación, obtención y envío de mensajes.....	33
5.2	Sistema traductor de mensajes.....	34
5.2.1	Canales de comunicación.....	34
5.2.2	Administración de preferencia de redes sociales.....	38

5.2.3	Interacción con sistemas externos	39
6	Evaluación de traductor de mensajes	40
6.1	Evaluación de la correctitud de la solución	40
6.1.1	Diseño experimental	40
6.1.2	Resultados Experimentales	42
6.2	Evaluación de la seguridad de la solución	44
6.2.1	Escenario 1: Amenazas en el servidor de aplicación	44
6.2.2	Escenario 2: Amenazas a la comunicación cliente-servidor	45
6.2.3	Escenario 3: Amenazas en la aplicación cliente	45
6.3	Evaluación de la usabilidad de la API	47
6.3.1	Diseño Experimental	47
6.3.2	Resultados Obtenidos	48
6.4	Discusión	50
7	Conclusiones y Trabajo Futuro	52
8	Bibliografía	54
Anexos	60
Anexo A:	Cuestionario para análisis de necesidades	60
Anexo B:	Sistema de preferencias	62
Anexo C:	Resultado de las pruebas de traducción de mensajes	64
Anexo D:	Pauta de entrevista a desarrolladores	70
Anexo E:	Formulario de consentimiento informado	71

Índice de Figuras

Figura 1: Evolución de los sistemas de social media	1
Figura 2: Dicotomía entre los escenarios de interacción social preferidos por las generaciones más jóvenes y más adultas de una comunidad familiar	2
Figura 3: Operación esperada del sistema traductor.....	5
Figura 4: Operación del sistema traductor con los distintos actores	5
Figura 5: Esquema de interacción social basada en un sistema mediador, el cual usa algún medio o canal dentro de la nube	9
Figura 6: Endpoints de comunicación con el traductor	19
Figura 7: UML de fábrica de traductores	24
Figura 8: Modelo de envío y recepción de mensajes	25
Figura 9: Modelo de relación entre usuarios y redes sociales	25
Figura 10: Modelo de interacción de usuarios.....	27
Figura 11: Modelo de preferencia de comunicación a través de redes sociales	29
Figura 12: Escenario de comunicación con el sistema de preferencias de redes sociales (o canales)	29
Figura 13: Ejemplo de método de Holt-Winters con tendencia y estacionalidad aditiva y multiplicativa.....	31
Figura 14: Estructura del escenario simulado para pruebas de correctitud.	41

Índice de Tablas

Tabla 1: Resumen de resultados de encuesta de necesidad de alinear canales.....	17
Tabla 2: Duración de token para las redes sociales soportadas.....	28
Tabla 3: Orden de preferencia de herramientas de interacción social.....	41
Tabla 4: Resumen de la prueba de traducción de mensajes realizada	43
Tabla 5: Resultados de consultas de usabilidad de la API	49

1 Introducción

Según la teoría matemática de la comunicación [1], un emisor envía un mensaje a un receptor a través de un canal; si alguno de estos elementos falla, la comunicación no puede llevarse a cabo. Es por esto que, si dos personas no comparten el mismo canal de comunicación, la interacción entre ellas no es factible, al menos en forma directa.

Debido a los avances de la tecnología, la forma en que las personas se comunican ha ido cambiando y diversificándose con el paso del tiempo. Es así como hoy en día existen más mecanismos de interacción (canales) que hace 10 años atrás, muchos más que hace 20 o 30 años, y los estudios indican que esta evolución y diversificación de canales de interacción continuará en el futuro inmediato [2].

Antes de la masificación de Internet, los principales canales de comunicación entre personas eran la interacción cara a cara, el teléfono y las cartas. Sin embargo, en la actualidad una gran cantidad de interacciones sociales se realizan a través de medios digitales, como por ejemplo, aplicaciones de mensajería instantánea, voz sobre IP (VOIP en inglés), videollamadas, servicios de redes sociales, etc. La Figura 1 muestra la evolución de los sistemas de social media. Allí se puede ver que no sólo aparecen nuevos sistemas con propósitos distintos, sino también que algunos de ellos redefinen su objetivo o evolucionan sus servicios para adaptarse a las nuevas necesidades de los usuarios o cubrir nuevas porciones del mercado. En ese sentido, las herramientas de apoyo a las interacciones sociales han estado en constante evolución, y todo indica que continuarán así en el futuro.

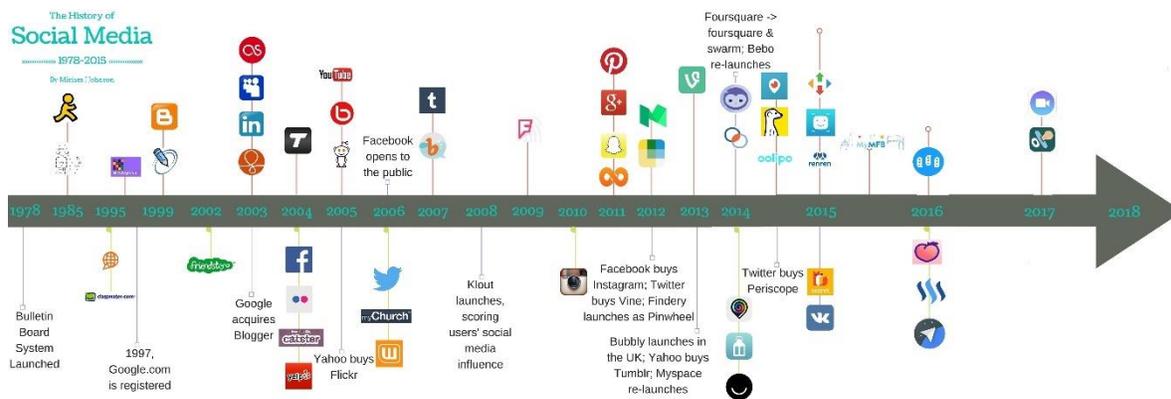


Figura 1: Evolución de los sistemas de social media
Fuente: <http://www.booksaresocial.com/timeline-social-media-2017/>

Por otra parte, la reducción de costos y las capacidades de cómputo de los dispositivos de computación móvil, como los teléfonos o relojes inteligentes, han permitido la ubicuidad de estas interacciones, haciendo que tanto dichos dispositivos, como los servicios que ellos proveen, se vuelvan parte de la vida cotidiana de las personas, especialmente de los más jóvenes.

En escenarios de interacción social mediada por computador, el uso de los servicios y los dispositivos es altamente volátil, ya que éstos se popularizan o caen en desuso en períodos muy cortos de tiempo. Este dinamismo genera en los usuarios la necesidad de mantenerse

constantemente actualizados, cambiando frecuentemente sus preferencias de medios digitales para interactuar con otros, lo cual se traduce en diversos inconvenientes para cualquier persona o sistema que no logre adaptarse a esta evolución tecnológica. En el caso de las personas los inconvenientes apuntan principalmente a su capacidad de mantenerse adoptando las nuevas soluciones, y en el caso de los sistemas esto se traduce en dificultades para mantener la compatibilidad con aquellos productos de software con los que el sistema interactúa.

Desde un aspecto social se puede apreciar cómo muchas personas, particularmente los usuarios precarios de tecnología (por ejemplo, los adultos mayores), no son capaces o no tienen intención de seguirle el ritmo a esta evolución tecnológica [2]. Debido a eso, se genera una asimetría de preferencia del canal de comunicación (o herramienta de interacción social) entre estas personas y el resto de su entorno social, que usualmente es más tecnológico [3] [4]. La asimetría de preferencia de canal se da cuando una persona prefiere un cierto canal para interactuar con otra, quien a su vez prefiere usar un canal diferente para dicho propósito.

En el presente trabajo, se considera como un “canal” a una aplicación específica (por ejemplo, correo electrónico), a través de la cual se pueden comunicar dos personas. La asimetría de preferencia de canal se da cuando una de las partes prefiere utilizar un cierto canal de comunicación, ejemplo email, mientras que la otra prefiere usar uno diferente, por ejemplo, Whatsapp. Lo cual implica una imposibilidad en la comunicación, utilizando los canales preferidos, debido a la discrepancia del canal preferente.

Un ejemplo concreto de esta situación es la que viven los adultos mayores y gran parte de la gente con baja adopción de tecnología computacional. Por un lado, estas personas prefieren utilizar interacciones cara-a-cara y eventualmente telefónicas, mientras que las generaciones más jóvenes de su familia preferirían utilizar aplicaciones como WhatsApp, Facebook, Instagram o similares [3] [4] [5]. La Figura 2 muestra un escenario de este tipo.



Figura 2: Dicotomía entre los escenarios de interacción social preferidos por las generaciones más jóvenes y más adultas de una comunidad familiar

Estas asimetrías de preferencia de canal entre los miembros de una comunidad familiar se traducen en una reducción de la cantidad de interacciones entre los involucrados (es decir, entre cada par de personas que tienen asimetría de preferencia de canal), lo cual usualmente aumenta el aislamiento social de aquellos que están limitados para usar tecnología; en este

caso, los adultos mayores. La literatura reporta que esta situación afecta la salud física, mental y social de los más desfavorecidos [6].

Esta asimetría de preferencia de canal se ve acrecentada por la evolución normal de las aplicaciones sociales, y representa un problema abierto, que involucra múltiples aristas, que van desde la seguridad y compatibilidad de los servicios que proveen, hasta aspectos sociales y emocionales de los usuarios [7] [8].

Este trabajo de tesis abordó parte de este problema social a través del desarrollo de una solución computacional, la cual permite la interoperabilidad entre múltiples servicios de interacción social a nivel de mensajes asíncronos; particularmente, esta solución soporta el intercambio asimétrico de mensajes multimedia entre los siguientes canales: email, Telegram, WhatsApp y Skype. La estrategia general de integración de estos servicios involucró la traducción automática y bidireccional de mensajes; función que es realizada por un sistema intermediario llamado SMT (Social Message Translator), el cual fue desarrollado en el marco de esta tesis. Este intermediario permite a aplicaciones sociales comerciales y sistemas ad hoc, interactuar entre ellos de forma fácil y transparente siempre que las aplicaciones involucradas cumplan ciertos requerimientos. En la siguiente sección se describe en detalle el problema computacional abordado y sus múltiples aristas.

1.1 Desafío Computacional Abordado

En la sección anterior se explica el problema general abordado en esta tesis, y las implicancias sociales que este problema genera. En esta sección se analiza el desafío computacional que se debe abordar durante el diseño de la solución de software que se propone para reducir la asimetría de preferencia de medios de interacción. Esta solución busca mejorar la utilidad de herramientas tecnológicas ad hoc por parte de los adultos mayores, para interactuar socialmente con el resto de su comunidad familiar, a pesar de que estos últimos prefieran utilizar una herramienta de comunicación social diferente a la del adulto mayor. Para que la solución tenga chances de mitigar el problema planteado, ésta debe abordar varias aristas del problema, las cuales representan requisitos o restricciones a la solución. A continuación se describen brevemente estas aristas a abordar.

- *Considerar el ecosistema de interacción completo.* Dado que el ecosistema social de un adulto mayor involucra personas de distintas generaciones (hijos, nietos, amigos, hermanos, etc.), cuando se diseña un software social para estas personas, no sólo hay que enfocarse en las necesidades de los adultos mayores, sino también en las del resto de su comunidad; o sea, todos aquellos que van a interactuar con el adulto mayor. Respetar las necesidades y preferencias de todos hará que las interacciones sean probablemente más frecuentes y naturales, lo que termina traducándose en una mayor integración social de los adultos mayores [5].
- *Evitar cambiar el paradigma de interacción humano-computador adoptado por el adulto mayor.* Es sabido que la mayoría de los adultos mayores tiene limitaciones para adoptar tecnología moderna de interacción social. Sin embargo, muchos de ellos se atreven a aprender a usar una cierta herramienta, siempre que dicho aprendizaje le

sirva por un período largo de tiempo. En otras palabras, el adulto mayor no quiere tener que estar aprendiendo a hacer lo mismo que ya sabía hacer, pero ahora de una manera distinta, pues la herramienta que él o ella usaba evolucionó y cambió su paradigma de interacción [9]. Por lo tanto, el diseño de herramientas ad hoc para estas personas debe mantener el paradigma de interacción aprendido por los adultos mayores, tanto como sea posible.

- *Asegurar la continuidad operativa de la aplicación.* Dado que los servicios comerciales de redes sociales evolucionan regularmente, los sistemas ad hoc que los usan deben también actualizarse para poder seguir interactuando con ellos. Por lo tanto, se vuelve necesario realizar actualizaciones de software en las aplicaciones ad hoc, las cuales logísticamente deben ser fáciles de actualizar por el parte del desarrollador, o por quien se encargue de la continuidad operativa de la aplicación. Una vez que el adulto mayor adoptó una cierta tecnología (considera hardware y software), ésta usualmente se vuelve una necesidad para él/ella, por lo tanto, el desarrollador debe considerar la continuidad operativa de dicha aplicación al momento de concebirla e implementarla [10].
- *Actualizar las aplicaciones facilitando la logística de dicha actividad y respetando la privacidad de los usuarios.* Si bien las actualizaciones son necesarias, esa tarea tiene que hacerse tratando de no intervenir físicamente el dispositivo del adulto mayor, pues una vez que ellos han adoptado un dispositivo y el software para interactuar, la mayoría sienten a eso como algo privado, y por lo tanto no quieren que otras personas toquen “sus cosas”. La información que manejan estos sistemas es usualmente sensible para el adulto mayor, por lo cual éste debe sentir que las actualizaciones de software no son un acto de vulneración de su privacidad [11].
- *Mantener la seguridad y privacidad de la información que se intercambia.* En este sentido se espera que la información intercambiada entre los miembros de una comunidad familiar, ya sea que usen herramientas comerciales o ad hoc para interactuar, esté suficientemente protegida y segura para que nadie acceda sin autorización a dicha información, o bien suplante a alguno de los participantes de la interacción social. La seguridad/privacidad real y la percibida afecta tanto a los adultos mayores, como al resto de su comunidad familiar [12].

1.2 Objetivos de la Tesis

El objetivo general de este trabajo de tesis es proponer una solución capaz de permitir interoperabilidad, a nivel de mensajes asíncronos, entre distintas herramientas de interacción social, ya sean éstas aplicaciones ad hoc o servicios de redes sociales (o similares). Para lograr ese objetivo se propuso una estrategia de traducción de mensajes que es ejecutada por un sistema intermediario, que permite la comunicación entre usuarios que utilizan canales de comunicación diferentes (Figura 3). Se espera que este traductor de mensajes sociales ayude a manejar el dinamismo en los servicios de interacción digital, y también la asimetría de preferencia de canal de quienes llevan a cabo la interacción.



Figura 3: Operación esperada del sistema traductor

Para lograr este objetivo general, se utilizará el SocialConnector [13] como aplicación cliente debido a que ha sido desarrollada por alumnos y profesores de la Universidad de Chile, por lo que existe mayor cercanía para coordinar cambios y mejoras a sus funcionalidades. Esta aplicación deberá utilizar el traductor para lograr invisibilizar los diversos canales soportados para interactuar con los adultos mayores (particularmente, Telegram, Skype, Gmail y Whatsapp), lo cual se puede ver en la Figura 4.

Debido a que el SocialConnector (aplicación ad hoc para adultos mayores) será quien interactuará en background con el traductor de mensajes proveyendo los servicios antes mencionados, los adultos mayores no deberían tener que estar periódicamente aprendiendo a utilizar nuevas herramientas para poder interactuar con sus familiares. Es decir, el sistema traductor debe ser capaz de manejar la comunicación con un conjunto diverso de canales de interacción, mediante una interfaz de programación (API) común, permitiendo abstraer a las aplicaciones ad hoc, de las particularidades propias de los medios de comunicación con los que ellas interactúan. Esta capa de aislación que implementa el traductor, permite finalmente a los adultos mayores ignorar tanto los canales a través de los cuales se llevan a cabo las interacciones, como las preferencias de canal de quienes interactúan con ellos.

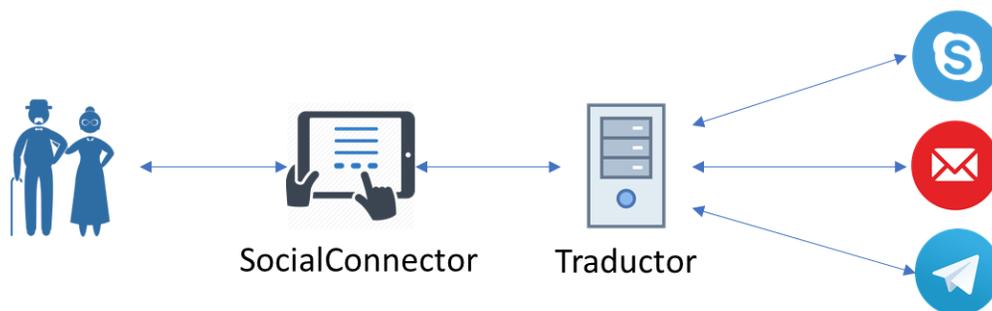


Figura 4: Operación del sistema traductor con los distintos actores

Los objetivos específicos que se desprenden del objetivo general son los siguientes:

- Crear un protocolo de comunicación entre el sistema traductor y los distintos medios de comunicación.

- Desarrollar un sistema de traducción bidireccional de mensajes que permita que las aplicaciones ad hoc (por ej. el SocialConnector) interactúen con diversos servicios o herramientas utilizando el protocolo antes definido. Los servicios considerados son Skype, Telegram, correo electrónico y Whatsapp.
- Evaluar la efectividad del sistema de traducción de mensajes y conectarlo con el sistema SocialConnector.

1.3 Metodología

Para el desarrollo del presente trabajo, se utilizó un enfoque incremental [14], de forma de tener prototipos funcionales y testeados al final de cada iteración. De esa forma era más fácil mantener el trabajo bajo control. El proceso general consideró las siguientes cinco etapas (o incrementos):

- a) *Verificar la Necesidad de Alinear Canales*: Si bien esta necesidad ha sido reportada en la literatura sobre interacción con adultos mayores, tanto en países desarrollados [4] [7] [8] como también en vías de desarrollo [3] [2], el primer paso a realizar en este trabajo de tesis consistió en verificar dicha necesidad en el escenario local; particularmente en la ciudad de Santiago, Chile. Este trabajo de verificación ayudó también a identificar los requisitos críticos con los que debía cumplir la solución propuesta, a fin de facilitar la interacción social entre diversos canales de comunicación.
- b) *Concepción de la Solución*: La idea de esta etapa fue explorar alternativas para atacar el problema, tratando de entender los pros y contras cada una de ellas. Una vez escogida la estrategia de solución, se procedió a diseñar la estructura de la misma y los mecanismos de comunicación entre los distintos elementos que la componen. Las actividades a destacar en esta etapa son las siguientes:
 - Revisión de alternativas de solución al problema planteado.
 - Diseño de los componentes y los mecanismos de seguridad del sistema.
 - Definición de los protocolos de comunicación entre los elementos del sistema. Esto incluye la entrada al sistema, y la traducción de formatos de mensajes para que el SocialConnector pueda interactuar con los canales de comunicación considerados.
- c) *Servicios de Traducción de Mensajes*: En esta etapa se identificaron diversos servicios, que usan la solución diseñada en la etapa anterior, y permiten intercambiar mensajes multimedia a través de los distintos canales de comunicación predefinidos por los usuarios involucrados. Estos servicios constituyen el traductor de mensajes que en la Figura 4 se indica como “traductor”. Para lograr este objetivo se debió:
 - Construir servicios de traducción de mensajes según el diseño realizado en la etapa anterior.
 - Evaluar el correcto intercambio de mensajes entre los servicios y las APIs de los canales de comunicación escogidos. Además, se evaluó el rendimiento de

los servicios, en términos de tiempos de envío y soporte de demanda de intercambio de mensajes, buscando que los tiempos de respuestas sean razonables y no perjudiquen la interacción entre los participantes.

- d) *Integración de Servicios al SocialConnector*: En esta fase se integraron al SocialConnector (SC), los servicios implementados en el marco de esta tesis. Para lo cual se debió:
- Entender funcionamiento del SC.
 - Modificar el SC para el uso efectivo de los servicios creados en el punto anterior.
 - Evaluar el correcto funcionamiento de las modificaciones realizadas al SC, junto con la comunicación con el sistema.
- e) *Evaluación Técnica de la Solución*. Aquí se evaluó la traducción de mensajes en un escenario de laboratorio. Para ello se probaron todas las combinaciones de canales posibles, y analizando la correctitud del proceso de traducción de mensajes y del protocolo de intercambio de estos. Por otra parte, se evaluó con un par de desarrolladores, la complejidad y el esfuerzo requerido para usar y extender el sistema de traducción propuesto.

1.4 Estructura del documento de tesis

El presente documento cuenta con 7 capítulos principales. En el Capítulo 2 se presenta el marco teórico y los trabajos relacionados. El Capítulo 3 presenta un estudio realizado con el fin de determinar la existencia de la necesidad de alinear los canales de comunicación. El Capítulo 4 describe del diseño de la solución, particularmente el modelo de datos, la arquitectura física, arquitectura lógica, y el diseño de los endpoints del sistema. Por otra parte, el Capítulo 5 muestra la implementación de la solución, tanto a nivel del desarrollo del sistema, como la comunicación entre éste y el SocialConnector. El Capítulo 6 explica las distintas evaluaciones realizadas al sistema en términos de su correctitud para la traducción de mensajes, y la seguridad y usabilidad del mismo. Finalmente, el Capítulo 7 presenta tanto conclusiones respecto al trabajo realizado, como las posibles labores a realizar en un futuro, de manera de seguir mejorando el sistema.

2 Marco Teórico

Existe una amplia variedad de trabajos en el ámbito de la comunicación social apoyada por tecnología, entre adultos mayores y su entorno familiar. Por ejemplo, sistemas para proveer apoyo emocional [15] usualmente a través de espacios digitales compartidos [3] [16] [17] [18], displays ambientales [3] [19], televisión socialmente inteligente [20] [21], o contacto físico simulado [22] [23]. A pesar de las ventajas que han traído estas propuestas (las cuales están reportadas en los artículos correspondientes) y de los avances realizados en este ámbito, hay muy poco trabajo hecho en lo que respecta a la alineación de canales de comunicación social entre estas partes. En este capítulo se presenta el marco teórico de este trabajo de tesis, el cual incluye también los trabajos relacionados con la alineación de canales de comunicación.

2.1 Desafíos de Socialización para los Adultos Mayores

En la literatura se reportan diversos problemas que presentan los adultos mayores que son propios del proceso de envejecimiento [24] [25] [26]. En lo que respecta a su contexto social, presentan problemas de socialización con sus familiares en múltiples dimensiones; en particular, este trabajo se enfoca principalmente en mejorar la comunicación con ellos utilizando medios digitales, y respetando sus preferencias de medio de comunicación.

Los problemas de socialización que presenta este grupo etario influyen directamente en su bienestar físico y psicológico [27] [28], haciendo que sea un aspecto importante a tratar de resolver para la sociedad. Cabe destacar también el interés que presentan los adultos mayores, tanto por mantener el contacto con sus familiares, como por utilizar tecnología simple para desarrollar dicha tarea [29].

Por otra parte, está evidenciado en distintos trabajos la importancia de abordar la asimetría de preferencia de medios de comunicación, en el diálogo entre un adulto mayor y sus familiares directos [6] [30] [31]. Allí se ven importantes diferencias entre los canales preferentes de un adulto mayor, respecto a los de sus hijos y nietos, principalmente enfocadas en la preferencia de medios de comunicación síncronos por parte de los adultos mayores frente a la preferencia de medios asíncronos entre los más jóvenes. En este escenario de interacción social, algunos miembros de la familia juegan un papel preponderante al actuar de proxies entre estas dos generaciones (es decir, abuelos y nietos) [3].

Junto con esto, la literatura también reconoce las distintas preferencias de medios digitales, según el segmento etario al cual pertenezcan los miembros de una familia. Por ejemplo, los jóvenes suelen preferir utilizar las redes sociales y mensajería instantánea, los adultos se inclinan por el correo y la telefonía móvil, mientras que los adultos mayores utilizan teléfonos convencionales, interacciones cara-a-cara y cartas [32] [33]. Cabe destacar la importancia que tiene para los adultos mayores las comunicaciones mediante video, las cuales mejoran la sensación de cercanía con sus familiares [34] y de cómo las comunicaciones de naturaleza asíncrona se perciben más flexibles [35], lo cual es importante para los jóvenes.

Por todo esto, la sociedad tiende a pensar que las limitaciones de comunicación que presentan actualmente los adultos mayores se acabarán cuando las generaciones jóvenes (“nativos digitales”) lleguen a dicha edad. Sin embargo, nuestra capacidad de asimilación tecnológica no puede seguirle el paso a la evolución de dichas tecnologías [36], por lo que este problema seguramente será una constante en el tiempo, pero diferente a como se vive en la actualidad.

En la actualidad no se ven sistemas que permitan manejar el dinamismo de los servicios que entregan los canales de comunicación digitales, o la volatilidad de las preferencias de canal de los usuarios. Esto provoca que se generen aplicaciones que terminan siendo tremendamente dependientes de un medio (canal) específico. Esta dependencia vuelve a estos servicios vulnerables frente a los cambios y evoluciones del medio utilizado, impidiendo un crecimiento independiente de la aplicación. Un problema aún más grave que se genera a partir de esta situación es el cese de las funciones de un medio o canal. En ese caso, las aplicaciones ad hoc que usaban dicho canal reducirán considerablemente su utilidad, y en el mejor de los casos se requerirá emplear gran cantidad de recursos para obtener una funcionalidad similar a la que se tenía antes del cese del servicio mencionado.

Algunas de las aplicaciones ad hoc diseñadas para apoyar la interacción social con adultos mayores son las siguientes: Tlatoque [3], BuildingBridges [37] y SocialConnector [13]. Éstas se encuentran vinculadas directamente a algún medio digital (o canal), y por lo tanto son tremendamente dependientes de esos servicios. La mayoría de estas aplicaciones proponen el uso de un mediador que oculta (o enmascara) a través de una interfaz simple, los servicios que le permiten a los adultos usar canales particulares de interacción social (Figura 5).

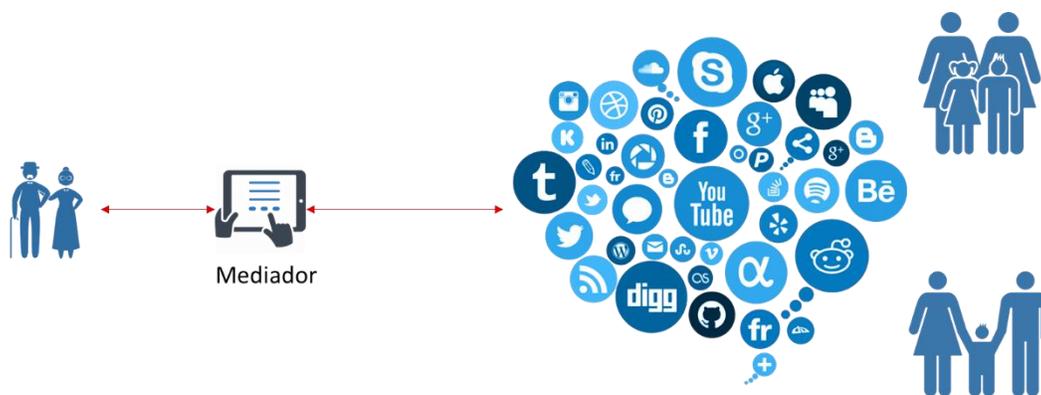


Figura 5: Esquema de interacción social basada en un sistema mediador, el cual usa algún medio o canal dentro de la nube

Aunque estos sistemas mediadores han demostrado ser útiles para integrar a los adultos mayores a sus comunidades familiares, todos ellos fallan en la forma en cómo manejan la evolución de las herramientas de interacción social a lo largo del tiempo; ya que, al estar fuertemente conectados a un canal en particular, se vuelven poco resistentes a los cambios.

Hoy en día cualquier cambio en el protocolo de intercambio de mensajes de una de estas herramientas, obliga a que se tenga que cambiar el código fuente del mediador. Lo mismo ocurre si se desea agregar soporte para un nuevo canal de comunicación al sistema mediador. Esta actualización del código en las aplicaciones cliente, si bien es factible, es muy costosa

y operativamente casi imposible de realizar por parte de los adultos mayores. Por lo tanto, es algo que se debería querer evitar a toda costa, pues esto le pone fecha de caducidad a las aplicaciones ad hoc, como ya se ha visto en la práctica; por ejemplo, Tlatoque [3] y Building Bridges [37] ya no funcionan con las versiones actuales de los servicios de interacción social que utilizan como base.

El problema fundamental que da origen a estas incompatibilidades y asimetrías tiene que ver con la falta de adherencia a estándares, por parte de los protocolos de intercambio de mensajes que utilizan estas herramientas [38]. En la actualidad cada herramienta define su propio protocolo y formato de mensaje, lo cual la hace potencialmente incapaz de conversar con otras herramientas que cumplen un rol similar. Esto ocurre con casi todos los servicios de comunicación digitales disponibles en la nube de medios, excepto para los servicios de email y de telefonía sobre IP (VoIP).

Por otra parte, la gente joven es capaz de manejar el dinamismo propio del avance tecnológico, y adoptar fácilmente las nuevas tecnologías. Sin embargo, en la medida que esa gente envejece, dicha capacidad se va perdiendo [30], hasta el punto donde, por ejemplo, una porción importante de los adultos mayores en Chile no quiere tener que enfrentarse al desafío de aprender a usar nuevos sistemas o dispositivos tecnológicos [2]. De esta forma se acrecienta el aislamiento social que sufre la población mayor, principalmente por la asimetría en la preferencia de medios de comunicación entre ellos y las generaciones más jóvenes.

Debido a que los adultos mayores no quieren enfrentarse a las nuevas tecnologías, se mantienen estáticos utilizando los canales de comunicación que ya saben usar, lo cual fuerza al resto de los familiares a tener que adecuarse a las preferencias del adulto mayor. Esta situación genera tensiones con el adulto mayor y los miembros de su familia debido a las limitaciones que los adultos mayores imponen para interactuar con ellos. Ambas cosas descritas repercuten directamente en la salud del grupo etario más vulnerable, que por tendencias demográficas tenderá a ser cada año más cuantioso [39] y, por lo tanto, se vuelve un problema aún más importante para la sociedad.

Muchos adultos mayores son capaces de aprender a utilizar una herramienta tecnológica por una única vez. Sin embargo, si esta herramienta cae en desuso o si evoluciona por ejemplo cambiando su paradigma de interacción, dichos usuarios probablemente la van a abandonar, puesto que la mayoría no están interesados en seguir el ritmo del avance tecnológico [40]. Esto indica que el problema a resolver tiene más que ver con cómo manejar la evolución tecnológica, que con lograr que los adultos mayores adopten tecnología para interactuar socialmente con otras personas.

Todo esto nos muestra un caso particular en donde el gran problema que existe respecto al dinamismo en canales de comunicación y la falta de adopción de estándares, repercute directamente en los servicios que se encuentran actualmente disponibles a través de dispositivos computacionales, e impacta aún más a un segmento etario cuyo porcentaje en la sociedad crece cada día, es decir, a los adultos mayores.

2.2 Herramientas de Apoyo a la Interacción Social con Adultos Mayores

Con el fin de tratar de enfrentar las circunstancias anteriormente descritas y darles apoyo a los adultos mayores frente a estas problemáticas, se han creado diversas herramientas. A continuación, se describen los trabajos más relevantes.

En Irlanda, fue desarrollado un proyecto enfocado en reducir los riesgos de soledad y aislamiento social de estas personas utilizando tecnologías de comunicación, al cual se le denominó Building Bridges [37]. Este sistema fue instalado en los hogares de adultos mayores en pos de mejorar su comunicación con otras personas. Building Bridges está basado en una pantalla táctil, audífonos y parlantes y permite realizar acciones como escuchar noticias, intercambiar mensajes con otras personas, hablar con otros, y participar en grupos de conversación públicos, entre otras. Es decir, este prototipo básicamente busca reducir los riesgos del aislamiento social acercando a desconocidos que tienen intereses en común, y permitiéndoles mantener conversaciones mediante el sistema.

Entre las herramientas que cumplen este rol también se encuentra Tlatoque [3], desarrollada en México, la cual permite estar pendiente de los familiares al recolectar sus fotos de Facebook y mostrarlas a los adultos mayores de forma de estar más al día respecto a las actividades de sus familiares. Junto con esto, el sistema tiene mecanismos para informar a los familiares sobre las acciones del adulto mayor, generando un flujo bidireccional de información entre los actores. Según los autores, el dispositivo proporciona los medios para facilitar la integración de los adultos mayores en redes sociales, mejorando las relaciones asimétricas que existen con los miembros más jóvenes de la familia.

InTouch [13] es una aplicación cuyos objetivos son similares a los anteriores ya que está enfocada en adultos mayores sobre los 80 años con riesgos de aislamiento social y soledad. Utilizando el sistema operativo Android como base y una interfaz basada en íconos (de forma de simplificar la interacción de los usuarios con el sistema), permite la comunicación tanto síncrona como asíncrona. Particularmente, permite enviar mensajes de texto predefinidos, al igual que fotos y mensajes de audio o video.

Finalmente, una de las herramientas que ha sido particularmente diseñada para abordar la comunicación intergeneracional en el ámbito familiar y ser usada por adultos mayores, es SocialConnector (SC) [13]. Esta aplicación permite a los usuarios comunicarse con sus familiares a través de Skype, Whatsapp y correo electrónico. También permite monitorear a los adultos mayores de forma de chequear su estado de ánimo y levantar alertas en casos de que se encuentren tristes o solos por periodos largos de tiempo.

2.3 Alternativas Tecnológicas para Alinear Canales de Comunicación

Para poder llevar a cabo la traducción de mensajes entre distintos canales de comunicación de forma transparente, es importante poder comunicarse con cada uno de esos canales. Durante el desarrollo de este trabajo, no se encontró en la literatura la descripción de un sistema con similares características y objetivos. Sin embargo, sí se encontró un protocolo estándar para el intercambio de mensajes entre sistemas de mensajería instantánea [41]. A

continuación, se escribe y discute dicho protocolo, y luego presenta un breve análisis de la situación actual y de las restricciones necesarias para diseñar un sistema que este pueda funcionar en la práctica.

2.3.1 Protocolo XMPP (Extensible Messaging and Presence Protocol)

XMPP es un protocolo abierto [41], mantenido por la XSF (XMPP Standards Foundation), el cual fue creado para la comunicación en tiempo real, ya sea de mensajería instantánea, chat de múltiples personas, llamadas de voz o de video, o notificaciones, entre otras. El protocolo nació inicialmente como una alternativa abierta y descentralizada a los servicios de mensajería instantánea de la época.

Las especificaciones principales de XMPP son desarrolladas en el Internet Engineering Task Force (IETF). Junto con esto, las extensiones del protocolo son desarrolladas por la XSF, en protocolos de extensión llamados XEP (XMPP Extension Protocol). Gracias a estos, es posible agregar funcionalidad y mantener vigente el protocolo con las nuevas exigencias de los usuarios o el mercado en sí.

A grandes rasgos, lo que permite XMPP es intercambiar fragmentos relativamente pequeños de datos estructurados en XML. Este protocolo es típicamente implementado con una arquitectura de cliente-servidor. El proceso donde el cliente se conecta con el servidor para intercambiar mensajes es explicado brevemente a continuación.

1. Identificar la dirección IP y puerto al cual conectarse. De forma de asegurar conocer la máquina con la cual se realizará el intercambio de mensajes.
2. Abrir una conexión TCP. Para poder conectar ambas máquinas y asegurar que se envíen/reciban los paquetes de datos sin errores y en el orden correcto.
3. Abrir un stream de datos XML sobre TCP. Así se logra enviar datos con el formato XML a través de la conexión TCP abierta en el paso anterior.
4. Encriptar el canal con TLS. De forma que luego de este punto los datos enviados entre ambas máquinas sean ilegibles para quien trate de interceptar la comunicación.
5. Autenticar utilizando SASL. Para los cual se envían los datos necesarios para permitir el intercambio de datos con el servidor de mensajes.
6. Enlazar el recurso al stream de datos de forma de identificar al cliente. Es decir, mandar información que logre identificar al usuario emisor de los mensajes.
7. Intercambiar un número indefinido de datos XML entre las entidades. Estos datos XML representan, por ejemplo, los mensajes multimedia que se quieran enviar.
8. Cerrar el stream de datos XML, una vez que se haya decidido dejar de enviar mensajes al servidor de mensajes.
9. Cerrar conexión TCP cuando ya no se quiera enviar datos al servidor de mensajes.

Este es el protocolo básico para intercambio de información, mayor detalle de esta conexión puede verse en el RFC 6120. En el caso de acciones como comunicación entre personas, entre grupos, u otras características, son especificados en cada XEP, y el funcionamiento del

protocolo varía según su cada caso, esto sin afectar el protocolo principal explicado anteriormente.

El protocolo base explicado anteriormente no habla en ningún momento de mensajes multimedia, solo de mensajes con formato XML. Esto se hace con el fin de permitir agregar fácilmente nuevas extensiones a éste (a través de los XEP), las cuales especifican los posibles formatos de XML a recibir y/o la interacción con estos. Por otra parte, la necesidad de conocer cada una de estas extensiones e implementarlas, afectan el uso del protocolo en caso de querer utilizar un servidor propio. En caso contrario se debe utilizar algunos de los servidores ya existentes en el mercado.

Por otro lado, existe poco uso en la práctica del protocolo por parte de las grandes empresas o aplicaciones de mensajería instantánea con gran cuota de mercado. En el caso específico de Whatsapp por ejemplo, se utiliza una versión modificada del protocolo, por lo que su proceso de intercambio de mensajes no es compatible con XMPP [42]. En otros casos se utilizan protocolos desarrollados por las mismas empresas, también incompatibles con XMPP. Esto probablemente sucede porque a las compañías les interesa tener dominio sobre el protocolo, los datos y los mismos usuarios, de forma de tener completo poder sobre tanto el flujo como la información misma, y así poder mejorar sus servicios o productos en base a estos datos.

2.3.2 Restricciones al funcionamiento de la solución

Un requerimiento básico para lograr implementar el sistema es la necesidad de intercomunicar distintos servicios, de forma que éstos se encarguen de realizar la comunicación, y que el usuario final no deba dirigirse a las distintas aplicaciones, es decir, abrirlas en su dispositivo computacional, para poder intercambiar mensajes con el destinatario. Todo esto con el fin de hacer el sistema transparente, y que el adulto mayor sólo deba utilizar una única interfaz de usuario para lograr la interacción deseada.

Para poder brindar estos servicios se hace necesario que los canales permitan realizar la comunicación con ellos de forma pública a través de Internet. Esto suele realizarse a través del uso de APIs Web que permiten recibir y enviar distintos mensajes, esta es la forma en la que mayoritariamente interactúan las aplicaciones móviles con los servidores.

Lamentablemente, se encontraron dos grandes dificultades en la práctica para poder realizar el intercambio transparente de mensajes. La primera es que a pesar de la similitud entre las funciones de los distintos servicios de comunicación digital, ninguna empresa utiliza un protocolo estándar para el intercambio de mensajes, excepto en los casos del correo electrónico y de la telefonía IP, en la cual existen protocolos claros y ampliamente difundidos y utilizados. La segunda dificultad que se ve en la industria, es que aun sabiendo de la existencia de formas de comunicarse con los diversos canales a través de algún protocolo, no todas las empresas permiten el uso de su API de forma pública o gratuita. Por ejemplo, grandes aplicaciones como WhatsApp no permiten el acceso [43] e incluso pueden prohibir de forma indefinida el uso de sus servicios. También existen casos como Facebook [44], que permite el uso de su API pública, pero hacen cobro por el uso de ésta. Finalmente, existen

servicios como Skype [45] y Telegram [46] que permiten acceder a sus servicios de forma gratuita y sus protocolos de comunicación son públicos.

Estas restricciones hacen que el sistema a diseñar sólo pueda comunicarse con aquellos servicios que posean una API pública, y por tanto, todos los otros servicios son inalcanzables para éste y cualquier otro sistema que pretenda comunicarse con ellos.

Finalmente, existen protocolos estándares de comunicación que permiten realizar todas las operaciones esperadas en este tipo de sistemas (como enviar mensajes, audios, imágenes, realizar videollamadas, etc.). A pesar que estos protocolos podrían brindar múltiples beneficios, por ejemplo, entregar más libertad a la hora de desarrollar nuevas funcionalidades a las aplicaciones, realizar optimizaciones específicas según cada compañía, o evitar la interconexión con otros sistemas sobre los que no se tiene control, estos protocolos no son utilizados en la práctica por parte de las aplicaciones que tienen una adopción importante por parte de la ciudadanía.

Luego de una revisión de los trabajos en el área, no se ha encontrado ningún sistema que permita la comunicación (asimétrica) a través de múltiples medios de comunicación, y que se pueda llevar a cabo de forma transparente al usuario. Por esa razón, y considerando el problema planteado, se ve la necesidad de crear una herramienta que soporte la interacción asimétrica entre medios de comunicación digital, la cual debe ser transparente para los distintos involucrados, y debe poder utilizarse por parte de aplicaciones de terceros (por ej., por herramientas de apoyo a la interacción social de adultos mayores como el SocialConnector). Como resultado de esto, se podría soportar la diversidad de preferencias de medios de comunicación de los distintos participantes, así como las limitaciones antes mencionadas (por ejemplo, que la herramienta usada por las personas tenga una API que permita el intercambio de mensajes). De esa manera se podría incrementar la utilidad de las aplicaciones para la tercera edad, más allá del ámbito planteado; por ejemplo, en el monitoreo de pacientes con enfermedades crónicas.

2.4 Aspectos involucrados en el problema a abordar

Considerando todo lo antes expuesto, el problema abordado en este trabajo de tesis involucra dos aspectos; uno *social* y otro *computacional*. El aspecto social tiene que ver con brindarle la posibilidad a los adultos mayores de interactuar socialmente con el resto de su familia a través de medios tecnológicos, a pesar de sus limitaciones para utilizar la tecnología. Por otra parte, el aspecto computacional a abordar contiene múltiples aristas, entre ellas un sistema de intercambio de mensajes transparente, la seguridad tanto en el intercambio como en el almacenamiento de la información y la heterogeneidad de formatos y protocolos de mensajes existentes hoy en día.

Los adultos mayores pueden lograr aprender a utilizar medios tecnológicos, sin embargo, debido a la rápida evolución y cambio en las redes sociales, es muy fácil que se vuelvan “huérfanos tecnológicos” al haber aprendido a utilizar una herramienta que cayó en desuso. Éste es un caso especialmente problemático debido a la frustración y desincentivo que genera

en los adultos mayores. Lo que esperan estas personas es aprender algo, y que dicho conocimiento les sea útil por mucho tiempo, de forma de no tener que volver a pasar por ese proceso de aprendizaje de nuevo. Este es un punto sumamente importante en el desarrollo de cualquier tecnología enfocada en dicho grupo etario.

Uno de los aspectos computacionales a abordar radica en cómo hacer para soportar el intercambio de mensajes entre distintas aplicaciones de manera tal que sea transparente para el usuario final. Además, el mecanismo dicho intercambio de mensajes debe poder actualizarse de manera simple, sin tener que intervenir las aplicaciones clientes. Esta capacidad se requiere dado que los servicios de mensajería cambian periódicamente, al igual que las APIs que ellos ofrecen, y por lo tanto estos cambios requieren realizar extensiones o ajustes a la actual solución de intercambio de mensajes. Debido a lo antes mencionado, esta evolución tecnológica debe poder ser absorbida por la solución que se proponga, haciendo al usuario inmune (tanto como se pueda) a estos cambios.

Junto con esto, se debe considerar el hecho de que los protocolos de intercambio de mensajes son diferentes en los distintos servicios, así que la solución que se proponga también deberá lidiar con estas diferencias, adaptando el formato de los mensajes y el protocolo de interacción con cada servicio. De esa manera el intercambio de mensajes entre múltiples servicios podrá realizarse de manera efectiva.

Otra de estos aspectos técnicos tiene que ver con la seguridad de la solución que se proponga; partiendo por la autenticación de los usuarios. Cada servicio de comunicación tiene su propio esquema de manejo y autenticación de usuarios, los cuales son incompatibles entre ellos; por lo tanto, cualquier solución que se proponga deberá poder manejar múltiples perfiles y mecanismos de autenticación y esto también debe ser transparente para el usuario.

Finalmente, hay un tema de privacidad que es sumamente delicado, y que tiene que ser abordado apropiadamente por cualquier solución. Particularmente, el almacenamiento de los mensajes intercambiados debe ser hecho de manera tal que si ocurriera una intrusión no deseada en el sistema, los datos que el intruso pudiera llevarse no deberían permitirle tomar el control de las cuentas de los usuarios; por ejemplo, para realizar suplantación de identidad.

En resumen, si bien hay un problema social importante asociado a la adopción tecnológica por parte de los adultos mayores (particularmente, a la adopción de herramientas de interacción social), el problema computacional que hay que abordar para intentar dar solución al desafío social es grande e involucra múltiples aspectos, tal como se ha explicado en esta sección.

3 Determinación de la Necesidad de Alinear Canales

Tal como se mencionó antes, la literatura muestra múltiples reportes acerca de la diversidad de preferencia de medios de comunicación entre miembros de una comunidad familiar; particularmente entre miembros de distintas generaciones [6]: adultos mayores, adultos, y jóvenes. En el caso chileno, y latinoamericano en general, hay muy pocos reportes al respecto, por lo que en el marco de este trabajo de tesis se comienza con la realización de un análisis de la necesidad que tienen los miembros de una comunidad familiar, de alinear sus canales de comunicación con aquellos que usan los adultos mayores. A continuación, se describe el estudio realizado para verificar esto, los resultados obtenidos, y se discuten las implicancias que estos resultados tienen para el diseño de la solución propuesta.

3.1 Descripción de los participantes

Para el presente estudio se tomaron 23 participantes (edad promedio de 34,7 años y desviación estándar de 14,5), de los cuales 12 eran mujeres y 11 hombres; y 11 eran adultos y 12 eran adultos jóvenes o adolescentes. En ambos casos los participantes declararon tener interacciones regulares (mínimo una vez a la semana) con al menos un adulto mayor de sus familias. Se escogieron a estas personas para realizar el estudio ya que forman parte del problema de asimetría antes planteado, y porque además todos ellos saben utilizar más de una herramienta para interactuar con los adultos mayores, por lo tanto, podrían eventualmente cambiar sus preferencias de canal, para así estar alineados con los adultos mayores. La predisposición de estas personas para llevar a cabo esta acción es parte de lo que se pretendía explorar.

3.2 Proceso de recopilación de información

Para la obtención de los datos se realizaron entrevistas semi-estructuradas en persona a los 23 involucrados. A todos ellos se les solicitó que proporcionaran su consentimiento libre, explícito e informado antes de comenzar las entrevistas. En el Anexo E se incluye la copia del formulario de consentimiento informado. En el caso de los participantes menores de edad, además de esto, se obtuvo el consentimiento formal de sus tutores legales (uno de sus padres en cada caso). Ningún entrevistado se negó a participar ni pidió retirar sus respuestas después de finalizar las entrevistas. El detalle de las preguntas realizadas puede verse en el Anexo A: Cuestionario para análisis de necesidades.

Las entrevistas tuvieron una duración de 20 minutos aproximadamente (con una media de 19,8 y una desviación estándar de 2,3) y cubrieron los siguientes temas:

1. Los medios de comunicación preferidos para comunicarse con el adulto mayor durante el tiempo libre.
2. Los medios de comunicación preferidos para comunicarse con el adulto mayor durante el horario laboral.
3. La influencia que ejerce "la importancia de un mensaje", en la elección del medio de comunicación escogido para iniciar una interacción. Es decir, si se quiere enviar un

mensaje importante o crítico, cómo afecta esto en la elección de medio de comunicación para enviar dicho mensaje.

4. La influencia que ejerce el “retraso en la respuesta de un mensaje”, en la preferencia del medio a través del cual se quiere recibir dicho mensaje de respuesta. Es decir, si se manda un mensaje, y el receptor se demora un tiempo considerable en responder, ver si la demora afecta el medio de comunicación por donde se prefiere recibir la respuesta.
5. La utilidad percibida de contar con servicios asistidos por computadora para mediar en la interacción entre el participante y el adulto mayor en su familia.

Los problemas de redacción en las preguntas de la entrevista y la posible ambigüedad en las respuestas obtenidas se controlaron a través de un estudio piloto a pequeña escala, que se realizó antes de llegar a los entrevistados seleccionados.

3.3 Resultados obtenidos

Cuando se les solicitó a los familiares del adulto mayor indicar el medio de comunicación preferido para iniciar una interacción social informal con los primeros durante su tiempo libre (por ejemplo, chatear o saludar a la otra parte), una gran mayoría de los participantes de ambas generaciones (jóvenes y adultos) declararon su interés en utilizar los servicios de mensajería instantánea móvil (93%), en particular la herramienta WhatsApp. Este resultado era medianamente esperado, debido a que la mayoría de los participantes utilizan ampliamente sus teléfonos inteligentes, especialmente WhatsApp o Telegram para comunicarse con otros miembros de sus propias redes sociales (principalmente con otros familiares y amigos).

Siguiendo esta misma línea, cuando la disponibilidad se vuelve más escasa; por ejemplo, cuando se intenta transmitir un mensaje importante mientras se está ocupado en el trabajo, la mensajería instantánea sigue siendo el canal de comunicación preferido entre las generaciones más jóvenes (100%). Cabe destacar que en este caso, los adultos declaran una mayor variedad de respuestas, incluidas llamadas telefónicas (36%) o correo electrónico (18%). Se debe tener en cuenta que estas respuestas corresponden al canal de comunicación preferido por cada parte, y no necesariamente a la herramienta que ellos utilizan para interactuar realmente con sus adultos mayores, dado que son conscientes de las limitaciones que tienen estos últimos para adoptar tecnologías modernas. La Tabla 1 muestra un resumen de los resultados obtenidos.

Tabla 1: Resumen de resultados de encuesta de necesidad de alinear canales

	Jornada Trabajo	Jornada Libre	Mensajes importantes	Mensajes poco importantes	Jóvenes	Adultos
Primera Preferencia	Whatsapp (89%)	Whatsapp (91%)	Whatsapp (91%)	Whatsapp (90%)	Whatsapp (92%)	Whatsapp (88%)
Segunda Preferencia	Llamada (23%)	Email (15%)	Llamada (34%)	Email (21%)	Llamada (14%)	Llamada (20%)

Cuando se les preguntó a los participantes acerca de la utilidad percibida de un potencial servicio para mediar en la interacción entre ellos y sus adultos mayores (como el diseñado y reportado en este trabajo), las opiniones eran ampliamente favorables. Los participantes también expresaron interés en que dicho servicio necesitaría poder transmitir con precisión el mensaje enviado.

Asimismo, los entrevistados valoraron la posibilidad de utilizar sus herramientas de preferencia, de una manera que les permita cambiar el canal de acuerdo con la situación social en la que se encuentren actualmente durante el día. Por ejemplo, ellos quieren poder seleccionar libremente los medios de comunicación mientras trabajan (correo electrónico o teléfono), y mensajes instantáneos (por ejemplo, Telegram o Whatsapp) o servicios de redes sociales (por ejemplo, Facebook) mientras están en casa durante su tiempo libre, sabiendo que sus mensajes llegarán correctamente a destino.

3.4 Implicancias de los resultados

En el contexto particular estudiado, y considerando las limitaciones de interoperabilidad de la herramienta de interacción (al menos en términos de capacidades de intercambio de mensajes), se hace evidente la necesidad de desarrollar un componente de software capaz de actuar como un centro de comunicación que agrupe diferentes redes sociales. Ante lo cual se pueden extraer los siguientes requerimientos:

- R1. Correctitud en el envío de mensajes por parte del sistema.
- R2. Comunicación asíncrona entre los interactuantes, la cual debe ser transparente. Es decir, que ellos sigan utilizando sus redes sociales como lo hacen habitualmente y no se enteren del funcionamiento del servicio.
- R3. Considerar la preferencia de redes sociales del destinatario de un mensaje, de forma que siempre le lleguen a su red preferida.
- R4. Soportar múltiples canales de comunicación de modo de darle flexibilidad a los miembros de la familia para poder utilizar el canal que prefieran.
- R5. Permitir conectar aplicaciones de terceros (por ejemplo, el SocialConnector) de modo de recibir y enviar mensajes a través de las múltiples redes sociales soportadas.

Considerando estos requerimientos, se inició el diseño del sistema propuesto, de modo de satisfacer las necesidades expresadas por los usuarios en el presente estudio.

4 Diseño de la solución

En esta sección se explica en detalle la forma en la que la solución fue diseñada de forma de lograr los objetivos planteados anteriormente. Para la solución se diseñó e implementó una API Web que permita la comunicación entre distintos entes, que pueden ser tanto personas como sistemas. La Figura 5 muestra de forma esquemática la estructura del escenario de intercambio de mensajes involucrando al traductor con la aplicación cliente y los distintos canales de comunicación.

En esta sección se explica la forma de comunicación con dicha API (endpoints), como se realiza la traducción de mensajes entre distintos canales de comunicación y finalmente, la forma en la que se analiza y decide la preferencia de uso de redes sociales.

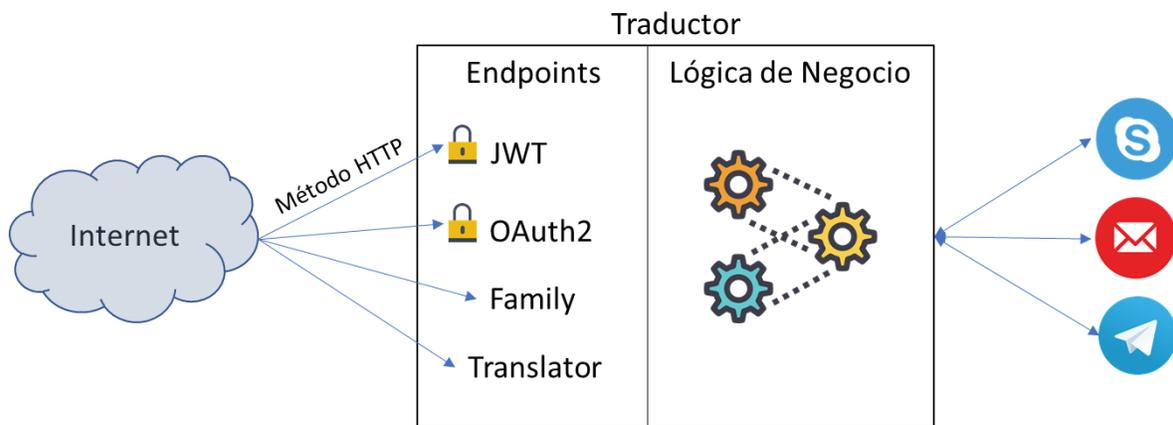


Figura 6: Endpoints de comunicación con el traductor

4.1 Comunicación con el Traductor

Como se mencionó anteriormente, el sistema se diseñó como una API Web por lo que recibe distintos métodos HTTP (GET, POST, etc.) en distintas URLs y esto produce resultados variables, ya sea autenticar al usuario, mandar un mensaje, chequear si existen nuevos mensajes y obtener estos últimos.

La idea de este diseño es que sea lo más genérico posible, de forma de imponer pocas restricciones a las aplicaciones clientes que requieran usar sus servicios. De esta manera, las aplicaciones clientes pueden ser una aplicación móvil, una aplicación web, un servidor, etc., debido a que las consultas a una API Web pueden ser realizadas en la actualidad por prácticamente cualquier lenguaje de programación. Esto va en directa relación con tratar de cumplir con el requerimiento R5 indicado en la sección 3.4.

Para poder comunicarse con la API del Traductor se requiere autenticarse con el sistema. En nuestro caso, dicha autenticación se realiza utilizando ya sea un JSON Web Token (JWT) o utilizando el estándar OAuth2, que son métodos seguros empleados en la actualidad para realizar dichas labores, y son utilizados por una amplia gama de empresas (como Google, Microsoft, Facebook, etc.).

Junto con esto, existen dos grupos de endpoints importantes que realizan la lógica de negocio de la aplicación. Éstos son los de comunidad y de traducción directa, los cuales son una combinación servicios comunes, junto con una lógica específica para cada uno de ellos. Tanto los endpoints de seguridad como los de lógica de negocio pueden verse en la Figura 6.

4.1.1 Protocolo de comunicación

Como se dijo anteriormente, todo el protocolo de comunicación con la API del traductor se basa en métodos HTTP y URLs, permitiendo realizar diferentes acciones sobre el sistema. Para poder realizar cualquier acción sobre el sistema, se requiere estar debidamente autenticado y autorizado. Esto se logra haciendo una consulta al endpoint “/auth/jwt/token” (asumiendo que se está utilizando JWT) y enviando los siguientes datos:

- Método HTTP: POST
- Tipo de datos de petición: JSON
- Ejemplo de cuerpo de petición:

```
{
  "username": "testUser",
  "password": "testUserPassword"
}
```

- Tipo de datos de la respuesta: JSON
- Ejemplo del cuerpo de la respuesta:

```
{
  "refresh":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhc2RmYXNkZiIsIm5hbWUiOiJUZXR0IFVzZXIiLCJpYXQiOiJlMTYyMzkwMjJ9.tF5gsr0NcHmn-FJYM2yVZtWT6cn5sEBqTsUPJzu4w0Y",
  "access":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
}
```

Al obtener el token, se puede consultar a los diferentes endpoints y llevar a cabo diversas acciones dependiendo de cuál sea éste, y también dependiendo de qué usuario se trate y sus permisos en el sistema. Luego, se podría por ejemplo, configurar una red social; para esto se requiere enviar al endpoint “/community/configure/” una petición con el siguiente formato:

- Header HTTP: “Authorization: JWT
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c”
- Método HTTP: POST
- Tipo de datos de petición: JSON
- Ejemplo de petición:

En ambos casos, la documentación completa de los endpoints se encuentra en la documentación del sistema [47]. Allí se indica el método HTTP utilizado, los parámetros de entrada y de salida, y el tipo de contenido aceptado.

4.1.3 Endpoints de comunidad

Este conjunto de endpoints está enfocado en usuarios que tengan ingresada una comunidad en el sistema. Una comunidad refiere a un grupo de usuarios con los que alguien se desea comunicar frecuentemente.

Para enviar un mensaje no requiere indicar la red social a través de la cual se le enviará dicho mensaje al destinatario, debido a que internamente el sistema analiza la preferencia del destinatario para decidir el canal a través del cual será finalmente enviado. De esta forma, se desliga al emisor de la elección del canal, dándole esta responsabilidad al traductor. Esto busca dar respuesta tanto a los requerimientos R2 y R3 (indicados en la sección 3.4) al considerar tanto la comunicación transparente con las distintas redes sociales como la preferencia de medios del destinatario. En la sección 4.3 se explica cómo esta preferencia es calculada.

Los endpoints soportados junto con una breve descripción, se presentan a continuación. En todos éstos, el nombre de usuario entregado al endpoint debe calzar con el indicado en el token de acceso. Información más específica como datos de entrada, método HTTP, tipo de contenido de la solicitud, tipo de contenido de la respuesta y ejemplos de estos dos últimos casos, son provistos en la documentación de la API [47].

- `/community/configure/`: Este servicio permite configurar un canal específico para un usuario en particular. Podría requerir contraseña o código de acceso.
- `/community/sendMessage/`: Este servicio permite enviar un mensaje de texto y/o multimedia a un destinatario específico.
- `/community/check/networks/`: Este servicio permite chequear el estado de los canales de comunicación para un usuario específico. Es decir, permite saber si dicho canal está funcionando o no. En caso de que no lo esté, se puede utilizar el endpoint de `"/community/configure/"` para configurar dicha red social y que ésta pueda ser utilizada.
- `/community/check/messages/`: Este servicio permite chequear la existencia de nuevos mensajes para el usuario consultado. En caso de que existan, indica la cantidad de nuevos mensajes.
- `/community/getMessages/`: Este servicio permite obtener los nuevos mensajes que tenga un usuario. También se puede indicar aquellos mensajes que fueron recibidos correctamente, los cuales son eliminados de la base de datos.
- `/community/register/communication/`: Este servicio permite registrar la interacción entre dos usuarios por canales que no son mediados por el traductor, por ejemplo, las videollamadas por skype.

4.1.4 Endpoints de traducción directa

Este conjunto de endpoints (los de traducción directa) está enfocado en usuarios que tengan acceso al sistema, pero no tengan una comunidad configurada en dicho sistema. Permite que un usuario pueda enviar un mensaje a otro, pero dado que no tiene configurada una comunidad, no se guardan las interacciones, y por ende, no se puede analizar la preferencia de canal, lo cual obliga al sistema a especificar la red social por la cual el mensaje debe ser enviado, así como el usuario en dicha red social de la persona que quiere contactar. Es decir, estos endpoints están enfocados en permitir envío de mensajes de forma esporádica o unilateral, en caso contrario debe emplearse los endpoints anteriormente listados.

El propósito de estos endpoints es darle soporte a otro tipo de aplicaciones externas, cuya lógica de negocio esté enfocada más en enviar información a sus contactos, que en recibir información de éstos. De esta forma se busca cumplir de mejor manera el requerimiento R5 (indicado en la sección 3.4), al permitir un abanico más grande de posibilidades a las aplicaciones externas.

Además de esto, permite la recepción de mensajes callback desde un servidor de autenticación OAuth2, y también recibe las notificaciones push enviadas por los servicios soportados con dicha capacidad.

Los endpoints soportados junto con una breve descripción, se presentan a continuación. Al igual que en el caso anterior, información más específica como datos de entrada, método HTTP, tipo de contenido de la solicitud, tipo de contenido de la respuesta y ejemplos de estos dos últimos casos, son provistos en la documentación de la API [47].

- `/translator/configure/`: Este servicio permite configurar una red social para un usuario específico. Este último debe ser el nombre de usuario de la red social a configurar.
- `/translator/sendMessage/`: Este servicio permite enviar un mensaje ya sea de texto o multimedia desde un emisor a un receptor, a través de una red social específica. Cabe destacar que el emisor y el receptor deben ser nombres de usuario del canal por el cual se enviará el mensaje, y que debe haberse configurado previamente dicho canal.
- `/translator/auth/{social}/`: Este servicio permite recibir un callback de autenticación desde sistemas de autenticación que utilicen OAuth2 para realizar dicha labor, por ejemplo, Gmail.
- `/translator/push-notif/{social}/`: Este servicio permite recibir notificaciones push desde distintas redes sociales, las cuales se especifican en la URL reemplazando “{social}” por el nombre corto soportado.

4.2 Traducción de mensajes

Como se describió anteriormente, en la sección 1.2 del presente documento, el fin del sistema es manejar el dinamismo de los servicios provistos por las redes sociales. Es por esto que es

importante en el diseño de la solución que el sistema sea flexible al cambio en estas redes, ya sea para añadir, modificar o eliminar nuevos conectores.

Para lograr esto, a nivel de código se utilizó el patrón de fábrica (*Factory*), el cual se encarga de entregar el traductor correcto según la red social indicada, lo cual es definido por la preferencia de red del usuario en el caso de los endpoints de comunidad o la exigida en la petición HTTP en el caso de los endpoints de traducción directa. La estructura de clases del código se puede ver en la Figura 7.

En dicha figura, también se puede ver cómo todos los traductores implementan la misma interfaz de programación, de forma de tener un estándar de uso de cualquier traductor y abstraer las acciones específicas que requiere cada uno de éstos para lograr el resultado esperado (como por ejemplo, mandar una imagen). Esto permite encapsular la lógica de cada traductor y mantener el resto del sistema independiente de los canales de comunicación soportados, haciendo escalable el sistema ya sea ante la modificación, eliminación o adición de redes sociales. Esto va en directa relación al cumplimiento del requerimiento R4 (indicado en la sección 3.4), al dotar al sistema de flexibilidad respecto a las redes sociales soportadas.

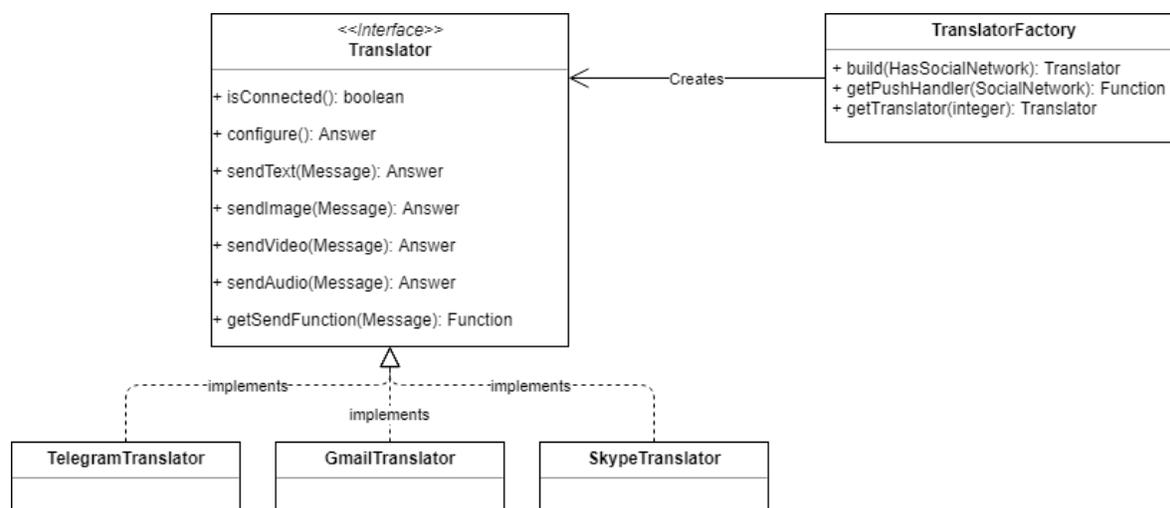


Figura 7: UML de fábrica de traductores

Las operaciones de envío de mensaje están asociadas a un objeto *Message* que, entre otras cosas, indica tanto el emisor, el receptor y la red social a través de la cual es enviado. Esto es guardado a nivel de base de datos, de forma de, por ejemplo, guardar los mensajes recibidos por un usuario a través de sus distintas redes sociales. La forma en que está guardado a nivel de modelo se puede ver en la Figura 8.

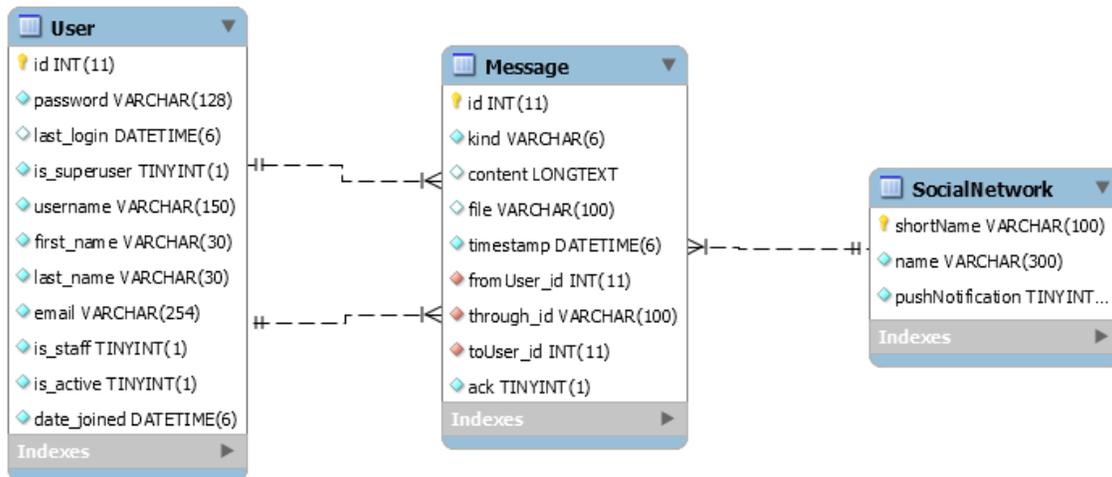


Figura 8: Modelo de envío y recepción de mensajes

Junto con esto es necesario soportar las redes sociales a nivel de base de datos, de modo de darles persistencia y poder enlazarlas con los usuarios. Para lo primero existe la tabla *SocialNetwork* que guarda todas aquellas redes sociales soportadas por el sistema; cabe destacar que si una red social no está ingresada en la base de datos, la aplicación no funcionará correctamente. Para lo segundo existe la tabla *HasSocialNetwork*, que indica la tenencia de una red social por parte de un usuario. Junto con esto, se guarda información respecto al nombre de usuario, alias y una flag que indica si el sistema debe guardar o no los mensajes de entrada. Todo esto puede verse en la Figura 9, que muestra estas entidades del modelo de datos junto a la entidad “usuario” y a sus relaciones.

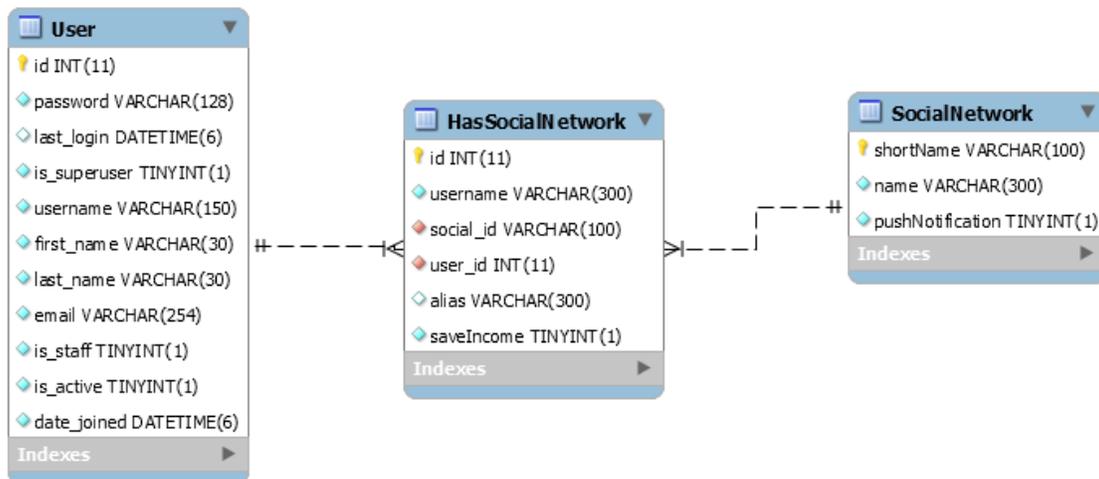


Figura 9: Modelo de relación entre usuarios y redes sociales

Como se mencionó anteriormente, el sistema no sólo debe encargarse de los mensajes enviados por un usuario, sino también de los mensajes recibidos por dicho usuario. De esta forma, se permite que no sólo exista una traducción de un protocolo estándar a los múltiples utilizados por la diversidad de redes sociales, sino también de esa misma diversidad hacia un

protocolo común, haciendo la traducción bidireccional. Es por esto que en la Figura 8 se puede ver que la tabla *Message* tiene un atributo llamado “kind”, el cual indica si dicho mensaje es de entrada o salida (I/O) respecto al sistema. Esta capacidad está relacionada con el requerimiento R2 indicado en la sección 3.4, ya que la comunicación bidireccional permite al sistema comportarse como cualquier servidor de chat tradicional, lo cual implica una comunicación asincrónica para el usuario final. Además, el soportar múltiples protocolos permite al sistema entregar la transparencia requerida por el usuario final.

Para la detección de nuevos mensajes, las dos técnicas usuales son las de pull y de push. En las primeras, la aplicación (en el presente trabajo, cada librería que hace de intermediaria) es la encargada de consultar a un servidor acerca de si ha habido cambios en los mensajes de un usuario (mensajes nuevos, modificados, eliminados, etc.). En las notificaciones push, el servidor de la red social es el encargado de avisar si hay cambios en los mensajes de un usuario específico.

Cada una de estas técnicas tiene sus ventajas y desventajas, y la posibilidad de implementarlas varía en cada red social. En particular, en el caso de pull, se requiere consultar constantemente al servidor de la red social por nuevos mensajes para cada usuario específico, lo cual implica uso de internet, procesamiento de información, consumo de RAM, etc. Por otro lado, este enfoque permite más control respecto a asegurar que la información no existe, y no es que haya habido algún problema con el envío o la recepción de la notificación push. Respecto a esta última, tiene el beneficio de ahorrar recursos al no requerir consultar a un servidor externo de forma constante, pero requiere abordar temas de arquitectura de redes, como contar con una IP/Dominio público a la cual pueda enviarse la notificación push. Esto genera mayores dificultades al hacer pruebas, y también puede existir la posibilidad de pérdida de dichas notificaciones por múltiples motivos.

Dado lo antes mencionado, se decidió que cada traductor debe encargarse de decidir y manejar el método de notificación. En el caso de las notificaciones de tipo push, es necesario que la aplicación reciba la llamada de la red social informando un cambio, por lo cual se tiene un endpoint (ver sección 4.1.4) que recibe la información entregada por una red social específica, y la reenvía de forma íntegra a la función del traductor correspondiente. Dicha función debe ser definida en el método `getPushHandler` de `TranslatorFactory` (ver Figura 7), el cual debe encargarse de obtener esta función dependiendo de la red social solicitada, de forma de saber cuál es el traductor correcto, y entregarla para que pueda ser ejecutada con la información provista por el servidor de la red social. Un ejemplo de este proceso puede verse en la implementación del conector de Gmail, el cual soporta notificaciones push para los correos electrónicos. Esto se hace con el fin de no restringir la funcionalidad de potenciales nuevas redes sociales soportadas por el sistema, y de esta forma, cumplir de mejor manera el requerimiento R4.

Junto con esto, todos los mensajes recibidos por cualquiera de estos métodos deben guardarse en la base de datos, hasta que éstos sean enviados al usuario correspondiente. Para controlar esto se usa el campo “ack” de la tabla *Message* (Figura 8), de forma de saber si cada mensaje fue recibido o no por el destinatario. Una vez recibido por el destinatario, el mensaje puede eliminarse de la base de datos, debido a que no son utilizados luego por el sistema. Esto se

hace también con un fin de seguridad, para no mantener mensajes privados en el sistema por más tiempo del necesario, y así protegerse de un posible ataque o fuga de información.

Por otra parte, la interacción entre los participantes debe ser guardada en la base de datos, de modo de saber por qué red social el emisor decidió contactar al receptor y así, poder ir aprendiendo las preferencias de usuario por una u otra red social a través del tiempo. Para esto se utiliza la tabla InteractionLog que se puede ver en la Figura 10. Esta tabla permite registrar una interacción entre dos usuarios, la red social por la que interactuaron, cuándo ocurrió dicha interacción, cuándo terminó dicha interacción (en caso de videollamadas), y qué tipo de mensaje fue enviado (texto, audio, video, etc.). Gracias a estos datos se puede hacer un análisis posterior de preferencia de uso de medios por los usuarios a medida que van utilizando la aplicación. Además, guardar esta información permitiría eventualmente hacer distinta clase de estudios respecto al comportamiento de los usuarios. En el caso de los adultos mayores podría ser interesante analizar la cercanía o lejanía con algún pariente, o determinar la sensación de soledad potencial del adulto mayor.

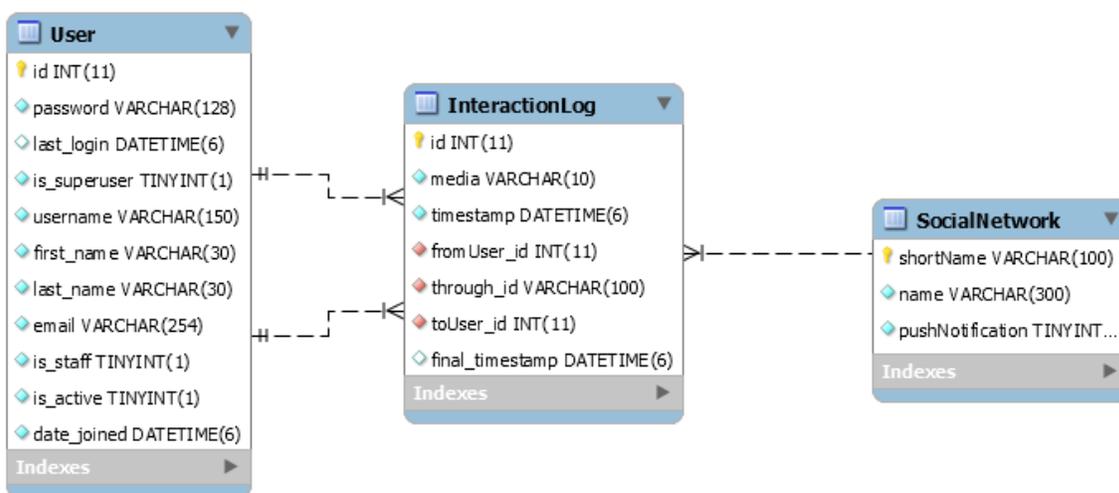


Figura 10: Modelo de interacción de usuarios

En la Figura 9, se puede ver que en ningún lado se guarda una clave para poder realizar la autenticación en las distintas redes sociales. Esto es debido a que el sistema creado no guarda las contraseñas de los usuarios, sino que debe recibirlas a través de un endpoint de configuración de red social, el cual se encarga de utilizar dicha clave para generar un token de acceso que luego es guardado en el sistema de archivos. La duración de estos token de acceso varía según la red social, en particular para las redes sociales soportadas actualmente, la duración puede verse en la Tabla 2.

Tabla 2: Duración de token para las redes sociales soportadas.

Red Social	Tiempo de duración de token de acceso
Telegram 	Indeterminado *
Gmail 	Indeterminado *
Whatsapp 	Indeterminado *
Skype 	24 horas

Nota: (*) Indeterminado debido a que el token es válido hasta que el usuario decida cancelarlo.

Es así como la aplicación cliente es la que debe encargarse, ya sea de solicitar cada cierta frecuencia la clave de acceso (para actualizar el token de Skype por ejemplo), o de guardarla de forma segura dentro de su aplicación. El sistema creado sólo debe encargarse de guardar dicho token y avisar a los clientes en caso de que haya una infiltración al sistema. Por ejemplo, en el caso de que alguien pueda haberse hecho pasar por el usuario, sin su real autorización.

4.3 Preferencias de redes sociales de los usuarios

El poder enviar un mensaje a la red preferente de un usuario es un requisito sumamente importante en el presente trabajo, ya que permite mejorar la comunicación entre el emisor y receptor de dicho mensaje, y además está directamente relacionado con el requerimiento R3 expresado por los usuarios encuestados (ver sección 3.4).

Esta preferencia no es sólo una relación entre un usuario y una red social, lo cual permitiría identificar la red social preferente de una persona, sino que además nos dice con qué red social prefiere enviar mensajes un emisor a un receptor específico. Para soportar este tipo de información, se utiliza la tabla *Preference* que se puede ver en la Figura 11. Esta tabla indica ambos interactuantes (usuarios), junto con la red social y la prioridad de este canal con respecto a los otros soportados.

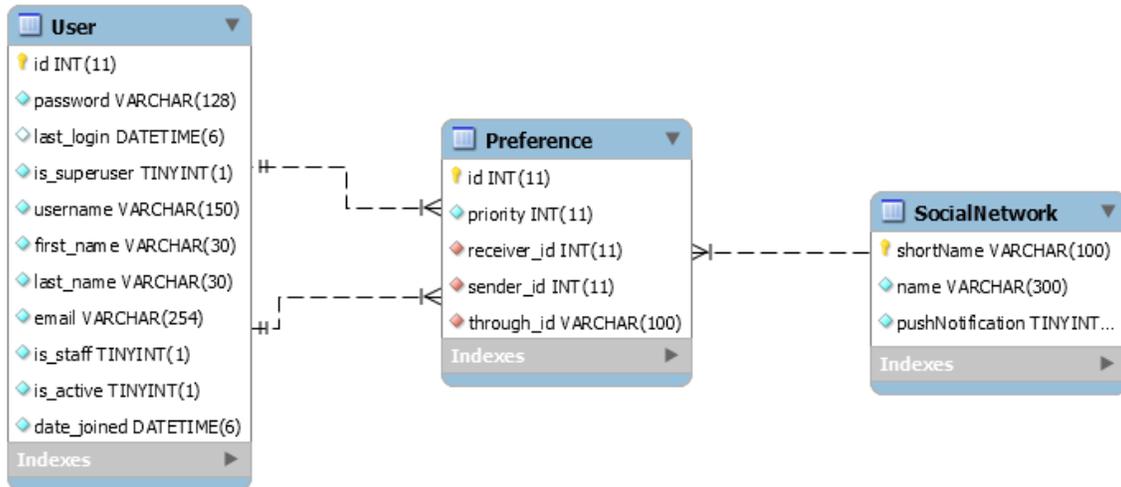


Figura 11: Modelo de preferencia de comunicación a través de redes sociales

Para identificar esta preferencia de canal (o red social), inicialmente el sistema se conecta a otro sistema desarrollado en el semestre de otoño 2017, en el curso de CC5401 - Ingeniería de Software 2 impartido en el Departamento de Ciencias de la Computación de la Universidad de Chile. De este sistema obtiene información de los miembros de una familia, sus usuarios en redes sociales, sus preferencias y sus relaciones de parentesco, todo esto a través de una API Web. Esta comunicación puede verse más claramente en la Figura 12 y una explicación detallada de cómo obtener la información de la API del sistema de preferencias puede verse en el Anexo B: Sistema de preferencias.

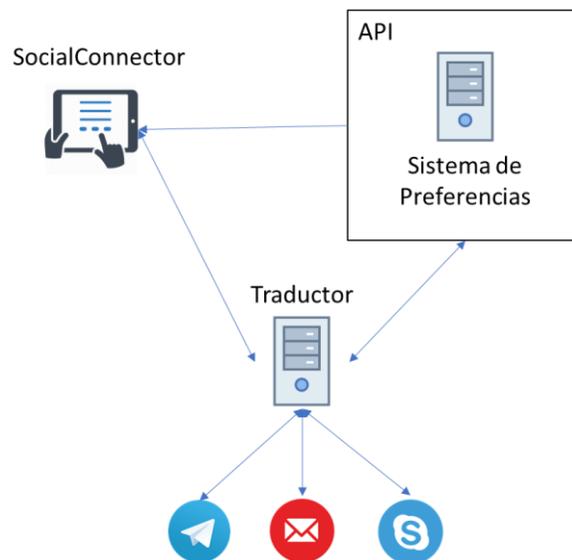


Figura 12: Escenario de comunicación con el sistema de preferencias de redes sociales (o canales)

Cabe destacar que debido al dinamismo que presentan dichas preferencias (por ejemplo, una persona deja de preferir una cierta herramienta, para pasar a preferir otra), este sistema externo sólo entrega información durante el inicio de nuestra aplicación, dado que las preferencias de uso de un canal se van modificando a través del tiempo, efecto causado por

el mismo dinamismo que afecta a las redes sociales. Es por esto que, ante el avance del tiempo, nuestra aplicación debe retroalimentar al sistema de preferencias externo, dotándolo de información más actualizada basada en el uso real de las distintas redes sociales.

Debido a los posibles cambios en el sistema de preferencias de los usuarios, y dado que éste no posee un mecanismo para enviar notificaciones de tipo push a nuestra aplicación, es necesario consultar periódicamente por cambios en la información otorgada. De esa manera, si se lleva a cabo un cambio en el sistema externo, el sistema actualizará automáticamente dicha información cuando llegue el deadline correspondiente. Dentro del mismo punto, es importante indicar que es necesario nutrir periódicamente al sistema externo con información actualizada respecto a las preferencias de usuario. De ese modo, ambos sistemas se mantendrán con la información actualizada, por lo cual cada vez que se determinen nuevas preferencias de uso para las redes sociales, se envía una actualización de esta información al sistema de preferencias de uso de estas redes.

Para determinar la preferencia del uso de una red social para la comunicación de un emisor con un receptor específico, se utiliza la información almacenada en la tabla *InteractionLog* (Figura 10), la cual permite saber cuándo se utilizó una red social en específico para llevar a cabo la comunicación entre los participantes. Dado que se tiene el registro de uso de cada una de las redes sociales en los distintos días del mes, se utilizó un modelo de pronósticos simple para estimar cuál será la red social que más se utilizará en el futuro (que representa la preferencia de dicha red). Debido a que el sistema es nuevo y no se tienen datos históricos, no se puede buscar un modelo que calce de forma más precisa con los datos empíricos, por lo cual se utilizó un modelo general bajo ciertos supuestos.

Uno de los supuestos utilizados es que existe una tendencia (ya sea al alza o a la baja) del uso de una red social, debido a que en el tiempo es usual que las redes se vuelvan más o menos populares, y por ende, lo mismo ocurre con su uso. Junto con esto, se espera una estacionalidad en los datos, de forma que exista cierta recurrencia en el comportamiento de la interacción de los usuarios durante los mismos días de una semana. Es decir, se espera un comportamiento similar durante los días lunes, por ejemplo, dado que las personas tienen cierta rutina en sus comportamientos semanales. Cabe destacar que este último supuesto puede llevar a errores, debido a feriados, inter-feriados, fecha de cumpleaños del receptor, o la ocurrencia de emergencias o catástrofes, entre otros.

Como método de pronóstico se utilizó el método de Holt-Winters con tendencia y estacionalidad [48], el cual es básicamente un modelo de suavizado exponencial triple. Es decir, que los datos más antiguos pesan menos que los más actuales, y además considera factores como tendencia en los datos y la estacionalidad. En particular se utilizó una estacionalidad aditiva y no multiplicativa, debido a que no se tienen datos para analizar este comportamiento y no se encontró tampoco ningún supuesto haga creer que la estacionalidad varía junto con los datos. Un ejemplo de ambas estacionalidades puede verse en la Figura 13.

Al utilizar este modelo, se puede pronosticar cual será la red que preferirá un usuario específico al comunicarse con otro, de forma de enviar los mensajes a la red preferente del destinatario. Esto con el fin de mejorar la experiencia del destinatario al considerar su predilección por una red social u otra.

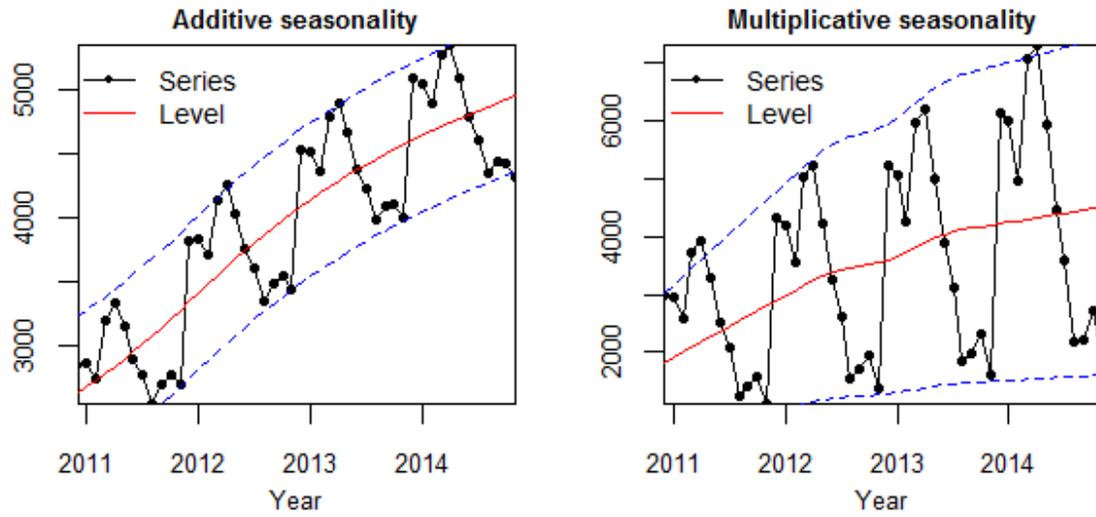


Figura 13: Ejemplo de método de Holt-Winters con tendencia y estacionalidad aditiva y multiplicativa. Fuente: <http://kourentzes.com/forecasting/2014/11/09/additive-and-multiplicative-seasonality/>

5 Implementación de la Solución

5.1 Descripción de la Aplicación Cliente: SocialConnector

Para este trabajo se utilizó el SocialConnector como aplicación cliente, debido a la cercanía con el equipo desarrollador y la posibilidad de introducir cambios en ella, debido a que es una aplicación del Departamento. Esta aplicación debió ser modificada para poder utilizar la API creada en el presente trabajo. A continuación, se describen los principales componentes del nuevo escenario que permite el funcionamiento conjunto, y los cambios que fueron necesarios realizar para poder conectar ambos sistemas.

5.1.1 Autenticación usando la API desarrollada

Para que el SocialConnector (o cualquier otra aplicación de terceros) pueda interactuar con las redes sociales regulares a través del traductor, lo primero que se necesita es crear una clase que refleje la conexión a la API a través de sus distintos endpoints. Para esto es necesario leer la documentación de dicha API, donde se especifica el tipo y la estructura de datos de entrada, así como los datos de retorno de cada endpoint. Sin embargo, esto no funcionará si en cada request no se le envía un token de autorización dado por la API. Para esto, se puede utilizar dos métodos de autenticación: OAuth2 o JWT. La aplicación cliente debe verificar cuál de éstos prefiere utilizar, y proceder en consecuencia. Para saber cómo utilizar estos métodos se debe leer la documentación de los endpoints de autenticación en la página de documentación del sistema [47].

Luego de obtener un token de acceso utilizando alguno de los dos métodos mencionados anteriormente, este token debe ser añadido en cada request realizado al sistema, ya sea como {Authorization: Bearer <access_token>} en caso de OAuth2, o bien como {Authorization: JWT <access_token>} en caso de que se haya decidido utilizar JWT. Así se podrán utilizar los distintos endpoints especificados en las secciones 4.1.3 y 4.1.4.

5.1.2 Configuración de las Redes Sociales

La aplicación cliente (en este caso, el SocialConnector) debe encargarse de mantener configuradas las conexiones con las distintas redes sociales que se pretenden utilizar. Para esto, el sistema soporta endpoints para chequear el estado de las conexiones, de forma que la aplicación cliente pueda informar o hacer labores por debajo (o sea, en background) para mantener estas conexiones activas.

Para lograr esto, se debe chequear el estado de ellas y en caso de que alguna no esté operativa, se debe solicitar al sistema creado (es decir, a la aplicación cliente) que configure dicha conexión. En caso de que el sistema no pueda realizar la labor, éste solicitará información adicional para poder habilitar la conexión con la red social en particular. Esta información puede ser la clave del usuario o un código de solicitud; esto varía según la red social de la cual se trate.

Dado que existe la situación en que la clave puede ser requerida con el fin de acceder a una red social particular, la aplicación cliente debe tener guardadas las claves del usuario o bien solicitarlas directamente al este último. En el caso de SocialConnector, la aplicación emplea ambos mecanismos. Debido a que Skype es la única red social soportada hasta el momento cuyo token de acceso vence de forma periódica, ésta es la única clave que el SocialConnector guarda en forma local. Esto se hace para evitar guardar claves del cliente, y así minimizar los riesgos de un potencial robo de claves o suplantación de identidad.

Para poder guardar la clave de forma segura en el dispositivo del usuario, se utilizó el sistema Android Keystore, el cual permite guardar de forma segura pares de claves pública y privada, lo cual hace difícil extraerlas para cualquiera, excepto para la aplicación que creó dichas claves. De esa forma, se puede guardar la llave de descryptación de la password del usuario, y guardar dichas password encriptadas en una base de datos simple.

Dada esa forma de funcionamiento, el SocialConnector chequea cada cierto tiempo el estado de las redes sociales. En caso de ser necesario, éste solicita información al usuario o la obtiene de la base de datos local, y además descrypta los datos utilizando las llaves guardadas en el sistema Keystore de Android [49]. En el caso de Skype es necesario hacer esta operación cada 24 horas, por lo que el SocialConnector realiza el chequeo cada 12 horas.

Debido a la posibilidad de un error de conexión en el momento de la verificación de las redes sociales, y para evitar que tengan que pasar 12 horas para que el sistema vuelva a tratar de configurar las redes, en caso de error de conexión durante este proceso se vuelve a verificar cada 1 hora en vez de 12. Esto se realiza una y otra vez hasta que el proceso se lleve a cabo de forma exitosa. Luego de esto, se vuelve al régimen usual del chequeo cada 12 horas.

5.1.3 Configuración de la comunidad del usuario

La comunidad del usuario no debe ser configurada a través de la aplicación Android del SocialConnector, sino a través de la aplicación Web desarrollada específicamente para tal fin. Esta aplicación cuenta con una API, a través de la cual el sistema creado se conecta con dicho sistema para poder realizar la carga de los grupos familiares, los cuales tienen acceso a la API traductora de mensajes.

Cabe destacar que, si el usuario del SocialConnector no se encuentra registrado en la aplicación Web, éste no podrá acceder al sistema creado debido a que este último obtiene los usuarios de esa aplicación Web. Por lo tanto, en caso de que haya un problema de autenticación de usuario, eso puede deberse a que no está la información del usuario en la aplicación Web, o a que la API aún no ha cargado la información al sistema.

5.1.4 Verificación, obtención y envío de mensajes

Luego de haber realizado todos los pasos anteriores, se podrá chequear la existencia de nuevos mensajes enviados por los miembros de la comunidad, obtener dichos mensajes y enviarles una respuesta. Cabe destacar que si trato de enviar mensajes a través de la API de comunidad (descrita en la sección 4.1.3) a alguien que no esté considerado dentro de dicha comunidad, el sistema creado enviará un código de error. El objetivo de esto es filtrar el

eventual spam que afecta a las redes sociales en general, y proteger a los adultos mayores de las interacciones ajenas a su comunidad.

El sistema debe chequear de forma constante los nuevos mensajes con el endpoint correspondiente, de esta forma se puede saber si hay mensajes nuevos sin tener que traerlos desde el servidor de forma inmediata. Cuando la aplicación indica que hay mensajes nuevos, se debe pedir al endpoint correspondiente que recupere y entregue dichos mensajes a la aplicación cliente. Es importante notar que si se vuelve a preguntar al sistema si hay mensajes nuevos, éste dirá que sí, hasta que no se le indique que se recibieron correctamente los mensajes que entregó previamente. Por lo tanto, se deben mandar los IDs de los mensajes recibidos, para indicar que éstos fueron recibidos correctamente y así poder marcarse como entregados, para que luego puedan ser eliminarlos del sistema intermediario (o sea, el traductor de mensajes).

Para enviar mensajes, se le debe indicar a la API quién es el usuario que envía el mensaje, quién lo debe recibir, y cuál es el contenido de éste. Un mensaje sólo puede tener un texto y/o un archivo. En el caso de querer enviar múltiples archivos, éstos se deben enviar como mensajes separados a través de la API creada en esta tesis. A su vez, para ambos usuarios se debe enviar el identificador que tiene en la aplicación Web de configuración familiar, ya que estos identificadores son los que guarda la API traductora de mensajes para poder realizar el tracking de las interacciones entre los usuarios.

5.2 Sistema traductor de mensajes

La implementación del prototipo del sistema traductor de mensajes sociales (SMT en inglés) desarrollado utilizando el lenguaje de programación Python, en conjunto con el framework Django y DjangoRestFramework. A continuación, se presentan los detalles específicos de la implementación de distintas secciones (o componentes) del software.

5.2.1 Canales de comunicación

Para conectarse a los distintos canales de comunicación se utilizaron diferentes librerías, las cuales requieren distintos mecanismos de configuración y habilitación de servicios. esto también lleva a que aparezcan diversas limitantes y diferencias debido a la escasa estandarización en la operación de estos canales.

Tal como se mostró en la Figura 7 (sección 4.2), todo traductor de un canal de comunicación específico debe implementar a la clase Translator. Además, dicho canal debe agregarse a TranslatorFactory de forma que pueda ser creado por la clase correspondiente, y finalmente, debe agregarse a la base de datos. Si alguna de estas tres cosas no es realizada, el sistema fallará.

Además de esto, cada uno de estos traductores específicos debe encargarse del manejo de sus tokens de acceso, de sus tests, y de sus clases o funciones auxiliares. Básicamente, cada traductor de un canal específico tiene gran nivel de autonomía acerca de cómo llevar a cabo su labor, siempre y cuando siga las especificaciones de su interfaz de programación. A

continuación, se describe brevemente las particularidades de la interacción con las distintas redes sociales (o canales soportados).

5.2.1.1 Interacción con Telegram

Telegram es una aplicación de mensajería instantánea multiplataforma y multidispositivo, según sus desarrolladores, está enfocada en la seguridad y la velocidad en el intercambio de mensajes. Esta herramienta posee una API pública y gratuita, lo cual permite utilizar sus servicios con pocas restricciones. Además, según algunas fuentes de información sobre uso de herramientas de mensajería instantánea, Telegram se encuentra entre las 10 aplicaciones de comunicación móvil más utilizadas en el mundo, con 200 millones de usuarios activos mensuales [50].

En el caso de la comunicación con la aplicación Telegram, se utilizó la librería Telethon [51], la cual permite abstraerse de las especificaciones de este canal, y utilizar directamente objetos y métodos de Python para realizar las distintas labores.

Para poder utilizar la API de Telegram, lo primero que debe hacerse es registrarse en la página oficial, ante lo cual se entregará un identificador y un hash los cuales autorizan el uso de la API por el sistema y a su vez, permiten a Telegram el tener claro las acciones que se están realizando por parte del sistema.

Telegram soporta el uso de token de acceso a la API, de forma de no tener que pedirle constantemente sus credenciales al usuario para acceder a su información. Dichos tokens pueden ser anulados por parte del usuario en cualquier momento, lo cual permite al dueño de la cuenta tener un nivel de seguridad adicional en caso de mal uso de esos permisos. Los tokens son guardados en el sistema de archivos del traductor, y son enviados en cada solicitud a la API de Telegram.

La reacción ante mensajes entrantes es llevada a cabo de forma autónoma por la librería, ya que ésta es la que se encarga de estar escuchando la recepción de nuevos mensajes o de otros eventos de interés para el traductor. De esta forma, a nivel de implementación sólo debe indicarse qué hacer ante la aparición del evento de interés específico.

5.2.1.2 Interacción con Gmail

Gmail es un servicio de correo electrónico gratuito proporcionado por Google, y es uno de los servicios de esta índole más conocidos. Gmail permite el acceso a su información, particularmente a los correos electrónicos asociados a una cierta cuenta de usuario, a través de una API pública y gratuita.

Para la comunicación con los servicios de Gmail se utilizaron las librerías que Google deja disponibles, con el fin de que los desarrolladores puedan llevar a cabo sus proyectos de intercambio de información con dicha herramienta. Para poder acceder a estos servicios, el desarrollador debe registrarse como tal, y así obtener acceso a las distintas APIs que disponibiliza Google. La mayoría de las indicaciones que muestran cómo utilizar Gmail, están documentadas utilizando ejemplo en distintos lenguajes, entre ellos Python.

Al realizar el proceso de intercambio de información a través de la API de Gmail, la aplicación cliente (o bien el servicio intermediario) deberá entregar las credenciales correspondientes, de forma de autenticar quién es el usuario que está tratando de acceder a los servicios de Google. Además de eso, es necesario que los usuarios Gmail permitan a la aplicación cliente leer sus correos y mandar correos en su nombre. En este caso, cuando un usuario desea configurar Gmail, se le entrega un link en el cual Google certifica que dicha persona desea entregarnos un token de acceso con los respectivos permisos. Una vez obtenido dicho token, éste se actualiza automáticamente y, al igual que en Telegram, puede ser cancelado por el usuario en cualquier momento que estime conveniente. Dicho token es guardado en el sistema de archivos, y enviado ante cualquier consulta a la API de Gmail.

Los nuevos mensajes recibidos, a diferencia de Telegram, son escuchados a través del servidor de notificaciones push de Google. Dicho servicio también debe ser configurado para poder utilizarse, y su uso requiere (entre otras cosas) un nombre de dominio y una IP pública, junto con un certificado que permita encriptar la comunicación. Si alguna de estas cosas no se cumple, no se puede habilitar el uso de este servicio.

Ante la llegada de un nuevo mensaje, Google manda una notificación a una URL predefinida. Así es posible agregar fácilmente nuevos traductores cuyo canal cuente con el servicio de notificaciones push. La URL predefinida tiene un formato predeterminado, el cual debe utilizarse de modo de no requerir añadir nuevas líneas de código a los elementos del sistema, que no sean el nuevo traductor en sí.

Debido a la limitación propia de los protocolos de correo electrónico, el tamaño de los archivos adjuntos es limitado, por lo cual, si se intenta mandar un archivo que supere ese tamaño máximo se generará un error en el sistema y el mensaje de correo no será enviado.

5.2.1.3 Interacción con Skype

La aplicación Skype, adquirida por la empresa Microsoft, permite la comunicación a través de texto, voz y video a través de Internet, ya sea entre personas individuales o grupos. Según Microsoft, más de 300 millones de usuarios a nivel mundial utilizan mensualmente esta herramienta [52]. Skype permite el uso de diferentes APIs, las cuales son públicas y gratuitas; sin embargo, para la comunicación interpersonal no existe una API oficial. Debido a eso, en este trabajo de tesis se utilizó la librería SkPy [53] desarrollada para Python, que realizó ingeniería inversa a la API utilizada por Skype en su versión Web. Esta librería permite utilizar clases y métodos de Python para utilizar Skype, proveyendo una capa de abstracción respecto a los endpoints específicos de su API.

Para utilizar dicha interfaz de programación, lo primero que es necesario hacer es autenticar al usuario, para lo cual se requiere su nombre de usuario y contraseña. Luego de esto, la API provee de un token de acceso para evitar tener que utilizar las credenciales del usuario de forma recurrente. Lamentablemente, a diferencia de Telegram, Whatsapp y Gmail, esta API no proporciona una forma de revalidar o refrescar dicho token, por lo cual después de su fecha de expiración (12 horas después de la solicitud) se debe volver a solicitar la clave y contraseña del usuario en cuestión.

Esto genera la necesidad de que cualquier aplicación cliente que desee utilizar Skype a través del SMT (es decir, el sistema traductor de mensajes sociales), enfrenta a grandes rasgos dos

opciones: (1) solicitar de forma reiterada al usuario su contraseña de Skype, o (2) guardar localmente la contraseña de dicho usuario para poder realizar la autenticación recurrente sin intervención de éste. Ambas opciones tienen distintos pros y contras. La última opción es la utilizada en el caso específico de SocialConnector para comunicarse con el SMT, utilizando el sistema keystore de Android [49] explicado anteriormente.

Una vez que el traductor de un usuario está instanciado, la librería se encarga de reaccionar ante la llegada de un nuevo evento, en nuestro caso esos son los de mensajes nuevos. Para ver qué eventos procesar, cada usuario se debe indicar cuáles le son relevantes, y luego el SMT utiliza una clase listener y un filtro para determinar cuáles eventos procesar y cuáles no. De esta forma, la implementación sólo debe encargarse de realizar lo que corresponda ante un evento específico, y no de realizar las acciones necesarias para identificar los distintos eventos generados por Skype.

La librería utilizada para interactuar con Skype presenta una limitante en el procesamiento de mensajes de voz enviados por los usuarios de esta red social (los mensajes de voz y los archivos de audios son considerados elementos diferentes ya que a nivel de interfaz se muestran de forma distinta por Skype). Los mensajes de voz son identificados por la librería al leer el mensaje entrante, pero ésta no permite recuperar el archivo de audio en sí. Por lo tanto, al recibir este tipo de mensajes el SMT le informa a la aplicación cliente que dichos mensajes no son soportados por el traductor. A su vez, los mensajes de voz son enviados por el SMT como simples archivos de audio a través de la librería a Skype. Para la recuperación y envío de archivos de audio convencionales la librería no presenta inconvenientes.

5.2.1.4 Interacción con WhatsApp

WhatsApp es una aplicación de mensajería instantánea adquirida recientemente por Facebook, la cual permite la comunicación a través de texto, mensajes multimedia (audios, fotos, videos, etc.), y llamadas de voz y video. Excepto estas últimas, todas las interacciones pueden realizarse tanto entre personas individuales, como entre grupos. En 2019 se estima que mensualmente más de 1500 millones de usuarios a nivel mundial usan la herramienta [50], lo cual es 7 veces más que telegram y 5 veces más que Skype, posicionándose así como la aplicación de mensajería instantánea más utilizada a nivel mundial. Esta aplicación no permite el uso de su API de forma oficial, de hecho, el uso no permitido de ésta podría implicar prohibición de uso de la aplicación de por vida.

Debido a las prohibiciones y al potencial peligro de utilizar la API de Whatsapp para los usuarios, se decidió utilizar la librería para Python WebWhatsappAPI [54] mantenida actualmente por un ex-alumno del Departamento de Ciencias de la Computación de la Universidad de Chile. Esta librería otorga una abstracción, a modo de API, de los servicios otorgados por la versión Web de Whatsapp [55]. Para lograr esto lo que hace la librería es levantar Selenium, que es un entorno de pruebas de software para aplicaciones basadas en la Web, el cual, dependiendo del método de la API invocado en Python, simula la realización de acciones sobre la aplicación Web. De esa forma se logra utilizar Whatsapp sin el riesgo de un mal uso de su API, ya que ésta no es utilizada.

El primer paso en la utilización de Whatsapp Web (y por ende, de la librería antes indicada) es enlazar el servidor con el teléfono, de modo que al primero se le otorguen los permisos para poder enviar los mensajes al segundo. Para lograr esto, la interfaz muestra un código

QR el cual debe ser escaneado por la aplicación Whatsapp del usuario; esto permite enlazar el servidor con el teléfono y autorizarlo. Para realizar esta acción, el traductor entrega una URL donde se puede ver la imagen del código QR para que pueda ser escaneado. Una vez escaneado, la aplicación es autorizada y dicha autorización no vence hasta que el usuario decida cancelarla, por lo tanto, a menos que se revoque dicho permiso, no será necesario escanear nuevamente un código QR.

Para la recepción de nuevos mensajes la librería no posee un método propio, por lo cual para realizar esta acción se levanta un proceso que verifica periódicamente si hay nuevos mensajes en la interfaz, en cuyo caso los guarda en la base de datos y los marca como leídos. Uno de los problemas de esto es que la librería, al marcar un mensaje como leído, no lo hace a nivel de la aplicación realmente, por lo tanto, si se reinicia el servidor, los mensajes no leídos podrían volver a procesarse. Para evitar esto se guarda en la base de datos los IDs de los mensajes ya leídos, evitando generar duplicados.

Finalmente, merece la pena comentar que, debido al funcionamiento de la librería, un pequeño cambio de la interfaz de Whatsapp Web puede implicar el cese de funcionamiento de la API. Por lo tanto, la necesidad de mantención de la librería es potencialmente muy alta, lo cual puede traer problemas de uso en el mediano y largo plazo. Lamentablemente, no existen otras opciones en este momento para poder utilizar Whatsapp de forma segura y transparente por los usuarios finales.

5.2.2 Administración de preferencia de redes sociales

Como se dijo anteriormente en el capítulo 4, el sistema administrador de preferencias es el encargado de guardar y analizar las preferencias de un usuario específico, para comunicarse con un miembro de su comunidad a través de un canal de comunicación definido. Es por esto que en el cálculo de la preferencia de cada canal incide directamente la opinión del emisor y del receptor de un mensaje.

El sistema soporta el ingreso de una preferencia inicial; sin embargo, en las operaciones sucesivas realizadas a lo largo del tiempo, esta preferencia se irá ajustando dependiendo de las interacciones que realicen los usuarios a través del sistema. En particular, para calcular la preferencia de un canal se utilizan los registros de comunicación de los últimos 365 días (modificable en la configuración del sistema) entre dos usuarios específicos, y se asume una estacionalidad fija de una semana.

Estos registros son agrupados por día, ante lo cual se obtienen 365 valores indicando cuántas veces hubo interacción entre los dos usuarios a través del canal analizado. Luego se procede a realizar el pronóstico, el cual se hace para los siguientes 30 días (parámetro configurable en el sistema) utilizando el método de Holt-Winters con tendencia y estacionalidad. Después se suma la cantidad de interacciones de esos 30 días y se ordenan los canales de comunicación en base al valor obtenido, estableciendo así una preferencia para ese par de usuarios.

Para ejecutar este proceso de actualización de preferencias, está disponible un comando que realiza distintas acciones. La persona que implanta el sistema debe asignar la labor de correr la tarea de forma periódica, para lo cual se cuenta con un cron (también desarrollado en el marco de esta tesis). Idealmente, este proceso se debe llevar a cabo cada mes, para que las

preferencias se vayan adecuando a los cambios en las conductas de comunicación entre los distintos miembros de una comunidad.

5.2.3 Interacción con sistemas externos

En particular, para poder enlazar el SMT con el SocialConnector se utilizó una aplicación ya existente que guardaba información de la familia de un adulto mayor (es decir, un usuario del SocialConnector), junto con sus redes sociales, usuarios y preferencia de canales. De esa forma es posible cargar automáticamente estos datos, y no es necesario que la aplicación móvil provea dicha información.

Para mantener el diseño de la aplicación mostrado en el capítulo 4, se implementaron servicios que realizan la labor de un ETL (Extraction, Transformation and Load); es decir, extraen información del servicio externo, la transforman, y finalmente, la cargan al sistema. En particular lo que se hizo fue extraer los datos y transformarlos para que calzaran con los servicios que ya existían en el sistema, de forma de reutilizar la lógica de negocio y asegurarse de que esta no se encuentra repartida en el sistema.

Debido a que no se esperaba generar múltiples ETLs para diversos sistemas externos que se quisieran conectar, el diseño de la aplicación no está particularmente pensado para soportar esta funcionalidad, por lo que la solución no es muy extensible en este sentido. Esto es así ya que se espera que aquellas aplicaciones que se quieran conectar al SMT, utilicen directamente los endpoints y las interfaces de programación de éste para realizar las labores necesarias para conectar ambos servicios.

6 Evaluación de traductor de mensajes

La evaluación del traductor de mensajes involucró tres aspectos: (1) determinar la correctitud del mecanismo de traducción de mensajes, (2) revisar el nivel de seguridad del protocolo de intercambio de mensajes, y (3) determinar la usabilidad de la API según los potenciales usuarios (es decir, desarrolladores de sistemas de software sociales). El primer tipo de evaluación permite mostrar el nivel de correctitud del traductor, no sólo para traducir mensajes, sino también para seleccionar los canales apropiados para distribuirlos. Esto último debe considerar el canal escogido por los usuarios para llevar a cabo las interacciones, y también las preferencias de canal de dichos usuarios.

El segundo tipo de prueba muestra que, si bien el sistema no es invulnerable, no es particularmente más vulnerable que la mayoría de las herramientas de interacción social disponibles para uso cotidiano. Finalmente, el último tipo de pruebas se enfocó en determinar la usabilidad de la API desarrollada. Particularmente, se buscó establecer si la API era suficientemente usable (comprensible), como para que desarrolladores acostumbrados a trabajar con este tipo de interfaces de software, pudieran usarla de forma autónoma (sin apoyo externo). Es importante recordar que la utilidad de los servicios que provee la API ya fue evaluada a través del estudio confirmatorio presentado en el capítulo 3. A continuación se detallan las evaluaciones hechas en los tres ámbitos indicados, así como los resultados obtenidos.

6.1 Evaluación de la correctitud de la solución

6.1.1 Diseño experimental

Esta evaluación consideró 40 escenarios de prueba, un mayor detalle de éstos se puede encontrar en el Anexo C: Resultado de las pruebas de traducción de mensajes. Los escenarios consideran la participación de cinco usuarios distintos, y cuatro herramientas de interacción social: Skype, email, Whatsapp y Telegram. Los usuarios 1 a 4 (U1 a U4) corresponden a los miembros de la comunidad familiar del adulto mayor, los cuales utilizan las versiones regulares de las herramientas de interacción social. El usuario 5 (U5) es el adulto mayor, que utiliza el SocialConnector System (SCS), e indirectamente usa el SMT (Social Message Translator) para interactuar los otros cuatro usuarios, tal como se muestra en la Figura 14.

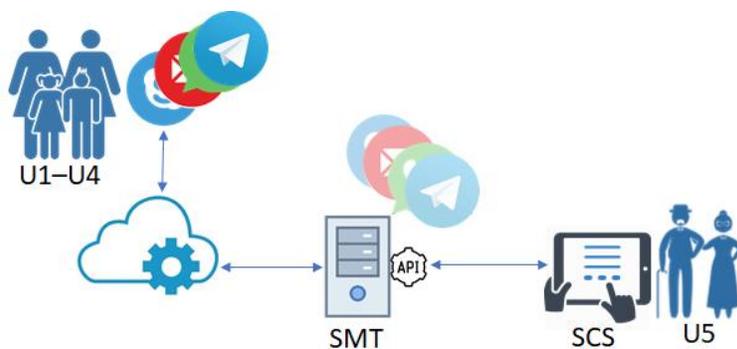


Figura 14: Estructura del escenario simulado para pruebas de correctitud.

El orden de preferencia de herramientas de cada uno de los usuarios puede verse en la Tabla 3. Como se mencionó anteriormente, U5 sólo usa el SCS (e indirectamente el traductor de mensajes).

Tabla 3: Orden de preferencia de herramientas de interacción social.

	1° Preferencia	2° Preferencia	3° Preferencia	4° Preferencia
Usuario 1	Skype	Email	Telegram	Whatsapp
Usuario 2	Email	Whatsapp	Skype	Telegram
Usuario 3	Telegram	Skype	Whatsapp	Email
Usuario 4	Whatsapp	Telegram	Email	Skype
Usuario 5	SCS			

Los mensajes que se intercambian entre los usuarios son de dos tipos: de texto y multimedia. Los primeros contienen sólo texto, y los segundos son iguales a los primeros, pero incluyen una imagen (500KB) o video (5MB) como archivos adjuntos.

Las pruebas 1 a la 16 refieren a la recepción de mensajes por parte de U5; los mensajes son enviados según las preferencias de los emisores. En particular, en las pruebas 1 a 4 y 9 a 12, los mensajes son enviados según la primera preferencia del emisor, y en las pruebas 5 a 8 y 13 a 16, los mensajes se envían a través de su segunda preferencia.

Las pruebas 17 a 32 involucran el envío de una respuesta a mensajes anteriores (enviados en las pruebas 1 a 16). Estas respuestas son enviadas antes de cumplir 24 horas luego de que se recibió el mensaje anterior. En este caso, el Traductor envía la respuesta a través del mismo canal por el cual se envió el mensaje anterior. Cabe recordar que este tiempo de 24 horas de espera por una respuesta a un mensaje, es configurable por el administrador del sistema.

Las pruebas 33 a 36 implican el envío de mensajes de respuesta, una vez vencido el período de 24 horas desde el envío anterior. En este caso, para el envío del mensaje se utiliza la primera prioridad de preferencia de canal que tenga el destinatario. Éste es también un comportamiento esperado por parte del traductor.

Las pruebas 37 a 40 son similares a las cuatro pruebas anteriores, con la salvedad de que posterior al período de 24 horas, el usuario destinatario se cambió su preferencia de canal para recibir mensajes. En el caso de estas pruebas, este cambio consistió en intercambiar la primera y segunda preferencia de los usuarios, cambiando así su red preferente.

Por otro lado, en las últimas ocho pruebas se utilizaron nuevas versiones de los traductores, esto con el fin de tratar de simular el proceso de añadir o modificar lógica a los traductores en SMT, dado que esa es la forma de evolución natural de estos módulos del sistema. En todas estas pruebas, tres variables fueron observadas: (1) la correctitud de los mensajes resultantes, (2) el tiempo transcurrido para que el mensaje llegue al destinatario, y (3) la correctitud en la selección del canal por parte de SMT para entregar cada mensaje.

6.1.2 Resultados Experimentales

La tabla 4 muestra un resumen con los resultados obtenidos en los 40 escenarios de prueba planteados anteriormente. Allí se especifican los datos del emisor, receptor, los canales de envío y recepción, los tiempos de transmisión y la correctitud en la selección del canal de envío para los distintos escenarios considerados.

Estos resultados muestran claramente que todos los mensajes fueron enviados correctamente a sus destinatarios según el diseño del traductor (ver área rotulada como “1” en la Tabla 4); es decir, el envío de mensajes respeta la preferencia del receptor, pero considera como prioritario el responder a través del canal por donde recibió el último mensaje. Esto permite ver que el requerimiento R1 expresado por los usuarios en la sección 3.4 se cumple, al enviar todos los mensajes de forma correcta a cada una de las redes sociales soportadas.

El tiempo requerido para el envío de mensajes mostró ser aceptable para lograr una comunicación asíncrona entre los interactuantes, dado que es similar a los encontrados en el uso diario de las aplicaciones; por tanto, el uso del traductor no afecta de forma notoria la experiencia de los usuarios finales.

También se puede ver que el tiempo de envío se ve más afectados por la calidad de la conexión a Internet, y las políticas de procesamiento de las diversas redes sociales, que por el procesamiento interno del Traductor. Esto puede verse por ejemplo en las pruebas 9 y 15, donde el recibir un mensaje de 5MB demora menos tiempo que uno de 500KB. Esto se debe a que dicho canal (Skype), por política interna, comprime los mensajes pesados antes de enviarlos a su destino, lo cual explica los resultados obtenidos. Un proceso similar ocurre con la aplicación Telegram, la cual también comprime los archivos antes de enviarlos, por lo cual los tiempos no son proporcionales al tamaño de los archivos.

El tiempo de conexión entre el servidor del canal y el Traductor también juega un papel importante, pues genera interferencia en el proceso, lo cual puede ser visto mediante las pruebas 18, 21, 26, 29, 34 y 37. En esas pruebas, el tiempo para mandar exactamente el mismo mensaje de 500kb a través de Gmail varía entre los 1,2 y los 4,4 segundos ($\sigma = 1,2$). Esto ocurre de forma similar en las otras redes sociales, como Telegram (en las pruebas 19, 24, 27, 32, 35, 40) y en Skype (en las pruebas 17, 23, 25, 31, 33, 39), donde los tiempos varían en el rango de 0,4 - 4,3 segundos ($\sigma = 2,2$) y entre 1,2 - 4,3 segundos ($\sigma = 1,3$)

respectivamente. Cabe destacar también que este patrón no se presenta en el caso de envío de archivos multimedia más pesados (5MB), ni tampoco en el caso de Whatsapp. En el caso de Whatsapp esta situación no se presenta debido a que la librería utilizada mantiene una sesión abierta con Whatsapp web (Cliente Web), haciendo más expedito el proceso de envío de mensajes.

Tabla 4: Resumen de la prueba de traducción de mensajes realizada

Escenario #	Usuarios Involucrados		Canales Involucrados		Resp. a Escenario	Tiempo de Transmision			Correctitud de Canal	
	Emisor	Receptor	Canal de Envío	Canal de Recepción	Respuesta #	Msg de texto	Txt + Pict (500 kB)	Txt + Video (5 MB)	Correctitud de Canal	Correctitud de Mensaje
1	U1	U5	Skype *	ClientApp	-	1 s.	1.9 s	-	✓	✓
2	U2	U5	Gmail *	ClientApp	-	1	2.8	-	✓	✓
3	U3	U5	Telegram *	ClientApp	-	1	0.9	-	✓	✓
4	U4	U5	Whatsapp *	ClientApp	-	8	12	-	✓	✓
5	U1	U5	Gmail	ClientApp	-	1	6.1	18.7 s	✓	✓
6	U2	U5	Whatsapp	ClientApp	-	10	11	-	✓	✓
7	U3	U5	Skype	ClientApp	-	1	1.8	-	✓	✓
8	U4	U5	Telegram	ClientApp	-	1	0.9	-	✓	✓
9	U1	U5	Skype *	ClientApp	-	-	4.7	4	✓	✓
10	U2	U5	Gmail *	ClientApp	-	-	4.9	11.9	✓	✓
11	U3	U5	Telegram *	ClientApp	-	-	2.8	6	✓	✓
12	U4	U5	Whatsapp *	ClientApp	-	-	10	28	✓	✓
13	U1	U5	Gmail	ClientApp	-	-	5.1	14.7	✓	✓
14	U2	U5	Whatsapp	ClientApp	-	-	12	34	✓	✓
15	U3	U5	Skype	ClientApp	-	-	4.1	4	✓	✓
16	U4	U5	Telegram	ClientApp	-	-	1.9	4.8	✓	✓
17	U5	U1	ClientApp	Skype *	1	1	1.2	-	✓	✓
18	U5	U2	ClientApp	Gmail *	2	1	1.2	-	✓	✓
19	U5	U3	ClientApp	Telegram *	3	1	0.4	-	✓	✓
20	U5	U4	ClientApp	Whatsapp *	4	1.5	4	-	✓	✓
21	U5	U1	ClientApp	Gmail	5	-	2.9	6.9	✓	✓
22	U5	U2	ClientApp	Whatsapp	6	-	3.9	8.3	✓	✓
23	U5	U3	ClientApp	Skype	7	-	4.3	7.2	✓	✓
24	U5	U4	ClientApp	Telegram	8	-	4.3	34.8	✓	✓
25	U5	U1	ClientApp	Skype *	9	-	3.6	6.3	✓	✓
26	U5	U2	ClientApp	Gmail *	10	-	4.1	6.1	✓	✓
27	U5	U3	ClientApp	Telegram *	11	-	4.6	34.7	✓	✓
28	U5	U4	ClientApp	Whatsapp *	12	-	4.1	7.5	✓	✓
29	U5	U1	ClientApp	Gmail	13	-	2.6	6	✓	✓
30	U5	U2	ClientApp	Whatsapp	14	-	4.4	8	✓	✓
31	U5	U3	ClientApp	Skype	15	-	3.7	10.2	✓	✓
32	U5	U4	ClientApp	Telegram	16	-	4.4	36.5	✓	✓
33	U5	U1	ClientApp	Skype * +	5 delayed	1	1.7	-	✓	✓
34	U5	U2	ClientApp	Gmail * +	6 delayed	1	1.4	-	✓	✓
35	U5	U3	ClientApp	Telegram * +	7 delayed	1	0.4	-	✓	✓
36	U5	U4	ClientApp	Whatsapp * +	8 delayed	1.3	4.5	-	✓	✓
37	U5	U1	ClientApp	Gmail ** +	9 delayed	1	1.2	-	✓	✓
38	U5	U2	ClientApp	Whatsapp ** +	10 delayed	1.2	4.4	-	✓	✓
39	U5	U3	ClientApp	Skype ** +	11 delayed	1	1.6	-	✓	✓
40	U5	U4	ClientApp	Telegram ** +	12 delayed	1	0.5	-	✓	✓

Nota: () Canal preferido por el destinatario, (**) Nuevo canal preferido después de un cambio de preferencia, (+) Una nueva versión del traductor fue utilizada.*

En la Tabla 4 se puede apreciar como los tiempos de procesamiento de mensajes recibidos (pruebas 1 a 16) son similares a los de envío (pruebas 17 a 40), exceptuando en el envío de archivos grandes, donde actúan políticas propias de las redes sociales (como la compresión de archivos grandes vista anteriormente). Estos números también muestran las diferencias entre las distintas redes sociales a la hora de enviar un mismo mensaje. Es por esto que se

puede esperar que el tiempo extra requerido por el traductor para procesar los mensajes, tanto de envío como de recepción, sea casi imperceptible para los usuarios finales, siendo inferior a un segundo en la mayoría de los casos.

En resumen, los resultados muestran que los mensajes enviados por el traductor son correctamente ruteados al canal que corresponde, y traducidos apropiadamente para que estos puedan ser interpretados por dicho canal. También se muestra que la demora producida por el procesamiento de los mensajes no afecta la experiencia de usuario; al menos no más que el uso regular de las herramientas involucradas en las interacciones. Finalmente, los resultados también muestran el sistema es capaz de manejar cambios en las preferencias de los usuarios, y en la implementación de los traductores.

6.2 Evaluación de la seguridad de la solución

Para evaluar la seguridad de la solución se plantearon distintos escenarios hipotéticos, y para cada uno de ellos se determinó el nivel de vulnerabilidad del sistema. Los escenarios involucran al servidor, a la aplicación cliente, a la comunicación entre el cliente y el servidor. Para cada escenario se consideró una amenaza específica y se determinó el nivel de vulnerabilidad del sistema para ese escenario y esa amenaza. A continuación, se detalla cada uno de ellos.

6.2.1 Escenario 1: Amenazas en el servidor de aplicación

Escenario 1.1 Ataque de denegación de servicio (DOS)

Descripción: Este tipo de ataque se caracterizan por inundar al servidor con peticiones con el fin que éste deje de funcionar al no poseer los recursos suficientes para poder responder tantas peticiones de forma simultánea.

Posibilidad de ocurrencia: Media, no requiere conocimientos o máquinas especiales para poder realizarlo, pero se requiere un incentivo.

Daño potencial: Bajo, implica la incapacidad temporal por parte del servidor para realizar cualquier acción, pero no fugas de información sensible.

Acciones paliativas: Las acciones orientadas a evitar este tipo de ataque van principalmente por el lado de la inclusión de componentes de infraestructura, a través de los cuales se busca bloquear las direcciones IP de los atacantes (por ej. mediante un firewall). Por esto, la aplicación fue implantada dentro del DCC, y de esa manera cuenta con la seguridad a nivel de infraestructura implementada por el Departamento.

Escenario 1.2 Escalamiento de privilegios

Descripción: Este tipo de ataque busca lograr obtener acceso al servidor o incrementar los privilegios que se tienen en éste, de forma de acceder al sistema de archivos y así, a la información sensible contenida.

Posibilidad de ocurrencia: Baja, pues requiere errores de implementación por parte del área de infraestructura, o un mal resguardo de credenciales de los administradores o encargados del servidor.

Daño potencial: Alto, puede implicar la obtención de los tokens de acceso de todos los usuarios del sistema a las distintas redes sociales, además de credenciales de acceso al servidor de la BD o a servicios externos (Gmail o Telegram).

Acciones paliativas: Permitir el acceso exclusivamente a través de canales cifrados (uso de SSH) e implantación dentro de la infraestructura del DCC, de forma de asegurar estándares de calidad.

6.2.2 Escenario 2: Amenazas a la comunicación cliente-servidor

Descripción: El tipo de ataque más típico en este escenario son los denominados “Man-in-the-middle” (MITM), los cuales permiten, leer, interceptar o inyectar nuevos mensajes ya sea hacia el emisor o receptor.

Posibilidad de ocurrencia: Media, requiere errores de implementación por parte del área de infraestructura de donde esté implantado el servidor.

Daño potencial: Medio, puede implicar el envío de mensajes maliciosos haciéndose pasar por el usuario o leer información sensible contenida en los mensajes intercambiados.

Acciones paliativas: Realizar la comunicación exclusivamente a través de canales cifrados (uso de HTTPS) e implantación dentro de la infraestructura del DCC, de forma de asegurar estándares de calidad.

6.2.3 Escenario 3: Amenazas en la aplicación cliente

Escenario 3.1. Keylogger

Descripción: Este escenario es cuando se tiene un software malicioso dentro del cliente (como su teléfono móvil), el cual se encarga de mantener un registro de todos los valores insertados por el usuario.

Posibilidad de ocurrencia: Media, pues requiere que el usuario ingrese a sitios maliciosos o descargue elementos maliciosos provenientes de fuentes como email.

Daño potencial: Medio, puede implicar la obtención de la clave del usuario o de los mensajes de texto enviados por éste.

Acciones paliativas: Ninguna, es responsabilidad del usuario mantener buenas prácticas de navegación en la web.

Escenario 3.2. Suplantación de identidad

Descripción: Este escenario ocurre cuando un malware logra acceder a la información de la aplicación cliente y obtiene el token de acceso para el traductor, permitiendo así leer o enviar mensajes haciéndose pasar por el usuario.

Posibilidad de ocurrencia: Baja, en Android implica lograr sobrepasar las barreras de seguridad implementadas por el sistema operativo para las aplicaciones corriendo sobre él.

Daño potencial: Medio, puede implicar el envío de mensajes maliciosos haciéndose pasar por el usuario o leer información sensible contenida en los mensajes intercambiados.

Acciones paliativas: Ninguna, es responsabilidad del usuario mantener buenas prácticas de navegación en la web.

Escenario 3.3. Acceso no autorizado

Descripción: Este escenario es cuando una persona distinta al usuario logra obtener acceso a la aplicación cliente, pudiendo por ende realizar cualquier acción como si el usuario se tratara.

Posibilidad de ocurrencia: Media, en Android implica lograr sobrepasar las barreras de seguridad implementadas por el sistema operativo u obtener las credenciales de acceso del usuario (contraseña, patrón, huella digital, etc.).

Daño potencial: Medio, puede implicar el envío de mensajes maliciosos haciéndose pasar por el usuario o leer información sensible contenida en los mensajes intercambiados.

Acciones paliativas: Ninguna, es responsabilidad del usuario mantener de forma segura sus credenciales de acceso a los dispositivos o de la aplicación cliente el mantener medidas para asegurar que el usuario este autorizado.

6.3 Evaluación de la usabilidad de la API

A continuación, se describe el diseño experimental utilizado para evaluar la usabilidad de la API con desarrolladores. Luego se presentan y discuten los resultados obtenidos.

6.3.1 Diseño Experimental

En el proceso de evaluación de la usabilidad de la API participaron 11 expertos en desarrollo de software, quienes trabajan en la industria. Todos ellos con formación universitaria completa y amplia experiencia en el uso de este tipo interfaces de programación de aplicaciones (APIs). Los expertos poseían en promedio 13,5 años de experiencia, con una desviación estándar de 8,5; además, sus años de experiencia en APIs web específicamente era en promedio de 5,6 y su desviación estándar de 3,7. La evaluación se llevó a cabo en distintas sesiones, y en cada sesión participaron 1 o 2 expertos. Las sesiones involucraron varias actividades, donde la última actividad fue la realización de una entrevista semi-estructurada para capturar sus opiniones. A continuación, se explica la secuencia de actividades realizadas en cada sesión.

- *Explicación introductoria:* Dado que los desarrolladores no necesariamente estaban familiarizados con el desarrollo de software que involucraba apoyo a la interacción asimétrica entre los usuarios, se les entregó una breve explicación del objetivo general de la API, y del escenario general en la que ésta brinda mayor valor. Dicha explicación tomó alrededor de 5 minutos e involucró una diapositiva.
- *Revisión breve de la API:* Con el objetivo de brindarles una visión panorámica de los servicios que ofrecía la API, se les mostró una diapositiva que indicaba las categorías de servicios implementados, y los nombres de los servicios específicos que estaban disponibles al interior de cada una de ellas. Esto involucró el uso de una diapositiva y una explicación de 3 minutos aproximadamente. Además, se les entregó a los participantes una hoja impresa con el resumen de los servicios provistos por dicha interfaz de programación, según lo indicado en la documentación de la misma [47].
- *Realización de un ejercicio de modelamiento:* A cada participante se le pidió que identificara los servicios de la API que utilizaría, así como la secuencia de estos, en el caso de tener que: (1) enviar un mensaje de texto desde una aplicación cliente a una red social comercial, y (2) hacer que la aplicación cliente recupere un mensaje desde una red social comercial. En ambos casos los participantes debían utilizar como intermediario el traductor desarrollado, y por lo tanto, debían indicar la secuencia de servicios que se requería invocar para realizar dichas operaciones. Esta actividad tomó en promedio 2 minutos, y el resultado que cada participante debía entregar era la secuencia de servicios a invocar en cada una de las dos operaciones a implementar.
- *Revisión de la documentación de la API:* Se les pidió a los participantes que revisaran la documentación de la API, para hacerse una idea del nivel de complejidad que tenía el uso de los servicios ofrecidos. Se les pidió particularmente que revisaran la

complejidad de agregar el soporte para nuevas plataformas sociales, como una forma de ver cuán extensible se percibía la API.

- *Realización de una entrevista semi-estructurada:* Una vez concluido el ejercicio y la revisión de la documentación de la API, se les hizo una entrevista semi-estructurada a los participantes, con el fin de determinar qué tan usable era la API según la percepción de estas personas. Dado que típicamente el uso de estas interfaces de programación tiene una complejidad inherente que es no menor, y que todos los participantes eran experimentados en el uso de ellas, la evaluación buscó también determinar si la API desarrollada era más o menos compleja de usar que una API típica. El formulario guía de la entrevista está disponible en el Anexo D: Pauta de entrevista a desarrolladores.

6.3.2 Resultados Obtenidos

En el caso de la actividad de identificar los endpoints para lograr las 2 acciones propuestas (enviar un mensaje y obtener los mensajes recibidos), la totalidad de los entrevistados logró indicar una sucesión de pasos que permitiera lograr cada una de las actividades. Definir ambas secuencias de invocaciones de servicios les tomó un tiempo promedio de 2 minutos ($\sigma=1.2$). Lo cual indica que la explicación y descripción de los endpoints es suficientemente entendible como para permitir una correcta elección de éstos.

En la entrevista semi-estructurada se le preguntaron varias cosas orientadas a determinar la usabilidad y utilidad de la API. La Tabla 5 resume las opiniones de los participantes para cada pregunta.

Gracias a este proceso de entrevista se puede decir que la mayoría de los entrevistados consideró que la documentación de la API era lo suficientemente usable, como para que un desarrollador pudiera entenderla y utilizarla, mostrando que no existe una gran barrera de entrada para desarrollar aplicaciones utilizando esta API. Además, indicaron que, en comparación con la documentación de otras APIs Web con las que han trabajado, SMT Web API presentaba una calidad de documentación similar o superior a las anteriores.

Por otro lado, la mayoría indicó que podría agregar un nuevo canal de comunicación a la aplicación. Esto indica que la documentación de la API permite a un desarrollador entender cómo funciona el sistema de traducción, y cómo lograr extenderlo, lo cual es fundamental en este tipo de aplicación donde se lucha en contra de los constantes cambios introducidos en los canales de comunicación soportados.

Respecto a la existencia de un software similar, y a si conocían otra forma de hacer la traducción de mensajes, la mayoría de los entrevistados indicó que no. Uno de los entrevistados, indicó que a nivel empresarial existen sistemas que hacen una labor similar al traductor. En particular, lo que hacen dichas soluciones es traducir mensajes de los usuarios a través de distintas redes sociales, convirtiéndolos en un ticket, para que luego sea resuelto por los miembros del área de soporte de la empresa.

Tabla 5: Resultados de consultas de usabilidad de la API

Pregunta respecto a la API implementada	Opiniones mayoritarias
¿Siente usted que podría utilizarla sin requerir apoyo externo?	10 de 11 entrevistados dijeron no requerir apoyo externo.
¿Sabría usted cómo agregar los servicios para soportar una nueva plataforma social a la API?	9 de 11 entrevistados indicaron que sabrían cómo lograrlo.
¿Conoce alguna otra herramienta/framework/API que realice este tipo de traducción de mensajes?	10 de 11 entrevistados no conocía un software que hiciera una labor similar.
¿Conoce alguna forma más simple y reusable para realizar esta traducción de mensajes?	Todos los entrevistados indicaron que no.
Pregunta respecto a la documentación de la API	Opiniones mayoritarias
Si uno tuviera sólo la documentación de la API, por favor indique ¿qué tan usable es ésta para un desarrollador con experiencia en APIs Web?	Todos los entrevistados opinaron que la usabilidad era alta.
Si tuviera que comparar la cantidad y calidad de la documentación de esta API, contra la de una API Web promedio, ¿cómo diría que es la documentación de la API desarrollada?	9 de 11 entrevistados consideraron que era mejor, y los otros 2 dijeron que era similar.

La principal diferencia en el funcionamiento de SMT con los indicados por este entrevistado, es que en primer lugar esos sistemas no proveen una API para su uso por parte de múltiples dispositivos. En segundo lugar, no contemplan la existencia de múltiples usuarios para los cuales traducir los mensajes hacia los distintos canales de comunicación. En tercer lugar, no consideran la preferencia de canales de comunicación, ni el cambio de estas preferencias a lo largo del tiempo. Estos puntos implican temas de seguridad, logística y diseño, diferentes a los presentes en los casos de las aplicaciones empresariales.

6.4 Discusión

Durante el proceso de validación del sistema se pudieron notar ciertas situaciones que merecen ser destacadas, dado que influyen en la funcionalidad del sistema. Sin embargo, en ningún caso implican un llamado a la no utilización del traductor.

Un ejemplo de esto es lo que sucede con el envío de archivos grandes desde el traductor al destinatario a través de Telegram. En este caso particular, las pruebas muestran un tiempo considerablemente mayor que en todos los otros canales de comunicación. La razón de tal alza de tiempos es intrínseco a la librería utilizada para realizar el envío de mensajes a través de esta red social. Esta librería, debido a su implementación, limita la velocidad de subida de los archivos, actuando como cuello de botella en el caso de archivos grandes. Esto puede ser visto en los escenarios de prueba 24, 27 y 32 (Sección 6.1.2).

Junto con esto, Whatsapp presenta tiempos de procesamiento de mensajes recibidos mucho más grandes que los otros canales. En particular, en los escenarios 4 y 6 se puede ver tiempos de 8 y 10 segundos respectivamente, siendo que los otros canales el tiempo de procesamiento es de un segundo. Esto se debe a que la librería utilizada para conectar con Whatsapp, por debajo (en background) levanta un navegador web, el cual se conecta con la versión web de este canal. Eso tiene como implicancia que el tiempo para la obtención de los mensajes nuevos depende de la tasa de refresco de la librería hacia la interfaz web, y de la tasa de refresco del navegador web hacia el servidor de Whatsapp. Por lo tanto, los tiempos son mayores que en los otros canales donde sólo existe una de estas tasas.

Otra carencia presente en SMT es la inhabilidad de éste para soportar la agregación, modificación o eliminación de los diversos elementos de dominio (Familias, Usuarios, Preferencia de Canales, etc.) a través de una API Web, debiendo cargarse dicha información a mano desde una interfaz, o bien utilizando la aplicación web de SocialConnector. Esto afecta el uso del Traductor por parte de sistemas externos, pero debido a que éste no fue el foco del desarrollo de esta primera versión de SMT, no se considera como un problema importante. Junto con esto, cabe destacar que el diseño de la solución permite soportar esta nueva funcionalidad sin ningún problema, por lo que no se requeriría modificar nada a nivel de diseño de componentes o de modelos de datos, simplemente se deben agregar nuevos endpoints al Traductor, junto con su lógica correspondiente.

Respecto a la seguridad de SMT, se debe dejar en claro que, a pesar de no poder proteger al usuario frente a cualquier intento de ataque malicioso, los más típicos fueron previstos y analizados tanto en su potencial daño, como en su probabilidad de ocurrencia. Es importante destacar esto, a fin de tranquilizar a potenciales desarrolladores y usuarios del sistema, pues la seguridad fue un factor considerado durante todo el desarrollo de la presente solución. Además, el software pretende ser abierto una vez liberado, de forma que éste pueda ser auditable por otros desarrolladores para encontrar potenciales riesgos a la seguridad de los usuarios.

Se puede ver también que a pesar de la existencia de aplicaciones con un procesamiento de mensajes similar a nivel empresarial, no existe algo pensado en un uso más doméstico,

enfocado en usuarios de múltiples naturalezas (empresas o personas naturales); y que además considere las preferencias de canal de los usuarios a la hora de enviar un mensaje.

Luego de estos tres procesos de evaluación de SMT, se puede decir que este sistema funciona correctamente en términos del envío, recepción, traducción y ruteo de mensajes sociales. Además, su participación como intermediario en la interacción entre dos personas, no impacta negativamente dichas interacciones. Finalmente, la API desarrollada para poder hacer uso de SMT es usable y comprensible, tanto por un futuro desarrollador externo como por uno interno.

7 Conclusiones y Trabajo Futuro

El rápido avance en las tecnologías de información y comunicaciones ha traído sin duda beneficios, pero junto con esto, también ha traído implicancias que no son tan simples de ver. Por ejemplo, existe una carencia de uso de estándares en el intercambio de mensajes en plataformas de comunicación social. Junto con esto, existe una gran cantidad de aplicaciones de comunicación digital, que usan diversos enfoques buscando capturar a distintos públicos objetivos. Este avance tecnológico, junto con la diversificación de las herramientas de apoyo a las interacciones sociales, genera asimetrías en las preferencias de medios de comunicación digital entre las personas.

La literatura reciente reconoce esta asimetría como un problema abierto que se vuelve más evidente en la comunicación social intergeneracional. Los trabajos relacionados muestran la existencia de múltiples herramientas que dan soporte a interacciones sociales entre nativos e inmigrantes digitales, y también sistemas que conectan punto a punto a estas personas a través de soluciones ad hoc.

A pesar de su utilidad, ninguna de estas aplicaciones aborda la asimetría de preferencia de canales de comunicación y la evolución natural de dichos canales. Esto implica que los usuarios de aplicaciones sociales ad hoc (usualmente adulto mayores) tienen que mantenerse al día con la evolución tecnológica, pues de otra forma estas personas en un corto período de tiempo ya no tendrán más chances de seguir utilizando las herramientas de interacción que utilizaban en el pasado. Esto reduce la capacidad de interacción sociales entre ellos y el resto de sus familiares.

Por otra parte, los desarrolladores de estas soluciones sociales ad hoc están forzados a su vez a actualizar de forma continua sus sistemas, debido no sólo a los cambios de sus propias plataformas, sino también a los cambios de otros canales de comunicación con los que estén conectados.

El sistema traductor antes presentado representa un paso adelante en pos de encontrar una solución completa y definitiva al problema de asimetría planteado, entregando mecanismos para la comunicación entre usuarios. El SMT considera en su diseño, tanto la asimetría de preferencia de canales, como la transparencia en el uso de estos canales, lo cual es particularmente útil para inmigrantes digitales, y también para desarrolladores de software sociales ad hoc. De esa manera SMT permite a estos sistemas mantener su funcionalidad y el paradigma para la interacción de usuarios en el tiempo, sin tener que considerar prácticamente la evolución de los canales de interacción social y de preferencia de los usuarios involucrados en las interacciones. Es decir, SMT permite que los desarrolladores de aplicaciones ad hoc no deban modificar el código fuente de sus aplicaciones, para poder seguirle el paso a la evolución de los canales de comunicación entregados por las diversas redes sociales.

La evaluación de la correctitud en la traducción de mensajes, y de los mecanismos de ruteo de estos, se realizó mediante la realización de 40 escenarios de interacción con asimetría de preferencias. Los resultados obtenidos muestran que el Traductor es capaz técnicamente de

hacerse cargo de la evolución de los canales de comunicación digital, y de la preferencia de los usuarios.

Respecto a la seguridad del sistema, el análisis realizado involucró a los tipos de ataque más frecuentes, su probabilidad de ocurrencia, su potencial daño y las medidas que se tomaron en el diseño de SMT para mitigar estos riesgos. Dicho análisis muestra qué desafíos de seguridad fueron considerados durante el diseño del traductor, y cómo se espera disminuir el potencial impacto de cada ataque. Además de esto, el código del software será liberado permitiendo el escrutinio público de pares, de forma de tener un cierto nivel de auditoría externa e imparcial en relación con la seguridad del sistema.

Durante la evaluación de la usabilidad y la mantenibilidad del sistema, los desarrolladores participantes entregaron comentarios altamente positivos, mostrando que este sistema (a través de su API) es una solución útil y usable para ellos. También indicaron que el diseño de la API, sus métodos y componentes eran entendibles y les podría permitir extender o mantener los traductores existentes. A pesar de los resultados positivos, este proceso de validación aún requiere el uso efectivo del sistema por parte de desarrolladores, para determinar así su verdadero impacto como solución intermedia al problema de asimetría de interacción social digital.

El trabajo futuro se puede dividir en tres intervalos de tiempo. En el corto plazo se podrían, agregar endpoints para permitir operaciones básicas (lectura, escritura, edición y eliminación) de los usuarios y las comunidades, de forma de no requerir utilizar la interfaz gráfica o acceder a la base de datos para hacerlo, y así poder realizar todas las acciones a través de la API Web. También se podría conectar SMT con aplicaciones ad hoc en un ambiente de producción o pre-producción, o agregar nuevos canales de comunicación considerando prioritarios los más usados por la población a la que se busca apoyar.

En el mediano plazo, se podría hacer una nueva evaluación de la plataforma, luego de que el sistema haya sido utilizado por diversos usuarios durante algunos meses. Considerando la existencia de un sistema social ad hoc que utilice al traductor como intermediario, se podría analizar el esfuerzo requerido por desarrolladores externos para utilizar la API y extender el sistema social. Esta evaluación permitiría determinar la utilidad de SMT para mediar interacciones sociales en ambientes de preferencia asimétrica, tanto desde el punto de vista de los usuarios como los desarrolladores.

En el largo plazo, al tener muchos datos de uso de la aplicación por parte de las personas, se podrían analizar éstos para mejorar los modelos de pronóstico de uso de los distintos canales. De esa forma se podría saber con mayor certeza cuáles son los canales actualmente preferidos por un individuo, y así mejorar su experiencia con el uso del sistema.

Finalmente, se puede decir que SMT permite tener una primera propuesta de mitigación del problema de asimetría de preferencia de canales de comunicación. Se espera además que este traductor facilite la investigación de este problema fuera del mundo netamente privado, y así mejorar la experiencia de todos, tanto desarrolladores como usuarios finales.

8 Bibliografía

- [1] C. Shannon, «A mathematical theory of communication,» *The Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, 1948.
- [2] F. J. Gutierrez y S. F. Ochoa, «Mom, i do have a family!: attitudes, agreements, and expectations on the interaction with Chilean older adults,» de *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, ACM, 2016, pp. 1402-1411.
- [3] R. Cornejo, M. Tentori y J. Favela, «Ambient Awareness to Strengthen the Family Social Network of Older Adults,» *Computer-Supported Cooperative Work*, vol. 22, pp. 309-344, 2013.
- [4] K. Tee, A. B. Brush y K. M. Inkpen, «Exploring communication and sharing between extended families,» *International Journal of Human-Computer Studies*, vol. 67, n° 2, pp. 128-138, 2009.
- [5] D. Muñoz, R. Cornejo, F. J. Gutierrez, J. Favela, S. F. Ochoa y M. Tentori, «A social cloud-based tool to deal with time and media mismatch of intergenerational family communication,» *Future Generation Computer Systems*, vol. 53, pp. 140-151, 2015.
- [6] D. Muñoz, R. Cornejo, S. F. Ochoa, J. Favela, F. Gutierrez y M. Tentori, «Aligning intergenerational communication patterns and rhythms in the age of social media,» de *Proceedings of the 2013 Chilean Conference on Human-Computer Interaction*, 2013, pp. 66-71.
- [7] C. Kim y J. Kook Lee, «Social Media Type Matters: Investigating the Relationship Between Motivation and Online Social Network Heterogeneity,» *Journal of Broadcasting & Electronic Media*, vol. 60, pp. 676-693, 2016.
- [8] T. Anttila y T. Oinas, «Patterns of Working Time and Work Hour Fit in Europe,» de *Family, Work and Well-Being*, M. Tammelin, Ed., 2018.
- [9] F. Gutierrez, S. Ochoa, R. Cornejo y J. Vassileva, «Designing Computer-Supported Technology to Mediate Intergenerational Social Interaction: A Cultural Perspective,» de *Perspectives on Human-Computer Interaction Research with Older People*, S. Sayago, Ed., 2019.
- [10] P. Santosa, «Measuring User Experience in an Online Store Using PULSE and HEART Metrics,» *Jurnal Ilmiah Kursor*, vol. 7, n° 3, pp. 145-153, 2014.

- [11] F. Rodríguez, S. Ochoa y F. J. Gutierrez, «Seamlessly Mediation of Social Interaction Services Respecting Communication Preferences,» de *Proceedings 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI'18)*, Punta Cana, República Dominicana, Dec. 2018.
- [12] F. Rodríguez, S. Ochoa y F. J. Gutierrez, «Supporting Asymmetric Interaction in the Age of Social Media,» En revisión en el *Journal of Ambient Intelligence and Humanized Computing*, 2019.
- [13] D. Muñoz, F. J. Gutierrez, S. F. Ochoa y N. Baloian, «Social connector: A ubiquitous system to ease the social interaction among family community members,» *Computer Systems Science and Engineering*, vol. 30, pp. 57-68, 2015.
- [14] R. S. Pressman, «The Incremental Model,» de *Software Engineering: A Practitioner's Approach 5th Edition*, McGraw-Hill Higher Education, 2001, pp. 35-36.
- [15] S. Frennert y B. Östlund, «Review: Seven Matters of Concern of Social Robots and Older People,» *International Journal of Social Robotics*, vol. 6, n° 2, pp. 299-310, April 2014.
- [16] T. K. Judge, C. Neustaedter, S. Harrison y A. Blose, «Family Portals: Connecting families through a multifamily media space,» *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1205-1214, May 2011.
- [17] E. D. Mynatt, J. Rowan, S. Craighill y A. Jacobs, «Digital Family Portraits: Supporting Peace of Mind for Extended Family Members,» de *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, USA, 2001, pp. 333-340.
- [18] C. Neustaedter, S. Harrison y A. Sellen, *Connecting Families: An Introduction*, 2013.
- [19] D. Hindus, S. D. Mainwaring, N. Leduc, A. E. Hagström y O. Bayley, «Casablanca: Designing Social Communication Devices for the Home,» de *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, USA, 2001, pp. 325-332.
- [20] M. Alaoui y M. Lewkowicz, «Struggling Against Social Isolation of the Elderly—The Design of SmartTV Applications,» 2012.

- [21] C. Müller, J. Schnittert, M. Walczuch, M. Alaoui, M. Lewkowicz, L. Wan y V. Wulf, «Impact Factors on Social TV Research in Real Elderly Persons' Households,» *Mensch und Computer 2015 – Proceedings*, pp. 213-222, 2015.
- [22] F. F. Mueller, F. Vetere, M. R. Gibbs, J. Kjeldskov, S. Pedell y S. Howard, «Hug over a distance,» de *CHI 05 extended abstracts on Human factors in computing systems - CHI 05*, 2005.
- [23] J. Werner, R. Wettach y E. Hornecker, «United-pulse,» de *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services - MobileHCI 08*, 2008.
- [24] G. Fitzpatrick, A. Hultgren, L. Malmberg, D. Harley y W. Ijsselsteijn, «Design for Agency, Adaptivity and Reciprocity: Reimagining AAL and Telecare Agendas,» de *Computer Supported Cooperative Work*, Springer London, 2015, pp. 305-338.
- [25] T. Lehnert, D. Heider, H. Leicht, S. Heinrich, S. Corrieri, M. Lupp, S. Riedel-Heller y H.-H. König, «Review: Health Care Utilization and Costs of Elderly Persons With Multiple Chronic Conditions,» *Medical Care Research and Review*, vol. 68, pp. 387-420, 2011.
- [26] J. Vines, G. Pritchard, P. Wright, P. Olivier y K. Brittain, «An Age-Old Problem,» *ACM Transactions on Computer-Human Interaction*, vol. 22, pp. 1-27, 2015.
- [27] J. S. House, «Social isolation kills, but how and why?,» *Psychosomatic medicine*, vol. 63, n° 2, pp. 273-274, 2001.
- [28] A. Karimi y C. Neustaedter, «From High Connectivity to Social Isolation: Communication Practices of Older Adults in the Digital Age,» de *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*, New York, NY, USA, ACM, pp. 127-130, 2012.
- [29] S. E. Lindley, R. Harper y A. Sellen, «Desiring to Be in Touch in a Changing Communications Landscape: Attitudes of Older Adults,» de *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, MA, USA, pp. 1693-1702, 2009.
- [30] D. Muñoz, F. J. Gutierrez, S. F. Ochoa y N. Baloian, «Enhancing Social Interaction between Older Adults and Their Families,» de *IWAAL*, pp. 47-54, 2013.

- [31] R. Cornejo, M. Tentori y J. Favela, «Enriching in-person encounters through social media: A study on family connectedness for the elderly,» *International Journal of Human-Computer Studies*, vol. 71, n° 9, pp. 889-899, 2013.
- [32] C. Plaisant, A. Clamage, H. B. Hutchinson, B. B. Bederson y A. Druin, «Shared family calendars: Promoting symmetry and accessibility,» *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, n° 3, pp. 313-346, 2006.
- [33] S. E. Lindley, R. Harper y A. Sellen, «Designing for elders: exploring the complexity of relationships in later life,» de *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, British Computer Society, pp. 77-86, 2008.
- [34] D. S. Kirk, A. Sellen y X. Cao, «Home video communication: mediating 'closeness',» de *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pp. 135-144, 2010.
- [35] X. Cao, A. Sellen, A. Brush, D. Kirk, D. Edge y X. Ding, «Understanding family communication across time zones,» de *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 2010, pp. 155-158.
- [36] J. Grudin y S. Poltrock, «Taxonomy and theory in computer supported cooperative work,» *The Oxford handbook of organizational psychology*, vol. 2, pp. 1323-1348, 2012.
- [37] J. Wherton y D. Prendergast, «The Building Bridges Project: Involving Older Adults in the Design of a Communication Technology to Support Peer-to-Peer Social Engagement,» de *Proceedings of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion*, Linz, Austria, Springer-Verlag, 2009, pp. 111-134.
- [38] M. A. Motoyama y G. Varghese, «CrossTalk: scalably interconnecting instant messaging networks,» de *Proceedings of the 2Nd ACM Workshop on Online Social Networks*, Barcelona, 2009, pp. 61-68.
- [39] Department of Economic and Social Affairs, Population Division, «World Population Prospects, The 2015 Revision: Key Findings and Advance Tables,» New York, 2015.
- [40] T. Mitzner, J. Savla, W. R. Boot, J. Sharit, N. Charness, S. Czaja y W. Rogers, «Technology Adoption by Older Adults: Findings From the PRISM Trial,» *The Gerontologist*, vol. 59, pp. 34-44, Sep. 2018.

- [41] XMPP Standards Foundation, «Extensible Messaging and Presence Protocol,» [En línea]. Available: <https://xmpp.org>. [Último acceso: 30 de Julio 2019].
- [42] F. Karpisek, I. Baggili y F. Breitinger, «WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages,» *Digital Investigation*, vol. 15, pp. 110-118, Dec. 2015.
- [43] Whatsapp, «No Api para Whatsapp,» [En línea]. Available: <http://mashable.com/2015/03/25/whatsapp-developers-api/#4YPSIE8ErOqw>. [Último acceso: 30 de Julio 2019].
- [44] Facebook, «API Graph de Facebook,» [En línea]. Available: <https://developers.facebook.com/docs/graph-api>. [Último acceso: 30 de Julio 2019].
- [45] Microsoft, «API Skype,» [En línea]. Available: <https://dev.skype.com>. [Último acceso: 30 de Julio 2019].
- [46] Telegram, «API Telegram,» [En línea]. Available: <https://core.telegram.org/>. [Último acceso: 30 de Julio 2019].
- [47] F. Rodríguez, «SocialTranslator Documentation,» [En línea]. Available: <https://farodrig.github.io/SocialTranslator/>. [Último acceso: 30 de Julio 2019].
- [48] P. J. Brockwell y R. A. Davis, *Introduction to time series and forecasting*, 2002.
- [49] Android, «Sistema Android Keystore,» [En línea]. Available: <https://developer.android.com/training/articles/keystore>. [Último acceso: 30 de Julio 2019].
- [50] «Statista,» 2019. [En línea]. Available: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps>. [Último acceso: 30 de Julio 2019].
- [51] «Telethon,» [En línea]. Available: <https://github.com/LonamiWebs/Telethon>. [Último acceso: 30 de Julio 2019].
- [52] «Microsoft,» 16 Marzo 2015. [En línea]. Available: <https://www.microsoft.com/en-us/microsoft-365/blog/2015/03/16/get-ready-for-skype-for-business/>. [Último acceso: 30 de Julio 2019].
- [53] «SkPy,» [En línea]. Available: <https://github.com/Terrance/SkPy>. [Último acceso: 30 de Julio 2019].

[54] «WebWhatsAPI,» [En línea]. Available: <https://github.com/Michotastico/WebWhatsAPI>. [Último acceso: 30 de Julio 2019].

[55] «Web Whatsapp,» [En línea]. Available: <https://web.whatsapp.com/>. [Último acceso: 30 de Julio 2019].

Anexos

A continuación, se presentan los anexos que entregan información detallada acerca de los instrumentos usados o generados en este trabajo de tesis.

Anexo A: Cuestionario para análisis de necesidades

Las preguntas realizadas a los participantes del estudio para identificar necesidades en la alineación de canales y el uso de canales preferentes fueron las siguientes:

1. Sexo
2. Edad
3. Relación con el adulto mayor (Niet@ o Hij@)
4. ¿Qué medio de comunicación utilizaría para enviar un mensaje importante durante su jornada laboral?
 - a. Si la respuesta a este mensaje llegará rápidamente (menos de 10 minutos), ¿por qué medio preferiría recibir la respuesta?
 - b. Si la respuesta a este mensaje fuera tardía (más de 4 horas), ¿por qué medio preferiría recibir la respuesta?
5. ¿Qué medio de comunicación usaría para enviar un mensaje de poca importancia durante su jornada laboral? Asuma que usted decidió enviar el mensaje.
 - a. Si la respuesta a este mensaje llegará rápidamente (menos de 10 minutos), ¿por qué medio preferiría recibir la respuesta?
 - b. Si la respuesta a este mensaje fuera tardía (más de 4 horas), ¿por qué medio preferiría recibir la respuesta?
6. ¿Qué medio de comunicación utilizaría para enviar un mensaje importante durante su tiempo libre?
 - a. Si la respuesta a este mensaje llegará rápidamente (menos de 10 minutos), ¿por qué medio preferiría recibir la respuesta?
 - b. Si la respuesta a este mensaje fuera tardía (más de 4 horas), ¿por qué medio preferiría recibir la respuesta?
6. ¿Qué medio de comunicación usaría para enviar un mensaje de poca importancia durante su tiempo libre? Asuma que usted decidió enviar el mensaje.
 - a. Si la respuesta a este mensaje llegará rápidamente (menos de 10 minutos), ¿por qué medio preferiría recibir la respuesta?
 - b. Si la respuesta a este mensaje fuera tardía (más de 4 horas), ¿por qué medio preferiría recibir la respuesta?
7. Si hubiera un servicio para mandarle un mensaje a su adulto mayor (u a otra persona) por cualquiera de estos medios, ¿qué tan útil lo considera? (del 1 al 10) ¿Por qué?

8. Si hubiera un servicio que redirige los mensajes del adulto mayor, hacia la herramienta preferida por usted (independientemente de la herramienta usada por el emisor), ¿qué tan útil lo considera? (del 1 al 10) ¿Por qué?

Anexo B: Sistema de preferencias

El sistema web de SocialConnector permite obtener las preferencias de redes sociales de la familia junto con sus nombres de usuario en las diversas redes sociales. Este sistema considera coordinadores de una familia, los cuales tienen la labor de mantener actualizada la información de los distintos miembros de la familia. El sistema provee interfaces enfocadas en ayudar al coordinador a mantener esta información de forma fácil y sencilla. Además de esto, el sistema se encarga de entregar a través de una API Web los datos de una familia, previo chequeo de tener los permisos correspondientes para obtener estos datos.

En el caso de que un sistema externo quisiera sacar la información de las familias (como es el caso de SMT), la aplicación posee endpoints para este fin. Lo primero que debe realizarse es solicitar acceso al administrador de la plataforma. Dicha persona deberá dirigirse al sitio de administración y agregar un nuevo usuario, asignándole un username y una contraseña, luego de esto, deberá dirigirse a la sección de “External Systems” y añadir uno nuevo, indicando el usuario que en realidad es un sistema externo, el nombre y la URL de la aplicación externa.

Luego de esto, el sistema externo podrá solicitar un token de acceso a la API utilizando el endpoint dedicado a esto, el cual es descrito a continuación:

- URL: /api/login/
- Método HTTP: POST
- Tipo de datos de petición: JSON
- Ejemplo de cuerpo petición:

```
{
  "username": "sistema.externo",
  "password": "password123"
}
```

- Tipo de datos de respuesta: JSON
- Ejemplo de respuesta:

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IjE2IiwiaWF0IjoiIiwidXNlcl9pZCI6ImtZ9.PrRqqnAYK8BHrGDvUVy6RuP0tU26KkfpWnH0dqCkL08"
}
```

Luego de esto, ya podemos hacer una consulta al endpoint principal para que nos entregue todas las familias, junto con su coordinador, abuelo, miembros y preferencias de cada uno de ellos. Este endpoint es el utilizado por SMT para hacer la carga inicial de las preferencias de todas las familias que posea la aplicación web de SocialConnector y es descrito a continuación:

- URL: /api/social/all
- Header HTTP: “Authorization: JWT
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IjE2IiwiaWF0IjoiIiwidXNlcl9pZCI6ImtZ9.PrRqqnAYK8BHrGDvUVy6RuP0tU26KkfpWnH0dqCkL08”

5zbGF0b3IiLCJleHAI0jE1NDYwMDQ4MTEsImVtYWlsIjoiIiwidXNlc19pZCI6MTZ9.
PrRqqnAYK8BHRGDvUVy6RuPOtU26KkfpWnH0dqCkL08”

- Método HTTP: GET
- Tipo de datos de respuesta: JSON
- Ejemplo de respuesta:

```
[
  {
    "coordinator": {
      "first_name": "Juan",
      "last_name": "Perez",
      "user": "juan.perez@gmail.com",
      "is_coordinator": true,
      "is_grandpa": false
    },
    "grandpa": {
      "first_name": "José",
      "last_name": "Perez",
      "user": "jose.perez@gmail.com",
      "is_coordinator": false,
      "is_grandpa": true
    },
    "members": [
      {
        "first_name": "Zunilda",
        "last_name": "Perez",
        "user": "zuny.perez@gmail.com",
        "is_grandpa": false,
        "is_coordinator": false,
        "preferences": [
          {
            "username": "+5697777777",
            "order": 1,
            "network": "Telegram"
          }
        ]
      },
      ...
    ]
  },
  ...
]
```

Anexo C: Resultado de las pruebas de traducción de mensajes

A continuación se presentan los escenarios de prueba utilizados en la traducción de mensajes, y los resultados obtenidos en cada uno de ellos.

N° Prueba	Descripción	Emisor	Receptor	Resultado	Tiempo (en segs)
Escenarios de envío de mensaje de texto a U5					
1	U1 le envía un mensaje de texto por Skype (canal preferido) a U5	U1	U5	OK	~1
2	U2 le envía un mensaje de texto por email (canal preferido) a U5	U2	U5	OK	~1
3	U3 le envía un mensaje de texto por Telegram (canal preferido) a U5	U3	U5	OK	~1
4	U4 le envía un mensaje de texto por Whatsapp (canal preferido) a U5	U4	U5	OK	~8
5	U1 le envía un mensaje de texto por email a U5	U1	U5	OK	~1
6	U2 le envía un mensaje de texto por Whatsapp a U5	U2	U5	OK	~10
7	U3 le envía un mensaje de texto por Skype a U5	U3	U5	OK	~2
8	U4 le envía un mensaje de texto por Telegram a U5	U4	U5	OK	~2
Escenarios de envío de mensaje multimedia a U5					
9	U1 le envía un mensaje multimedia por Skype (canal preferido) a U5	U1	U5	OK	~5
10	U2 le envía un mensaje multimedia por email (canal preferido) a U5	U2	U5	OK	~5

11	U3 le envía un mensaje multimedia por Telegram (canal preferido) a U5	U3	U5	OK	~3
12	U4 le envía un mensaje multimedia por Whatsapp (canal preferido) a U5	U4	U5	OK	~10
13	U1 le envía un mensaje multimedia por email a U5	U1	U5	OK	~5
14	U2 le envía un mensaje multimedia por Whatsapp a U5	U2	U5	OK	~12
15	U3 le envía un mensaje multimedia por Skype a U5	U3	U5	OK	~4
16	U4 le envía un mensaje multimedia por Telegram a U5	U4	U5	OK	~2
Escenarios de envío de mensaje por U5 en respuesta a escenarios pasados					
17	U5 le responde el <u>primer mensaje</u> de U1 (Prueba 1), enviándole un mensaje de texto. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U1	OK	1
18	U5 le responde el <u>primer mensaje</u> de U2 (Prueba 2), enviándole un mensaje de texto. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U2	OK	1
19	U5 le responde el <u>primer mensaje</u> de U3 (Prueba 3), enviándole un mensaje de texto. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U3	OK	1
20	U5 le responde el <u>primer mensaje</u> de U4 (Prueba 4), enviándole un mensaje de texto. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U4	OK	1.5

21	U5 le responde el <u>segundo mensaje</u> de U1 (Prueba 5), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U1	OK	2.9
22	U5 le responde el <u>segundo mensaje</u> de U2 (Prueba 6), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U2	OK	3.9
23	U5 le responde el <u>segundo mensaje</u> de U3 (Prueba 7), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U3	OK	4.3
24	U5 le responde el <u>segundo mensaje</u> de U4 (Prueba 8), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U4	OK	4.3
25	U5 le responde el <u>tercer mensaje</u> de U1 (Prueba 9), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U1	OK	3.6
26	U5 le responde el <u>tercer mensaje</u> de U2 (Prueba 10), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U2	OK	4.1
27	U5 le responde el <u>tercer mensaje</u> de U3 (Prueba 11), enviándole un mensaje	U5	U3	OK	4.6

	multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.				
28	U5 le responde el <u>tercer mensaje</u> de U4 (Prueba 12), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U4	OK	4.1
29	U5 le responde el <u>cuarto mensaje</u> de U1 (Prueba 13), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U1	OK	2.6
30	U5 le responde el <u>cuarto mensaje</u> de U2 (Prueba 14), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U2	OK	4.4
31	U5 le responde el <u>cuarto mensaje</u> de U3 (Prueba 15), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U3	OK	3.7
32	U5 le responde el <u>cuarto mensaje</u> de U4 (Prueba 16), enviándole un mensaje multimedia. El mensaje debería ser recibido a través del último canal utilizado por el destinatario.	U5	U4	OK	4.4
Para los siguientes escenarios, se utiliza una nueva versión del traductor					

33	U5 le manda un nuevo mensaje de texto a U1. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U1	OK	1
34	U5 le manda un nuevo mensaje de texto a U2. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U2	OK	1
35	U5 le manda un nuevo mensaje de texto a U3. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U3	OK	1
36	U5 le manda un nuevo mensaje de texto a U4. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U4	OK	1.3
Para los siguientes escenarios, se intercambia la primera y la segunda preferencia de los usuarios					
37	U5 le manda un nuevo mensaje de texto a U1. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U1	OK	1
38	U5 le manda un nuevo mensaje de texto a U2. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U2	OK	1.2
39	U5 le manda un nuevo mensaje de texto a U3. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U3	OK	1

40	U5 le manda un nuevo mensaje de texto a U4. El mensaje debería ser recibido a través del canal preferido por el destinatario.	U5	U4	OK	1
-----------	---	----	----	----	---

Anexo D: Pauta de entrevista a desarrolladores

A continuación se presenta la pauta de entrevista utilizada en la evaluación de la API por parte de los desarrolladores.

Entrevista Semiestructurada a Desarrolladores

Pauta de entrevista

1. Indique la cantidad de años de experiencia que tiene desarrollando software.
2. Indique la cantidad de años de experiencia que tiene utilizando APIs Web.

En cada una de las preguntas siguientes, los participantes deben indicar y justificar (al menos brevemente) sus respuestas. Si se va a grabar las entrevistas, entonces se debe pedir permiso para ello.

3. Respecto a la API implementada:
 - 3.1. ¿Siente usted que podría utilizarla sin requerir apoyo externo?
 - 3.2. ¿Sabría usted cómo agregar los servicios para soportar una nueva plataforma social a la API?
 - 3.3. ¿Conoce alguna otra herramienta/framework/API que realice este tipo de traducción de mensajes?
 - 3.4. ¿Conoce alguna forma más simple y reusable para realizar esta traducción de mensajes?
4. Respecto a la Documentación de la API:
 - 4.1. Si uno tuviera sólo la documentación de la API, por favor indique ¿qué tan usable es ésta?
 - 4.1.1. Para un desarrollador promedio.
 - 4.1.2. Para un desarrollador que ya ha trabajado antes con APIs Web.
 - 4.2. Si tuviera que comparar la cantidad y calidad de la documentación de esta API, contra la de una API Web promedio, ¿cómo diría que es la documentación de la API desarrollada?

Anexo E: Formulario de consentimiento informado

(Entrevistas)

Solicitamos a usted, colaborar con el proyecto titulado “*A Comprehensive Framework for Designing Computer-Mediated Interaction Mechanisms to Support Informal Elderly Caregiving*” (Un Marco Integral para el Diseño de Mecanismos de Interacción Mediados por Computador para Apoyar el Cuidado Informal de Adultos Mayores), presentado a CONICYT en su concurso FONDECYT Regular 2019.

La colaboración que solicitamos consiste en participar en una entrevista que busca indagar de manera general sobre las prácticas de interacción familiar que tienen los participantes de este estudio; especialmente entre los adultos mayores y los miembros más jóvenes de la familia.

A continuación le proporcionamos información más detallada del proyecto, y de los términos concretos que involucra su participación en esta entrevista. Esto lo hacemos con el fin de permitirle decidir, de manera informada, si desea o no colaborar con este proyecto.

Información sobre el proyecto

1. El proyecto “*A Comprehensive Framework for Designing Computer-Mediated Interaction Mechanisms to Support Informal Elderly Caregiving*” tiene los siguientes objetivos: (i) proponer un marco de referencia para caracterizar la estructura familiar, así como identificar las capacidades, necesidades, expectativas y prioridades de sus miembros en términos de la provisión de cuidado informal a sus adultos mayores; (ii) proponer un conjunto de guías de diseño contextualizadas que apoyen la concepción de tecnología fácil de adoptar por adultos mayores, con la finalidad de apoyar su cuidado informal; (iii) desarrollar un sistema de interpretación de mensajes, capaz de interactuar con varios servicios de redes sociales en línea y software social, de manera de poder intercambiar contenido de manera asíncrona en un formato común; y (iv) desarrollar mecanismos de persuasión sensibles al contexto, que permitan motivar a los miembros de la familia a tomar acción (o no) cuando se requiera.
2. Este proyecto proporcionará información de relevancia para el desarrollo de **soluciones computacionales**. En ningún caso el estudio implica ingerir líquido o sólido alguno, ni medicamentos, ni tratamientos médicos.
3. Este Proyecto está dirigido por los Prof. Sergio Fabián Ochoa y Francisco Javier Gutiérrez, investigadores del Departamento de Ciencias de la Computación de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile. Participan en el estudio, como entrevistadores, alumnos y profesionales de la misma universidad.
4. Los datos recopilados dentro del Proyecto serán administrados resguardando la estricta confidencialidad de la identidad de las personas participantes. Sólo tendrán acceso a ellos el director del proyecto y sus ayudantes directos. Las personas responsables del resguardo de la información son el director y el co-director del proyecto, Prof. Sergio F. Ochoa y Prof. Francisco Gutiérrez respectivamente.

5. Los datos recopilados dentro del Proyecto podrán ser usados para ser publicados, en forma anónima y respetando la privacidad de los participantes involucrados, en revistas científicas del área de investigación, libros especializados y en congresos académicos en Chile y en el extranjero.

Información sobre su participación:

El contexto general de la entrevista se refiere a las prácticas de interacción social de los miembros más jóvenes de la familia, para con sus adultos mayores. Para ello, un entrevistador miembro del equipo de investigación, le formulará una serie de preguntas sobre este tema.

La participación en la entrevista no contempla ningún tipo de remuneración económica ni retribución material de ningún tipo, por parte del equipo de investigadores, ni de Conicyt.

Su participación en el estudio es totalmente voluntaria. Si no acepta, no tiene ninguna consecuencia negativa para Ud. Si acepta, usted puede dejar de participar en el estudio en cualquier momento que lo desee sin que eso tenga consecuencias negativas para usted. Por favor, tómese todo el tiempo que necesite para decidir. A continuación se presenta un conjunto de preguntas y respuestas típicas, que le pueden ayudar a decidir mejor su eventual participación en este estudio.

¿Para qué se firma este documento?

La firma de este documento indica que usted voluntariamente desea participar en este estudio, y que conoce los términos de su participación, los cuales están expresados en el texto de este documento.

¿Por qué se está haciendo este estudio de investigación?

Queremos saber más sobre cómo ayudar a las personas en riesgo de aislamiento social. Este estudio nos ayudará a aprender más sobre cómo se lleva a cabo la interacción social con los adultos mayores, y de qué manera se puede diseñar soluciones computacionales efectivas en este escenario.

¿Qué pasa si digo “sí, quiero participar en el estudio”?

Si dice que sí, le preguntaremos sobre las acciones que usted lleva a cabo para interactuar con los adultos mayores, y los eventuales sistemas que utiliza para esto. Esta información la recopilaremos en forma de entrevista personalizada e individual. Las preguntas que le haremos no tienen respuestas correctas o incorrectas. Usted tiene el derecho de saltarse cualquier pregunta si no quiere contestarla.

¿Quién verá mis respuestas?

Las únicas personas autorizadas para ver sus respuestas son las que trabajan en el estudio; que son las mismas que se aseguran de que dicho estudio se realice de manera correcta. Sus respuestas en la entrevista y una copia firmada de este documento se mantendrán bajo llave en nuestros archivos. Cuando compartamos los resultados del estudio, por ejemplo en reportes técnicos o revistas especializadas de investigación, podremos utilizar citas tuyas, pero en ningún caso incluiremos su nombre o cualquier otra información que lo identifique. En tal caso, usted será contactado/a por el equipo de investigación y dispondrá de un derecho

a veto o retracto sobre la publicación de citas textuales que hayan sido obtenidas durante el proceso de recolección de datos, por ejemplo, en entrevistas o grupos focales. Por lo tanto, haremos todo lo posible para asegurar el anonimato y privacidad de la información que usted provea como parte de este estudio.

Participar en el estudio, ¿me ayudará de alguna manera?

Participar en este estudio no le ayudará necesariamente de manera directa, pero ayudará a diseñar mejor tecnología para apoyar la interacción social entre los adultos mayores y el resto de su comunidad familiar.

Participar en este estudio, ¿podría ser malo para mí, de alguna manera?

No, al menos no se ve ninguna causa evidente para ello. En todo caso, si usted piensa que lo que se le está preguntando involucra información sensible, entonces está en su derecho de no responder a esa pregunta.

¿Qué debo hacer si tengo preguntas?

Por favor, contacte al equipo responsable si tiene alguna pregunta sobre el estudio o tiene preguntas sobre sus derechos. Los datos de contacto están más abajo. Las preguntas asociadas al contenido de la entrevista pueden ser respondidas por el entrevistador.

¿Estoy obligado/a firmar este documento?

No. Fírmelo solamente si desea participar en el estudio.

¿Qué debo hacer si quiero participar en el estudio?

Tiene que firmar tres copias de este documento, nosotros le entregaremos una copia del mismo, una va para el equipo investigador, y otra para Fondecyt.

¿Puedo dejar de participar en el estudio?

Usted es libre de dejar el estudio a voluntad, en el momento que usted lo decida. Para ejercer este derecho, le solicitamos que por favor nos contacte para poder llevar a cabo una breve entrevista sobre las razones que lo motivaron a dejar el estudio, con la finalidad de poder dejar registro de esta situación. Los datos que usted haya generado a la fecha, directa o indirectamente, serán destruidos y no serán procesados.

¿Qué obtengo al final del estudio?

Al terminar este estudio, usted tiene derecho a solicitar una copia de los reportes técnicos y publicaciones derivadas de este proyecto de investigación.

Al firmar este documento está diciendo que:

- Está de acuerdo con participar en el estudio.
- Le hemos explicado la información que contiene este documento y hemos contestado todas sus preguntas.

En conocimiento del proyecto y de acuerdo a las condiciones antes descritas, indique con una cruz su decisión de participar:

- No acepto** colaborar con el proyecto “*A Comprehensive Framework for Designing Computer-Mediated Interaction Mechanisms to Support Informal Elderly Caregiving*”.
- Si acepto** colaborar con el proyecto “*A Comprehensive Framework for Designing Computer-Mediated Interaction Mechanisms to Support Informal Elderly Caregiving*”.

Entrevista:

- Si acepto**
- No acepto**

Grabaciones de audio:

- Si acepto**
- No acepto**

Los Prof. Sergio Fabián Ochoa y Francisco Javier Gutiérrez, investigador principal y co-investigador del proyecto respectivamente, suscriben el compromiso de respetar cabalmente las condiciones detalladas más arriba. Los datos de contacto de estas personas se detallan a continuación:

Prof. Sergio F. Ochoa, Profesor Asociado, Departamento de Ciencias de la Computación, FCFM, Universidad de Chile. Teléfono: 22978 4879, Correo electrónico: sochoa@dcc.uchile.cl.

Prof. Francisco J. Gutiérrez, Profesor Experto en Docencia, Departamento de Ciencias de la Computación, FCFM, Universidad de Chile. Teléfono: 22978 4633, Correo electrónico: frgutier@dcc.uchile.cl.

Nombre del participante: _____

Teléfono de contacto: _____

Correo electrónico: _____

Firma: _____

Fecha: _____

Nombre del entrevistador: _____

Teléfono de contacto: _____

Correo electrónico: _____

(Se firman tres copias, quedando una en poder de cada firmante y una para FONDECYT)