



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SISTEMA DE APOYO AL ANÁLISIS DE ACTIVIDADES SOCIALES DE ADULTOS
MAYORES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

TOMÁS JOAQUÍN ESTÉVEZ LENZ

PROFESOR GUÍA:
FRANCISCO GUTIÉRREZ FIGUEROA

MIEMBROS DE LA COMISIÓN:
SERGIO OCHOA DELORENZI
SANDRA DE LA FUENTE GONZÁLEZ
NELSON BALOIAN TATARYAN

Este trabajo ha sido parcialmente financiado por el Proyecto FONDECYT N° 1191516

SANTIAGO DE CHILE
2021

Resumen

Durante la última década, la tecnología ha irrumpido con fuerza en el ámbito de las comunicaciones, cambiando en gran medida la forma en cómo los seres humanos nos comunicamos los unos con los otros. Hoy en día, el uso de aplicaciones de videoconferencia y mensajería instantánea es el medio de comunicación por defecto de gran parte de la población mundial, más aún en el contexto actual de pandemia. Este cambio de paradigma en la comunicación no ha sido fácil de asimilar por los grupos de personas cuya relación con la tecnología no es tan fluida, como es el caso de los adultos mayores, pues ellos crecieron y se desarrollaron en una sociedad donde el principal canal de comunicación era cara-a-cara o por teléfono. El no estar acostumbrados a interactuar mediante medios digitales los ha perjudicado, poniéndolos en riesgo de caer en una situación de aislamiento social, debido a la brecha tecnológica que existe entre las nuevas generaciones y las más antiguas.

Dentro de este contexto, han surgido algunas aplicaciones que apuntan a simplificar la relación de los adultos mayores con la tecnología, con el objetivo de evitar un eventual aislamiento social de los mismos. Entre estas aplicaciones, está el sistema SocialConnector, cuya principal característica es que integra algunas de las plataformas de videoconferencia y mensajería instantánea más utilizadas (como WhatsApp, Skype y Telegram). A través de una única interfaz este sistema le permite a un adulto mayor, comunicarse con miembros de su familia o amigos, sin tener que aprender a manejar cada aplicación por separado.

SocialConnector fue diseñado con fines de investigación, particularmente para ayudar a entender las interacciones sociales que estas personas llevan al interior de sus hogares. Para ello, el sistema SocialConnector almacena información de las interacciones digitales (mensajes) que se llevan a cabo a través de él, así como también almacena información de las visitas (presenciales) que recibe el adulto mayor en su hogar. Si bien esta información se almacena de manera correcta en la plataforma, el sistema no implementa servicios que faciliten el análisis científico de las interacciones por parte de los investigadores que estudian este proceso.

Este trabajo de título abordó esta problemática, a través de la extensión del modelo de contexto de SocialConnector y la implementación de visualizaciones de información de interacciones con fines científicos. Los servicios implementados fueron evaluados con información proveniente de una base de datos de prueba, puesto que SocialConnector aún no cuenta con información real. A pesar de no utilizar datos de producción, se simularon diversas realidades y se pudo comprobar que todos los desarrollos tenían el comportamiento esperado. Además, que eran útiles para el análisis científico de los datos. En resumen, se logró cumplir los objetivos establecidos en este trabajo de título.

*Dedicado a mi querida Tite,
cuyo recuerdo será eterno.*

Agradecimientos

En primer lugar, agradezco principalmente a mis padres, Soledad y Mario, quienes han sido un pilar fundamental a lo largo de toda mi vida, apoyando cada una de las decisiones que me llevaron a hoy en día ser quien soy. El hecho de que yo hoy día pueda estar escribiendo este documento es debido al fruto del esfuerzo que hicieron por mí, este logro también es de ustedes. También me gustaría agradecer a mis hermanas, Javiera, por ser ejemplo de lucha y convencerme del gran valor que representan los adultos mayores para la sociedad actual a pesar de la gran cantidad de prejuicios que existen hacia ellos, y Camila, por ser ejemplo de valentía, superación y una gran motivación para haber elegido la carrera que escogí.

Agradecer a mi pareja, Fernanda, por acompañarme durante todo el camino universitario (y bastante del colegio), siempre brindándome el apoyo, comprensión y cariño necesario, aún en los momentos más difíciles o incluso en los momentos donde no me lo merecía. Estaré eternamente agradecido de tu apoyo negra.

A mis grandes amigos de la U, Nicolás, Felipe, Rodrigo, Cristián y Agustín, por dejarme compartir con ellos y hacerme sentir acompañado en este camino, estoy muy agradecido de haberlos conocido y espero que el lazo que formamos perdure por siempre. A mis amigos del colegio (sobre todo a Vienna), porque a pesar de que cuesta juntarnos, siempre que lo logramos me brindan su amistad y buena onda. Muchas gracias por ser la gran fuente de espacios de dispersión y relajo de los estudios.

Quisiera también agradecer a las dos compañeras felinas más grandes que tuve a lo largo de mi vida, Ofelia y Rafaella, quienes a pesar de no ser conscientes de lo importantes que eran para mí, me acompañaron durante mis largas tardes de estudio y programación, me dieron su afecto incondicional y siempre me brindaron un cálido ronroneo cuando lo necesitaba, aún en sus últimos días.

A mis profesores guías, Francisco y Sergio, grandes docentes, con gran vocación por la investigación y la docencia, y por sobre todo, muy pedagógicos. Muchas gracias por su guía, sus comentarios, sus correcciones y por sobre todo por haber confiado en mí para tomarme bajo su tutela en este trabajo, aún cuando ni siquiera me conocían.

A mi profesor del colegio, Pedro Unda, quien con su vocación logró transmitirme esa pasión por la física y que con su estilo que mezcla exigencia con sus ganas de enseñar me ayudó a aprender a empujar mis límites. También fue el único profesor (del colegio) que alentó mi decisión por la Universidad de Chile, lo cual le agradezco hasta el día de hoy.

Por último, y no por eso menos importantes, quiero agradecer a 2 grandes adultas mayores con las que tuve el gusto de relacionarme. La primera, mi abuela Silvia, una gran mujer, extraordinariamente culta, sin dudas la persona más inteligente que he conocido en mi vida, pionera en la lucha por la igualdad de género, siendo de las primeras mujeres que fueron a la Universidad. Siempre preocupada de sus nietos y manteniendo su sed de conocimiento hasta sus últimos días. Como me gustaría que pudieras haber presenciado este momento Tite querida, espero estés orgullosa donde quiera que estés. La segunda, Eliana, quien es un ejemplo de liderazgo y fortaleza, siendo el pilar fundamental de su familia, donde es un ejemplo de experiencia y sabiduría, además de ser muy querida por todos. Gracias a estas dos grandes mujeres, que (junto a mi hermana Javiera) me ayudaron a cambiar mi visión sobre la vejez, y darme cuenta que los adultos mayores aún tienen mucho que aportar a la sociedad.

Este trabajo de memoria ha sido parcialmente financiado por el proyecto FONDECYT N° 1191516.

Tabla de Contenido

1. Introducción	1
1.1. Contexto de trabajo	2
1.2. Problema abordado	3
1.3. Objetivos del trabajo	4
1.4. Resumen de la solución desarrollada	4
1.5. Estructura del documento	6
2. Marco Teórico	7
2.1. Aislamiento social en adultos mayores	7
2.2. Sistema SocialConnector	7
2.3. Sistema Visitrack	9
2.4. Registro de Interacciones Sociales	10
2.5. Visualización de datos	11
2.5.1. Plataforma Tableau	11
2.5.2. Biblioteca D3.js	12
2.5.3. Biblioteca Chart.js	13
2.6. Minería de datos	13
2.6.1. Series de tiempo	14
2.6.2. Clustering	16
3. Extensión del Modelo de Contexto	18
3.1. Identificación de conceptos	18
3.2. Definición de indicadores	21
3.3. Extensión del modelo de contexto	22
3.3.1. UserProfile	24
3.3.2. DailyContext	24
3.3.3. Pets	25
3.3.4. Events	25
3.3.5. InteractionWindows	26
3.3.6. WeeklyInteractions	27
3.4. Implementación del modelo de contexto	27
3.4.1. Modificación de los modelos en backend	27
3.4.2. Implementación del sistema de procesamiento automático de datos	28
3.4.3. Creación de vistas en frontend	33
4. Diseño e implementación de visualizaciones	42

4.1. Heatmap	43
4.1.1. Objetivo de la visualización	43
4.1.2. Descripción	43
4.2. Boxplot	50
4.2.1. Objetivo de la visualización	50
4.2.2. Descripción	50
4.3. Clustering	54
4.3.1. Objetivo de la visualización	54
4.3.2. Descripción	54
5. Diseño e implementación del Dashboard	58
5.1. Definición de información relevante	58
5.2. Diseño del dashboard	60
5.3. Implementación del dashboard	64
6. Validaciones	68
6.1. Extensión al modelo de contexto	69
6.1.1. DailyContext	69
6.1.2. InteractionWindows	70
6.1.3. WeeklyInteractions	72
6.2. Implementación de visualizaciones	73
6.2.1. Heatmap	73
6.2.2. Boxplot	76
6.2.3. Clustering	77
6.3. Resumen Semanal	78
6.4. Evaluación de utilidad	80
6.4.1. Metodología de evaluación	81
6.4.2. Evaluador N°1	82
6.4.3. Evaluador N°2	83
6.4.4. Discusión	83
7. Conclusiones y trabajo futuro	85
Bibliografía	88
Anexos	92
A. Roles dentro de una familia	92
B. Criterios de unión de clusters	93
C. Código modelo de contexto	94
C.1. UserProfile	94
C.2. Events	95
C.3. DailyContext	95
C.4. InteractionWindows	96
C.5. WeeklyInteractions	96
C.6. Pets	97
D. Detalle de la generación de registros InteractionWindows	98
D.1. Procesamiento de ventanas de no interacción físicas	98
D.2. Procesamiento de ventanas de no interacción digitales	99

E.	Otras visualizaciones exploradas	100
E.1.	Gráfica multilínea	100
E.2.	Serie de Tiempo con Autocorrelación	101

Índice de Tablas

6.1. Adultos mayores con sus respectivas ubicaciones, utilizados para la validación	69
6.2. Registros generados por la ejecución del <i>cron-job</i>	70
6.3. Datos climáticos y de feriados obtenidos desde los sitios AccuWeather y TimeAndDate	70
6.4. Interacciones de Tite Escalante el día 11/10/2020	71
6.5. Registros esperados del procesamiento de las ventanas sin interacción	71
6.6. Registros obtenidos del procesamiento de las ventanas sin interacción	72
6.7. Resumen de las interacciones registradas en la semana 28/12/20 - 03/01/21	72
6.8. Registros esperados de la ejecución del <i>cron-job</i>	73
6.9. Registros obtenidos de la ejecución del <i>cron-job</i>	73
6.10. Resultados esperados para el heatmap del escenario de ejemplo	74
6.11. Resultados obtenidos al consultar las series de tiempo dentro del heatmap del escenario de ejemplo	75
6.12. Resultados esperados para el boxplot del escenario de ejemplo	76
6.13. Series de tiempo esperadas para cada semana	78
6.14. Series de tiempo obtenidas para cada semana	79
6.15. Detalle de las interacciones registradas durante la semana 28/12/20-03/01/21	79
6.16. Valores esperados para el resumen semanal	80
1. Algunos de los métodos más utilizados unir clusters con algoritmos de clustering jerárquico [21]	93

Índice de Ilustraciones

1.1. Escenario de trabajo a abordar	2
1.2. Diagrama de flujo de la información capturada por el sistema	5
2.1. Interfaz de creación de usuarios de tipo “contacto” del adulto mayor	9
2.2. Interfaz <i>drag-and-drop</i> de Tableau	12
2.3. Diferencias de alineamiento entre el algoritmo de distancia euclideana y <i>Dy- namic Time Warping</i> (extraído de [47])	15
2.4. Ejemplo de una visualización de tipo dendrograma, extraído de [2]	17
3.1. Modelo de Contexto previo al desarrollo de la memoria (extraído de [40])	18
3.2. Modelo de contexto extendido	23
3.3. Formulario de edición de perfil antiguo	34
3.4. Interfaces de los nuevos formularios	35
3.5. Vista del detalle de un adulto mayor sin mascotas ni eventos	36
3.6. Formulario para añadir una mascota	36
3.7. Vista del detalle de un adulto mayor con mascotas	37
3.8. Vista de las mascotas de un adulto mayor	37
3.9. Formulario para añadir un evento	38
3.10. Vista del detalle de un adulto mayor con eventos	38
3.11. Vista de los eventos de un adulto mayor	39
3.12. Modal para actualizar el estado de las mascotas registradas en el sistema	39
3.13. Modal mostrado al usuario coordinador para actualizar los eventos de un adul- to mayor	40
4.1. Definición de bloques horarios	43
4.2. Heatmap vacío	44
4.3. Formulario para la gráfica de Heatmap	44
4.4. Calendario de Febrero 2021	45
4.5. Escala de color seleccionada	46
4.6. Visualización de heatmap resultante	46
4.7. Ejemplo de heatmap con un miembro de la familia desactivado	47
4.8. <i>Tooltip</i> desplegado al poner el cursor sobre un bloque horario	48
4.9. Ejemplo de serie de tiempo desplegada al hacer click en un bloque horario	49
4.10. <i>Tooltip</i> que muestra el detalle del valor de cada punto de la serie de tiempo	50
4.11. Ejemplo de gráfica de boxplot, extraído de [25]	51
4.12. Formulario para la gráfica de Boxplot	51
4.13. Visualización de boxplot resultante	52

4.14. Detalle de la gráfica de <i>outliers</i> dentro de los boxplot	53
4.15. <i>tooltip</i> que muestra el detalle de un <i>outlier</i>	53
4.16. Vista de detalle diario, mostrado al clickear un punto de la gráfica	54
4.17. Visualización de dendrograma resultante del algoritmo de clustering jerárquico	56
4.18. <i>Tooltip</i> que muestra a qué semana corresponde el nodo	56
4.19. Detalle de la serie de tiempo de una semana específica	57
4.20. Vista detalle de dos semanas del mismo cluster, desplegadas al mismo tiempo	57
5.1. Interfaz de dashboard extraída de [23]	60
5.2. Interfaz de dashboard extraída de dribbble.com	61
5.3. Interfaz de dashboard extraída de behance.net	61
5.4. Vista del dashboard sin resumen de adulto mayor	62
5.5. Vista del dashboard con resumen del adulto mayor	64
5.6. Implementación de la primera versión del dashboard en html	65
5.7. Versión final del dashboard en html	66
6.1. Heatmap generado para el caso de ejemplo	75
6.2. Detalle del bloque horario de interés	75
6.3. Información mostrada por el <i>tooltip</i> en el caso de ejemplo	76
6.4. Gráficas de Boxplot, solo para el bloque horario Miércoles 12:40-17:20	77
6.5. Resumen semanal generado para “Tite Escalante”	80
1. Diagrama del procesamiento de ventanas sin interacción física	98
2. Diagrama del procesamiento de ventanas sin interacción digital	99
3. Gráfica multilínea de interacciones físicas	100
4. Gráfica de la serie de tiempo junto con su desfase de mayor correlación . . .	102

Capítulo 1

Introducción

A lo largo de la historia humana, la tecnología ha sido vista como una herramienta de progreso, tanto en materias de producción, comunicación o transporte, por mencionar algunas. Durante el siglo pasado hubo un aumento sustancial en la cantidad de avances tecnológicos, lo cual trajo consigo muchos cambios en la vida de las personas. Por ejemplo, llegaron los computadores a las casas, se masificó el uso de automóviles y aumentó la expectativa de vida de las personas. Sin embargo, los avances tecnológicos no son tan fáciles de asimilar para algunos individuos, especialmente para los adultos mayores.

Los adultos mayores actuales crecieron en una época donde la mayoría de las tareas se realizaban a mano, y el principal medio de comunicación era cara-a-cara. Justamente, este último ítem es el que ha experimentado un cambio radical debido a la explosión de las nuevas tecnologías, reemplazando la comunicación cara-a-cara por una basada en su mayoría en interacciones digitales (por ejemplo, a través de aplicaciones de mensajería instantánea y servicios de redes sociales) [24]. Este escenario de interacción digital se ha convertido en un problema para los adultos mayores, debido a que (en su mayoría) su relación con la tecnología no es tan “fluida” como la de las nuevas generaciones [13]. Además, la incorporación de nuevas tecnologías a su vida cotidiana no es trivial, y supone una curva de aprendizaje importante [4].

Debido a esto, se termina generando una especie de “brecha comunicacional”, en la cual las generaciones más antiguas se van aislando cada vez más de las más jóvenes, ya que estos últimos van incorporando tecnologías que las otras no [20]. Dicha “brecha tecnológica” ha quedado particularmente expuesta en el contexto de la pandemia del COVID-19 [36], la cual ha llevado a adoptar medidas de distanciamiento social y restricciones para realizar reuniones sociales. Es en este contexto que la comunicación digital ha cobrado especial relevancia, planteándose como un medio seguro (en el sentido de que no requiere cercanía física) para mantener una comunicación constante y fluida con las personas cercanas [15].

Así pues, se vuelve clave que los adultos mayores sean incluidos dentro del paradigma de la comunicación digital [29]. Esto es, con el objetivo de evitar un aislamiento social que termine impactando de forma negativa en su salud, incrementando las posibilidades de sufrir demencia e incluso afecciones cardíacas [7].

1.1. Contexto de trabajo

Con el propósito de disminuir el riesgo de aislamiento social entre los adultos mayores, es que surgen las aplicaciones SocialConnector [28] y Visitrack [16]. Por un lado, SocialConnector es una aplicación que simplifica la relación del adulto mayor con las aplicaciones de mensajería instantánea y videollamadas (como WhatsApp, Skype o Telegram, entre otras), ofreciéndole una interfaz amigable para facilitar su comunicación con personas de su círculo cercano. Por otro lado, Visitrack es un sistema de monitoreo no invasivo, que permite registrar las visitas físicas que recibe el adulto mayor en su casa, mediante el registro de rostros a través de la cámara de un Tablet PC.

Además de ser una interfaz entre el adulto mayor y las distintas *apps* de mensajería, SocialConnector permite registrar las interacciones digitales que tiene un adulto mayor mediante la plataforma. Con esto, es posible notar que con el trabajo en conjunto de ambas aplicaciones (SocialConnector y Visitrack), se puede registrar la gran mayoría de las interacciones sociales que tenga el adulto mayor (salvo las que ocurran fuera de SocialConnector o fuera del hogar del adulto mayor). Esto abre una gran oportunidad para desarrollar sistemas que sean capaces de analizar dichas interacciones, ya sea para generar conocimiento sobre las relaciones interpersonales de las personas mayores, o para monitorear sus niveles de interacción, a fin de mantenerlos en un “nivel sano”.

En la Figura 1.1 se puede observar la arquitectura del escenario de trabajo de esta memoria. Allí se puede ver a los sistemas SocialConnector y Visitrack como los registradores de la información sobre interacciones sociales (digitales y físicas) de los adultos mayores, la cual se almacena en un repositorio central. En este trabajo de título se extendió dicho repositorio, y se desarrollaron dos sistemas satélites: (1) el generador de indicadores de interacción social y (2) el monitor de interacciones sociales. En resumen, el alcance de este trabajo de memoria se limitó únicamente a los componentes mostrados en amarillo.

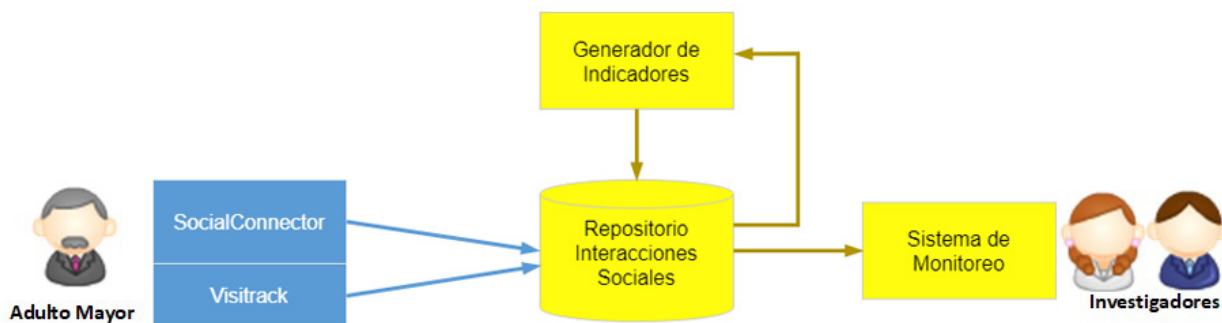


Figura 1.1: Escenario de trabajo a abordar

Este conjunto de sistemas se construyó con fines científicos, como una forma de monitorear la actividad social de los adultos mayores de manera objetiva, sin necesidad de tener que instrumentar el hogar de estas personas. Por lo tanto, los beneficiarios directos de este trabajo son los investigadores que estudian las actividades sociales de los adultos mayores; en este caso, con el fin de que eso les ayude a comprender de mejor manera la relación de estos con su círculo cercano.

1.2. Problema abordado

Tanto SocialConnector como Visitrack son capaces de registrar las interacciones sociales del adulto mayor, ya sean estas virtuales o físicas, almacenando la información en bases de datos independientes. Recientemente se realizó un trabajo de título [40], cuyo propósito fue integrar la información de ambas fuentes, y así conformar un repositorio central (mostrado en la Figura 1.1) que facilite el procesamiento de dicha información.

Para evitar confusiones, por “estado actual del sistema” nos referiremos a la última versión del sistema, la cual corresponde justamente al producto final de la memoria de Sebastián Toro [40]. Esta versión del sistema es capaz de almacenar las interacciones digitales y físicas de los adultos mayores en un repositorio de datos centralizado y fue la base de todos los desarrollos implementados durante este trabajo de título.

El repositorio de información implementado en dicha memoria permite alcanzar la primera etapa del modelo planteado por Gutiérrez et al. [19]. Este modelo plantea una estrategia para mediar las asimetrías dentro de la comunicación familiar (derivadas de la brecha tecnológica existente) y contempla cuatro etapas: (1) recopilación de datos, (2) monitoreo, (3) toma de decisiones e (4) intervención. Actualmente, el sistema es capaz de integrar la información extraída de ambas aplicaciones, homogeneizando y almacenando los datos en el repositorio.

A pesar de que se tuvieron buenos resultados en la implementación de la primera etapa (es decir, construir un repositorio central con información integrada), aún quedaba pendiente el desarrollo del resto de los componentes que permiten interpretar dichos datos, con la finalidad de prevenir un eventual aislamiento social del adulto mayor. Si bien la información era integrada y almacenada de forma correcta, esta solo contemplaba datos concretos sobre interacciones sociales del adulto mayor estudiado, no incorporando otras variables que pudieran ayudar a entender el contexto en el cual ocurren dichas interacciones. Ejemplos de estas variables de contexto son el clima, los días feriados/festivos, y la ubicación de las personas involucradas en la interacción.

También se presentaba el problema de que los datos almacenados por el sistema no eran sometidos a ningún tipo de procesamiento, por lo cual, en caso de querer monitorear efectivamente la salud social del adulto mayor, era necesario hacer dicho procesamiento a mano. Al respecto, según la Universidad Internacional de Valencia (VIU) [46], el concepto de salud social puede ser definido como “la habilidad para la adaptación y la autogestión ante los cambios y desafíos del entorno, así como la capacidad para desarrollar relaciones satisfactorias con otras personas”. Así pues, en este documento nos referiremos a la salud social de un adulto mayor como una escala, asociada a la cantidad de interacciones físicas y digitales que tiene una persona con otros miembros de su círculo cercano.

La falta de mecanismos de procesamiento automático de los datos de interacciones sociales limita la capacidad de los sistemas para enviar notificaciones, tomar acciones paliativas de forma temprana, o bien contar con un diagnóstico informal de la salud social del adulto mayor en tiempo real. Esto también limita las oportunidades de usar la información capturada por las aplicaciones antes mencionadas, para realizar investigación; por ejemplo, estudios de psicología, sociología o ciencias sociales por mencionar algunas.

Lo antes mencionado muestra que el sistema actual carecía de infraestructura para facilitar el análisis de los datos sobre las interacciones sociales de los adultos mayores. Esto se debe a que existen fuentes de datos que se encuentran parcialmente integradas, sobre las cuales faltaba definir marcadores semánticos de contexto, a fin de poder extraer información de ellas y generar conocimiento útil para el área del cuidado informal de adultos mayores.

El trabajo de memoria reportado en este documento apuntó a resolver estas falencias, mediante la extensión del modelo de datos contextual actual, para que permitiera capturar variables relevantes para la comprensión de los datos almacenados por los sistemas SocialConnector y Visitrack. Además, se diseñó e implementó un mecanismo de procesamiento automático de los datos, con la finalidad de brindar a los investigadores la posibilidad del monitoreo en tiempo real de los niveles de interacción social del adulto mayor, y la identificación de patrones no triviales dentro de la información capturada. Todo esto, considerando que el sistema mantenga su naturaleza no invasiva del espacio personal del adulto mayor.

1.3. Objetivos del trabajo

El objetivo general de este trabajo de memoria es extender el modelo de contexto que posee actualmente el sistema e implementar un mecanismo de procesamiento automático de los datos de interacciones almacenadas por las aplicaciones SocialConnector y Visitrack. Con esto se busca poder exponer información relevante en un dashboard, que permita realizar un diagnóstico informal de la “salud social” de un adulto mayor.

A fin de alcanzar el objetivo general enunciado anteriormente, se plantearon los siguientes objetivos específicos:

1. Capturar variables que sean relevantes para la comprensión de las interacciones sociales del adulto mayor.
2. Explorar, diseñar e implementar visualizaciones que utilicen los datos almacenados por los sistemas SocialConnector y Visitrack, con la finalidad de apoyar el entendimiento de los mismos.
3. Diseñar e implementar un sistema capaz de procesar los datos almacenados por los sistemas SocialConnector y Visitrack para alimentar tanto el modelo de contexto como las visualizaciones implementadas.
4. Definir indicadores útiles para analizar el nivel de interacción social de los adultos mayores, y verificar la correctitud del cálculo de dichos indicadores.
5. Implementar y validar un dashboard que reporte métricas relevantes para el monitoreo de la actividad social del adulto mayor.

1.4. Resumen de la solución desarrollada

Como se mencionó en la sección 1.2, el sistema actual se compone de cuatro etapas. La solución desarrollada apuntó a la intervención de las etapas (1) *recopilación de datos* y (2) *monitoreo* de las actividades sociales de las personas de la tercera edad.

Para ilustrar de mejor manera el alcance de este trabajo de memoria, en la Figura 1.2 se

exhibe una versión simplificada del sistema. En esta figura se puede apreciar el flujo que sigue la información capturada por el mismo. Actualmente, las componentes de SocialConnector (en color verde) y Visitrack (en color celeste) se encuentran implementadas y en funcionamiento, por lo cual se encuentran fuera del alcance de la memoria. En amarillo se muestra el sistema de procesamiento y monitoreo de los datos recolectados, que involucra los componentes que se construyeron/intervinieron en este trabajo.

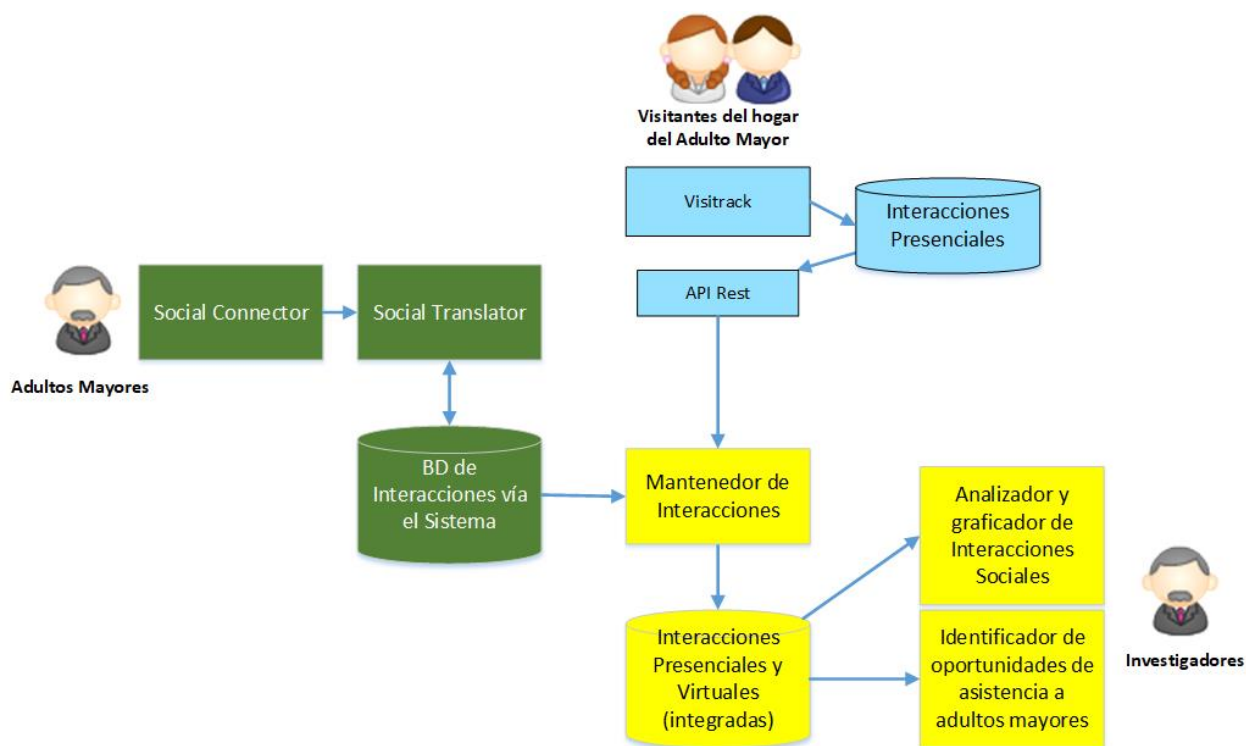


Figura 1.2: Diagrama de flujo de la información capturada por el sistema

A nivel de la etapa de recopilación de datos, se extendió el modelo de contexto con variables de relevancia para el análisis de las interacciones sociales. Para determinar dichas variables, en primer lugar se definió un conjunto inicial de estas, para luego iterar sobre ellas y así descartar algunas o incorporar nuevas, en base a lo que indicó la literatura del área y la discusión de las mismas con expertos de dominio.

Respecto a la etapa de monitoreo, se incorporó un sistema de procesamiento automático de los datos de interacciones almacenados por SocialConnector, Visitrack, y las variables de contexto que se definan, para su posterior almacenamiento en un repositorio centralizado. También se exploró el diseño de visualizaciones que pudieran favorecer una mejor comprensión de los datos procesados.

Finalmente, se definieron indicadores que describen y caracterizan las interacciones de un adulto mayor, y que permiten realizar un diagnóstico informal de su salud social. Estos indicadores fueron representados visualmente en un dashboard, el cual es el producto final de cara al usuario.

Este dashboard es utilizado para el análisis de las interacciones según lo que los investigadores estimen conveniente. Debido a ello, durante el desarrollo de este trabajo de título se

trabajó con investigadores (potenciales usuarios de este sistema) para determinar la utilidad y correctitud de la información reportada por el sistema, así como la entendibilidad de la información entregada a través de las visualizaciones y el dashboard.

En resumen, la solución desarrollada en este trabajo de título apuntó a resolver el problema planteado en la sección 1.2, a través de la inclusión de cuatro mecanismos: (1) la extensión del modelo de contexto de la herramienta, (2) el desarrollo de visualizaciones que utilizan la información almacenada por el sistema y facilitan la comprensión de la misma, (3) la definición e implementación de indicadores que permitan obtener un mayor entendimiento del escenario en el cual se dan las interacciones sociales del adulto mayor, y (4) el desarrollo de un dashboard que facilita el entendimiento y el análisis de las interacciones. Haciendo uso de esta información es posible realizar un diagnóstico informal, en tiempo real, de la “salud social” de un adulto mayor.

1.5. Estructura del documento

El resto del documento se estructura de la siguiente forma:

- En el Capítulo 2 se expone el marco teórico, donde se explican algunos conceptos que fueron claves durante la realización de este trabajo de título. Además, describe el funcionamiento de las aplicaciones SocialConnector y Visitrack.
- En el Capítulo 3 se explica el proceso de concepción e implementación de la extensión al modelo de contexto del sistema SocialConnector.
- En el Capítulo 4 se ahonda en el proceso de exploración, diseño e implementación de las visualizaciones que utilizan los datos de las interacciones registradas. Además, se explica el proceso de cada visualización implementada, sus objetivos, cómo estas usan los datos y sus interacciones con el usuario.
- El Capítulo 5 describe el proceso de diseño e implementación del dashboard. Esto corresponde principalmente a la definición de métricas relevantes para incluir en el dashboard, el diseño de dicho dashboard, y su implementación en código.
- El Capítulo 6 presenta las validaciones realizadas para legitimar los desarrollos realizados durante el trabajo de título. Se analiza la correctitud de los datos resumen entregados por el sistema, el comportamiento de las visualizaciones y del dashboard frente a una base de datos de prueba. También se exhiben los resultados de la evaluación de utilidad realizada con usuarios del sistema.
- En el Capítulo 7 se exponen las conclusiones y el trabajo futuro relativo a esta memoria. Además, se explica en qué medida los desarrollos implementados contribuyen a la comprensión de los datos almacenados, y qué mejoras se pueden realizar en un futuro cercano.

Capítulo 2

Marco Teórico

Este capítulo presenta los conceptos, tecnologías y sistemas que fueron más relevantes durante el desarrollo de este trabajo de título.

2.1. Aislamiento social en adultos mayores

Con la finalidad de mantener su independencia, muchos adultos mayores optan por envejecer en su propio hogar, en lugar de trasladarse a vivir con familiares o a residencias para personas de tercera edad [49]. Mantener esta sensación de independencia es clave para estas personas, ya que ese aspecto está estrechamente relacionado con el bienestar, la autoestima y la buena salud mental de ellos [44]. Sin embargo, si dicha independencia es fomentada sin preocuparse de mantener al adulto mayor en contacto con sus seres queridos, puede llegar a ser incluso contraproducente, empujándolo a una situación de aislamiento social [6].

El aislamiento social en los adultos mayores es un problema que puede traer múltiples consecuencias para la persona, pudiendo causar trastornos de ansiedad, depresión, el aumento de los niveles de cortisol en la sangre, alteración de los patrones de sueño, e incluso aumenta el riesgo de sufrir problemas cardíacos [1, 39]. Esta problemática social se acentúa aún más en los tiempos actuales de pandemia, ya que debido a las medidas de distanciamiento social, los adultos mayores han visto disminuidas sus interacciones físicas con su círculo cercano, teniendo que forzarse a utilizar medios digitales para comunicarse y mantenerse en contacto [36]. Así, se hace necesario ponerle un especial énfasis a este tópico en los tiempos actuales, a fin de velar por la buena salud mental y un “buen envejecer” de las personas de la tercera edad.

2.2. Sistema SocialConnector

SocialConnector [28] es un sistema que, como se mencionó anteriormente, simplifica la interacción del adulto mayor con las aplicaciones de mensajería instantánea, agregando una capa de abstracción adicional. Este sistema tiene dos componentes principales, una aplicación móvil y una aplicación web.

En cuanto a la aplicación móvil de SocialConnector, es la que hace de nexo entre el adulto mayor y las *apps* de mensajería. Su principal característica es que quien la utilice, no necesita saber el canal específico por el cual se realiza la comunicación (por ejemplo, WhatsApp, Skype o Telegram). Esto es porque la aplicación reúne todos los canales de comunicación disponibles en una única interfaz, ahorrándole al adulto mayor el esfuerzo de aprender a manejar cada aplicación independientemente.

Para que dicha capa de abstracción funcione, SocialConnector utiliza otro sistema llamado Social Translator [33], el cual se encarga de identificar a qué canal específico pertenece cada mensaje. Esto es, actúa como un nexo entre SocialConnector y las distintas *apps* de mensajería soportadas, además de registrar las interacciones digitales del adulto mayor en una base de datos.

Por otro lado, la aplicación web de SocialConnector sirve como una plataforma de administración del sistema. Esta aplicación web está implementada sobre el *framework* de desarrollo web Django, el cual utiliza Python como su lenguaje base. Dentro de esta plataforma, hay tres tipos de usuarios, el usuario administrador, el usuario coordinador y el usuario contacto. A continuación se describe brevemente el rol de cada uno de estos usuarios:

- **Usuario Administrador:** Como su nombre lo indica, este usuario actúa como administrador de la plataforma. Por lo tanto, tiene la capacidad de gestionar (añadir/editar/borrar) usuarios coordinadores y las redes sociales que forman parte de la plataforma. Además, puede acceder a los datos almacenados en la plataforma (en forma de tablas) y exportarlos (en CSV, Excel o PDF).
- **Usuario Coordinador:** Antes de definir el rol del usuario coordinador en la plataforma, es necesario explicar cómo se estructura una familia dentro de SocialConnector. Dentro de la plataforma, una familia está compuesta por un adulto mayor, un coordinador de familia y cero o más miembros de la comunidad familiar.

El flujo actual de creación de una familia requiere que el usuario coordinador agregue a un adulto mayor dentro de la plataforma. La creación de este usuario automáticamente genera una nueva familia en el sistema SocialConnector, donde están incluidos el usuario coordinador y el adulto mayor recién creado.

El usuario coordinador puede además crear contactos del adulto mayor (estos son otros usuarios), como se muestra en la Figura 2.1. Estos contactos pueden ser agregados a las familias de adultos mayores existentes. Al añadir un nuevo contacto a la familia de un adulto mayor, es necesario señalar qué rol cumple esta persona dentro de la familia. Los roles que pueden tomar los contactos están detallados en el anexo A. En síntesis, el usuario coordinador asume el rol de gestionar las familias dentro de la plataforma mediante la creación, edición o eliminación de adultos mayores y/o contactos, junto con poder incluir/remover contactos de una familia.

- **Usuario Contacto:** El usuario contacto solo puede acceder a la plataforma para modificar la información relativa a su perfil de usuario. Este puede ser un adulto mayor o un integrante de su familia y no tiene otros permisos más que la edición de su información personal. En resumen, no se espera que este tipo de usuario use la plataforma muy frecuentemente.

Es necesario mencionar que el repositorio central de interacciones se almacena en la apli-

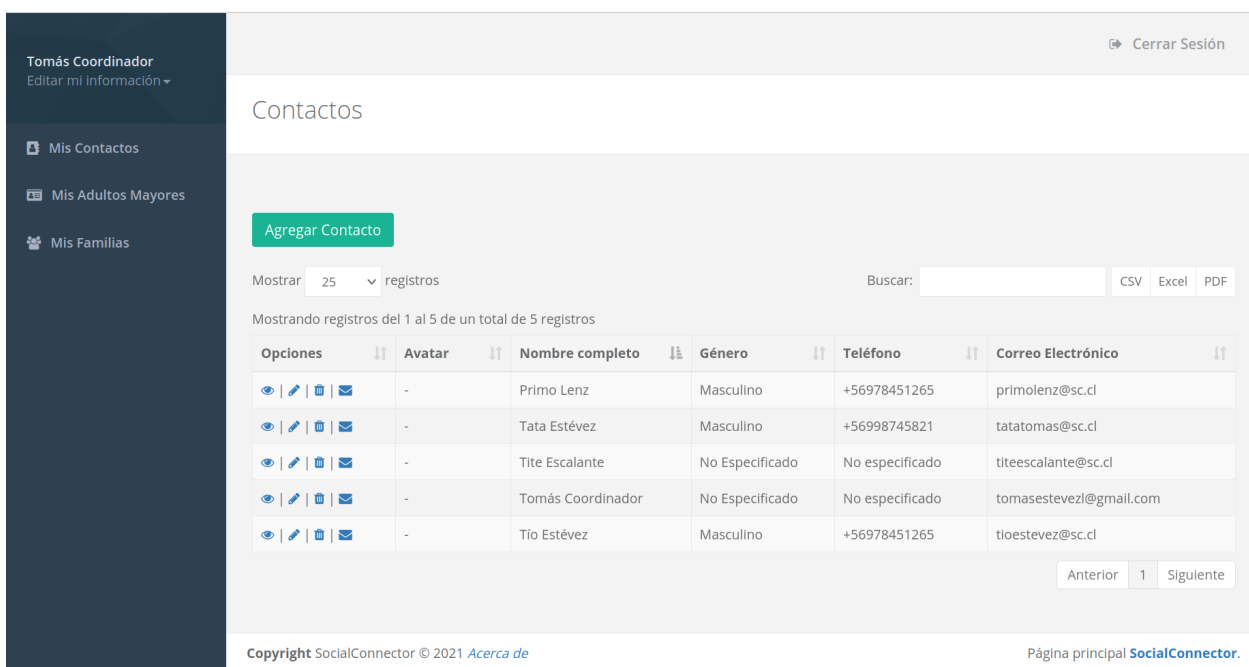


Figura 2.1: Interfaz de creación de usuarios de tipo “contacto” del adulto mayor

cación web de SocialConnector. Esto es de especial relevancia, debido a que este trabajo de título se basa en dicho repositorio.

2.3. Sistema Visitrack

Visitrack [16] es una aplicación para Tablet PC que registra las visitas físicas que recibe un adulto mayor en su hogar, mediante el reconocimiento facial a través de la cámara de dicho dispositivo. Esta aplicación permite reconocer si la visita pertenece o no al círculo cercano del adulto mayor, comparándolo con los miembros del círculo familiar registrados en el sistema. Para lograr este objetivo, Visitrack ejecuta las siguientes tareas:

1. Monitorea constantemente el sector del hogar del adulto mayor donde está ubicado el Tablet PC (la aplicación corre en segundo plano, por lo cual el monitoreo es permanente). Si durante este monitoreo, se detecta una o más personas, entonces el sistema captura una imagen con la cámara frontal, recorta los rostros que aparecen en la imagen, y los envía al servidor de la aplicación para su posterior análisis.
2. En el servidor, Visitrack analiza las imágenes recibidas e intenta identificar a quién corresponde cada rostro; este reconocimiento se realiza considerando solo los contactos del adulto mayor. En caso de que el rostro no sea reconocido por el sistema, el coordinador de la familia puede “taggear” manualmente la imagen, indicándole al sistema el nombre de la persona que aparece en la imagen. De esa manera, el sistema podrá identificar en el futuro a esa persona, cuando ella realice nuevas visitas.
3. Por último, Visitrack guarda el registro de cada visita, con la hora de inicio, hora de término, y los participantes de la misma (los rostros que detectó el sistema).

Para capturar de forma correcta las visitas, el sistema divide el tiempo de monitoreo en

bloques de igual duración (por ejemplo, en bloques de 2 o 3 minutos). Si Visitrack detecta un rostro en un bloque, y luego lo vuelve a detectar en un bloque de tiempo cercano (por ejemplo, en un rango de 1-4 bloques posterior a la primera detección), entonces Visitrack asume que se trata de la misma visita. En caso contrario, el sistema las registra como visitas distintas [16]. Esta corrección es importante, ya que como el sistema se basa en la cámara frontal del Tablet para capturar rostros, y este está en una posición fija dentro de la casa, no necesariamente toda la visita transcurrirá en un lugar visible para la cámara del dispositivo.

Un punto importante de mencionar, es que pueden existir dilemas éticos de privacidad con los datos que genera Visitrack. Esto, debido a que el sistema al estar monitoreando constantemente el hogar del adulto mayor, los datos que captura sobre las visitas no siempre tienen el consentimiento explícito de quien visita al adulto mayor. Sin embargo, este dilema ético escapa al alcance de este trabajo de título, ya que durante este solo se trabajó en el procesamiento de los datos provenientes de Visitrack y no sobre el proceso de recolección de dichos datos.

Además, dicho dilema ético es planteado por los creadores de Visitrack [16], donde reconocen que si bien existen dichas brechas de privacidad, estas se encuentran controladas en el manejo y la gestión de datos dentro del sistema. También señalan que la percepción de uso de la herramienta Visitrack por parte de los usuarios finales sobrepasa la brecha de privacidad identificada.

2.4. Registro de Interacciones Sociales

Tanto las interacciones digitales como las interacciones físicas registradas por ambos sistemas (es decir, SocialConnector y Visitrack) son procesadas, homogeneizadas en su formato y almacenadas en la base de datos de SocialConnector. Para realizar esta integración de información de las interacciones, se diseñó una estructura de datos que sirviera para guardar tanto interacciones físicas, como las digitales. Esta estructura fue diseñada e implementada como parte del trabajo de título de Sebastián Toro [40]. Los campos que posee dicha estructura son los siguientes:

- **elderly:** Corresponde al adulto mayor que participa de la interacción.
- **counterpart:** Este es el miembro de la comunidad familiar con el que interactuó el adulto mayor.
- **type:** Indica el tipo de la interacción (física o digital) registrada.
- **source:** Indica la fuente de la interacción, es decir, desde qué plataforma se recibió/envió el mensaje (por ejemplo, WhatsApp, Telegram, o Skype). Si la interacción es física, entonces se marca Visitrack como fuente.
- **isIncoming:** Indica si la interacción es saliente o entrante; es decir, si el adulto mayor es quien llama a la otra persona, o es al revés. Cuando se trata de una visita, el sistema siempre marca a esas interacciones físicas como entrantes.
- **init:** Este es un timestamp que indica cuándo inicia la interacción.
- **end:** Representa el timestamp en el cual termina la interacción.
- **duration:** Indica la duración total de la interacción en segundos.

- **fileType:** Indica el tipo de archivo que se adjuntó al mensaje, si es que hay alguno. El tipo de archivo puede tomar cuatro valores, “audio”, “video”, “imagen” o “ninguno” (en caso de no haber adjunto, o ser una interacción física).

Así, cada vez que se hable acerca de “interacciones” en el resto del documento, se estará haciendo referencia a esta superestructura.

2.5. Visualización de datos

El concepto de visualización de datos refiere a las técnicas que se emplean para transmitir o comunicar información contenida en datos a través de elementos visuales [5]. Esto hace más comprensible la información para el cerebro humano.

Este campo de la computación ha tomado especial relevancia el último tiempo. Debido al avance de las tecnologías y la digitalización de tareas y/o sistemas que antes eran manuales, el volumen de datos que se crea cada día en el mundo ha ido en aumento, y todo indica que debería seguir creciendo conforme pase el tiempo [48]. Este gran volumen de información se suma al hecho de que los estudios señalan que los humanos somos seres intrínsecamente visuales [34], lo cual lleva a que la visualización de datos sea fundamental para poder generar conocimiento en base a los datos almacenados.

A continuación se enuncian algunas herramientas útiles a la hora de generar visualizaciones de datos. Por temas prácticos, se mencionan únicamente las herramientas que fueron relevantes en el contexto de esta memoria.

2.5.1. Plataforma Tableau

Tableau¹ es una plataforma de procesamiento y visualización de datos, capaz de generar dinámicamente diversas representaciones visuales, que permiten al usuario hacer cruces de datos de manera interactiva. La interfaz gráfica de Tableau permite cargar y conectar bases de datos, ver las columnas/atributos que tienen las entidades de datos, e implementa un sistema de *drag-and-drop*, donde el usuario puede seleccionar la columna que desea, y arrastrarla hasta la sección de la visualización deseada. Por ejemplo, si se desea que una columna actúe como “eje X” de un gráfico, se puede tomar la columna y arrastrarla hasta el eje X de la visualización. En la Figura 2.2 se puede observar la interfaz mencionada.

Como se mencionó anteriormente, Tableau es una herramienta capaz de generar visualizaciones con un sistema *drag-and-drop*. Este punto es particularmente interesante para este trabajo de título, debido al potencial que presenta para generar cruces de datos y explorar distintas visualizaciones sin la necesidad de invertir tiempo en la programación de las mismas. Así, esta herramienta podría ser de suma utilidad para generar conocimiento en base a los datos almacenados por SocialConnector.

Aunque es útil y muy usado [14], Tableau presenta dos limitaciones importantes. La primera, es que es una aplicación pagada, llegando a costar desde \$70 USD mensuales² según

¹Tableau - Home Page

²Tableau - Pricing

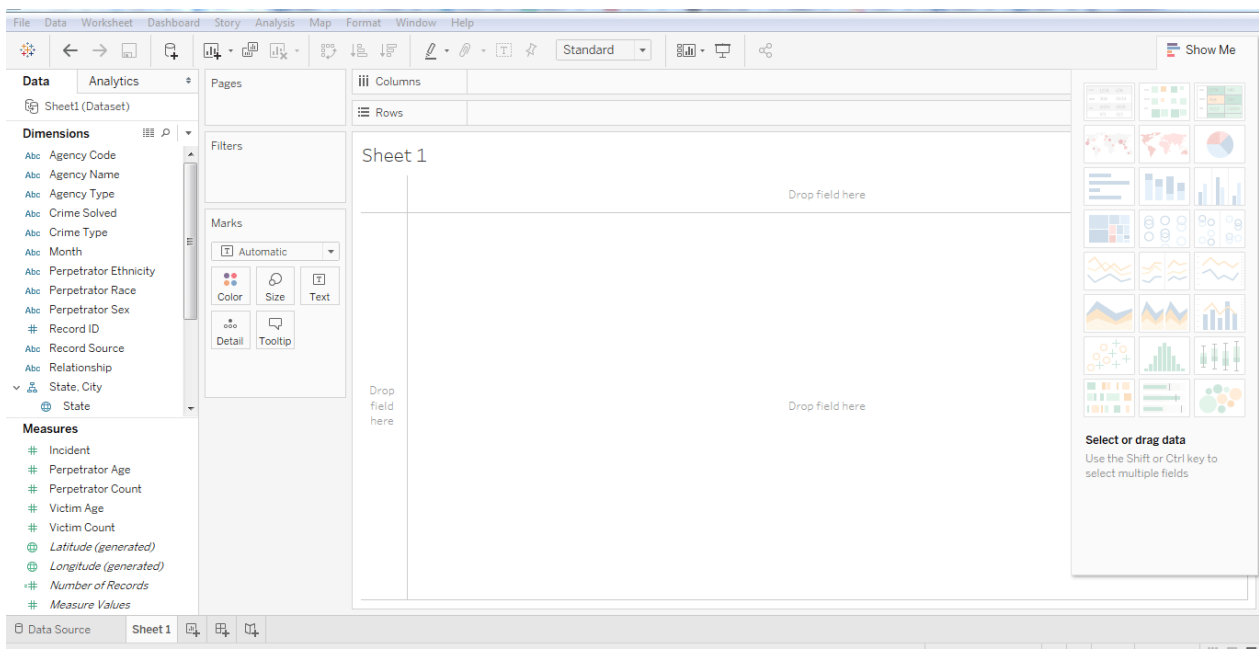


Figura 2.2: Interfaz *drag-and-drop* de Tableau

las funcionalidades que se necesiten, encareciendo así el costo de operación del sistema. Otra limitación es que, al ser una aplicación externa al sistema, es necesario desarrollar una infraestructura que permita la integración entre SocialConnector y Tableau, lo cual puede ser no trivial. Además, la integración de un sistema externo aumenta las dependencias con el mismo, lo cual puede no ser conveniente; por ejemplo, en el caso de que Tableau presente intermitencias en su servicio.

2.5.2. Biblioteca D3.js

D3.js es una biblioteca de JavaScript cuya principal característica es que da al programador la posibilidad de manipular los DOM html en base a datos; esto se conoce como *Data-Driven Document*. En otras palabras, la biblioteca permite definir comportamientos sobre la vista, los cuales son dependientes de los datos con los que se alimente dicha vista. Esto hace que la librería sea naturalmente útil para crear visualizaciones que ayuden al usuario a comprender la información que se le muestra de manera gráfica y dinámica.

Al ser una biblioteca de JavaScript, D3.js es sumamente útil a la hora de incorporar gráficas dentro de una aplicación web. Además, al ser de naturaleza *open-source*, su código puede ser usado e incluso descargado de forma gratuita, lo cual hace que su integración con sistemas ya existentes sea fácil y sencilla, ya que solo bastaría descargar el código fuente e incorporarlo al directorio de la aplicación.

A diferencia de Tableau, para incorporar D3.js a SocialConnector no es necesario el desarrollo de una infraestructura para la integración de ambos sistemas, debido a la característica mencionada en el párrafo anterior. Sin embargo, la implementación de visualizaciones basadas en D3.js requiere un mayor esfuerzo de programación y manejo de tecnologías de *front-end* (HTML y JavaScript principalmente), debido a que es necesario programar el comporta-

miento de las visualizaciones y su presentación dentro de la aplicación (por ejemplo, posición dentro del DOM). En este aspecto, D3.js es diferente a Tableau, ya que este último oculta dicha capa de programación al usuario con su dinámica de *drag-and-drop*.

Un punto a favor de D3.js es que esta biblioteca manipula directamente el DOM de la página donde se inserta. Esto implica que las visualizaciones que se pueden generar son ilimitadas, debido a que no es una biblioteca de “visualizaciones predefinidas” (como sí lo es Chart.js por ejemplo), sino que tiene la capacidad de añadir dinámicamente elementos geométricos (SVG, *rect*, *circle*, etc.) al cuerpo del html. En pocas palabras, puede “dibujar” directamente en el html según los datos que se le entreguen. Sin embargo, este mayor grado de libertad para generar visualizaciones requiere de una curva de aprendizaje mucho más pronunciada con respecto a otras opciones disponibles.

2.5.3. Biblioteca Chart.js

Chart.js es una biblioteca de JavaScript enfocada en generar gráficas con los datos que le son facilitados. A diferencia de D3.js, Chart.js no manipula directamente el DOM, sino que la biblioteca contempla una serie de gráficas preestablecidas, las cuales “esconden” al programador la complejidad de dibujar en el DOM directamente. Entre las gráficas que incluye esta biblioteca están los gráficos de línea, de barras, de radar, de torta, de área, bubble plot, scatter plot y mixtas (combinación de gráficas)³.

Si bien Chart.js no tiene la flexibilidad que sí entrega D3.js, la complejidad de aprender a utilizar esta librería es mucho menor, debido a que las gráficas predefinidas actúan como plantillas. A estas sólo es necesario entregarle los datos que se quieren graficar, y con eso basta para generar la visualización. Además, cada plantilla tiene predefinida su interacción con el usuario (a través de tooltips, eventos de click, etc.), por lo cual le ahorra al programador gran parte del trabajo de el diseño de la interacción entre la gráfica y el usuario final. Para usuarios avanzados, esta biblioteca ofrece la posibilidad de personalizar esta configuración, y otras más.

Chart.js apunta a hacer fácil la generación de visualizaciones más comunes, por ejemplo, gráficas de barras, líneas y todas las mencionadas anteriormente. Esto la convierte en una buena alternativa para generar gráficas simples. Sin embargo, esto puede no ser lo más óptimo si lo que se requiere es una gráfica más compleja, o una visualización con un alto nivel de personalización.

2.6. Minería de datos

La minería de datos se puede definir como el proceso de analizar grandes volúmenes de datos con la finalidad de generar conocimiento sobre estos [42]. Este análisis puede realizarse con el objetivo de identificar patrones, resolver problemas o detectar anomalías, entre otros.

Al igual que la anteriormente mencionada visualización de datos, la minería de datos ha cobrado gran importancia en las últimas décadas, beneficiándose de los avances en hardware, los que permiten llevar a cabo proyectos de minería de datos de gran escala [10]. A pesar

³ChartJS Documentation

del gran volumen de datos disponible en la actualidad [48], muchas veces los datos no son fácilmente interpretables o esconden patrones, anomalías o tendencias que no son detectables a simple vista. Es en estos casos en los que la aplicación de técnicas de minería de datos se vuelve sumamente relevante; es decir, para poder comprender los datos y descubrir características que pueden estar ocultas en ellos.

A continuación, se mencionarán ciertos conceptos relativos a la minería de datos, los cuales fueron útiles a lo largo de este trabajo de título.

2.6.1. Series de tiempo

Una serie de tiempo es una colección de medidas cuantitativas sobre una determinada variable, la cual es medida en intervalos de tiempo regulares, y las mediciones están ordenadas cronológicamente [35]. Mas informalmente, es un arreglo de valores donde cada valor tiene una marca de tiempo asociada (esta puede ser una hora, una fecha, un mes, un año, etc.), y cada punto representa el valor que la variable de interés tenía en su marca de tiempo asociada.

Las series de tiempo son muy útiles para poder observar la evolución de una variable o indicador a través del tiempo. Un ejemplo de ello, es su uso en economía, donde una práctica habitual es el análisis de la “historia” (serie de tiempo) de un indicador económico para realizar proyecciones sobre su comportamiento futuro.

Dentro de la minería de datos, es habitual encontrarse con datos organizados en forma de series temporales. Algunos de los estudios más comunes realizados sobre series de tiempo son los siguientes:

- **Promedios móviles:** Sea P una serie de tiempo cualquiera, el promedio móvil de tamaño de ventana x del punto a , se define como: $P_a = \frac{\sum_{i=0}^{x-1} P_{a-i}}{x}$. Esta medición es potencialmente útil para detectar tendencias (simples) en los datos. Sin embargo, esta es dependiente del tamaño de la ventana (x) que se elija.
- **Autocorrelación:** La autocorrelación de una serie de tiempo implica calcular la correlación de la serie de tiempo, con una copia de la misma, pero desfasada en x pasos. Es decir, el punto correspondiente al casillero 1 en la serie original, correspondería al casillero $1+x$ en la serie x -desfasada. Según el valor que tome la correlación, se puede analizar si los valores de la serie de tiempo dependen de valores pasados (correlación ~ 1 o ~ -1) o no (correlación ~ 0).
- **Detección de Anomalías:** La detección de anomalías en una serie de tiempo se puede realizar de múltiples formas, las cuales generan diversos resultados. Dentro de las formas de detección de anomalías dentro de una serie, existen enfoques estadísticos, por ejemplo, definiendo que cualquier punto que tome un valor fuera del rango $[\mu - 2\sigma, \mu + 2\sigma]$ es una anomalía. También existen enfoques basados en *Machine Learning*, como pueden ser técnicas basadas en clustering (explicada en la siguiente subsección).

Por otra parte, cuando se tienen múltiples series de tiempo, es interesante compararlas y ver que tan similares o distintas son unas con otras. Una forma de medir la similitud entre dos series de tiempo es el método llamado *Dynamic Time Warping*, que se describe a

continuación.

2.6.1.1. Dynamic Time Warping

Dynamic Time Warping (DTW) (o alineamiento temporal dinámico) es un método que compara dos series de tiempo, y tiene la capacidad de detectar similitudes entre ellas a pesar de que estén desfasadas u ocurran a distintas velocidades [32]. Por ejemplo, puede ocurrir que ambas series tengan un *peak* similar, pero que en una de ellas este *peak* sea más duradero que en la otra.

Su principal diferencia con respecto al algoritmo de distancia euclidiana, es que DTW permite hacer un mapeo no-lineal entre las series de tiempo [32]. En la Figura 2.3 es posible observar cómo el algoritmo euclidiano compara cada punto de la serie azul, con su símil (punto en la misma posición) de la serie roja. Por otra parte, el algoritmo DTW puede mapear un punto de la serie azul, con un punto que no necesariamente sea su símil en la serie roja. Específicamente, para cada punto de la serie azul se busca un punto en la serie roja con el cual su distancia sea mínima (según una medida de distancia definida por el investigador). El cálculo de esta distancia se explica a continuación.

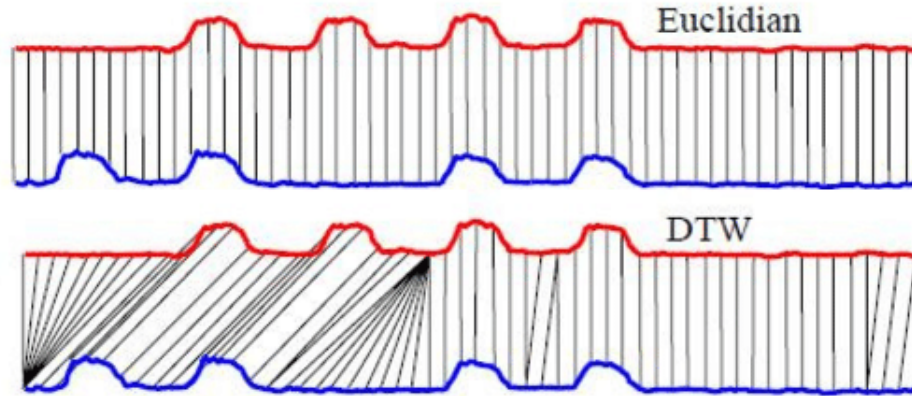


Figura 2.3: Diferencias de alineamiento entre el algoritmo de distancia euclideana y *Dynamic Time Warping* (extraído de [47])

Teniendo dos series arbitrarias A y B, de largos n y m , para calcular la matriz de distancia entre ellas usando DTW, se sigue el siguiente procedimiento:

1. Se inicializa una matriz M de dimensiones $n \times m$.
2. Primero, se completa el casillero $M_{(0,0)}$ con el valor $|A_0 - B_0|$.
3. Luego, para casilla en M , se establece que:

$$M_{(i,j)} = |A_i - B_j| + \min\{M_{(i-1,j-1)}, M_{(i-1,j)}, M_{(i,j-1)}\}$$

Para los casos de borde, en que $i - 1 < 0$, se ignoran los valores $M_{(i-1,j-1)}$ y $M_{(i-1,j)}$ en la ecuación. Esto es similar para el caso en que $j - 1 < 0$, donde se ignoran los valores $M_{(i-1,j-1)}$ y $M_{(i,j-1)}$. Cabe hacer notar que los casos $i - 1 < 0$ y $j - 1 < 0$ no se pueden dar conjuntamente, ya que corresponde al punto $(0,0)$ de la matriz, el cual ya calculamos en el paso 2.

4. Una vez terminada la computación de la matriz de distancia, el casillero $M_{(n,m)}$ contiene la distancia total entre las series de tiempo A y B.

2.6.2. Clustering

Clustering es una técnica de minería de datos donde los datos a trabajar son clasificados en grupos según su similitud respecto a una métrica definida por quién analiza la información. En palabras más simples, es “agrupar datos que sean similares entre sí” y que los clusters sean, simultáneamente, lo más disímiles posibles el uno del otro.

Existen distintas técnicas de clustering. Sin embargo, en el contexto de esta memoria sólo será relevante la técnica de clustering jerárquico, debido a que fue la técnica de clustering utilizada durante este trabajo de título. Esta técnica es explicada a continuación.

2.6.2.1. Clustering Jerárquico

El clustering jerárquico es un procedimiento para realizar clustering que recibe un conjunto de datos, los cuales pueden ser puntos en el espacio o vectores. Existen dos métodos para realizar clustering jerárquico: el método aglomerativo y el método divisivo. El método aglomerativo considera que cada observación es un cluster, y conforme se va iterando, se van mezclando distintos grupos en nuevos clusters. El método divisivo es lo opuesto al anterior; es decir, todas las observaciones comienzan en un mismo gran cluster, y conforme el algoritmo va iterando, va separando a las observaciones en nuevos clusters más pequeños.

Para efectos de esta trabajo de título, solo será relevante el método aglomerativo, que es el utilizado en la sección 4.3 Clustering. Este método funciona de la siguiente manera:

1. Cada observación es definida como un cluster de un solo elemento.
2. Se calcula la matriz de distancia entre cada punto (cluster inicial). También, le puede entregar al algoritmo la matriz ya computada, como una forma de ahorrarse este paso. Existen múltiples formas de calcular la distancia entre dos puntos, como lo es distancia euclidiana, dynamic time warping (en caso de series de tiempo o vectores), y distancia máxima (vectores), entre otras.
3. Una vez calculada la matriz de distancia, se define el criterio de enlace del algoritmo. Esto es, indicar qué criterio se utilizará para seleccionar los dos clusters que serán unificados en cada iteración del algoritmo. El detalle de algunos de los criterios más usados para este paso puede consultarse en la tabla 1 del anexo B.
4. Luego, se itera sobre la matriz de distancias, buscando los dos clusters que cumplan con el criterio para ser unificados. La iteración se acaba cuando se haya alcanzado el número de clusters deseados, o hasta que todas las observaciones estén en el mismo cluster.

El resultado de este procedimiento es usualmente visualizado en un dendrograma, como el que se muestra en la Figura 2.4. En esta Figura, es posible observar 3 clusters. Cada unión de elementos tiene una determinada altura, la cual representa la similitud entre dichos elementos. A menor altura, menor es la diferencia entre ellos. En este caso de ejemplo, podemos observar que los elementos S1 y S2 están unidos en el Cluster 1, el cual tiene la altura “más baja”. Esto se puede interpretar, como que los elementos S1 y S2 son los “más parecidos” entre ellos.

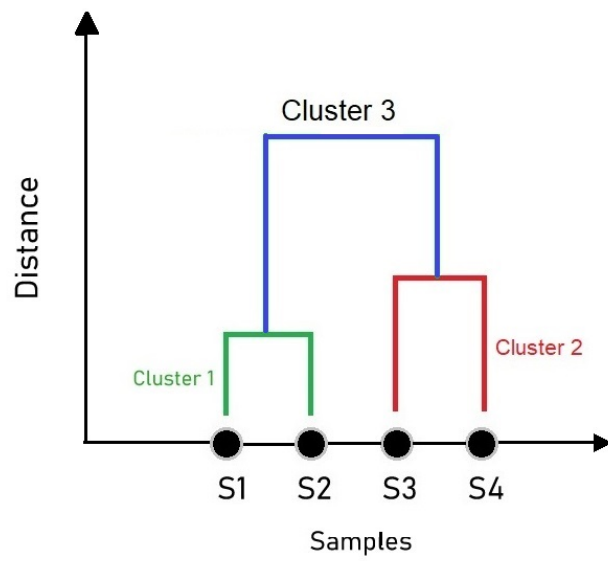


Figura 2.4: Ejemplo de una visualización de tipo dendrograma, extraído de [2]

Capítulo 3

Extensión del Modelo de Contexto

El modelo de contexto previo al desarrollo de esta memoria es el que se muestra en la Figura 3.1, el cual fue diseñado e implementado por Sebastián Toro [40] durante su trabajo de título. En la figura es posible observar que este modelo captura únicamente variables relativas a la interacción adulto mayor-contraparte en sí (tales como fecha, duración y canal de interacción), sin incorporar elementos que puedan entregarle una mayor semántica a los datos recolectados.

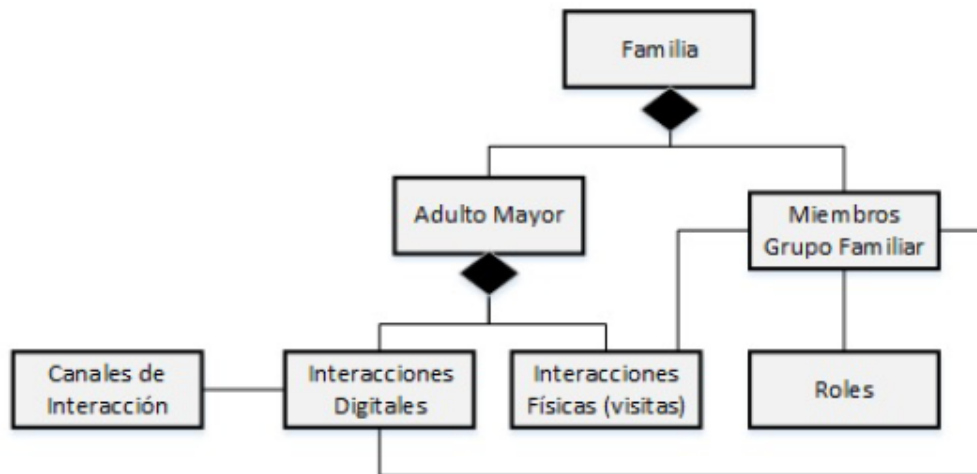


Figura 3.1: Modelo de Contexto previo al desarrollo de la memoria (extraído de [40])

Con la finalidad de poder darle un mayor contexto a los datos, es que se diseñó una extensión de este modelo, la cual apuntó principalmente a capturar variables relativas al entorno del adulto mayor. Los pasos que se siguieron para concebir e implementar esta extensión se detallan a continuación.

3.1. Identificación de conceptos

Para el proceso de identificar conceptos potencialmente relevantes para su incorporación al modelo de contexto, en primer lugar se realizó una lluvia de ideas con el fin de definir un

conjunto inicial de variables candidatas. Estas variables son las que se describen a continuación:

- **Nivel educacional del adulto mayor:** Refiere al nivel educacional que alcanzó el adulto mayor a lo largo de su vida (educación básica, media o superior). Mientras más nivel educativo haya alcanzado, probablemente habrá desarrollado una mayor cantidad de conexiones neuronales [30], las cuales pueden mantener la actividad mental a niveles saludables por un mayor período de tiempo [11].
- **Hobbies:** Refiere a las actividades que el adulto mayor realiza en su tiempo libre. El objetivo es identificar la naturaleza de los hobbies del adulto mayor (interior/ exterior). En el caso de que sus hobbies sean en su mayoría de interior, es más posible que esta persona pueda seguir desarrollándolos, y de esa forma mantener una “ocupación activa” de su tiempo. Lo interesante de poder incorporar esta información en el modelo de contexto es hay múltiples estudios [3, 12, 31] que señalan a los hobbies como factores que reducen el estrés y ayudan a mantener la salud mental. Esto podría contribuir a mitigar las consecuencias de una eventual disminución de las interacciones sociales del adulto mayor.
- **Personalidad del adulto mayor:** Se busca saber si el adulto mayor posee una personalidad más introvertida o extrovertida. De esta forma, se puede plantear la posibilidad de que el adulto mayor extrovertido prefiera un volumen de interacciones mayor que en el caso de un adulto mayor introvertido [38].
- **Mascotas:** Las mascotas son una buena compañía para las personas, e incluso pueden ser de gran ayuda al momento de crear vínculos con otros seres humanos [37]. Esto puede impactar de manera positiva en la vida social de un adulto mayor, en particular de aquellos que viven solos, debido a que las mascotas pueden ayudar a reducir el estrés, combatir la depresión y reducir la sensación de soledad [26].
- **Cantidad de pastillas ingeridas diariamente por el adulto mayor:** Los adultos mayores en general consumen medicamentos, debido a que su salud física tiende a debilitarse con el paso del tiempo [45]. Si un adulto mayor consume una elevada cantidad de medicamentos, puede ser necesario que su familia deba mantener un mayor control sobre él, a fin de monitorear su salud “más de cerca”. Dicho monitoreo requiere de un mayor nivel de interacción social, el cual puede ser desde un simple mensaje preguntándole cómo se ha sentido, hasta visitarlo físicamente.
- **Calendario de fechas importantes:** Se busca mantener un registro de fechas importantes para el adulto mayor, tales como cumpleaños, santorales, aniversarios (de matrimonio, de defunción, etc.). De esta forma, se pueden generar recordatorios o verificar si el adulto mayor tuvo una mayor interacción social dichos días; por ejemplo, si lo saludaron para su cumpleaños.
- **Situación económica del adulto mayor:** Es sabido que en Chile, un gran problema de los adultos mayores son las bajas pensiones que reciben [41]. Este bajo ingreso puede eventualmente causar que los hijos (o algún otro parentesco) asuman un rol de “mantenedor” del adulto mayor, ayudando a sus padres durante sus últimos años. Si se da el caso de que un hijo (u otro familiar) asume el rol de “mantenedor”, en general se esperaría que este (el mantenedor) tuviera una mayor cantidad de interacción con el adulto mayor.
- **Condiciones meteorológicas del día:** La idea es capturar las condiciones meteorológicas

lógicas en la ubicación del adulto mayor a diario. Esto es útil, debido a que en días donde haya mal tiempo (lluvias, nieve o temporales), es posible que se reduzcan (o no haya) las visitas físicas al adulto mayor. Si las condiciones meteorológicas adversas se extienden por varios días, entonces los miembros de la comunidad familiar deberán estar más atentos a la salud social de sus adultos mayores.

Una vez definido el conjunto inicial de variables de contexto a agregar, se evaluó (en conjunto con expertos de dominio) la utilidad de estas para ayudar a monitorear la salud social de los adultos mayores, la factibilidad de medición de cada variable y la necesidad y facilidad de darle seguimiento (mantenerla actualizada). Como resultado de dicha evaluación, se descartaron algunas variables y surgieron algunas consideraciones que se deben tomar en cuenta para registrar los valores de estas. Por lo tanto, el conjunto final de variables, junto a sus consideraciones, quedó definido de la siguiente forma:

- **Nivel educacional del adulto mayor:** Se le pedirá al usuario coordinador del grupo familiar que ingrese el nivel educacional del adulto mayor. A nivel de base de datos, este valor puede ser 0 (educación básica incompleta), 1 (educación básica completa), 2 (educación media completa), 3 (educación superior-técnica completa), 4 (educación superior-profesional completa) o 5 (educación superior con postgrado completa).
- **Personalidad del adulto mayor:** Se le pedirá al usuario coordinador que indique si el adulto mayor es introvertido, extrovertido, ambivertido (mezcla de introvertido y extrovertido), o indeterminado/personalidad desconocida. De esta forma, en la base de datos esta variable será un nuevo campo de la tabla adulto mayor, que tomará el valor 0 en caso de que el adulto mayor sea de personalidad introvertida, 1 en caso de extrovertida, 2 en caso de ambivertida y 3 en caso de indeterminada/desconocida.
- **Mascotas:** Al crear, editar o ver el detalle de una persona de la tercera edad, el usuario coordinador podrá ingresar o editar las mascotas con las que vive el adulto mayor, con su fecha de llegada (puede ser aproximada) y su nombre. Esta información se almacenará en una nueva tabla junto con los campos id (único, auto-generado), fecha de muerte de la mascota (nulo si sigue viva) y el id del adulto mayor al cual corresponde. Debido a que las mascotas tienen una expectativa de vida más corta que los seres humanos, es posible que fallezcan mientras el adulto mayor esté vivo, por lo cual se le pedirá al coordinador del grupo familiar que actualice el estado de la mascota cada 3 meses.
- **Calendario de fechas importantes:** Al crear, editar o ver el detalle de un adulto mayor, el usuario coordinador del grupo familiar podrá ingresar las fechas que puedan ser importantes para el adulto mayor (como aniversarios de boda, de muerte, nacimiento de nietos, bisnietos, entre otros). Cada hito dentro de este calendario considerará los campos título (nombre del evento), fecha (fecha del evento), connotación (un evento puede tener una connotación negativa o positiva), recurrencia (`true`, si es un evento que se repite cada año, `false`, si es puntual) y el id del adulto mayor asociado. Toda la información relacionada a estos eventos se almacenarán en una tabla dentro de la base de datos.
- **Condiciones climáticas del día:** Las condiciones climáticas pueden ser capturadas por el sistema sin necesidad de requerir un ingreso de información manual por parte de los usuarios. Para este fin, se utilizaron librerías de Python (*PyOWM*¹ específicamente)

¹PyOWM documentation

para guardar la información de dicho día en una tabla de la base de datos. Se optó por guardar la información en la base de datos de la aplicación, debido a que la cantidad de registros generados no exige una capacidad muy grande de almacenamiento (un registro diario por cada ubicación distinta entre los adultos mayores). Además, esto permite simplificar la obtención de la información a la hora de efectuar un análisis, y aumentar la disponibilidad del sistema ahorrando el consultar esta información a APIs externas en el momento del análisis.

El modelamiento de cada variable en la base de datos se verá en detalle en la subsección 3.3: Extensión del modelo de contexto.

3.2. Definición de indicadores

Una vez establecidas las variables con las cuales se extendería el modelo de contexto actual, se procedió a definir los indicadores que permiten monitorear la actividad social de los adultos mayores. Para estos efectos, se definió un plan muy similar al que se utilizó para la identificación de conceptos, donde en una primera etapa se realizó una lluvia de ideas.

A continuación, se listan las propuestas de indicadores que surgieron de dicha lluvia de ideas, junto con una breve descripción de cada uno de ellos:

- **Cantidad y tipo de interacciones semanales, por cada miembro del grupo familiar:** Con esto se busca cuantificar el número de interacciones que cada miembro del grupo familiar tenga con el adulto mayor. La importancia de este indicador tiene que ver con los roles definidos en el anexo A. Estos son los roles que se le asignan a cada miembro familiar, según su relación con el cuidado del adulto mayor. Si bien este rol se puede configurar en la base de datos, puede que en la realidad dicha persona no actúe de acuerdo a su “rol teórico” (o sea, el almacenado en la base de datos). Por ejemplo, si un familiar tiene asignado el rol de “monitor”, pero no interactúa de ninguna forma con el adulto mayor, es claro que existe una discordancia entre el rol “teórico” y el “real”. Entonces, en base al nivel de actividad que tenga la persona con el adulto mayor, se puede inferir cuál es el rol “real” que juega en este ámbito el miembro del grupo familiar en cuestión. Hay que entender también que el rol de una persona podría cambiar en el tiempo debido a diversas razones.
- **Duración de las visitas físicas semanales, por cada miembro del grupo familiar:** El objetivo de este indicador es saber cuánto tiempo pasa un miembro de la familia con el adulto mayor (físicamente). Este indicador complementa al anterior, en el sentido de que puede ayudar a ver qué rol “real” asume el miembro de la familia respecto al cuidado del adulto mayor. Esto en sentido que, mientras más tiempo pase una persona con el adulto mayor, existe una mayor posibilidad que la relación entre ambos sea más personal o estrecha, a menos que se trate de alguien contratado específicamente para asistir al adulto mayor.
- **Ventanas de tiempo sin interacción física, en el período entre las 8:00 - 22:00 hrs:** Este indicador tiene por objetivo la detección de patrones de visitas al hogar del adulto mayor (o rutinas en estas visitas) entre las 8:00 y las 22:00 hrs. Se asume que en el resto del día muy probablemente no se recibirán visitas, salvo en ocasiones especiales

o emergencias.

- **Ventanas de tiempo sin interacción digital, en el período entre las 8:00 - 22:00 hrs:** Al igual que el punto anterior, con este indicador se busca poder detectar ciertos patrones de interacción entre el adulto mayor y los miembros de su grupo familiar. También se establece el período entre las 22:00 y las 8:00 del día siguiente, como la ventana de tiempo donde probablemente la actividad digital será nula o casi nula, debido a que es horario de descanso.
- **Índice de interacción digital semanal (cantidad de mensajes enviados/cantidad de mensajes recibidos):** Este indicador apunta a ver qué tan “balanceada” es la interacción digital entre las partes (adulto mayor - familiar). Por balanceada, nos referiremos a que los mensajes que se envían sean respondidos, y vice versa. Este indicador se calcula como la cantidad de mensajes enviados sobre la cantidad de mensajes recibidos del adulto mayor. El indicador se calcula semanalmente y considera las interacciones entre el adulto mayor y cada miembro de su grupo familiar. El aspecto más interesante de este indicador es poder detectar si ambas partes se esfuerzan en mantener la comunicación, o si existe alguna anomalía en ella. Un ejemplo de anomalía sería cuando este indicador tiene a ∞ (el miembro de la familia ignora al adulto mayor o tiene problemas para comunicarse) o cuando tiende a 0 (el adulto mayor ignora al miembro familiar o tiene problemas para comunicarse).

Al igual que en el caso de las variables de contexto, el siguiente paso consistió en validar los indicadores con expertos de dominio. Durante este proceso se realizaron observaciones menores; sin embargo, la estructura y la idea general de los indicadores planteados se mantuvo.

Con los indicadores ya validados, se diseñaron estructuras de datos que fueran capaces de guardar dichos datos. Estas estructuras serán explicadas en la siguiente subsección.

3.3. Extensión del modelo de contexto

Como se mencionó al comienzo de este capítulo, el modelo de contexto diseñado es una extensión al implementado por Sebastián Toro (ver Figura 3.1) [40]. Por lo tanto, el modelo de contexto que se plantea en esta sección contiene las tablas que ya están incorporadas en la aplicación, junto con las tablas con las cuales extendió el modelo de contexto. Para evitar confusiones, las tablas que constituyen la extensión del modelo de contexto son enunciadas a continuación:

- **DailyContext**
- **Pets**
- **Events**
- **InteractionWindows**
- **WeeklyInteractions**

Estas tablas están nombradas en inglés y en formato PascalCase, para no romper con los estándares de la aplicación (el resto del modelo poseía ese formato)². Además de estas

²Definición de PascalCase

tablas que se incorporan al modelo de contexto, se modificó la tabla *UserProfile* (que es la que contiene la información relativa a los adultos mayores), con la finalidad de que pudiese capturar algunas de las variables definidas en la sección 3.1 (Identificación de conceptos); en particular, las variables relativas al nivel educacional y personalidad del adulto mayor. En la Figura 3.2 se presenta el diagrama del modelo de contexto extendido, en conjunto con una breve descripción de las tablas añadidas/modificadas.

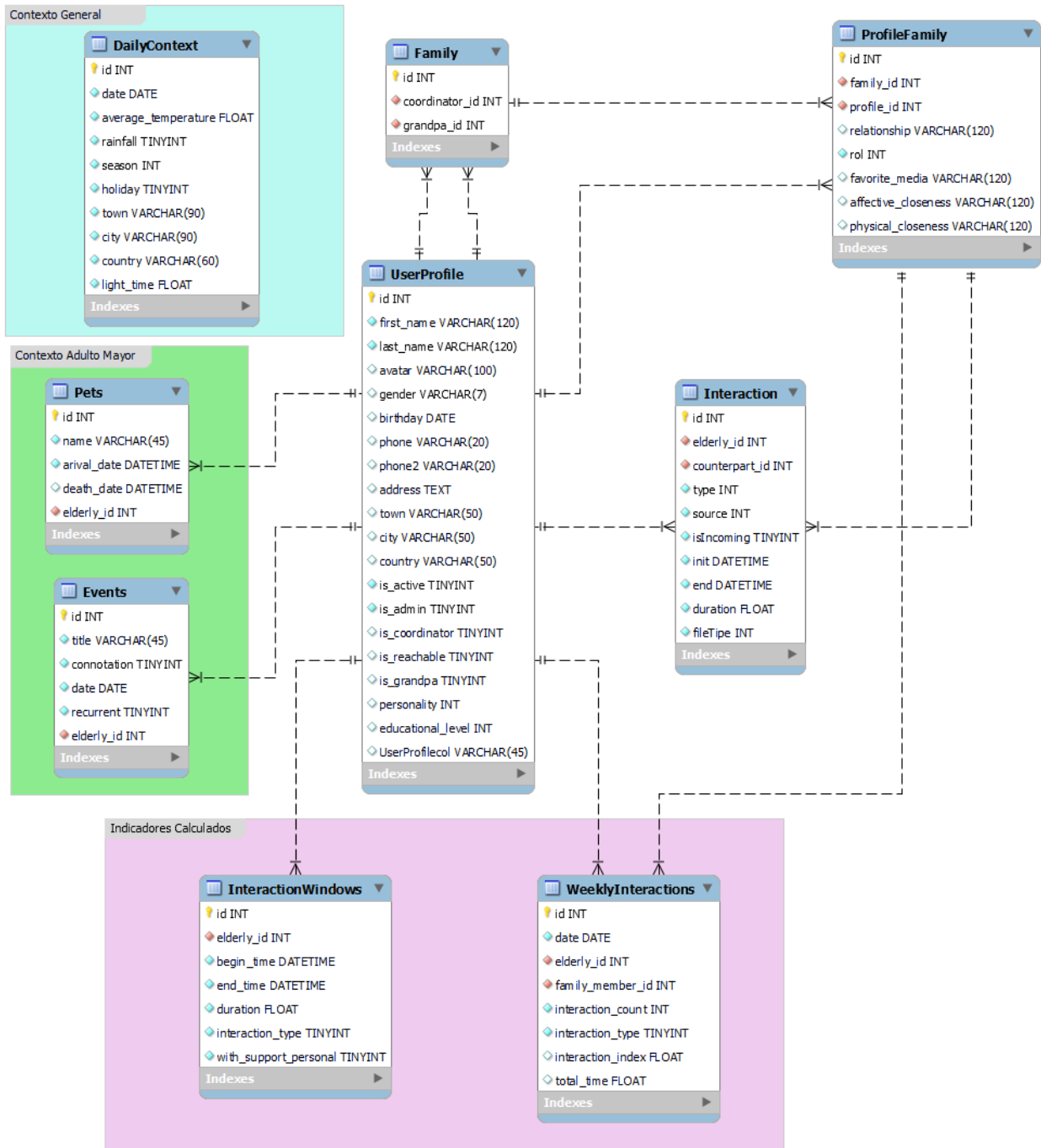


Figura 3.2: Modelo de contexto extendido

3.3.1. UserProfile

La tabla `UserProfile` fue modificada para agregar los siguientes campos:

- **personality**: Se agrega este campo para incorporar la información respecto a la personalidad del adulto mayor en el sistema. Como se mencionó anteriormente, este campo puede tener valores 0 (introvertido), 1 (extrovertido), 2 (ambivertido) o 3 (desconocido/indeterminado).
- **educational_level**: Corresponde al nivel educacional del adulto mayor, mencionado en la subsección 3.1 Identificación de conceptos.
- **town**: Corresponde al nombre de la comuna de residencia del adulto mayor. Para obtener esta información, se le pedirá al usuario coordinador del grupo familiar que ingrese la comuna donde vive el adulto mayor a la hora de crear a este usuario.
- **city**: Corresponde al nombre de la ciudad de residencia del adulto mayor. Al igual que para la comuna, el valor será ingresado por el usuario coordinador.
- **country**: Corresponde al nombre del país de residencia del adulto mayor. Al igual que para la comuna, el valor de este campo también será ingresado por el usuario coordinador.

Es importante notar que se separaron los campos *town*, *city* y *country* para evitar errores en el formato de ingreso de esta información; por ejemplo, puede haber usuarios que estén acostumbrados a escribir las ubicaciones separadas por “,”; otros pueden hacerlo con “-” o simplemente con un espacio en blanco. La importancia de esta información será explicada más en detalle en la siguiente sección.

3.3.2. DailyContext

Esta tabla se creó con el objetivo de almacenar un resumen diario de las condiciones climáticas, y el contexto relativo a la naturaleza del día; por ejemplo, si es feriado o no, si es día de semana o fin de semana. Esta tabla posee los siguientes campos:

- **id**: Identifica a la instancia de contexto diario en la plataforma (autogenerado).
- **date**: Indica la fecha a la cual corresponde el registro.
- **average_temperature**: Registra la temperatura promedio del día (medida en °C).
- **rainfall**: Indica si es que hubo precipitaciones ese día. Toma el valor 0 si es que no hubo, y 1 si es que hubo.
- **season**: Indica a qué estación del año corresponde el día en cuestión. Puede tomar los siguientes valores: 0 (Verano), 1 (Otoño), 2 (Invierno) o 3 (Primavera).
- **holiday**: Almacena si el día actual fue festivo o no. Toma el valor 0 si no, y 1 si lo es.
- **light_time**: Refiere al período de luz que hubo durante el día, medido en segundos. Poco tiempo de luz puede incidir en visitas físicas más cortas.
- **town**: Indica la comuna a la cual pertenece el contexto diario. Como en las comunas donde residen los miembros de la familia podría eventualmente haber climas distintos, se requiere este campo para diferenciar a las variables de clima según la comuna registrada.
- **city**: Esta es la ciudad a la cual pertenece el contexto diario. Al igual que el campo *town*, permite diferenciar las localidades.

- **country**: Es el país al cual pertenece el contexto diario. Al igual que el campo *town*, permite diferenciar las localidades.

Es importante mencionar que los datos con los cuales se alimenta esta tabla son extraídos desde APIs que ofrecen alternativas gratuitas para la entrega de dicha información. Un mayor detalle de esto será expuesto en la sección 3.4 (Implementación del modelo de contexto).

3.3.3. Pets

La tabla **Pets** se creó con la finalidad de guardar información relacionada a las mascotas que se encuentran bajo el cuidado del adulto mayor. Los campos de dicha tabla son los siguientes:

- **id**: Identificador de la mascota en la plataforma (autogenerado).
- **name**: Nombre de la mascota.
- **arrival_date**: Fecha de llegada (o nacimiento) de la mascota.
- **death_date**: Fecha de defunción (o pérdida) de la mascota. Puede ser nulo mientras el animal siga vivo.
- **elderly**: Identificador del adulto mayor al cual pertenece la mascota.
- **created_date**: Fecha de creación de la mascota en la base de datos.
- **updated_date**: Fecha de la última modificación de la mascota en la base de datos.

Como se mencionó en la subsección 3.1 (Identificación de conceptos), las mascotas domésticas usualmente tienen una expectativa de vida mucho menor a la de los seres humanos, por lo cual con esta tabla es necesario tener cuidado de mantenerla actualizada. Por ello, se le pedirá al usuario coordinador del grupo familiar que actualice el estado de esta tabla cada 3 meses. Esto es, añadir nuevas mascotas en caso de adoptar una, o marcar como “no activa” a una mascota cuando haya fallecido o se haya perdido.

3.3.4. Events

Esta tabla se incorporó para almacenar el calendario de fechas importantes para el adulto mayor. Los campos que registra esta tabla son los siguientes:

- **id**: Identificador del evento dentro de la plataforma (autogenerado).
- **title**: Título que recibe el evento.
- **connotation**: Carga afectiva del evento (para el adulto mayor), indicado como 0 si es un evento considerado “negativo”, 1 si es considerado “positivo”.
- **date**: Fecha del evento.
- **recurrent**: Indica si el evento es recurrente o no. Toma el valor 0 si el evento no es recurrente, o 1 si lo es.
- **elderly**: Identificador del adulto mayor al cual está relacionado el evento.
- **created_date**: Fecha de creación del evento en la base de datos.
- **updated_date**: Fecha de la última modificación del evento en la base de datos.

Durante su vida, los adultos mayores van experimentando hitos que son importantes, los cuales pueden tener una connotación positiva o negativa, como lo puede ser el nacimiento de nietos/bisnietos, defunciones de seres queridos/mascotas o aniversarios (de matrimonio o defunción). Como dichos eventos van surgiendo “aleatoriamente” a través del tiempo, se le pedirá al usuario coordinador del grupo familiar que cada tres meses incorpore los nuevos (si los hay).

3.3.5. InteractionWindows

La tabla `InteractionWindows` corresponde a la agrupación de los indicadores ventanas de tiempo sin interacción física, y ventanas de tiempo sin interacción digital.

- **id**: Identifica la instancia del indicador dentro de la plataforma (autogenerado).
- **begin_time**: Es el timestamp (fecha y hora) en el cual se inició la ventana sin interacciones.
- **end_time**: Es el timestamp en el cual terminó la ventana sin interacciones.
- **duration**: Duración de la ventana de tiempo sin interacciones, medida en segundos (`end_time - begin_time`).
- **interaction_type**: Tipo de interacción que representa la ventana. Toma el valor 0 si se trata de ventanas de tiempo sin interacción física, y el valor 1 si se trata de una ventana de tiempo sin interacción digital.
- **with_support_personal**: Refiere a si, durante este proceso, el adulto mayor tuvo interacción con personal de apoyo (ya sea física o digitalmente). Por personal de apoyo nos referiremos a personas remuneradas que cuidan o asisten al adulto mayor. Se asume que estas personas están relacionadas con el adulto mayor a través de un contrato de trabajo. Se hace esta distinción, ya que en la interacción adulto mayor-cuidador, no está necesariamente presente la dimensión afectiva de ambas partes. Toma el valor 0 si no ocurrió ninguna interacción con personal de apoyo durante la ventana, y 1 en caso contrario.
- **elderly**: Identificador del adulto mayor para el cual se calculó el indicador.

En cuanto a las ventanas de tiempo sin interacción digital, es necesario tener en cuenta que estas pueden estar basadas en mensajería asíncrona. Por esto, es que para calcular los intervalos de tiempo sin interacciones digitales, se tomará el cuidado de establecer un período de tiempo posterior al intercambio del último mensaje digital registrado, antes de establecer el comienzo de una ventana de tiempo sin interacciones digitales. Dicho período de tiempo se fijó en 20 minutos.

A la hora de calcular las ventanas sin interacción, se tomó en consideración las interacciones con personal de apoyo (cuidadores). En estos casos, si se detecta una interacción con dicho personal dentro de una ventana sin interacción, esta interacción no se tomará como una interrupción de la ventana actual, sino que se marcará el flag correspondiente a *with_support_personal*, el cual tomará el valor 1. El detalle de la implementación del sistema de procesamiento de datos que alimenta a esta tabla será explicado con mayor detalle en siguientes secciones.

3.3.6. WeeklyInteractions

La tabla `WeeklyInteractions` engloba los indicadores de cantidad y tipo de interacciones semanales por cada miembro del grupo familiar, duración de las visitas físicas semanales por cada miembro del grupo familiar, e índice de interacción digital semanal.

- **id**: Identifica la instancia del indicador dentro de la plataforma (autogenerado).
- **date**: Fecha en la cual se calculó el indicador. Para efectos de cálculo de este indicador (semanal), se registra el domingo de cada semana.
- **elderly**: Identifica al adulto mayor para el cual se calculó el indicador.
- **family_member**: Identifica al miembro de la familia que interactuó con el adulto mayor.
- **interaction_count**: Registra la cantidad de interacciones detectadas por el sistema durante la semana.
- **interaction_type**: Tipo de interacción que representa el indicador. Toma el valor 0 si se trata de interacciones físicas, y el valor 1 si se trata de interacciones digitales.
- **interaction_index**: El índice de interacción corresponde al indicador mencionado en la subsección 3.2 (Definición de identificadores). Este corresponde al cociente entre $\frac{\text{mensajes enviados}}{\text{mensajes recibidos}}$. Toma el valor nulo si el indicador corresponde a interacciones físicas (es decir, si `tipo_interaccion = 0`).
- **total_time**: Corresponde al indicador de duración de las visitas físicas semanales y se calcula para cada miembro del grupo familiar (medido en segundos). Es la suma de todas las visitas físicas que el miembro de la comunidad familiar realizó durante la semana. Toma el valor nulo si es que el indicador corresponde a interacciones digitales (es decir, si `interaction_type = 1`).

Como observación general al modelo, es posible ver que con la extensión realizada se puede tener un mayor conocimiento del contexto en el cual ocurren las interacciones sociales, ya que se almacenan datos relativos al entorno donde vive el adulto mayor (como lo es el clima o sus mascotas), en conjunto con los indicadores calculados. Esto entrega más herramientas para el análisis por parte de los investigadores, así como también hace posible el estudio histórico/evolutivo de los indicadores en el tiempo, debido a la persistencia de estos en la base de datos.

3.4. Implementación del modelo de contexto

Para implementar esta extensión al modelo de contexto, fue necesario hacer cambios tanto a nivel de backend como a nivel de frontend de la aplicación `SocialConnector`. Con la finalidad de hacer una descripción más ordenada de las modificaciones introducidas, se explicarán por separado los cambios al backend y al frontend.

3.4.1. Modificación de los modelos en backend

El primer paso en la implementación del modelo de contexto fue la modificación de los modelos en el backend de `SocialConnector`. Este sistema basa su funcionamiento en el framework `Django`, por lo cual, para lograr este objetivo se necesita crear nuevas clases y editar

las existentes (esto es para el caso de las modificaciones al modelo del adulto mayor). La única tabla que fue necesario modificar fue la respectiva a los datos de los adultos mayores, debido a que el resto de las tablas (mencionadas en la sección anterior) no existían antes del desarrollo de este trabajo de título.

En SocialConnector, el modelo que contiene la información relativa a los adultos mayores es un modelo llamado `UserProfile`. Este es utilizado para guardar la información del perfil de los usuarios en SocialConnector, por lo cual las instancias de este modelo no son únicamente adultos mayores; también pueden ser contactos de ellos. Bajo este escenario, los campos que se añadieron a este modelo (*town*, *city*, *country*, *personality* y *educational_level*) se añadieron junto con el flag `null = true`, debido a que si una instancia de `UserProfile` no corresponde a un adulto mayor entonces no es de relevancia capturar las variables de contexto antes mencionadas.

Con respecto al resto de las tablas, no existían implementaciones previas de las mismas, por lo cual se implementaron las clases correspondientes desde cero. El código implementado a nivel de backend puede ser visto en el anexo C.

3.4.2. Implementación del sistema de procesamiento automático de datos

Luego de extender el modelo de contexto de la aplicación, fue necesario implementar el sistema de procesamiento automático de datos que debía alimentar a las tablas `DailyContext`, `InteractionWindows` y `WeeklyInteractions`. Recordemos que las dos primeras tablas requieren procesar diariamente datos de clima e interacciones respectivamente, mientras que la tabla `WeeklyInteractions` requiere procesar datos relativos a interacciones una vez a la semana.

3.4.2.1. DailyContext

Para que la implementación del sistema pudiera procesar la información requerida por la tabla `DailyContext` (clima y feriados), en primer lugar se debió investigar acerca de dos tipos de APIs: una que entregue los datos del clima dada una fecha y ubicación geográfica, y la otra que indique si una cierta fecha corresponde o no a un feriado para una cierta localidad. Los criterios de selección sobre qué API utilizar en cada caso se basaron principalmente en la gratuidad de su uso, la calidad de la información, la facilidad de su integración con SocialConnector, y el soporte que tenga la API durante el tiempo (no tendría sentido utilizar APIs deprecadas o por deprecar). De esta forma, las APIs seleccionadas fueron las siguientes:

- Tiempo: *OpenWeatherMap API*. *OpenWeatherMap* (OWM) es una API que ofrece una alternativa gratuita, que permite realizar casi 1.000 requests por día para obtener datos climáticos en una localidad³. Para obtener la información, basta con enviar una request al *endpoint* de la API *One-Call* de OWM⁴. OWM ofrece una librería de Python (PyOWM), que se puede utilizar para una mayor comodidad, principalmente para no tener que procesar el resultado de la request a mano. La respuesta de esta API es un

³OWM Pricing

⁴OWM One-Call API

archivo json, que incluye datos como temperatura máxima, temperatura mínima, hora de salida del sol, hora de puesta de sol, y cantidad de lluvia, entre otros.

- Feriados: *Holiday API*. Esta es una API que permite al usuario realizar consultas acerca de si una determinada fecha es feriado en un determinado país. Para ello, ofrece una alternativa gratuita, para realizar hasta 10.000 requests al mes (un poco más de 300 requests por día). El formato de consulta requiere que el usuario facilite la fecha y el país a consultar, y el resultado se entrega en archivo json.

Durante el desarrollo del sistema se descubrió que OWM utiliza coordenadas geográficas para recibir requests en su API, mientras que Holiday API utiliza los códigos ISO⁵ de los países para su API. Debido a esto, se debió incorporar una tercera API de geolocalización, la cual fuese capaz de, dada una ubicación del estilo “Comuna, Ciudad, País”, retornar su latitud, longitud, y código ISO del país donde se ubica. La API elegida para esta tarea fue positionstack⁶, la cual permite realizar 25.000 requests por mes (poco menos de 1.000 por día), y entrega los datos solicitados (es decir, latitud, longitud y código ISO del país).

Una vez definidas las APIs a utilizar, se diseñó e implementó el *cron-job* que realiza las consultas pertinentes a las APIs, y crea y almacena todas las instancias necesarias para alimentar la tabla `DailyContext`. El *cron-job* implementado sigue el procedimiento descrito a continuación:

1. Se consultan todos los usuarios adultos mayores registrados en SocialConnector.
2. Se inicializa una lista de Python vacía, donde se guardan las localidades ya consultadas.
3. Para cada adulto mayor:
 - (a) Se extraen los campos `town`, `city` y `country` correspondiente al domicilio del adulto mayor, y se formatean para que queden en formato capitalizado (es decir, primera letra mayúscula y todo el resto minúscula).
 - (b) Se consulta en la lista de localidades si es que ya se consultó la localidad del adulto mayor actual. Si no se ha hecho, se pasa al paso (c). Por el contrario, si ya fue consultada, entonces se pasa al siguiente adulto mayor.
 - (c) Se añade a las localidades consultadas la ubicación del adulto mayor actual.
 - (d) Se consultan las APIs para solicitar los datos requeridos.
 - (e) Se procesan las respuestas de las APIs para generar un nuevo objeto `DailyContext`.
 - (f) Se guarda el nuevo objeto de `DailyContext` y se continúa con el siguiente adulto mayor.

Para terminar, se configuró este *cron-job* para que corra una vez al día a las 1:00 AM. A pesar de que no se espera un uso masivo de la plataforma, se elige una hora en la cual se estima que el uso de la plataforma sea bajo, para que la ejecución de esta tarea no influya en el desempeño/rendimiento de la aplicación.

⁵Códigos ISO

⁶positionstack home page

3.4.2.2. InteractionWindows

Antes de explicar el sistema de procesamiento encargado de generar los registros correspondientes a las ventanas de no interacción (tanto físicas como digitales), es necesario mencionar las estructuras de datos utilizadas como apoyo, debido a que su comprensión es fundamental para el correcto entendimiento del algoritmo que procesa los datos. Estas estructuras son tres, y su detalle es explicado a continuación.

- **daily_interactions**: Este es un diccionario de Python pensado para registrar todas las interacciones del día, organizadas por adulto mayor y tipo de interacción. Las llaves de este diccionario corresponden al `id` del adulto mayor y el valor de cada llave corresponde a otro diccionario. Este último contiene los campos `physical` y `digital`, cuyos valores son una lista con las interacciones físicas y digitales del adulto mayor respectivamente. Por ejemplo, en el caso de adulto mayor cuyo `id = 24`, `daily_interactions[24][physical]` contendrá una lista con sus interacciones físicas, mientras que `daily_interactions[24][digital]` contendrá una lista con sus interacciones digitales.
- **physical_parameters**: Este es un diccionario, diseñado para guardar parámetros relativos las visitas físicas de cada adulto mayor. Posee los siguientes campos:
 - `support_visit_begin`: Corresponde al comienzo de la última visita de personal de apoyo (en caso de ocurrir). Se inicializa con el valor 8:00 AM.
 - `last_visit`: Corresponde a la hora de término de la última visita procesada. Se inicializa con el valor 8:00 AM.
 - `support_staff`: Corresponde a si la última visita fue realizada por personal de apoyo o no. Toma el valor `True` en caso de que sí, y `False` si es que no.
- **digital_parameters**: Este es un diccionario, diseñado para guardar parámetros relativos las interacciones digitales de cada adulto mayor. Contiene los siguientes campos:
 - `support_mssge_begin`: Corresponde al comienzo de la última interacción digital con personal de apoyo (en caso de ocurrir). Se inicializa con el valor 8:00 AM.
 - `last_mssge`: Corresponde a la hora de término de la última interacción digital procesada. Se inicializa con el valor 8:00 AM.
 - `support_staff`: Indica si la última visita fue realizada por personal de apoyo o no. Toma el valor `True` en caso de que sí, y `False` si es que no.

Estas estructuras juegan un rol fundamental dentro del procesamiento de datos, debido a que gracias a ellas es posible ordenar las interacciones para su correcto procesamiento, e identificar cuando ocurren ventanas de tiempo sin interacción. Con estas estructuras, que utiliza el sistema de procesamiento de datos, a continuación se detallan los pasos que sigue este algoritmo para generar los registros de `InteractionWindows`.

1. Consulta todas las interacciones que han sido registradas durante el último día.
2. Se inicializa el diccionario `daily_interactions`, y en él se guarda las interacciones separadas por adulto mayor y por tipo de interacción.
3. Para cada adulto mayor en el diccionario:
 - (a) Se inicializan las estructuras `physical_parameters` y `digital_parameters`.
 - (b) Se ordenan sus interacciones físicas por su timestamp de inicio, para evitar errores en los cálculos del sistema.

- (c) Para cada interacción física:
 - i. Se verifica si es que la contraparte de la interacción es personal de apoyo.
 - ii. Se verifica la última hora de visita al adulto mayor (`physical_parameters[last_visit]`).
 - iii. Se verifica si es que la última visita la realizó personal de apoyo (`physical_parameters[support_staff]`).
 - iv. Verificando estas tres situaciones, el sistema sabe cuáles son las ventanas de no interacción que debe guardar. El detalle de este procedimiento puede ser visto en el diagrama expuesto en el anexo D.1.
 - (d) Se ordenan sus interacciones digitales por su timestamp de inicio.
 - (e) Para cada interacción digital:
 - i. Se verifica si es que la interacción corresponde a un mensaje o a una llamada.
 - ii. Se verifica si es que la contraparte de la interacción es personal de apoyo.
 - iii. Se verifica la última hora de interacción con el adulto mayor (`digital_parameters[last_mssge]`).
 - iv. Se verifica si es que la última interacción ocurrió con personal de apoyo (`digital_parameters[support_staff]`).
 - v. Verificando estas cuatro situaciones, el sistema sabe cuáles son las ventanas de no interacción que debe guardar. El detalle de este procedimiento se muestra en el diagrama presentado en el anexo D.2.
 - (f) Luego de procesar todas las interacciones del adulto mayor se revisan los valores de los parámetros de `physical_parameters` y `digital_parameters`, con la finalidad de guardar ventanas de no interacción que puedan haber quedado pendientes.
4. Finalmente, se consultan todos los adultos mayores activos presentes en la base de datos. Se recorre esta lista, y si se encuentra un adulto mayor que no está presente en el diccionario `daily_interactions` (es decir, que no registró interacciones en el último día), entonces se guardan dos ventanas de no interacción correspondientes a todo el día (8AM - 10PM), una de no interacción digital y otra de no interacción física.

Con el algoritmo que procesa los datos ya implementado, el último paso fue configurar el *cron-job* que lo ejecuta. Este, se programó para correr a las 3 AM cada día, para evitar afectar el desempeño/rendimiento de la aplicación.

3.4.2.3. WeeklyResume

Para el procesamiento de datos de la tabla `WeeklyResume`, primero fue necesario el diseño de una estructura de datos que fuese útil para almacenar los datos procesados antes de guardarlos en la base de datos. Esta “estructura intermedia” es necesaria, debido a que para generar registros para la tabla `WeeklyResume` se necesitan calcular campos sobre las interacciones que se procesen. Para el cálculo de dichos campos, es indispensable procesar todas las interacciones; de lo contrario, los registros generados podrían contener información incompleta (y por ende, contener datos erróneos).

La estructura diseñada consiste en una serie de diccionarios anidados, los cuales permiten organizar las estadísticas de cada par adulto mayor - contraparte. Esta posee cuatro niveles de diccionarios, los cuales son explicados a continuación.

- **interactions_resume**: Este es un diccionario que contiene toda la información de los resúmenes semanales de todos los adultos mayores. Se inicializa vacío, sus llaves corresponden al `id` de cada adulto mayor y el valor de cada llave corresponde a un diccionario con las estadísticas del adulto mayor al cual corresponda el `id`.
 - **elderly_resume**: Este es un diccionario que contiene toda la información del resumen semanal de un determinado adulto mayor. Se inicializa vacío, sus llaves corresponden al `id` de cada contraparte con la cual haya interactuado el adulto mayor durante la semana y el valor de cada llave corresponde a un diccionario con las estadísticas específicas de la contraparte (a la cual corresponda el `id` de su llave).
 - * **counterpart_resume**: Es un diccionario que contiene toda la información del resumen semanal de una contraparte (respecto a un determinado adulto mayor). Posee dos llaves, **physical** y **digital**. El detalle de estos se presenta a continuación:
 - **physical**: Este es un diccionario que alberga las estadísticas de interacciones físicas entre el adulto mayor y la contraparte. Tiene únicamente dos campos: **interactions** que almacena el número de interacciones físicas (visitas) que realizó la contraparte al adulto mayor y **total_time**, que almacena el tiempo total de visita que registró la contraparte (es la suma de la duración de cada una de sus visitas).
 - **digital**: Este es un diccionario que guarda las estadísticas de interacciones digitales entre el adulto mayor y la contraparte. Posee tres campos: **interactions** que almacena el número de interacciones digitales entre la contraparte y el adulto mayor, **incoming** que guarda la cantidad de interacciones digitales entrantes (es decir, de la contraparte al adulto mayor) y **outgoing** que indica la cantidad de interacciones digitales salientes (o sea, desde el adulto mayor hacia la contraparte).

Para explicar de mejor forma la estructura diseñada, a continuación se describe un caso de uso. Para este ejemplo, tomaremos como adulto mayor a “Tite Escalante” (`id = 24`), y como contraparte a “Primo Lenz” (`id = 76`)⁷. Estableceremos que durante la última semana, “Primo Lenz” visitó en tres oportunidades a “Tite Escalante”: el martes de 17:10 - 19:10 (2 horas), el jueves de 18:00-19:00 (1 hora) y el domingo de 13:00-20:00 (7 horas).

En este escenario ficticio, una vez finalizado el procesamiento de los datos y almacenado el resultado en la estructura intermedia `interactions_resume`, el valor de `interactions_resume[24][76][‘physical’]` debería ser igual a `{‘interactions’: 3, ‘total_time’: 36000}`. Esto se interpreta de la siguiente manera:

“Durante la última semana, la contraparte `id = 76` (“Primo Lenz”) visitó 3 veces al adulto mayor `id = 24` (“Tite Escalante”), con un tiempo total de visita de 36.000 segundos (10 horas).”

Una vez diseñada la estructura que almacena los datos procesados, se procedió a implementar el sistema de procesamiento de los datos que, justamente, alimentará dicha es-

⁷Estos son nombres y los datos son de fantasía, únicamente con fines ilustrativos

estructura y posteriormente generará y guardará los registros correspondientes para la tabla `WeeklyResume`. Este sistema de procesamiento funciona de la siguiente forma:

1. Consulta todas las interacciones que han sido registradas durante la última semana (7 días).
2. Inicializa la estructura intermedia (`interactions_resume`) como un diccionario en blanco.
3. Para cada interacción:
 - (a) Se identifica el id del adulto mayor de la interacción.
 - (b) Se consulta por el id del adulto mayor en `interactions_resume`. En caso de no existir, se inicializa el diccionario (`elderly_resume`).
 - (c) Se identifica a la contraparte de la interacción.
 - (d) Se consulta por el id de la contraparte en `elderly_resume`. En caso de no existir, se inicializa el diccionario (`counterpart_resume`).
 - (e) Se identifica el tipo de la interacción (si es física o digital).
 - (f) Si la interacción es física, se selecciona el diccionario `physical` correspondiente a la contraparte y se suma su duración a `total_time` y 1 a `interactions`. Si la interacción es digital, entonces se selecciona el diccionario `digital` correspondiente a la contraparte, se suma 1 a `incoming/outgoing` (dependiendo si la interacción es entrante/saliente respectivamente) y 1 a `interactions`.
4. Con todas las interacciones ya procesadas, y la información almacenada en la estructura `interactions_resume`, se recorre cada entrada de dicho diccionario, generando un registro por cada tupla distinta (adulto mayor, contraparte, tipo de interacción).

Finalmente, se configuró un *cron-job* que ejecuta este algoritmo de procesamiento todos los lunes a las 0:00 hrs. Este horario fue escogido por la misma razón planteada tanto en `DailyResume` como en `InteractionWindows`.

3.4.3. Creación de vistas en frontend

Para que los cambios realizados en el backend se vean reflejados en la aplicación de Social-Connector (de cara a sus usuarios), es necesario adaptar el frontend de la aplicación a estos. Para ello, en primer lugar se identificaron los elementos del modelo que debían ser expuestos a nivel de frontend, pues no todos los elementos necesitan interactuar con los usuarios de la plataforma. Con esto nos referimos a que la gran mayoría de los cambios en el modelo de contexto fueron diseñados considerando alimentación automática por parte de sistemas de procesamiento de datos y por lo tanto no requieren manipulación de los mismos por parte de usuarios de la plataforma. Sin embargo, los cambios al modelo correspondientes a la tabla `UserProfile`, y elementos correspondientes a `Events` y `Pets`, sí requieren que el usuario coordinador sea capaz de administrarlos (es decir, añadir, editar o borrar estos componentes del contexto). En resumen, los cambios que se realizaron en frontend se enfocan únicamente en estos tres componentes de la extensión al modelo de contexto: `UserProfile`, `Events` y `Pets`.

Respecto a los cambios realizados al modelo `UserProfile`, primero se identificaron las pantallas donde se podría ingresar o editar la información de un usuario de la plataforma.

Esto, debido a que como se mencionó anteriormente, el modelo `UserProfile` almacena la información del perfil de un usuario (sea o no adulto mayor) en la plataforma `SocialConnector`. En la Figura 3.3 se muestra el formulario de edición de perfil de un usuario de `SocialConnector` en un estado previo al desarrollo de este trabajo.

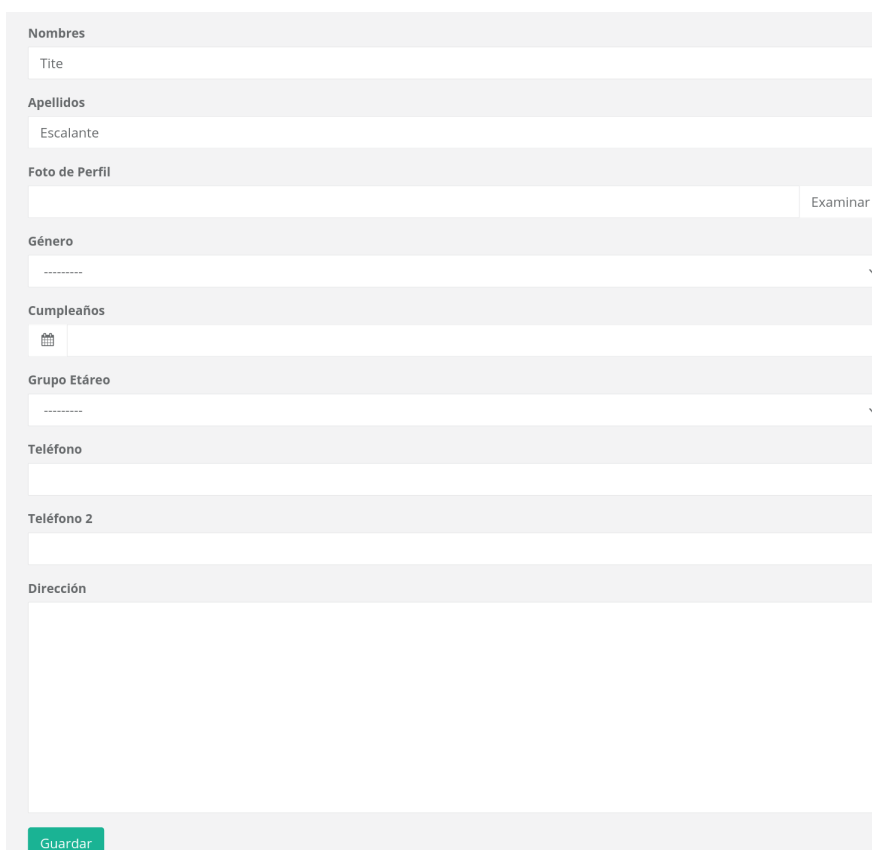
El formulario de edición de perfil antiguo se muestra en un diseño vertical con secciones encabezadas por títulos en gris. Las secciones y sus campos son: 'Nombres' con un campo 'Tite'; 'Apellidos' con un campo 'Escalante'; 'Foto de Perfil' con un campo de imagen y un botón 'Examinar'; 'Género' con un menú desplegable; 'Cumpleaños' con un ícono de calendario; 'Grupo Etnico' con un menú desplegable; 'Teléfono' con un campo de texto; 'Teléfono 2' con un campo de texto; 'Dirección' con un campo de texto grande. En la parte inferior hay un botón verde 'Guardar'.

Figura 3.3: Formulario de edición de perfil antiguo

El formulario mostrado en la Figura 3.3 era único, es decir, los usuarios adultos mayores y no adultos mayores veían el mismo formulario al editar su perfil. Allí es posible ver que en este no existen los campos que se añadieron en la extensión del modelo de contexto. Para corregir eso, se aprovechó la característica de los templates de Django, los cuales permiten añadir bloques condicionales a los html que se muestran en pantalla.

Sumado a lo anterior, la tabla `UserProfile` posee un campo llamado `is_grandpa`, el cual toma el valor `True` si es que el usuario es un adulto mayor. Ese campo permite filtrar fácilmente la información a mostrar en el formulario, en función de si el usuario es o no adulto mayor. Así, las vistas de edición del perfil para usuarios adultos mayores y para los contactos queda como se muestra en la Figura 3.4.

Allí se puede observar que en los nuevos formularios, el usuario adulto mayor tiene más campos por rellenar (Figura 3.4 (a)), que son justamente las variables que se añadieron en el modelo de contexto. Por otro lado, se ve que el usuario “contacto” sigue teniendo el mismo formulario de antes (Figura 3.4 (b)), ya que las variables planteadas en la extensión del modelo de contexto apuntan solo a capturar información respecto al adulto mayor, no siendo estas relevantes para los demás usuarios (particularmente, para los contactos).

(a) Nuevo formulario para el usuario adulto mayor

(b) Nuevo formulario para el usuario contacto

Figura 3.4: Interfaces de los nuevos formularios

Con respecto a **Events** y **Pets**, se crearon vistas para que el usuario coordinador del grupo familiar pueda administrar (agregar, editar y borrar) estas instancias de un adulto mayor. En el caso que un adulto mayor no tenga mascotas registradas, al ver el detalle de su perfil se muestra la pantalla correspondiente a la Figura 3.5.

En esta vista se puede apreciar que se añadió una sección de “Mascotas Actuales”, en la cual hay un botón “Agregar Mascota”. Al hacer click en dicho botón, se desplegará la vista para añadir una mascota. Esta vista contiene un formulario con los campos necesarios para añadir una mascota a la plataforma (Figura 3.6). Si se clickea el botón guardar, la mascota se registrará en la plataforma, y quedará asociada al adulto mayor.

Por el contrario, si el adulto mayor tiene mascotas registradas en la plataforma, al consultar el detalle de su perfil ya no aparecerá un botón con el texto “Agregar Mascota” en la sección correspondiente a las mascotas (como en la Figura 3.4). En su lugar, se mostrará la lista de mascotas que están asociadas al adulto mayor junto con el botón “Editar Mascotas” (Figura 3.7). Al hacer click en el botón “Editar Mascotas”, se muestra la vista que contiene la lista de mascotas asociadas al adulto mayor, con la opción de editarlas, borrarlas o agregar una nueva (Figura 3.8)

Tomás Coordinador
 Editar mi información >

Mis Contactos
 Mis Adultos Mayores
 Mis Familias

Cerrar Sesión

Datos de Tite Escalante

Datos personales Contacto no tiene foto de perfil.

Nombre	Tite Escalante
Género	No Especificado
Cumpleaños	No especificado
Grupo Étareo	No Especificado
Teléfono	No especificado
Teléfono 2	
Dirección	No especificado

Editar Datos Personales

Redes Sociales
 No tiene redes sociales
 Agregar Redes Sociales

Mascotas Actuales
 No tiene mascotas registradas
 Agregar Mascota

Eventos
 No tiene eventos registrados
 Agregar Evento

Figura 3.5: Vista del detalle de un adulto mayor sin mascotas ni eventos

Añadir mascota

Nombre

Fecha de Llegada

Fecha de Defunción

Agregar Mascota

Figura 3.6: Formulario para añadir una mascota

El flujo para administrar los eventos de un adulto mayor es prácticamente idéntico al recién expuesto para las mascotas. Si un adulto mayor no tiene eventos registrados, en su perfil aparecerá el botón “Agregar Evento” (Figura 3.5). Al hacer click en dicho botón se muestra el formulario para añadir un evento (Figura 3.9). Haciendo click en guardar, el evento se registra en la plataforma.

Tomás Coordinador
 Editar mi información ▾

Mis Contactos

Mis Adultos Mayores

Mis Familias

Cerrar Sesión

Datos de Tite Escalante

Datos personales Contacto no tiene foto de perfil.

Nombre	Tite Escalante
Género	No Especificado
Cumpleaños	No especificado
Grupo Étnico	No Especificado
Teléfono	No especificado
Teléfono 2	
Dirección	No especificado

Editar Datos Personales

Redes Sociales

No tiene redes sociales

Agregar Redes Sociales

Mascotas Actuales

Mascota	Fecha de Llegada
Gato Juanito	13 de Abril de 2020

Editar Mascotas

Eventos

No tiene eventos registrados

Agregar Evento

Figura 3.7: Vista del detalle de un adulto mayor con mascotas

Tomás Coordinador
 Editar mi información ▾

Mis Contactos

Mis Adultos Mayores

Mis Familias

Cerrar Sesión



Mascotas de Tite Escalante

Agregar Nueva Mascota

Mostrar 25 registros

Buscar: CSV Excel PDF

Mostrando registros del 1 al 1 de un total de 1 registros

Opciones	Nombre	Fecha de Llegada	Fecha de Muerte
 	Gato Juanito	13 de Abril de 2020	

Anterior 1 Siguiente

Figura 3.8: Vista de las mascotas de un adulto mayor

Añadir evento

Título

Fecha

Connotación

Recurrente

Agregar Evento

Figura 3.9: Formulario para añadir un evento

Por otro lado, si el adulto mayor tiene eventos registrados, en su perfil aparecerá la lista de eventos que tiene junto con el botón “Editar Eventos” (Figura 3.10). Al hacer click en este botón, el usuario es redirigido hacia la vista que muestra la lista de eventos del adulto mayor (Figura 3.11).

Tomás Coordinador
Editar mi Información
Cerrar Sesión

Datos de Tite Escalante

Datos personales

Nombre	Tite Escalante
Género	No Especificado
Cumpleaños	No especificado
Grupo Etéreo	No Especificado
Teléfono	No especificado
Teléfono 2	
Dirección	No especificado

Editar Datos Personales

Contacto no tiene foto de perfil.

Redes Sociales

No tiene redes sociales

Agregar Redes Sociales

Mascotas Actuales

No tiene mascotas registradas

Agregar Mascota

Eventos

Fecha	Título	Connotación	¿Es recurrente?
18 de Febrero de 2000	Aniversario Muerte Papá	Negativo	Sí

Editar Eventos

Figura 3.10: Vista del detalle de un adulto mayor con eventos

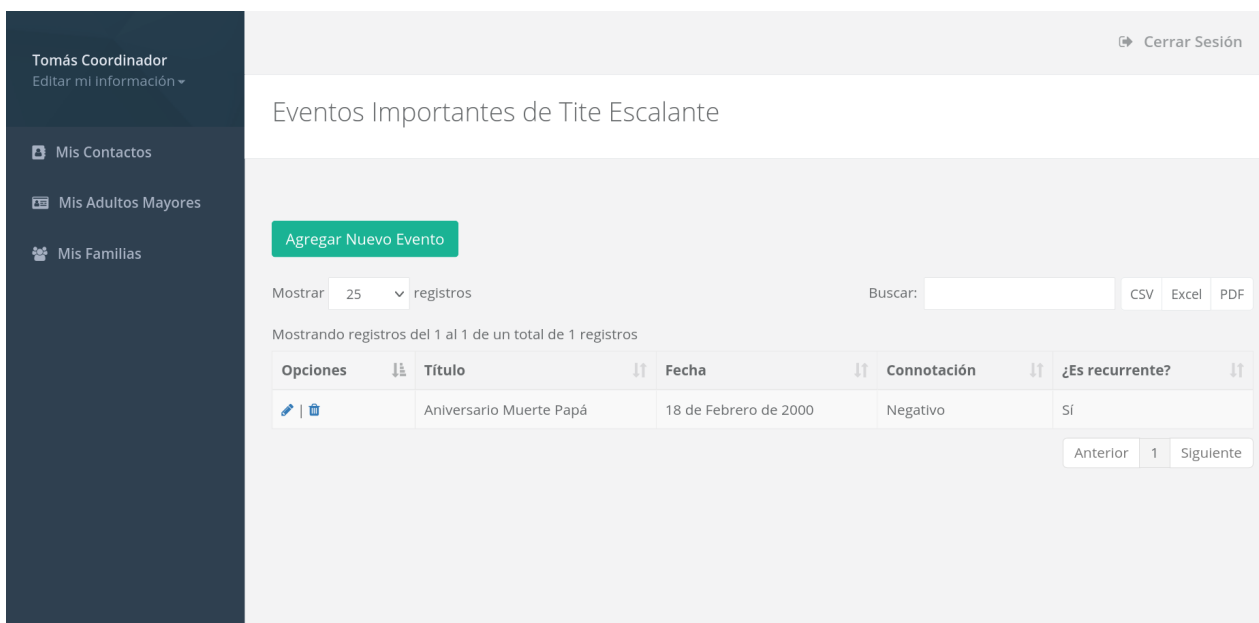


Figura 3.11: Vista de los eventos de un adulto mayor

Por último, se implementaron dos pantallas con la finalidad de mantener actualizada la información correspondiente a las mascotas y eventos de un adulto mayor. Este mecanismo consta de dos modales html, los cuales son explicados en mayor detalle a continuación.

El primer modal tiene por objetivo actualizar el estado de las mascotas. Este consta de un formulario, el cual contiene todas las mascotas cuya fecha de última actualización (campo `updated_date`) sea mayor a tres meses de antigüedad, que tengan el campo `death_date = null` (es decir, que sigan vivas) y que pertenezcan a adultos mayores cuyo grupo familiar sea coordinado por el usuario actual (al que se le muestra el modal). Este modal puede ser visto en la Figura 3.12.



Figura 3.12: Modal para actualizar el estado de las mascotas registradas en el sistema

Es posible observar que este modal se organiza como una tabla que posee 3 columnas, correspondientes al adulto mayor, el nombre de su mascota y un *dropdown menu* para actualizar su estado. Este *dropdown menu* contiene dos opciones; “SI” y “NO”. La idea es que el usuario coordinador pueda responder con alguna de estas dos opciones si es que la mascota sigue viviendo con el adulto mayor. En caso de seleccionar “SI”, entonces el estado de la mascota no se modifica. Por el contrario, si se selecciona la opción “NO”, entonces se marca la fecha actual como fecha de defunción de la mascota (`death_date`). En ambos casos se modifica la fecha de última modificación (`updated_date`) de la mascota, a fin de evitar preguntar en intervalos inferiores a tres meses. Por último, al pulsar “Actualizar”, se valida que ninguna selección sea nula y se envía el formulario para actualizar el estado de las mascotas.

El segundo modal tiene por objetivo recordar al usuario coordinador que mantenga actualizados los perfiles de los adultos mayores (de los grupos familiares que coordina). A diferencia del modal implementado para el seguimiento de las mascotas, este modal no posee un formulario para añadir/modificar eventos (ya que se asume que los eventos que ya están registrados en la plataforma no sufrirán cambios). En cambio, este modal funciona como un recordatorio, mostrando un mensaje al usuario coordinador para que recuerde actualizar el perfil de los adultos mayores cada tres meses. Este mensaje puede ser visto en la Figura 3.13.

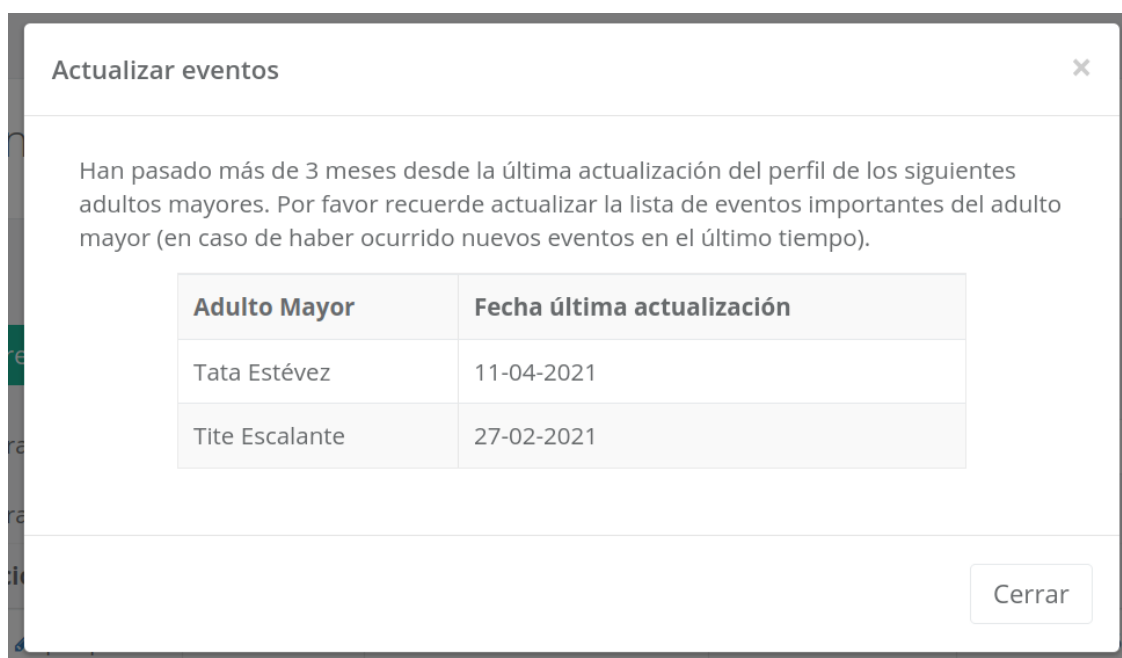


Figura 3.13: Modal mostrado al usuario coordinador para actualizar los eventos de un adulto mayor

Al igual que como sucede con las mascotas, en este modal se muestran únicamente los adultos mayores que formen parte de los grupos familiares coordinados por el usuario actual (al que se le muestra el modal) y cuya última actualización de perfil (`updated_date`) sea mayor a los tres meses de antigüedad. También, una vez que se muestra este mensaje, el campo `updated_date` es actualizado a la fecha actual, para evitar recordar al usuario actualizar el perfil de un adulto mayor a intervalos de tiempo menores a tres meses.

Estos dos modales son mostrados al usuario coordinador al iniciar sesión en la plataforma

web de SocialConnector. El flujo para mostrar dichos modales consiste en mostrar primero el modal para actualizar el estado de las mascotas, y una vez se cierre dicho modal, mostrar el modal para actualizar los eventos. En caso de no haber mascotas por actualizar (todas las mascotas tienen fecha `updated_date` menor a tres meses de antigüedad), se muestra directamente el modal para actualizar eventos. Por el contrario, si es que no existen perfiles de adultos mayores por actualizar (pero si mascotas), se muestra solamente el modal correspondiente a las mascotas. Por último, si no existen ni mascotas ni perfiles de adultos mayores por actualizar, no se muestra ningún modal.

Capítulo 4

Diseño e implementación de visualizaciones

Una vez extendido el modelo de contexto del sistema, se investigaron tipos de visualizaciones que pudiesen ayudar a los investigadores a comprender de mejor manera los datos relativos a la interacciones almacenadas. Con este objetivo, se investigaron varias de las bibliotecas de JavaScript más utilizadas para generar visualizaciones web.

De esta investigación, se definió que se utilizarían las bibliotecas D3.js y Chart.js, debido a que son *open-source*, cuentan con una amplia base de desarrolladores que las utilizan (lo que se traduce en un amplio soporte), y además el memorista tiene experiencia utilizándolas. Con las bibliotecas a utilizar definidas, se revisitó material del curso Visualización de Información (CC5208, dictado por el profesor Benjamín Bustos) y se visitaron las páginas web de D3.js y Chart.js en busca de visualizaciones de ejemplo, enfocándose principalmente en gráficas basadas en series de tiempo.

Luego, se definieron ciertas bases comunes sobre las cuales se construirían todas las visualizaciones. Estos puntos (bases comunes) fueron los siguientes:

- Solo se considerarán interacciones ocurridas entre el período 8:00 - 22:00 hrs. Esto permite mantener la consistencia con el horario de los indicadores *ventanas de tiempo sin interacción física/digital* definidos en la subsección 3.2 (Definición de indicadores).
- Los días se dividirán en tres bloques de tiempo, con la finalidad de poder realizar un análisis más fino. Los tres bloques tendrán igual duración (4:40 hrs) y corresponderán a mañana (8:00 - 12:40), tarde (12:40 - 17:20) y noche (17:20 - 22:00). La definición de los bloques también puede ser vista en la Figura 4.1.

Desde este punto en adelante, nos referiremos a los pares “día de la semana - bloque horario”, por ejemplo, martes en la tarde o jueves en la noche, simplemente como “bloque horario”.

A continuación, se muestran las visualizaciones implementadas en la aplicación. Para cada visualización se explicará su objetivo específico y los detalles de su implementación. Además,



Figura 4.1: Definición de bloques horarios

en el anexo E se exponen otras visualizaciones exploradas que no fueron incorporadas en el producto final.

4.1. Heatmap

4.1.1. Objetivo de la visualización

Esta visualización tiene dos principales objetivos:

1. Observar qué tan frecuente es la interacción entre un adulto mayor - contraparte en cada bloque horario.
2. Observar la frecuencia total de interacción (es decir, la suma de todas las contrapartes) en cada bloque horario.

Esto se calcula entre un período de fechas especificadas, sobre las cuales se genera el análisis.

4.1.2. Descripción

Un heatmap (o mapa de calor) es una visualización 2D que permite graficar la intensidad de un fenómeno a través de una escala de colores. Esta es básicamente una matriz (no necesariamente cuadrada), donde cada casilla (i,j) está coloreada según qué tan intenso es el fenómeno entre las variables (i,j) . Esta escala de color es definida en base a la intensidad del fenómeno que se busca modelar.

En este caso en específico, se diseñó un heatmap, donde cada columna corresponde a un bloque horario distinto y cada fila corresponde a un miembro del grupo familiar del adulto mayor que se seleccione para el análisis. Además, se añadió una fila adicional que corresponde a un resumen total. Esta última fila es la suma acumulada de todas las contrapartes; por ejemplo, la casilla resumen total del bloque horario martes 8:00 - 12:40, contiene la suma de las interacciones de todas las contrapartes en ese bloque horario. Así, si el adulto mayor tiene registrados tres miembros de la familia en la plataforma, el heatmap generado tendrá cuatro filas (tres miembros + resumen total) y 21 columnas (las columnas son siempre fijas). Un ejemplo de la matriz generada (vacía, sin colores) puede ser visto en la Figura 4.2.

Para filtrar la información que alimenta al heatmap, se implementó un formulario donde



Figura 4.2: Heatmap vacío

el usuario debe ingresar el adulto mayor a observar, el tipo de interacción y el rango de fechas sobre el cual quiere generar el heatmap. Un ejemplo de este formulario puede ser visto en la Figura 4.3.

(a) Formulario con fechas predefinidas

(b) Formulario con fechas variables

Figura 4.3: Formulario para la gráfica de Heatmap

La descripción detallada de cada campo del formulario se indica a continuación:

- **Adulto Mayor:** Este campo corresponde al adulto mayor sobre el cual se generará la gráfica. El sistema despliega un *dropdown menu* con todos los adultos mayores registrados en la plataforma.
- **Tipo de Interacción:** Corresponde al tipo de interacción que se analizará. Este campo tiene tres opciones, “Físicas” (visitas), “Digitales: Llamadas” (llamadas o videollamadas) y “Digitales: Mensajes” (mensajes).
- **Tiempo de Análisis:** Es la fecha de interés para el análisis. Para esta casilla, el sistema despliega un menú que ofrece 5 opciones predefinidas al usuario las cuales son “Última Semana”, “Último Mes”, “Último Trimestre”, “Último Semestre” y “Último Año”. Además, ofrece una sexta opción, la cual es “Otras Fechas”. Al seleccionar esta opción, el sistema muestra otros dos campos en el formulario “Fecha Inicio” y “Fecha Término” (Figura 4.3 (b)), que corresponden justamente a la fecha de inicio y término del período de interés.

En la Figura 4.3 es posible observar el formulario cuando se selecciona una opción predefinida (a), y cuando se seleccionan fechas variables (b).

Una vez definidas las bases del heatmap, se procedió a explorar las distintas alternativas de escalas de colores que se podrían utilizar para colorear las celdas de la matriz.

4.1.2.1. Análisis de alternativas de representación de colores

Para este proceso, se identificaron tres decisiones claves que fueron necesarias para determinar que escala de color utilizar. Estas decisiones fueron las siguientes:

1. Qué variable se mapeará al color.
2. Qué colores se utilizarán para la escala.
3. Determinar si se utilizará una escala de colores continua o discreta.

Con respecto al punto 1, la variable que se seleccionó para mapear en el heatmap corresponde al porcentaje de días en los cuales hubiera interacción entre el adulto mayor y una determinada contraparte. Para explicar de mejor manera la variable elegida, tomaremos como ejemplo el período de tiempo que va entre el 1 de febrero de 2021 y el 28 de febrero de 2021.

FEBRERO 2021						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Figura 4.4: Calendario de Febrero 2021

Para ser más específicos, estudiaremos el bloque horario de martes 12:40 - 17:20. En el calendario es posible observar que durante el período de estudio (01/02/21 - 28/02/21) hay cuatro días martes (2, 9, 16, 22). Si queremos saber cuál es el porcentaje de días en que hubo interacción entre un adulto mayor y una determinada contraparte, particularmente en el bloque del martes en la tarde (12:40 - 17:20), entonces debemos calcular la cantidad de martes en que hubo interacción entre el adulto mayor y la contraparte en dicho bloque, y a ese valor hay que dividirlo por el total de días observados (en este caso, sobre los cuatro martes antes indicados). Así, por ejemplo, si es que solo hay interacciones en dos de los cuatro martes, el porcentaje de interacción sería igual a 50 %.

Por lo tanto, en términos generales podemos establecer que para cada contraparte y cada bloque horario, su porcentaje de interacción es igual a la cantidad de días en los cuales hubo interacción, dividido en la cantidad total de días.

Para el punto 2, en una primera instancia se optó por una escala que fuera desde el color rojo (nula/baja frecuencia de interacción) hacia el verde (alta frecuencia de interacción). Sin embargo, esta escala de colores confundía al usuario debido a la presencia de distintos colores (rojo, amarillo y verde); en este caso la información se hacía difícil de comprender. Debido a ello, se optó por usar una escala monocromática de tonalidades azules.

Se escogió el color azul por ser un color que transmite calma y tranquilidad [27], a diferencia del rojo que puede relacionarse con el peligro [8] o el color verde que puede relacionarse con correctitud [9] (y con ello introducir un sesgo en la visualización).

El último paso fue decidir si la escala de colores sería continua o discreta. En este caso, se optó por una escala discreta, de cinco pasos, como se muestra en la Figura 4.5. Esta decisión se tomó principalmente para evitar confundir al usuario final y así poder presentar la información de manera más clara. Por ejemplo, el usuario puede tener problemas para distinguir distintos bloques horarios si es que existen múltiples tonalidades de colores en pantalla, situación que puede ocurrir en el caso de usarse una escala continua.

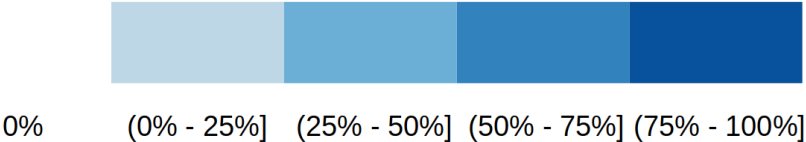


Figura 4.5: Escala de color seleccionada

En la figura se ve que esta escala toma pasos de 25%. Se opta por dejar el 0% sin color alguno (transparente) para minimizar el ruido que pueda llegar a ocasionar el mostrar casilleros con dicho valor. Con todo lo definido anteriormente, un ejemplo de resultado final de la gráfica implementada es el mostrado en la Figura 4.6.

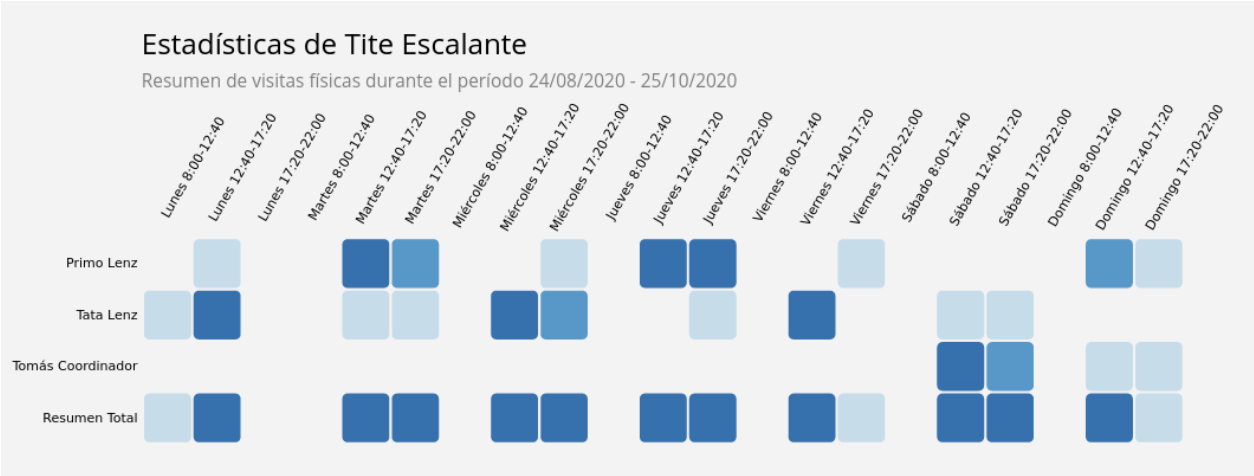


Figura 4.6: Visualización de heatmap resultante

En este ejemplo, es posible ver que las casillas que no tienen interacciones (por ejemplo, toda la columna martes 8:00 - 12:40) no están coloreadas, mientras que aquellas casillas que tienen patrones de interacción fuertes, están en tonalidades azul oscuro (por ejemplo, Primo Lenz, los martes de 12:40 - 17:20).

4.1.2.2. Implementación de interacciones de usuario con las gráficas

Para finalizar, se implementaron las interacciones entre el usuario y la gráfica. Estas interacciones son principalmente dos, la activación/desactivación de familiares en el heatmap, y la visualización del detalle de un bloque horario.

En el caso de la interacción para activar/desactivar familiares, esta surgió en conversación con expertos de dominio. Básicamente le da al usuario la posibilidad de elegir qué contrapartes serán sumadas en el resumen total. Recordar que la fila resumen total, corresponde a la suma de las interacciones de todas las contrapartes en cada bloque horario. Dado eso, la opción de desactivar a un familiar del heatmap, esconde la fila correspondiente a sus datos e ignora las interacciones de dicho familiar para el resumen total. Un ejemplo de esto puede ser visto en la Figura 4.7.

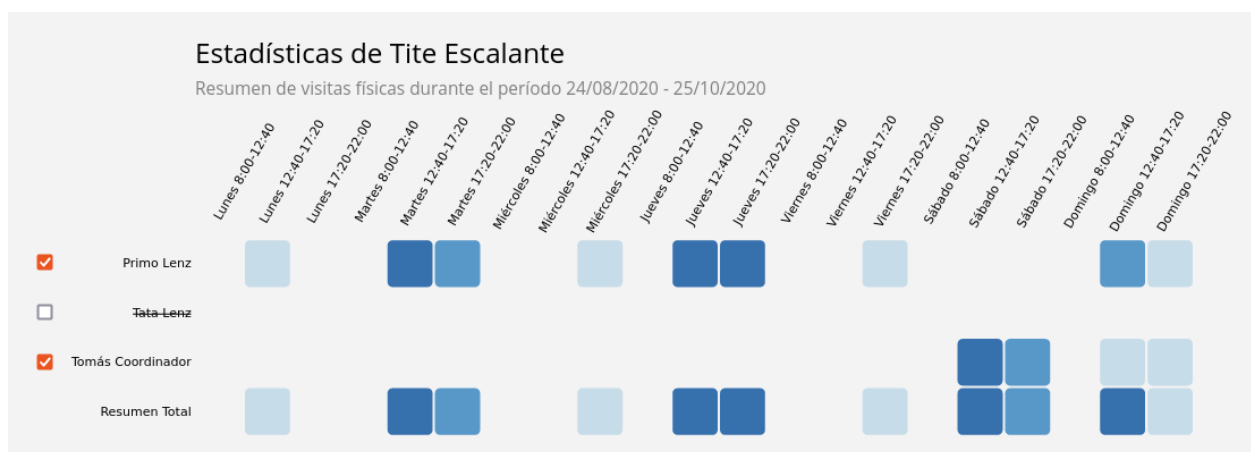


Figura 4.7: Ejemplo de heatmap con un miembro de la familia desactivado

Tomando esta Figura como ejemplo, es posible observar que en este caso se desactiva al usuario “Tata Lenz”. Como se explicó anteriormente, esto oculta las casillas correspondientes a su fila en el heatmap y no considera sus interacciones de la fila de resumen total. En este caso, con “Tata Lenz” desactivado, la fila de resumen total corresponde únicamente a la suma de las interacciones de “Primo Lenz” y “Tomás Coordinador”. Allí es posible observar las diferencias de la gráfica que contiene al heatmap con todos los usuarios activos (Figura 4.6) con la recién expuesta, principalmente a nivel de la fila resumen total, donde se ve claramente que esta ahora no considera las interacciones de “Tata Lenz”.

Como se puede ver en la Figura 4.7, se implementaron *checkboxes* para la activación / desactivación de usuarios en el heatmap. Al generar la gráfica, todos los *checkboxes* están activados (marcados) y el usuario tiene la opción de desmarcar aquellos que quiera excluir del análisis.

Al desmarcar un usuario, se ocultan sus interacciones en el heatmap, se tacha su nombre de la gráfica, e ignoran sus interacciones para el cálculo del resumen total. Al volver a activar dicho usuario, se restituye su estado a su estado inicial (es decir, muestra sus datos en el heatmap, su nombre sin tachar y se incluyen sus interacciones en el resumen total).

Esta funcionalidad es particularmente útil para poder estudiar el comportamiento de un

grupo determinado de personas. Por ejemplo, dentro de la familia del adulto mayor podría haber muchos integrantes de distinto parentesco (por ejemplo, hermanos, hijos, nietos, cuñados, etc.). En el caso de querer estudiar el comportamiento de algún grupo en específico (por ejemplo, los nietos), bastaría únicamente con desactivar a las personas que no forman parte del grupo de interés, para así poder realizar el análisis específico.

Por último, es necesario mencionar que esta funcionalidad de activar/desactivar usuarios en el heatmap no implica activar/desactivar usuarios a nivel de la plataforma SocialConnector. Es decir, esta funcionalidad solo está pensada para modificar la gráfica, no para activar/desactivar el perfil del usuario a nivel base de datos.

Otra funcionalidad que se implementó para interactuar con el usuario es la de mostrar el detalle de un bloque horario. Esta funcionalidad consiste básicamente en mostrar el total de interacciones y la serie de tiempo asociada a un determinado bloque horario. Esta funcionalidad se compone de dos elementos principales. El primero es un *tooltip*, el cual al poner el cursor sobre un determinado bloque horario, despliega el total de interacciones en dicho horario. Un ejemplo de esto puede ser visto en la Figura 4.8.



Figura 4.8: *Tooltip* desplegado al poner el cursor sobre un bloque horario

En dicha imagen se puede observar el *tooltip* desplegado en el caso de un heatmap de interacciones físicas. Allí se muestra el tiempo total de visita, es decir, la suma de la duración todas las visitas que hayan ocurrido en ese bloque horario para esa contraparte en específico.

En el caso de analizar las interacciones digitales de tipo llamada, este *tooltip* mostrará la misma información (cantidad de horas de llamada). Por otro lado, si se analizan interacciones digitales de tipo mensaje, el *tooltip* mostrará la cantidad de mensajes intercambiados entre ambas partes.

El segundo elemento de esta funcionalidad está relacionado con el texto “Click para ver detalle”, que se muestra en la Figura 4.8. Este entrega al usuario la opción de ver en detalle la evolución del nivel de interacción entre una contraparte y el adulto mayor en el bloque horario que se ha seleccionado. En ese caso el sistema despliega la serie de tiempo correspondiente a las interacciones entre el adulto mayor y la contraparte dentro del bloque horario clickeado. Un ejemplo de esto puede ser visto en la Figura 4.9.

Para una mejor comprensión de esta imagen, es necesario explicar cómo se construye esta serie de tiempo. Particularmente, en el eje X se genera un punto por cada día de interés que haya dentro del período a analizar. Por día de interés nos referiremos al día al cual

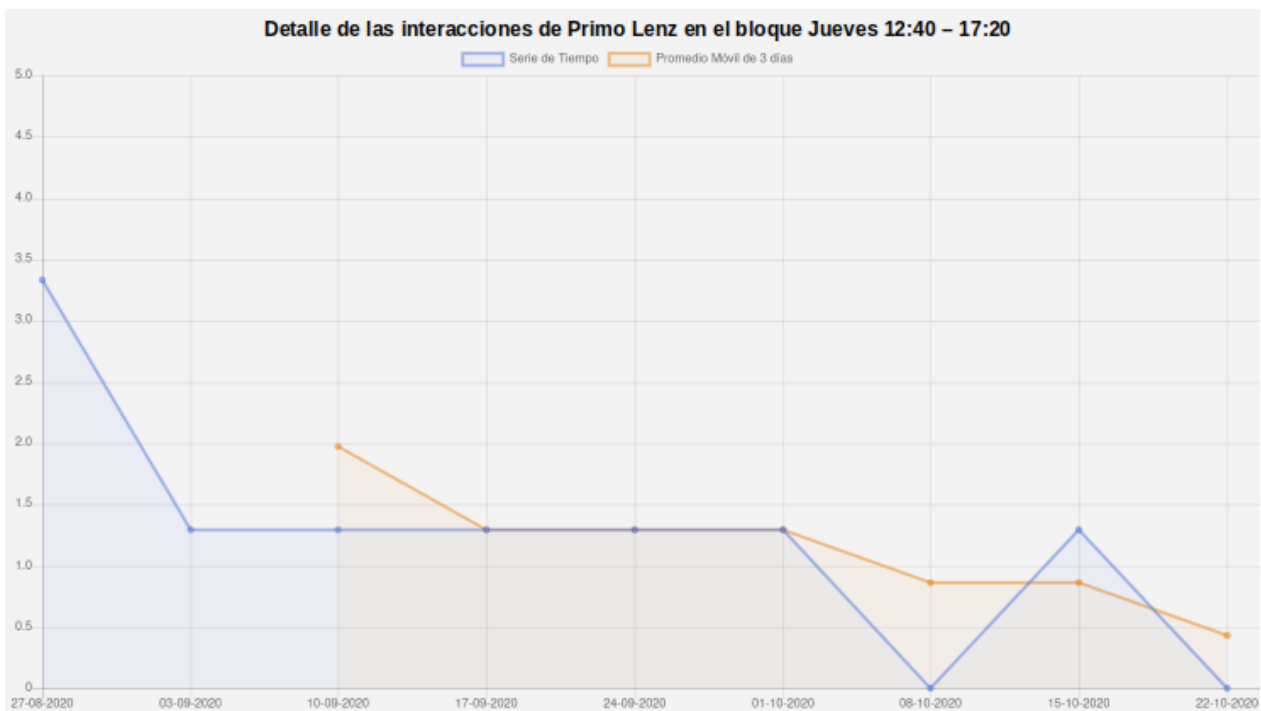


Figura 4.9: Ejemplo de serie de tiempo desplegada al hacer click en un bloque horario

pertenece el bloque horario que fue seleccionado. Por ejemplo, si se selecciona el bloque horario correspondiente al jueves 12:40 - 17:20 (como en la Figura 4.9), entonces el día de interés es el día jueves.

El valor que toma cada punto en la serie de tiempo depende de qué tipo de interacción se esté analizando. En el caso de las interacciones continuas (físicas o digitales tipo llamada), cada punto de esta serie toma el valor de la cantidad de horas de interacción que la contraparte haya tenido con el adulto mayor en el día determinado. Esto se puede visualizar en el ejemplo de la Figura 4.9, donde si se observa el punto correspondiente al 27/08/2020, podemos ver que tiene un valor cercano a 3.3, lo cual se interpreta como que el día 27/08/2020, “Primo Lenz” visitó durante 3 horas y 20 minutos al adulto mayor analizado (en este caso, a “Tite Escalante”).

Por otro lado, en el caso de las interacciones discretas (digitales tipo mensajes), el valor que toma cada punto en la serie corresponde a la cantidad de mensajes intercambiados entre el adulto mayor y la contraparte respectiva en el bloque horario seleccionado. El detalle del valor de cada punto de la serie se muestra mediante un *tooltip*, como se puede ver en la Figura 4.10.

En la gráfica es posible ver dos líneas: la azul es la serie de tiempo original (descrita en el párrafo anterior), mientras que la serie naranja representa el promedio móvil con una ventana de 3 días, que se incluye en dicha gráfica para ayudar al usuario que vea la gráfica a detectar la tendencia en los datos. Sin embargo, si el usuario quisiera tener una gráfica “más limpia”, existe la posibilidad de desactivar ciertas líneas del gráfico, clickeando en la leyenda correspondiente.



Figura 4.10: *Tooltip* que muestra el detalle del valor de cada punto de la serie de tiempo

La posibilidad de mostrar este detalle es muy importante para el sistema, ya que la variable que se seleccionó para mapear a la visualización (porcentaje de días en los que hay interacción) posee una potencial debilidad. Esta es que por su concepción, solo indica qué tan frecuente es que un adulto mayor interactúe con una determinada contraparte en un bloque horario, sin considerar el volumen de dicha interacción. Por ejemplo, podría darse el caso extremo en que una contraparte visite a un adulto mayor todos los jueves en el bloque 12:40 - 17:20, pero lo haga solamente por dos minutos, lo cual claramente no es una visita “muy duradera”.

Para ese caso, el heatmap colorearía el casillero correspondiente a esa contraparte los jueves 12:40 - 17:20 con una tonalidad correspondiente al 100%. Sin embargo, a pesar de que el patrón es muy fuerte (está presente en todas las semanas), el volumen de interacción sigue siendo muy bajo (2 minutos por visita), por lo cual el investigador que analiza los datos, al ver el detalle, tiene la posibilidad de observar dicho escenario y discernir si el patrón de interacción tiene un volumen que considere “significativo”.

4.2. Boxplot

4.2.1. Objetivo de la visualización

Esta visualización tiene por objetivo estudiar el comportamiento histórico e identificar anomalías en las interacciones adulto mayor - contraparte, para cada bloque horario durante un período de tiempo definido.

4.2.2. Descripción

Un boxplot (o diagrama de caja) es una visualización 2D, la cual usualmente se utiliza para estudiar una variable cuantitativa. Es una gráfica estandarizada para estudiar la distribución de una variable en base a cuatro parámetros: el primer cuartil (Q_1), la mediana, el tercer cuartil (Q_3) y su distancia inter-cuartil ($Q_3 - Q_1$). Con estos parámetros se puede identificar anomalías a nivel estadístico, además de poder decir qué tan simétrica y qué tan compacta es la distribución de los datos [17].

En la Figura 4.11 podemos observar un ejemplo de boxplot, donde el mínimo (*min*) corresponde a $Q_1 - 1,5 * (Q_3 - Q_1)$ y el máximo (*max*) corresponde a $Q_3 + 1,5 * (Q_3 - Q_1)$. Cada punto fuera del rango $[min, max]$ puede ser etiquetado como un *outlier* o anomalía.

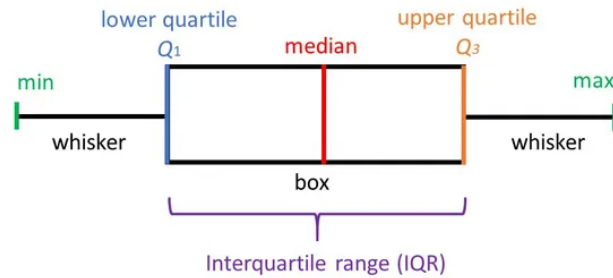


Figura 4.11: Ejemplo de gráfica de boxplot, extraído de [25]

Para la aplicación desarrollada en esta memoria se diseñó una visualización que tuviera un boxplot por cada bloque horario en la semana (o sea, 21 boxplots). Es decir, el eje X corresponde a los bloques horarios, mientras que el eje Y corresponde a las horas de visita/llamada (en el caso de que se analicen interacciones físicas o digitales tipo llamadas) o cantidad de mensajes intercambiados (en el caso de que se analicen interacciones digitales de tipo mensajes).

Al igual que en el caso del heatmap, se utiliza un formulario muy similar al exhibido en la Figura 4.3 para filtrar la información que grafican los distintos boxplot. La única salvedad es que para generar las gráficas de boxplot, se requiere adicionalmente indicar la contraparte específica que se va a analizar. Este formulario se puede observar en la Figura 4.12.

Figura 4.12: Formulario para la gráfica de Boxplot

El campo “contraparte” que se añade despliega un *dropdown menú* con todos los miembros de la familia del adulto mayor seleccionado, además de ofrecer la opción “todos”, que como su nombre lo dice, incluye a todos los miembros registrados de la familia. Este campo “contraparte” tiene por objetivo permitir filtrar todas las interacciones que no sean entre el adulto mayor y la contraparte seleccionada. En el caso de seleccionar “todos”, se traen todas las interacciones que haya tenido el adulto mayor durante el período de tiempo seleccionado, independiente de su contraparte.

Los datos que utiliza cada boxplot son similares a los utilizados para crear la serie de tiempo del heatmap, la cual se muestra en la Figura 4.9. Esto es, para un determinado bloque horario, se genera un punto por cada día de interés que haya en el período de fechas a analizar. Cada punto tomará un valor dependiendo del tipo de interacción que se esté analizando. En

el caso de las interacciones físicas y digitales tipo llamadas, este punto tomará el valor de las horas totales de interacción entre el adulto mayor y la contraparte seleccionada durante el respectivo día. Para las interacciones digitales tipo mensajes, este punto tomará como valor la cantidad de mensajes intercambiados entre el adulto mayor y la contraparte. En el caso de seleccionar “todos” como contraparte, entonces para cada bloque horario, el sistema generará un punto por cada día de interés y cada miembro de la comunidad familiar.

Para explicar de mejor forma lo expuesto en el párrafo anterior, ejemplificaremos nuevamente con el período 01/02/21 - 28/02/21 (Figura 4.4), y nos enfocaremos únicamente en el bloque del jueves entre 12:40 - 17:20. Es posible notar que existen cuatro jueves (día de interés para el bloque horario seleccionado) en el período de estudio. Para el campo “contraparte”, tenemos dos casos posibles, se elige un miembro familiar en específico, o se elige la opción “todos”. A continuación se explica brevemente cada opción.

- **Contraparte específica:** Si se escoge un miembro familiar en específico (y no la opción “todos”), entonces para el bloque jueves 12:40 - 17:20 se generarán cuatro puntos de datos (uno para cada jueves). Cada punto tendrá su valor correspondiente (número de horas o número de mensajes intercambiados entre la contraparte - adulto mayor).
- **Todos:** En el caso de escoger el campo “todos”, entonces el sistema realizará el proceso mencionado en **contraparte específica** para cada miembro familiar. Es decir, generará 4 puntos por cada miembro de la familia, donde cada punto corresponderá a las horas/mensajes intercambiados entre la contraparte - adulto mayor.

Una vez que se procesan los datos para alimentar la gráfica, se calculan los parámetros necesarios para cada boxplot (Q_1 , *mediana*, Q_3 y $Q_3 - Q_1$) y se genera la visualización. Un ejemplo de la visualización generada puede ser visto en la Figura 4.13.

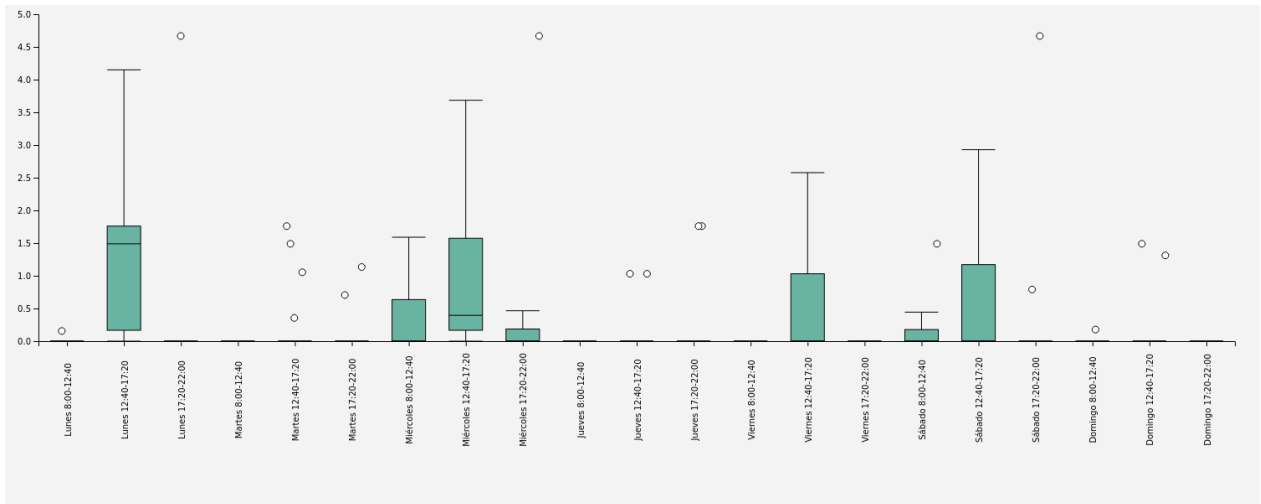


Figura 4.13: Visualización de boxplot resultante

En la figura se observa que los *outliers* son graficados como puntos blancos en sus respectivos bloques horarios. Por razones de diseño, los puntos que no corresponden a anomalías no son graficados explícitamente dentro de la gráfica final, debido a que interfieren con su correcta visualización. Además, se tomaron medidas para reducir la posibilidad de que los puntos graficados se superpongan entre sí, introduciendo un pequeño factor aleatorio que se

le suma/resta (dependiendo del signo que tenga) a la posición X del punto. Sin este factor, todos los puntos se graficarían en la parte central de su respectivo boxplot (Figura 4.14 (a)), mientras que con el factor, se grafican en una posición aleatoria, pero siempre dentro de los márgenes de su respectivo boxplot (Figura 4.14 (b)).



(a) Puntos gráficos sin factor aleatorio en su posición X

(b) Puntos gráficos con factor aleatorio en su posición X

Figura 4.14: Detalle de la gráfica de *outliers* dentro de los boxplot

Debido a que el sentido de la gráfica es justamente el estudio de anomalías, se implementó un *tooltip* para que el usuario pueda explorar las anomalías que se vean en la gráfica. Este *tooltip* se muestra al usuario cuando sitúa el cursor encima de uno de los puntos blancos. El detalle de la información que se le muestra al usuario en el *tooltip* puede ser visto en la Figura 4.15.



Figura 4.15: *tooltip* que muestra el detalle de un *outlier*

Tal como se ve en la figura, en el *tooltip* se muestra información relativa a qué contraparte corresponde el punto, en qué fecha ocurrió y cuál es su valor. En este caso, como el ejemplo se realizó sobre interacciones físicas, se muestra el tiempo total de visitas. Por el contrario, si el ejemplo se hubiera realizado sobre mensajes, entonces se mostraría el total de mensajes intercambiados entre ambas partes.

Además de mostrar un pequeño resumen en el *tooltip*, se le da la opción al usuario de clicar los puntos, lo cual despliega mayor información del día en cuestión. Concretamente, al hacer click en un *outlier*, el sistema muestra un resumen del clima y de los eventos (en caso de haber) que el adulto mayor tenga registrados en esa fecha. Un ejemplo de este detalle, se muestra en la Figura 4.16.

Este resumen diario es desplegado debajo de la gráfica de los boxplot (en la misma pantalla) y su idea es que pueda darle al investigador un mayor contexto sobre la anomalía que este clickee. Contando con mayor información de contexto, el investigador tiene la opción

Clima		Eventos importantes		
Fecha:	26-08-2020	Evento	Fecha	Connotación
Temperatura Promedio:	19.4°C	Aniversario Muerte Papá	26-08-1996	Negativo
Lluvia?	Sí			
Estación:	Invierno			
Feriado?	No			
Comuna:	Providencia			
Ciudad:	Santiago			
País:	Chile			
Tiempo de Luz:	9:58:21			

Figura 4.16: Vista de detalle diario, mostrado al clicar un punto de la gráfica

de explorar las eventuales relaciones entre la anomalía y la información expuesta en el resumen. Un ejemplo de ello puede ser el cumpleaños del adulto mayor, donde pueden generarse anomalías debido a la alta cantidad de visitas generadas (si es que sus familiares concurren a visitarlo). Si este evento está registrado en la plataforma, entonces al hacer click en estas anomalías el sistema entregará dicha información en el resumen de eventos diarios, lo cual permite explicar el fenómeno.

4.3. Clustering

4.3.1. Objetivo de la visualización

El objetivo de esta visualización es ayudar a identificar la existencia patrones semanales de comportamiento (en cuanto a cantidad de interacciones) entre una tupla adulto mayor - contraparte, durante un período de tiempo definido.

4.3.2. Descripción

Como se explicó en la sección 2.6.2, las técnicas de clustering tienen por objetivo encontrar agrupaciones naturales sobre los datos que se procesan. Es decir, el clustering agrupa datos que se parecen respecto a la métrica que se defina para medir la similitud entre los elementos.

Para este caso, se implementó un procesamiento de clustering jerárquico para la serie de tiempo de interacciones entre un adulto mayor - contraparte. Antes de explicar el procedimiento de clustering realizado, es necesario describir cómo se construyen las series de tiempo sobre las cuales corre el algoritmo.

La serie de tiempo correspondiente a las interacciones entre un adulto mayor y una determinada contraparte contiene un punto por cada bloque horario dentro del período de tiempo a analizar. Por ejemplo, si se quisiera analizar el período de tiempo correspondiente a febrero

del presente año (01/02/21 - 28/02/21), entonces la serie de tiempo tendría exactamente 28 (cantidad de días) * 3 (cantidad de bloques en un día) = 84 puntos.

El valor que toma cada punto de la serie depende de qué tipo de interacción se esté estudiando. Si el tipo de interacción a analizar son visitas físicas o interacciones digitales tipo llamadas (interacciones continuas), entonces cada punto en la serie de tiempo es la cantidad de horas totales que la contraparte interactuó con el adulto mayor en dicho bloque horario. Por otro lado, si el tipo de interacciones a estudiar es digital de tipo mensajes (interacciones discretas), cada punto representa la cantidad de mensajes que la contraparte intercambió con el adulto mayor durante dicho bloque horario.

Finalmente, para generar las series de tiempo que serán clusterizadas, se “quiebra” la serie antes generada en series de tiempo de tamaño 21. Es decir, en series de tiempo que representan exactamente una semana (por los 7 días de la semana que están divididos en 3 bloques cada uno $7 * 3 = 21$).

El algoritmo de clustering que se implementó primero calcula la matriz de distancia entre las series de tiempo semanales, utilizando el algoritmo de Alineamiento de Tiempo Dinámico, para luego entregarle la matriz calculada al algoritmo de clustering para que este forme los clusters. En este caso se utilizó la librería `dtaidistance`¹ de Python.

Para definir la información con la que se construirán las series de tiempo, el formulario es idéntico al implementado para la visualización de boxplot (Figura 4.12). Es necesario señalar que, como el clustering se realiza sobre semanas, si el usuario ingresa un período de tiempo que no puede ser dividido en semanas exactas, el sistema extiende las fechas hasta hacerlas calzar con fechas que sí sean semanas exactas. Es decir, si reutilizamos el ejemplo de febrero 2021 (Figura 4.4), un período de tiempo que no puede ser dividido en semanas exactas es 04/02/21 - 28/02/21, el cual es un rango de fechas que tiene exactamente 25 días, lo cual no es divisible por 7 (cantidad de días en una semana).

Para lidiar con estos casos, y garantizar que todas las series de tiempo semanales tengan la misma cantidad de datos, el sistema extiende hacia atrás (es decir, hacia la fecha de inicio) el período de tiempo hasta que el total de días si pueda ser dividido en semanas. En el caso del ejemplo, el período 04/02/21 - 28/02/21 sería extendido a 01/02/21 - 28/02/21 por el sistema.

El resultado del algoritmo de clustering realizado en el backend es mostrado al usuario mediante una gráfica de dendrograma, la cual fue implementada usando las capacidades de la librería D3.js. En la Figura 4.17 se muestra un ejemplo de gráfica final.

En esta visualización los nodos naranjos representan clusters, mientras que los nodos verdes representan cada una de las semanas dentro del período de tiempo que se estableció para el análisis. A simple vista no es posible ver a qué semana corresponde cada nodo. Sin embargo, posicionando el cursor sobre un nodo verde (semana), se despliega un *tooltip* que muestra cuál es la semana (ver Figura 4.18).

Además, existe la posibilidad de hacer click en los nodos semana (verdes), para ver en

¹dtaidistance clustering algorithm

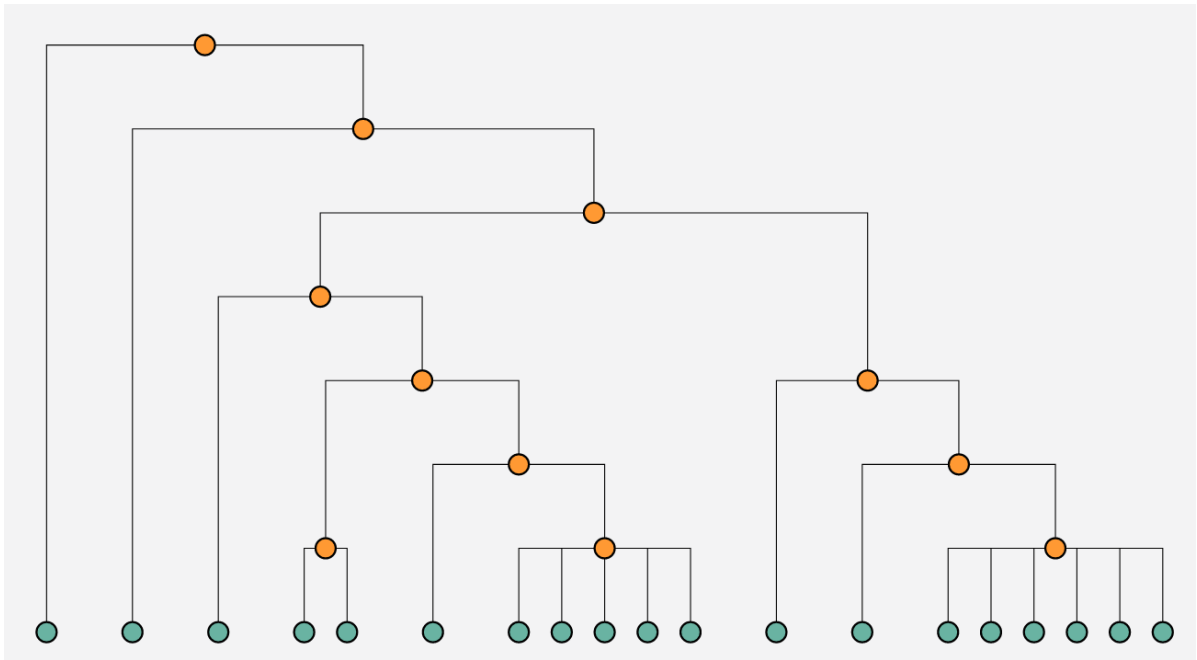


Figura 4.17: Visualización de dendrograma resultante del algoritmo de clustering jerárquico

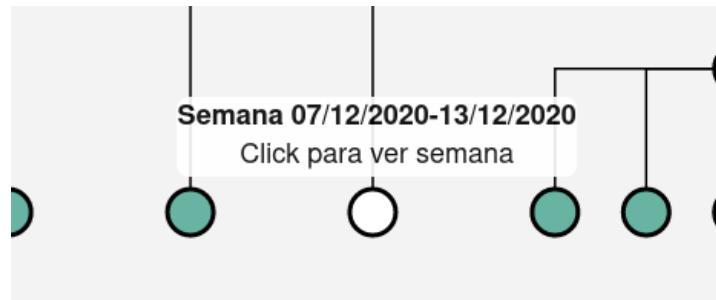


Figura 4.18: *Tooltip* que muestra a qué semana corresponde el nodo

detalle la serie de tiempo de dicha semana (Figura 4.19). Este detalle corresponde a la serie de tiempo de la semana sobre la cual se haga click.

Se puede tener el detalle de varias semanas desplegadas al mismo tiempo, con la finalidad de que el usuario pueda comparar visualmente las distintas semanas si es que lo desea, ya sean estas pertenecientes a los mismos o distintos clusters. Por ejemplo, en la Figura 4.20 se pueden observar dos semanas desplegadas al mismo tiempo. En este ejemplo, ambas semanas son idénticas y corresponden al mismo cluster.

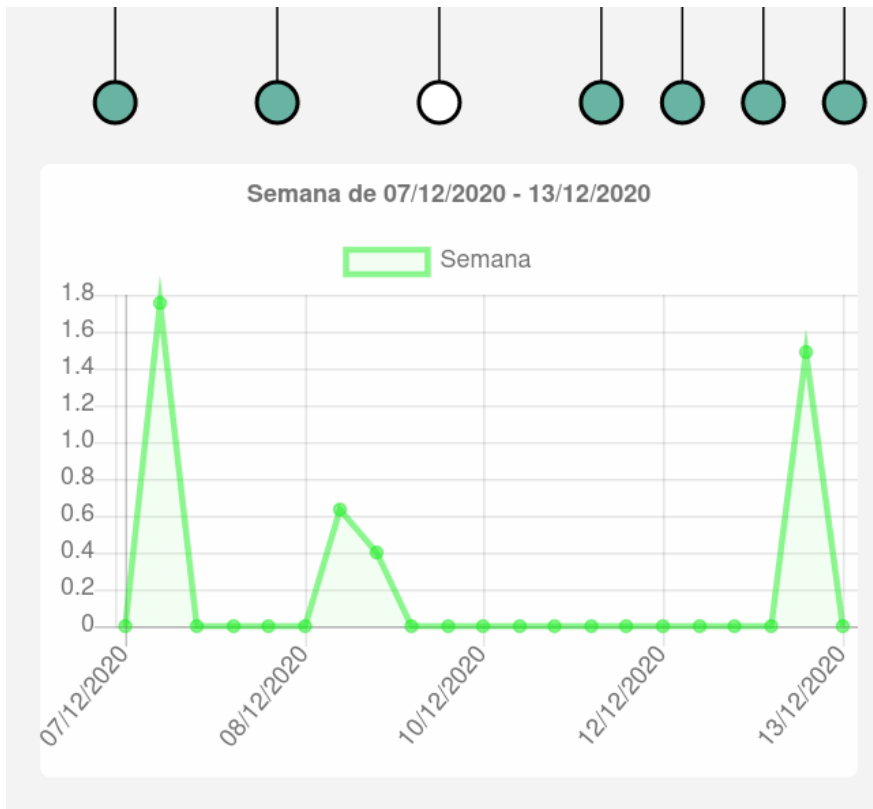


Figura 4.19: Detalle de la serie de tiempo de una semana específica



Figura 4.20: Vista detalle de dos semanas del mismo cluster, desplegadas al mismo tiempo

Capítulo 5

Diseño e implementación del Dashboard

El diseño e implementación del dashboard fue la etapa final de programación de este trabajo de título. El dashboard debía agrupar todas las visualizaciones, además de generar un resumen de la última semana con algunos datos potencialmente interesantes para los investigadores. El proceso de desarrollo del dashboard se dividió en tres etapas, las cuales son descritas en mayor detalle a continuación.

5.1. Definición de información relevante

Esta etapa del desarrollo consistió en definir un conjunto inicial de información para mostrar en el resumen semanal del dashboard. La idea de esto es que dicho resumen permita realizar un diagnóstico rápido de los indicadores más relevantes de cada adulto mayor y entregar algunos indicadores a nivel global (a nivel de la plataforma, no específicamente de un adulto mayor).

Para la definición del conjunto inicial de información potencialmente relevante se utilizó el software Tableau. Es importante aclarar que la versión de Tableau que se utilizó es la versión *Desktop*, la cual puede ser obtenida vía *Tableau for Students*, programa que entrega licencias gratuitas para estudiantes de instituciones reconocidas por Tableau. Esta herramienta solamente fue utilizada de forma *offline* para la exploración de datos, debido a que las licencias estudiantiles no aplican para *Tableau Server*, que es la aplicación que se requería para incorporar Tableau a SocialConnector.

Para explorar qué información de la base de datos puede ser relevante para exhibir en este resumen semanal, se realizó un *dump* de los datos de la aplicación en formato *json*, con el objetivo de poder cargar la base de datos a Tableau. Esta acción de volcar los modelos a un archivo *json* puede ser fácilmente realizada con el comando `Django-admin dumpdata`.

Una vez cargada la base de datos de SocialConnector en Tableau, se procedió a experimentar con distintos cruces de datos y generar campos calculados. Esto se hizo con la finalidad de encontrar cuáles de ellos pudieran ser interesantes de incorporar en el resumen semanal del dashboard.

De este proceso de exploración resultaron dos tipos de indicadores: los de tipo global y los de tipo local. Los indicadores de tipo global son aquellos que apuntan a realizar un diagnóstico a nivel de plataforma, mientras que los indicadores de tipo local corresponden a indicadores calculados sobre un adulto mayor en específico. A continuación se presentan los indicadores globales definidos.

- **Interacciones registradas:** Corresponde a la cantidad total de interacciones registradas por la plataforma durante la última semana.
- **Adultos mayores activos:** Muestra la cantidad de adultos mayores activos dentro de la plataforma.
- **Duración promedio de visitas:** Duración promedio de las interacciones de tipo físicas (independiente del adulto mayor) durante la última semana.
- **Duración promedio de llamadas:** Duración promedio de las interacciones digitales de tipo llamada (independiente del adulto mayor) durante la última semana.
- **Mensajes promedio recibidos:** Corresponde a la cantidad promedio de mensajes que recibió un adulto mayor durante la última semana.

Por otra parte, los indicadores locales definidos son los que se indican a continuación:

- **Duración promedio de visitas:** Duración promedio de las interacciones de tipo físicas del adulto mayor durante la última semana.
- **Duración promedio de llamadas:** Duración promedio de las interacciones digitales de tipo llamada del adulto mayor durante la última semana.
- **Mensajes recibidos:** Cantidad de mensajes recibidos por el adulto mayor durante la última semana.
- **Mensajes enviados:** Cantidad de mensajes enviados por el adulto mayor durante la última semana.
- **Duración promedio de ventanas sin interacción física:** Se calcula el promedio de la duración de las ventanas sin interacciones físicas. Este indicador apunta a mostrar qué tan espaciadas están las visitas que recibe el adulto mayor.
- **Duración promedio de ventanas sin interacción digital:** Este indicador muestra la duración promedio las ventanas sin interacción digital de un adulto mayor. Esta apunta principalmente a estudiar qué tan espaciadas están las interacciones digitales que recibe el adulto mayor.
- **Desglose de interacciones según tipo:** La idea de este punto es poder implementar una gráfica de barras simple donde se desglose el tipo de interacciones que haya tenido el adulto mayor.
- **Contraparte con más interacciones físicas:** Se muestra quién es la contraparte con más cantidad de interacciones físicas con el adulto mayor durante la última semana.
- **Contraparte con más interacciones digitales:** Se muestra quién es la contraparte con más cantidad de interacciones digitales con el adulto mayor durante la última semana.
- **Día de mayor interacción:** Se muestra cuál es el día de la semana donde el adulto mayor registró la mayor cantidad de interacciones.

Con los indicadores del resumen semanal ya definidos, el siguiente paso fue diseñar el dashboard.

5.2. Diseño del dashboard

Una vez definida la información del resumen semanal, se procedió a diseñar la interfaz donde se volcaría. Para ello, se llevó a cabo un proceso de diseño que consistió en la construcción de un bosquejo en Balsamiq (wireframe de baja fidelidad), el cual fue posteriormente implementado en html.

El propósito de realizar el bosquejo en Balsamiq fue para establecer, de manera preliminar, cómo se organizaría toda la información dentro del dashboard. Es decir, cómo y dónde se mostrarían las visualizaciones implementadas y cómo se mostraría la información en el resumen semanal principalmente.

Previo al diseño del bosquejo del dashboard, se consultaron distintos sitios web en búsqueda de ejemplos de dashboards. Esto permitió encontrar elementos comunes en sus diseños, y seleccionar algunas vistas que pudiesen servir de base para la construcción del dashboard de este trabajo de título.

En las Figuras 5.1, 5.2 y 5.3 se muestran algunas pantallas que sirvieron de inspiración para el proceso de diseño de este dashboard.

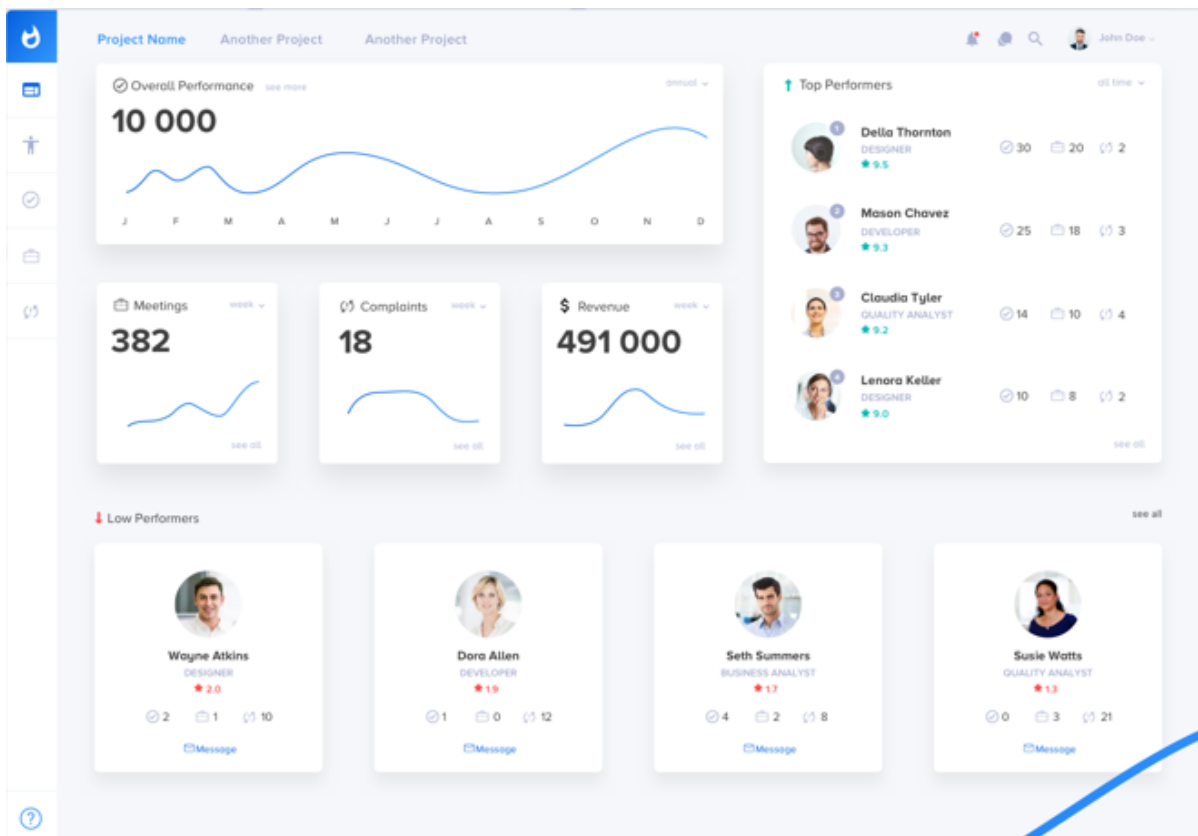


Figura 5.1: Interfaz de dashboard extraída de [23]

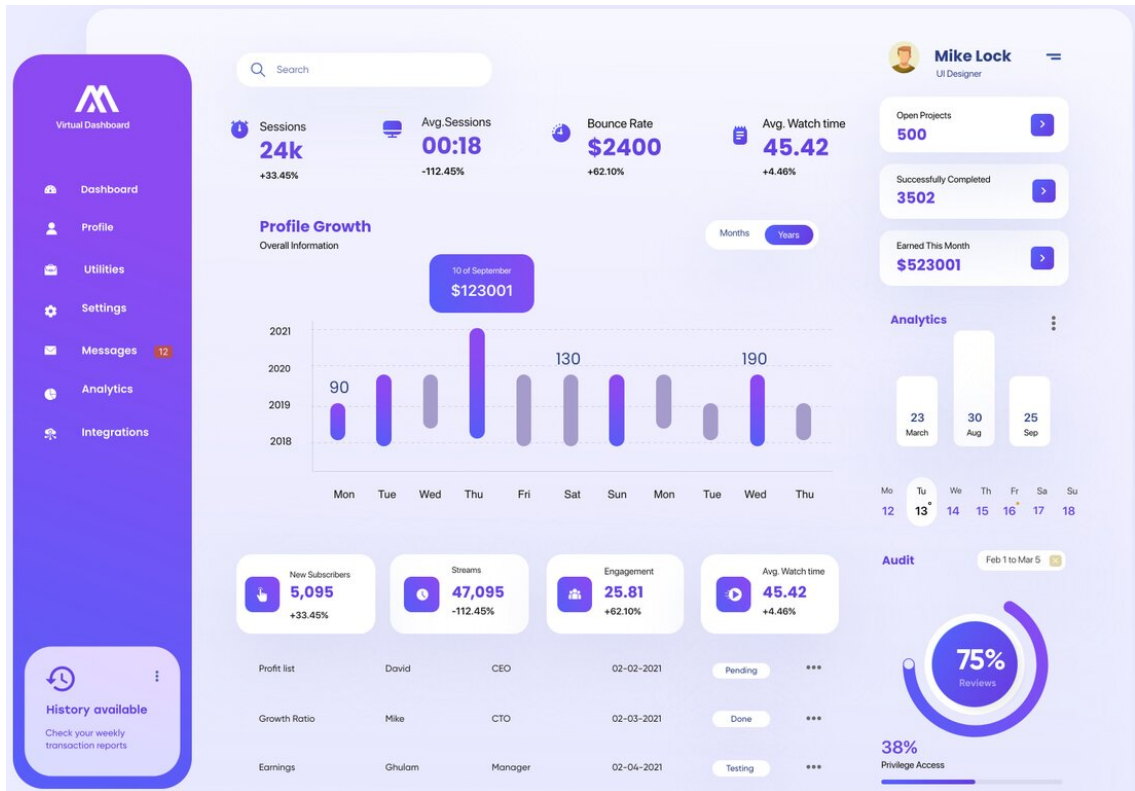


Figura 5.2: Interfaz de dashboard extraída de dribbble.com



Figura 5.3: Interfaz de dashboard extraída de behance.net

En base a estas propuestas se pudieron identificar algunos elementos comunes, como por ejemplo: el uso de un tamaño de letra grande para resaltar indicadores importantes, el uso de contrastes para resaltar la información importante (por ejemplo, poner indicadores en cuadrados con fondo blanco, mientras el fondo de la pantalla es gris), y en algunas ocasiones, el uso de bordes redondos para los contenedores de la información.

Con los elementos recientemente mencionados, el prototipo de dashboard diseñado puede ser visto en la Figura 5.4.

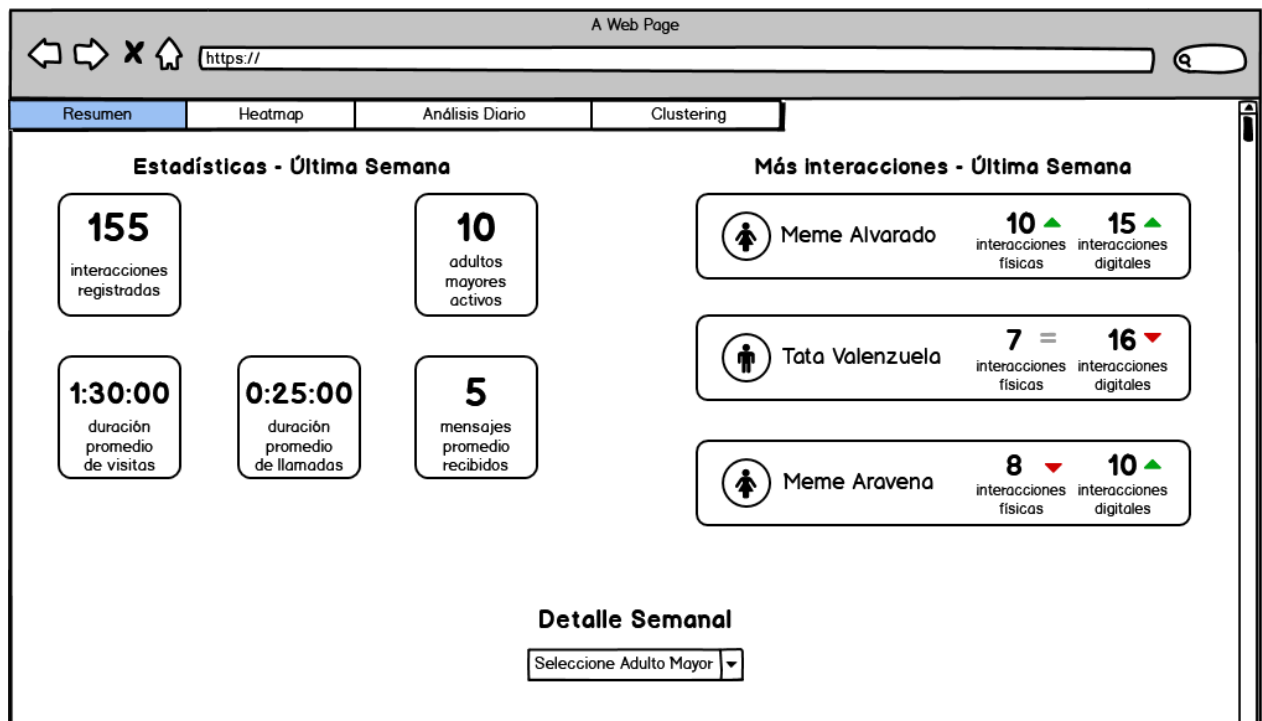


Figura 5.4: Vista del dashboard sin resumen de adulto mayor

En este prototipo se puede ver que las visualizaciones están organizadas en pestañas, donde la pestaña principal corresponde al resumen semanal del dashboard. La decisión de separar las visualizaciones en pestañas se tomó con el objetivo de no abrumar al usuario con muchas gráficas en una sola pantalla.

En la parte superior del resumen semanal se muestra un diagnóstico rápido del estado de la plataforma durante la última semana. Este resumen se divide en dos partes: la izquierda y la derecha.

A la izquierda se visualiza un resumen de todas las interacciones registradas por la plataforma (independiente del adulto mayor) durante la última semana, como por ejemplo, la cantidad total de interacciones registradas, el tiempo promedio de todas las llamadas registradas o la cantidad de adultos mayores activos entre otros. Por otro lado, la parte derecha presenta una gráfica de tipo ranking, donde se muestran los tres adultos mayores que más interacciones registraron durante la última semana. Además se muestra un indicador al lado de cada número, que representa la evolución de dicho número con respecto a la semana anterior (flecha verde si aumentó, roja si disminuyó, y signo igual si se mantuvieron los valores).

Además del resumen de la plataforma, existe la posibilidad de analizar en detalle la semana de un adulto mayor específico, que es lo que se muestra en la parte inferior de la imagen. Aquí aparece un *dropdown menu*, con todos los adultos mayores de la plataforma, donde la idea es que el usuario que utilice el dashboard pueda seleccionar el adulto mayor que le interesa analizar, para así poder generar dicho resumen (Figura 5.5). La inclusión del *dropdown* se consideró necesaria debido a que a priori no se sabe cuál puede ser el adulto mayor de interés para el investigador. Además, se optó por dejar el bloque inferior sin contenido mientras no haya un adulto mayor seleccionado en dicho *dropdown*, ya que la otra opción era mostrar las estadísticas de un determinado adulto mayor por defecto. Sin embargo, se estimó que esto podría tener el riesgo de confundir al usuario, el cual podría pensar que se trataba de estadísticas generales, por ejemplo.

Una vez terminado el diseño del *mockup* en Balsamiq se procedió a implementarlo en un archivo html (sin el procesamiento de datos), para poder visualizarlo en un formato más cercano a lo que sería el producto final y así poder evaluarlo de mejor manera con expertos de dominio. La implementación de la primera versión del dashboard se muestra en la Figura 5.6.

Con el bosquejo ya implementado en html, este fue revisado con expertos de dominio, donde se identificaron dos reparos. En primer lugar se señaló la poca utilidad del resumen correspondiente al diagnóstico de la plataforma (parte superior del resumen semanal). Se indicó que tener información agregada respecto de las interacciones de los adultos mayores no tenía mayor valor para realizar un análisis científico de la información, y tampoco lo tenía el generar un ranking con los adultos mayores con más interacciones.

Esta información carecía de utilidad, principalmente porque cada adulto mayor tiene su propio “estándar” (o punto de referencia) en cuanto a volúmenes de interacción aceptables para ellos. Dado eso, ni la agregación ni la comparación de los volúmenes de interacciones de distintos adultos mayores tenía sentido.

La segunda observación tenía relación con la distribución de la información del resumen semanal, en un sentido que, el hecho de que los indicadores estuvieran organizados en contenedores (`<divs>`) generaba una mayor dispersión de la información. Esto se debía a que los contenedores, al estar separados, generan una gran cantidad de “espacio muerto” o inutilizado dentro de la gráfica. Esto hace que el usuario que vea la gráfica pierda más tiempo ubicando donde están los indicadores de su interés, versus una gráfica que contenga la misma información en un estilo más compacto.

Para atender estas observaciones se optó por retirar el resumen referente al diagnóstico de la plataforma (parte superior del resumen semanal), dejando únicamente el resumen semanal por adulto mayor. También se optó por cambiar la distribución de los indicadores de una vista basada en contenedores a una basada en tablas html. Así, el diseño final del resumen semanal del dashboard quedó como se muestra en la Figura 5.7.

En el diseño final, además de incorporar información relativa a las interacciones del adulto mayor, se incorpora información relativa a su contexto; por ejemplo, temperatura promedio, mascotas activas, número de familiares registrados en la plataforma y eventos recientes. Esta información proviene en su mayoría de tablas añadidas durante la extensión al modelo de

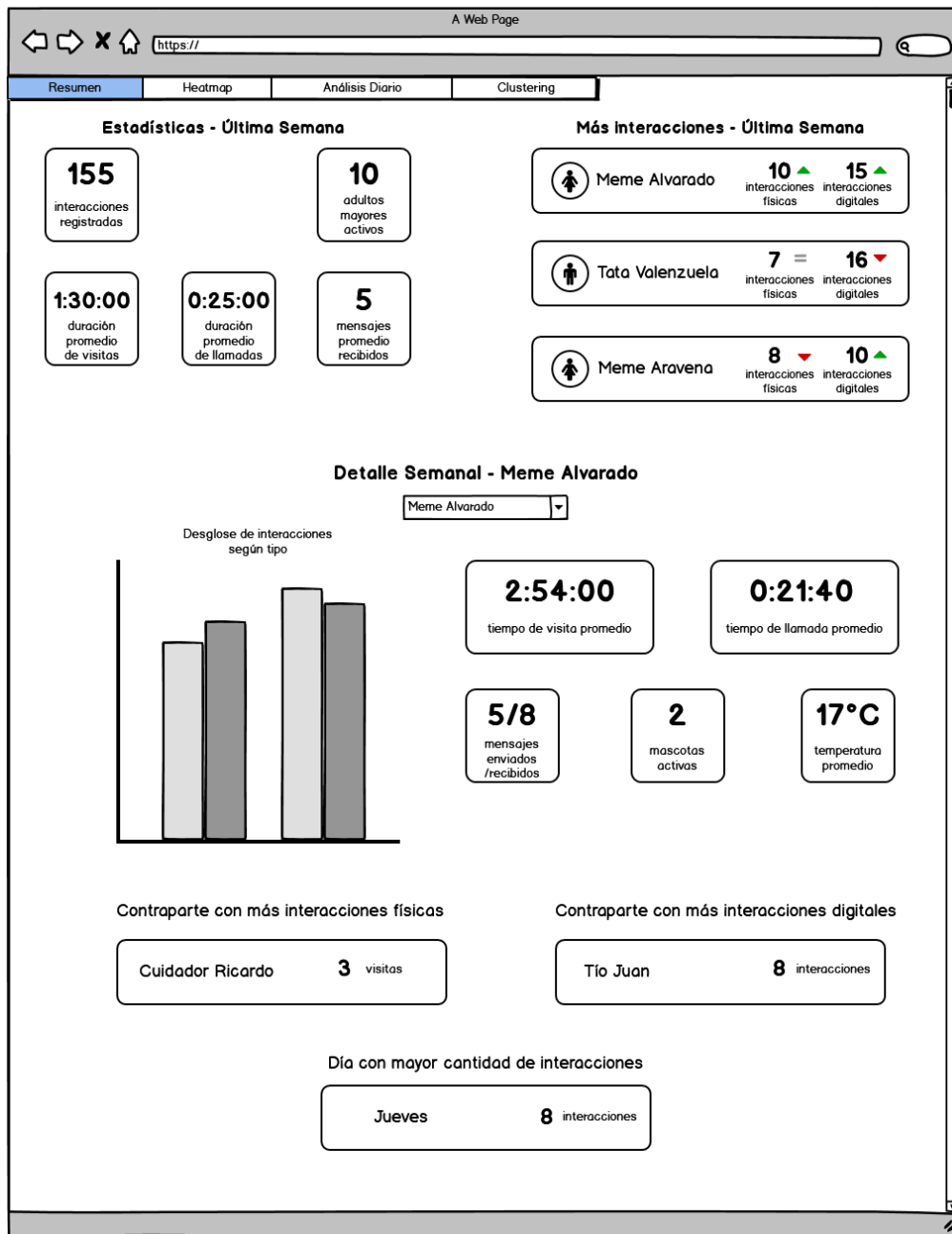


Figura 5.5: Vista del dashboard con resumen del adulto mayor

contexto.

5.3. Implementación del dashboard

Con el diseño ya corregido en base a las indicaciones realizadas por los expertos de dominio, la última etapa involucró implementar toda la lógica encargada del dashboard. Para ello, se definieron tres tareas claves: la implementación del sistema de datos que alimentaría el resumen semanal, la organización de las distintas visualizaciones en pestañas, y la unión del código JavaScript de las diferentes visualizaciones.

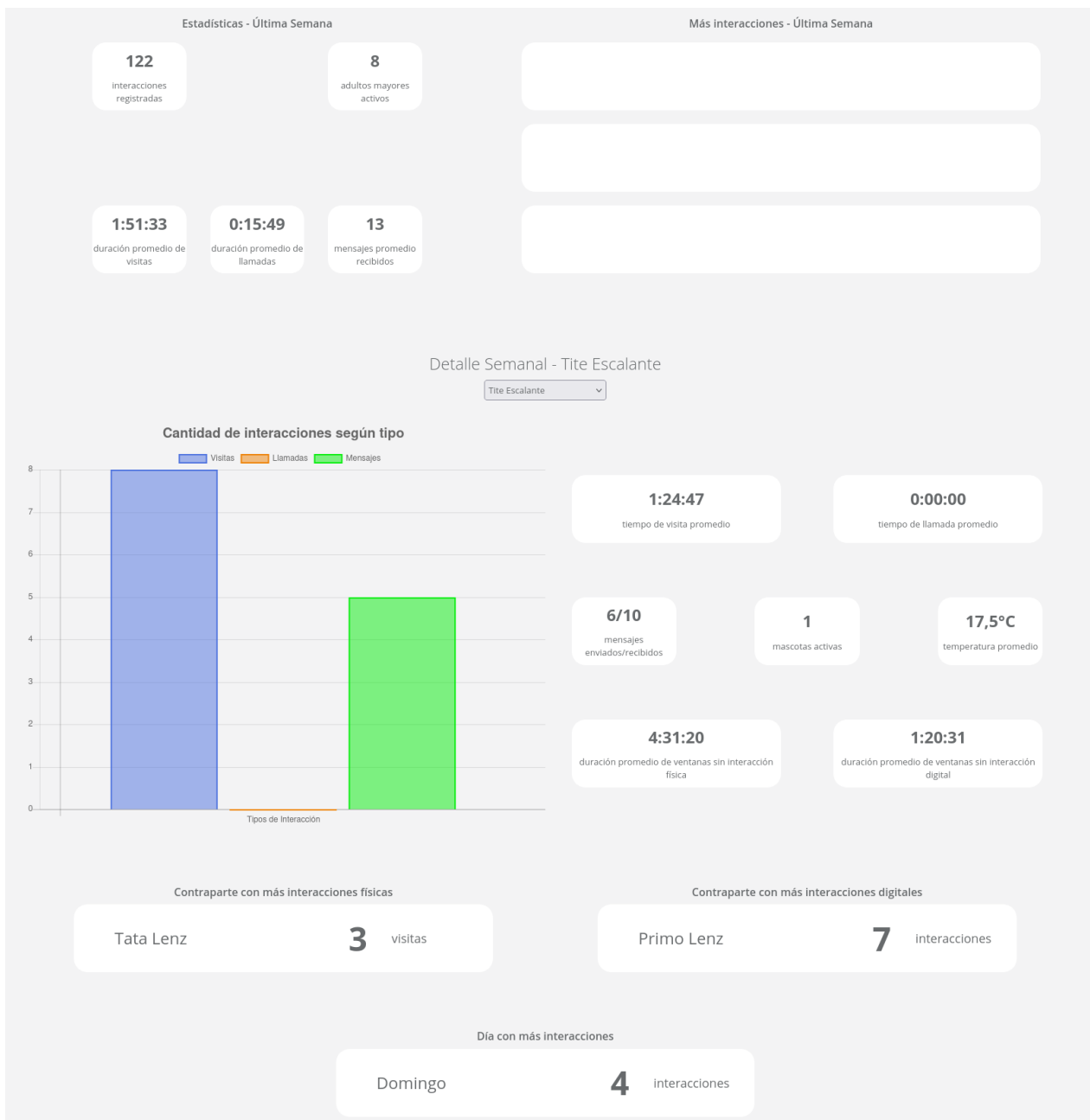


Figura 5.6: Implementación de la primera versión del dashboard en html

En primer lugar, se implementó el sistema de procesamiento de datos encargado de procesar la información para generar el resumen semanal. Este sistema ejecuta las siguientes tareas:

1. Se consultan por todas las interacciones, las ventanas sin interacción (física y digital) y los contextos diarios (clima) asociados al adulto mayor que se hayan registrado durante los últimos 7 días.
2. Las interacciones son procesadas para calcular los tiempos promedio de visita, tiempo promedio de llamada, cantidad de mensajes recibidos, cantidad de mensajes enviados (con sus respectivas las contrapartes) y el día con mayor cantidad de interacciones.

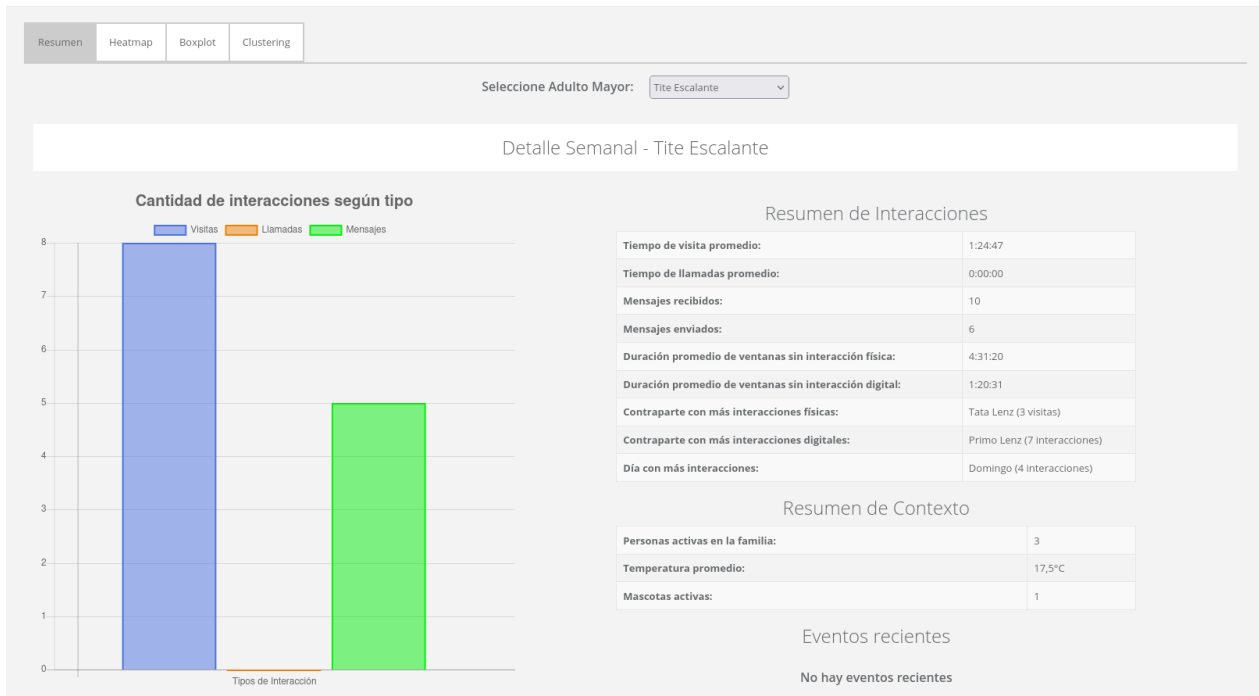


Figura 5.7: Versión final del dashboard en html

3. Se procesan las ventanas sin interacción y se calcula la duración promedio tanto de las ventanas sin interacción física, como de aquellas sin interacción digital.
4. Se consultan los eventos registrados del adulto mayor.
5. Se procesan los eventos del adulto mayor, verificando si alguno de ellos ocurrió en la última semana. También se verifica si se celebró un evento recurrente, por ejemplo, un aniversario.
6. Se procesan los datos del clima con el fin de calcular el promedio de temperatura de la semana.
7. Se consultan las mascotas activas del adulto mayor, para conocer cuántas hay activas.
8. Se consultan los miembros familiares activos del adulto mayor, para conocer cuántos hay activos.
9. Se guarda toda la información procesada en un archivo `json` y se envía como respuesta a la request.

Para organizar las distintas visualizaciones en pestañas se utilizaron botones para simular los tabs de las pestañas y se organizó el contenido de cada pestaña en un contenedor (`<div>`) para cada visualización. De esta forma, capturando el click en los botones (tabs) se puede controlar que contenido (pestaña) mostrar al usuario mediante funciones JavaScript (utilizando las funciones `.show()` y `.hide()` de JQuery), logrando así que el usuario vea la pantalla organizada en pestañas. En cuanto al código html de cada visualización, se aprovechó la funcionalidad `{% include %}` de Django. Esta funcionalidad permite incrustar el contenido de un archivo html dentro de otro archivo html. De esa manera, solo bastó realizar modificaciones mínimas a los archivos html de las distintas visualizaciones e incrustar su

contenido dentro del archivo final (Figura 5.7).

Por último, se unió el código JavaScript de las distintas visualizaciones implementadas (resumen semanal, heatmap, boxplot y clustering). Debido a que todas las visualizaciones fueron implementadas por separado, en un comienzo se debió hacer un refactoring del código JavaScript, ya que existían varias funcionalidades que eran compartidas entre algunas visualizaciones. Este refactoring fue necesario porque, pese a que las visualizaciones se encuentran organizadas en pestañas, esta vista sigue siendo una única pantalla (un solo archivo html). Por lo tanto, en algunas ocasiones además de código duplicado (derivado de las funcionalidades similares implementadas por separado), existían alcances de nombre con algunas funciones. Esto generaba comportamientos no esperados en las visualizaciones.

Capítulo 6

Validaciones

Una vez implementado el dashboard se procedió a validar la correctitud de los sistemas de procesamiento de datos implementados, tanto los que se encargan de generar los registros de las tablas del modelo de contexto como los que alimentan las visualizaciones y el resumen semanal. Para realizar estas validaciones, se utilizó una base de datos de prueba, la cual contenía 1.746 interacciones, repartidas entre cuatro adultos mayores y comprendidas en un período de tiempo de 19 semanas (entre 24/08/2020 y el 03/01/2021). Cada adulto mayor corresponde a un grupo familiar distinto.

Si bien las validaciones se realizaron para los cuatro grupos familiares presentes en la base de datos, este capítulo expondrá únicamente las validaciones realizadas sobre la familia del adulto mayor “Tite Escalante”, con la finalidad de poder explicar de una forma más ordenada las validaciones realizadas. Dicha familia posee los siguientes integrantes e interacciones con el adulto mayor:

- **Integrantes de la Familia:**
 - Tite Escalante (Adulto mayor)
 - Tomás Coordinador (Coordinador familiar)
 - Tata Lenz
 - Primo Lenz
- **Cantidad de interacciones registradas por este grupo familiar en el período 24/08/20 - 03/01/21: 216**

La idea de acotar las validaciones expuestas en este capítulo a este grupo familiar en particular, es para poder ejemplificar de mejor forma el procedimiento que se realizó para validar cada desarrollo. Además, se busca plantear los ejemplos siempre en torno a un mismo contexto para evitar confusiones en el lector. A continuación se presentan las validaciones realizadas para cada desarrollo implementado.

6.1. Extensión al modelo de contexto

Las validaciones realizadas para la extensión del modelo de contexto se pueden dividir en tres, una para cada *cron-job* implementado (explicados en la sección 3.4.2). A continuación se explica en detalle los procedimientos realizados para cada *cron-job*.

6.1.1. DailyContext

Para validar el sistema de procesamiento de datos encargado de generar los registros de la tabla `DailyContext` se diseñó una metodología simple, que consistió en ejecutar el *cron-job* encargado de realizar las consultas a las APIs y guardar los registros generados. Luego, observar los registros generados y verificar su correctitud, comparándolos contra la información climática disponible en internet.

Como este *cron-job* no depende de las interacciones registradas en la plataforma, sino que depende de los adultos mayores actualmente presentes en la plataforma (recordemos que genera un registro por cada ubicación distinta que detecte), se configuró el siguiente escenario para realizar la validación.

Tabla 6.1: Adultos mayores con sus respectivas ubicaciones, utilizados para la validación

Adulto Mayor	Ubicación
Meme Madriaza	Godoy Cruz, Mendoza, Argentina
Tata Rodríguez	Los Vilos, Los Vilos, Chile
Meme Alvarado	Las Condes, Santiago, Chile
Meme Aravena	Las Condes, Santiago, Chile
Meme Palma	La Florida, Santiago, Chile
Tata Pineda	La Florida, Santiago, Chile
Tata Estévez	Iquique, Iquique, Chile
Tite Escalante	Puerto Montt, Puerto Montt, Chile

Como se puede ver en la tabla, en este escenario hay ubicaciones repetidas, las cuales deberían ser manejadas por el *cron-job* para evitar generar registros con información redundante. Así, se esperaría que por cada ejecución del *cron-job* se generen únicamente seis registros, ya que del total de ocho ubicaciones, existen dos que se repiten. Además, podemos ver que hay una ubicación fuera de Chile, específicamente en Mendoza, Argentina. Esta ubicación se seteó para probar la capacidad del sistema de manejar casos que no correspondiesen a Chile, para asegurar que las APIs responderán de igual forma tanto en casos nacionales como en casos internacionales. Así, al ejecutar el *cron-job* se obtuvieron los registros que se muestran en la Tabla 6.2.

Estos resultados fueron obtenidos corriendo el *cron-job* el día 21/07/2021, por lo tanto estos se deben comparar contra los datos climáticos de dicha fecha. Para realizar esta comparación, se utilizaron datos recuperados desde AccuWeather¹. Estos datos relativos al clima fueron complementados con los datos de la página TimeAndDate² para verificar si el día ana-

¹AccuWeather HomePage

²TimeAndDate Holidays

Tabla 6.2: Registros generados por la ejecución del *cron-job*

Ubicación	T° Promedio	Lluvia	Estación	Tiempo de Luz [seg]	Feriado
Godoy Cruz	10,6°C	False	Invierno	37033	False
Los Vilos	13,5°C	False	Invierno	37243	False
Las Condes	17°C	False	Invierno	36905	False
La Florida	17°C	False	Invierno	36905	False
Iquique	17°C	False	Invierno	39875	False
Puerto Montt	10°C	False	Invierno	34596	False

lizado era feriado o no. Los datos extraídos desde ambas páginas para la fecha 21/07/2021 fueron los que se muestran en la Tabla 6.3.

Tabla 6.3: Datos climáticos y de feriados obtenidos desde los sitios AccuWeather y TimeAndDate

Ubicación	T° Promedio	Lluvia	Estación	Tiempo de Luz [seg]	Feriado
Godoy Cruz	11°C	No	Invierno	37080	No
Los Vilos	15°C	No	Invierno	37200	No
Las Condes	13°C	No	Invierno	36900	No
La Florida	13°C	No	Invierno	36900	No
Iquique	16°C	No	Invierno	39840	No
Puerto Montt	9,5°C	No	Invierno	34560	No

Comparando los valores reportados en ambas tablas podemos observar que existen pequeñas diferencias en cuanto a las temperaturas registradas. En la mayoría de los registros, la diferencia no es significativa. Sin embargo, en el caso de Las Condes y La Florida (ambas comunas de la Región Metropolitana), la diferencia alcanza 4°C entre los registros esperados y obtenidos. Esta diferencia puede deberse al hecho de que al consultar dos fuentes de información distintas (es decir, la API que consulta el *cron-job* y la página web utilizada para los resultados esperados), naturalmente pueden existir diferencias en los datos suministrados. De todas formas, es recomendable hacer un seguimiento de los datos aportados por la API para verificar su correctitud en el futuro. A pesar de ello, el resto de la información extraída desde la API es casi idéntica a la información esperada, e incluso se obtienen resultados muy similares para el caso del tiempo de luz de cada locación (diferencias de máximo 47 segundos).

6.1.2. InteractionWindows

En cuanto a las validaciones realizadas para la tabla `InteractionWindows`, se utilizó el contexto de la familia del adulto mayor “Tite Escalante” (enunciado al comienzo del capítulo). Como el procesamiento de datos de esta tabla está diseñado para correr una vez al día (al final de cada jornada), las condiciones bajo las cuales se efectuó el proceso de validación de esta parte del sistema fueron las siguientes:

- Se calcularon las ventanas de no interacción para el día 11/10/2020, el cual posee seis interacciones: dos físicas (visitas) y cuatro digitales (mensajes).
- Se marcó al “Primo Lenz” (miembro de la familia) como personal de apoyo, únicamente

con la finalidad de testear si el sistema puede distinguir entre interacciones con el personal de apoyo, de aquellas que se llevan a cabo con miembros de la familia.

El detalle de las interacciones registradas durante el día 11/10/2020 se muestra en la Tabla 6.4.

Tabla 6.4: Interacciones de Tite Escalante el día 11/10/2020

Contraparte	Tipo Interacción	Inicio	Fin	Duración
Primo Lenz	Física	10:28:00	12:57:00	8940
Tomás Coordinador	Digital	13:00:00	13:00:00	0
Tomás Coordinador	Digital	13:12:00	13:12:00	0
Tomás Coordinador	Digital	13:20:00	13:20:00	0
Tata Lenz	Física	15:32:00	18:14:00	9720
Tomás Coordinador	Digital	19:56:00	19:56:00	0

En base a las interacciones expuestas en la Tabla 6.4, se procedió a establecer los registros que debiera generar el sistema una vez ejecutado el *cron-job*. Estos registros son los mostrados en la Tabla 6.5.

Tabla 6.5: Registros esperados del procesamiento de las ventanas sin interacción

begin_time	end_time	duration	interaction_type	support_personal
8:00:00	10:28:00	8880	0	0
10:28:00	12:57:00	8940	0	1
12:57:00	15:32:00	9300	0	0
18:14:00	22:00:00	13560	0	0
8:00:00	13:00:00	18000	1	0
13:40:00	19:56:00	22560	1	0
20:16:00	22:00:00	6240	1	0

En estos registros esperados (Tabla 6.5) podemos observar que se generan siete registros, cuatro de los cuales corresponden a ventanas sin interacción física, y los otros tres corresponden a ventanas sin interacción digital. Existe una ventana sin interacción física que corresponde a la visita de “Primo Lenz” al adulto mayor. Este es un comportamiento esperado por el sistema, debido a que el “Primo Lenz” fue etiquetado como personal de apoyo, por lo cual su visita no se cuenta como una interrupción de la ventana sin interacción física, sino que se marca dicho período con el flag `with_support_personal`, tal como se señaló en la sección 3.3.5 `InteractionWindows`. Además, es necesario recordar que para evitar la generación de múltiples ventanas digitales de ínfima duración, el sistema establece un período de 20 minutos posterior a la última interacción digital registrada, para establecer el inicio de un período sin interacción digital. Esto se hace debido a la no continuidad de estas interacciones, y así evitar generar registros entre mensajes que estén separados por segundos o pocos minutos, por ejemplo.

Con el resultado esperado ya definido, se ejecutó el *cron-job* modificado para que considerara las interacciones del día 11/10/2020 y se consultó la base de datos para verificar los

resultados obtenidos. Los registros generados por el sistema pueden ser vistos en la Tabla 6.6.

Tabla 6.6: Registros obtenidos del procesamiento de las ventanas sin interacción

begin_time	end_time	duration	interaction_type	support_personal
8:00:00	10:28:00	8880	0	0
10:28:00	12:57:00	8940	0	1
12:57:00	15:32:00	9300	0	0
18:14:00	22:00:00	13560	0	0
8:00:00	13:00:00	18000	1	0
13:40:00	19:56:00	22560	1	0
20:16:00	22:00:00	6240	1	0

Con los resultados expuestos en la Tabla 6.6, podemos verificar que estos obedecen al comportamiento esperado señalado en la Tabla 6.5. Por lo cual la prueba realizada para esta parte del sistema se califica como exitosa.

6.1.3. WeeklyInteractions

Para validar la función a cargo de generar los registros de la tabla `WeeklyInteractions`, se utilizó el contexto correspondiente a la familia del adulto mayor “Tite Escalante”. Para esto, se configuró el *cron-job* para que corriera sobre la última semana del período en el que hay interacciones (esto es, del 28/12/20 al 03/01/21). En esta semana existen nueve interacciones dentro del grupo familiar estudiado, de las cuales cinco son visitas físicas y cuatro son mensajes (todos son mensajes entrantes). El detalle de las nueve interacciones registradas puede ser visto en la Tabla 6.7.

Tabla 6.7: Resumen de las interacciones registradas en la semana 28/12/20 - 03/01/21

Contraparte	Interacciones Físicas	Tiempo total	Interacciones Digitales Entrantes	Interacciones Digitales Salientes
Primo Lenz	2	4:04:11	2	0
Tata Lenz	2	2:47:26	2	0
Tomás Coordinador	1	1:01:01	0	0

Recordando lo visto en la sección 3.3.6, esta tabla genera un registro por cada par “contraparte - tipo de interacción” distinto. En este caso en específico, podemos ver que “Tomás Coordinador” no tiene interacciones de tipo digital registradas durante el período de estudio, por lo cual se esperaría que se generen solo cinco registros, y no seis. Así, los cinco registros esperados de la ejecución del *cron-job* se pueden ver en la Tabla 6.8.

Es importante recordar que los campos `interaction_index` y `total_time` solo aplican para registros de tipo digital y físico respectivamente, por ello algunos registros tienen un “No Aplica” (N/A) como parte de su resultado esperado. Así, una vez ejecutado el *cron-job*, se obtuvieron los registros que se muestran en la Tabla 6.9.

Tabla 6.8: Registros esperados de la ejecución del *cron-job*

Contraparte	interaction _count	interaction _type	interaction _index	total_time
Primo Lenz	2	0	N/A	14651
Primo Lenz	2	1	0	N/A
Tata Lenz	2	0	N/A	10046
Tata Lenz	2	1	0	N/A
Tomás Coordinador	1	0	N/A	3661

Tabla 6.9: Registros obtenidos de la ejecución del *cron-job*

Contraparte	interaction _count	interaction _type	interaction _index	total_time
Primo Lenz	2	0	N/A	14651
Primo Lenz	2	1	0	N/A
Tata Lenz	2	0	N/A	10046
Tata Lenz	2	1	0	N/A
Tomás Coordinador	1	0	N/A	3661

Los registros generados por el sistema (Tabla 6.9) coinciden perfectamente con los registros esperados (Tabla 6.8). Por lo cual se establece que esta sección del sistema se comporta de acuerdo a lo esperado.

6.2. Implementación de visualizaciones

Como se mencionó anteriormente, la validación de la implementación de las visualizaciones se enfocó en verificar la correctitud del procesamiento de los datos que las alimentan. Para esto, la metodología que se utilizó consistió en realizar el procesamiento de los datos en papel, definir un comportamiento esperado de cada visualización en base a dicho procesamiento manual, y compararlos con los resultados obtenidos en cada una. A continuación, se exponen las validaciones realizadas para cada visualización.

6.2.1. Heatmap

Para validar la implementación del heatmap, se definieron tres aspectos claves a evaluar para verificar si la visualización generada coincide con el comportamiento esperado o no. Estos aspectos fueron los siguientes:

- Que el color asignado a cada casilla del heatmap (en base a la escala de color) sea el correcto.
- Que la serie de tiempo que muestra el detalle de cada bloque horario contenga información correcta.
- Que el *tooltip* contenga información correcta.

Para evaluar el comportamiento de la visualización en estos tres aspectos, el primer paso

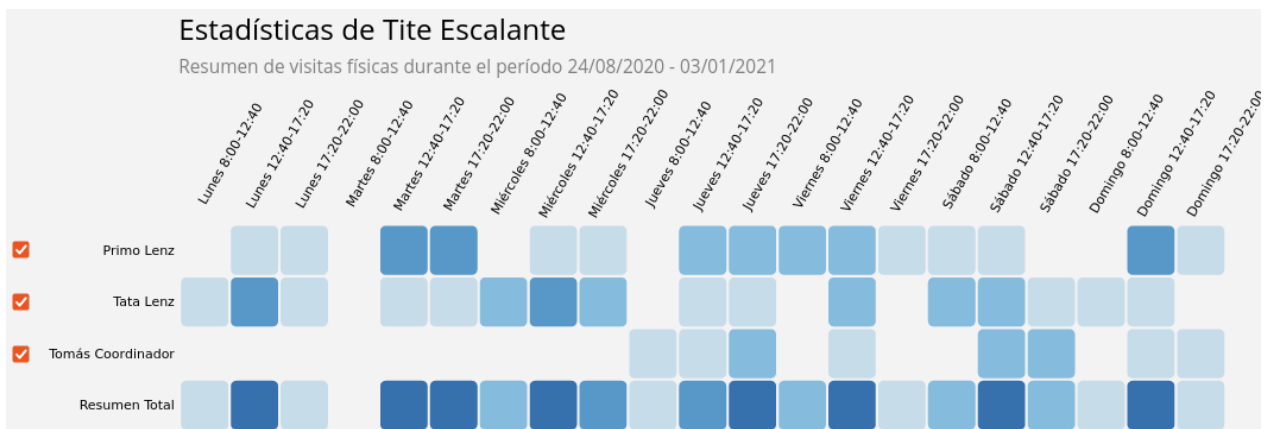


Figura 6.1: Heatmap generado para el caso de ejemplo

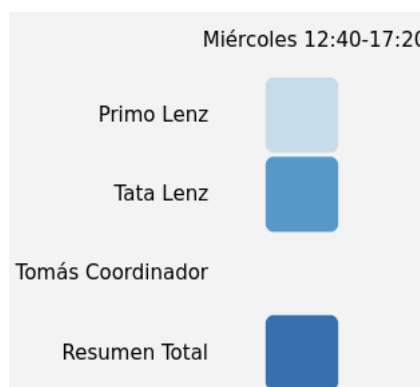


Figura 6.2: Detalle del bloque horario de interés

Por último, se verificó la correctitud de las series de tiempo mostradas al hacer click en un bloque del heatmap. Para ello, se comparó las series de tiempo mostradas por el sistema versus las series de tiempo planteadas en el comportamiento esperado. Los resultados obtenidos al hacer click en cada bloque se muestran en la Tabla 6.11.

Tabla 6.11: Resultados obtenidos al consultar las series de tiempo dentro del heatmap del escenario de ejemplo

Contraparte	Serie de tiempo
Primo Lenz	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.04, 0, 0.23]
Tata Lenz	[2, 1.57, 1.57, 1.57, 1.57, 1.57, 0, 1.57, 0, 0, 0.33, 0.4, 0.4, 0.4, 0.4, 0.4, 0, 0.4, 0]
Tomás Coordinador	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Resumen Total	[2, 1.57, 1.57, 1.57, 1.57, 1.57, 0, 1.57, 0, 0, 0.33, 0.4, 0.4, 0.4, 0.4, 0.4, 2.04, 0.4, 0.23]

Los resultados obtenidos (Tabla 6.11) coinciden con los enunciados en los resultados esperados (Tabla 6.10). Por lo cual se concluye que la información mostrada por el sistema a la hora de hacer click en el detalle de cada bloque horario es la correcta.

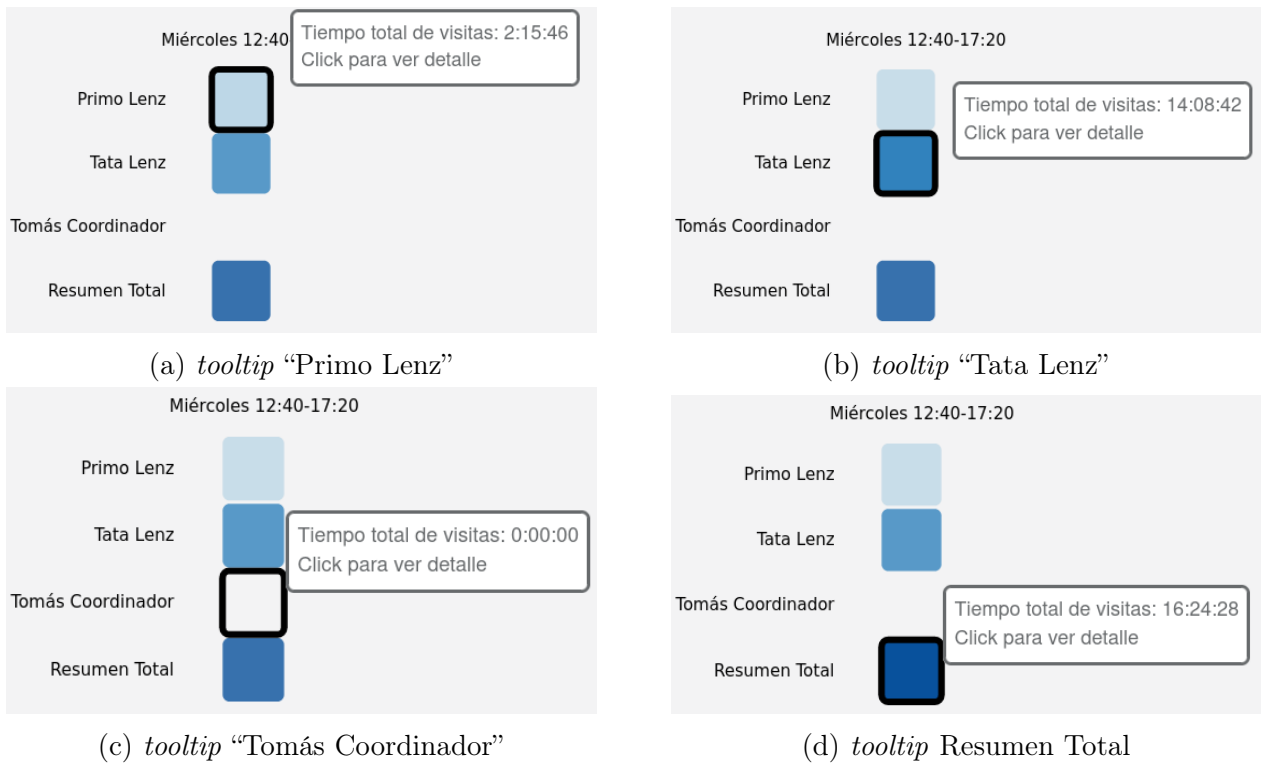


Figura 6.3: Información mostrada por el *tooltip* en el caso de ejemplo

6.2.2. Boxplot

Para validar el sistema de procesamiento de datos del boxplot se utilizó una metodología similar a la que se usó para validar el heatmap. Esta consiste en definir aspectos claves a evaluar. En este caso, se definió un único punto a evaluar, el cual consiste en verificar la correctitud del cálculo de las estadísticas que utiliza el boxplot. Debido a que el boxplot basa su comportamiento en los parámetros Q_1 , Mediana, Q_3 , min y max , para validar la implementación se calcularon dichos parámetros a mano y se contrastaron con lo obtenido al generar la visualización.

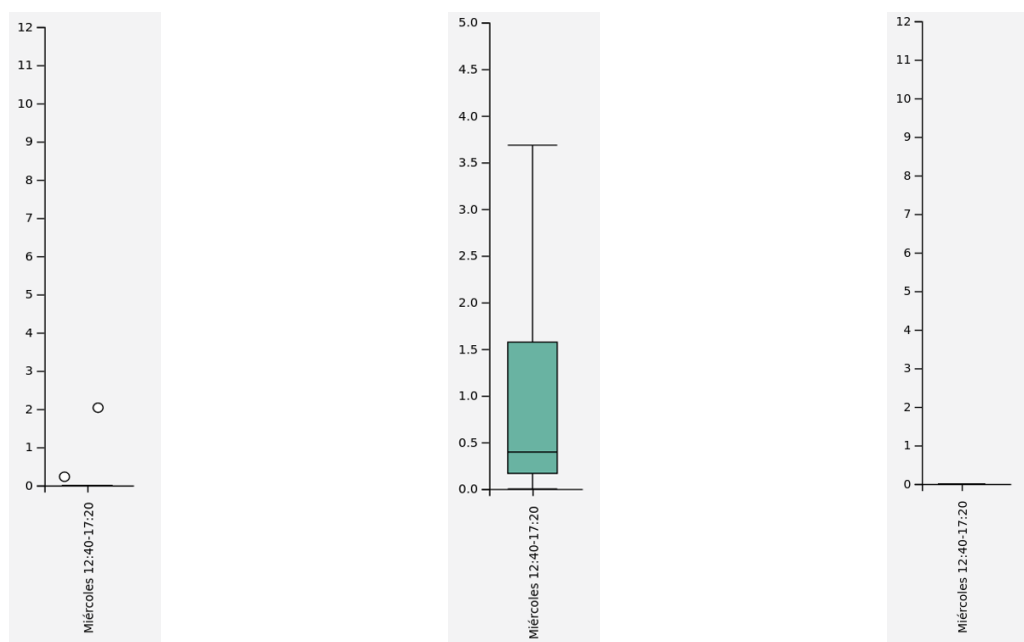
Para definir el comportamiento esperado de la gráfica, se calcularon los parámetros antes mencionados para cada miembro de la familia perteneciente al adulto mayor “Tite Escalante”. Ídem a lo realizado para el heatmap, se tomó el período de tiempo 24/08/21 - 03/01/21, se analizaron las interacciones físicas, y se tomó como ejemplo el bloque horario miércoles 12:40 - 17:20. Así, los resultados esperados son los que se muestran en la Tabla 6.12.

Tabla 6.12: Resultados esperados para el boxplot del escenario de ejemplo

Contraparte	Q_1	Mediana	Q_3	min	max
Primo Lenz	0	0	0	0	0
Tata Lenz	0.16	0.4	1.57	0	3.69
Tomás Coordinador	0	0	0	0	0

Con estos resultados definidos, se generó el boxplot para cada contraparte. En la Figura 6.4 es posible observar el resultado de estos. En las imágenes solo se incluye la sección

correspondiente al bloque horario miércoles 12:40 - 17:20, para facilitar la comprensión de las mismas.



(a) Boxplot de “Primo Lenz” (b) Boxplot de “Tata Lenz” (c) Boxplot de “Tomás Coordinador”

Figura 6.4: Gráficas de Boxplot, solo para el bloque horario Miércoles 12:40-17:20

Los valores en estas imágenes (Figura 6.4) coinciden con el comportamiento definido en la Tabla 6.12. Vemos que tanto el “Primo Lenz” (Figura 6.4 (a)) como “Tomás Coordinador” (Figura 6.4 (c)) tienen su correspondiente boxplot con todos sus parámetros en 0, mientras que en la gráfica generada para “Tata Lenz” (Figura 6.4 (b)) se puede apreciar que los parámetros de la caja dibujada están acorde a los resultados esperados. Es decir, su *min* en 0, su límite inferior está alrededor de 0.16, su mediana en torno a 0.4, su límite superior cerca del 1.57, y el *max* cercano a 3.69. Por lo tanto, se estima que el procesamiento de datos relativo a esta gráfica funciona acorde a lo esperado.

6.2.3. Clustering

En cuanto a las validaciones realizadas para la visualización de clustering, estas se ejecutaron solo a nivel de la generación de las series de tiempo para el clustering, debido a que tanto el algoritmo que calcula la matriz de distancia como el que realiza el proceso de clustering pertenecen a una biblioteca externa; por ende, su desarrollo es ajeno a este trabajo de título. Entonces, para esta validación se generaron las series de tiempo y se contrastaron con las obtenidas (entregadas por el sistema) al analizar las interacciones físicas entre la adulto mayor “Tite Escalante” y su contraparte “Tata Lenz” durante el período de tiempo 24/08/2020 - 03/01/2021. Se eligió a “Tata Lenz” debido a que es la contraparte con mayor cantidad de interacciones en el período de tiempo estudiado (54 interacciones).

Así, el primer paso fue generar las series de tiempo sobre las cuales se ejecuta el algoritmo de clustering. Como el período de tiempo a analizar consta de 19 semanas, entonces se deben

generar 19 series de tiempo (una para cada semana). Las series de tiempo esperadas son las que se muestran en la Tabla 6.13.

Tabla 6.13: Series de tiempo esperadas para cada semana

N° de Semana	Serie de Tiempo Esperada
Semana 1	[0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0]
Semana 2	[0, 1.5, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 3	[0, 1.5, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 4	[0, 1.5, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 5	[0, 1.5, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 6	[0, 1.5, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 7	[0, 0, 0, 0, 0.35, 1.13, 0, 0, 0, 0, 0, 1.76, 0, 0, 0, 0, 1.03, 0, 0, 0, 0]
Semana 8	[0.16, 1.33, 0, 0, 0, 0, 0, 1.57, 0.2, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 9	[0, 0, 0, 0, 1.5, 0, 0, 0, 0, 0, 1.76, 0, 0, 0, 0, 0.24, 0.79, 0, 0, 0, 0]
Semana 10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.33, 4.67, 0, 0, 0, 0]
Semana 11	[0, 0.33, 4.67, 0, 0, 0, 0, 0.33, 4.67, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 12	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 13	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 14	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 15	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 16	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.5, 0]
Semana 17	[0, 0, 0, 0, 1.76, 0, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 1.5, 0, 0, 0, 0, 0]
Semana 18	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0, 0]
Semana 19	[0, 0, 0, 0, 1.05, 0.71, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Con los resultados esperados ya planteados, se ejecutó el algoritmo que procesa los datos para el clustering. Para efectos de esta validación, el algoritmo se modificó para que mostrara las series de tiempo calculadas. Así, las series de tiempo obtenidas son la presentadas en la Tabla 6.14.

Comparando los resultados de ambas tablas es posible observar pequeñas diferencias. Por ejemplo, en las semanas 2, 3, 4, 5 y 6, la segunda casilla toma el valor 1.5 en los resultados esperados, mientras su valor es de 1.49 en los resultados obtenidos. Estas pequeñas diferencias se deben principalmente a aproximaciones realizadas al calcular las series de tiempo a mano, por lo cual no significan un comportamiento anormal del sistema. Por lo tanto, se concluye que la generación de las series de tiempo para el algoritmo de clustering se realiza de forma correcta por parte del sistema.

6.3. Resumen Semanal

La última parte del proceso de validación de los reportes implementados fue validar la información mostrada en el resumen semanal del dashboard. Para ello se utilizó el contexto del adulto mayor “Tite Escalante”, y se generó un resumen semanal para la semana del 28/12/20 al 03/01/21. Esto corresponde a la última semana en la cual existen interacciones en la base de datos.

Tabla 6.14: Series de tiempo obtenidas para cada semana

N° de Semana	Serie de Tiempo Obtenida
Semana 1	[0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0]
Semana 2	[0, 1.49, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 3	[0, 1.49, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 4	[0, 1.49, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 5	[0, 1.49, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 6	[0, 1.49, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 7	[0, 0, 0, 0, 0.35, 1.13, 0, 0, 0, 0, 0, 1.76, 0, 0, 0, 0, 1.03, 0, 0, 0, 0]
Semana 8	[0.16, 1.33, 0, 0, 0, 0, 0, 1.57, 0.19, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0]
Semana 9	[0, 0, 0, 0, 1.49, 0, 0, 0, 0, 0, 1.76, 0, 0, 0, 0, 0.24, 0.79, 0, 0, 0, 0]
Semana 10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.33, 4.67, 0, 0, 0, 0]
Semana 11	[0, 0.33, 4.67, 0, 0, 0, 0, 0.33, 4.67, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 12	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 13	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 14	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 15	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 16	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.49, 0]
Semana 17	[0, 0, 0, 0, 1.76, 0, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 1.49, 0, 0, 0, 0, 0]
Semana 18	[0, 1.76, 0, 0, 0, 0, 0.64, 0.4, 0, 0, 0, 0, 0, 0, 0.18, 1.31, 0, 0, 0, 0]
Semana 19	[0, 0, 0, 0, 1.05, 0.71, 0, 0, 0, 0, 1.03, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

El resumen de las interacciones registradas durante dicha semana fue expuesto en la Tabla 6.7. Sin embargo, en la Tabla 6.15 se da un detalle más fino de las interacciones transcurridas durante dicha semana.

Tabla 6.15: Detalle de las interacciones registradas durante la semana 28/12/20-03/01/21

Contraparte	Día	Tipo Interacción	Hora Inicio	Hora Fin
Primo Lenz	Lunes	Digital	14:45:35	14:45:35
Primo Lenz	Lunes	Física	17:06:12	19:08:16
Tata Lenz	Martes	Física	16:16:15	18:02:20
Primo Lenz	Miércoles	Física	17:06:21	19:08:28
Tata Lenz	Jueves	Física	14:05:33	15:07:24
Primo Lenz	Jueves	Digital	16:46:06	16:46:06
Tomás Coordinador	Viernes	Física	14:03:03	15:04:04
Tata Lenz	Viernes	Digital	16:02:04	16:02:04
Tata Lenz	Viernes	Digital	17:29:21	17:29:21

Con este detalle de las interacciones registradas se puede calcular los valores esperados para el resumen semanal de dicha semana, mostrados en la Tabla 6.16.

Una vez definido el comportamiento esperado, se consultó la pantalla correspondiente al resumen semanal de “Tite Escalante”, con la finalidad de comprobar cuáles son los datos entregados por el sistema, y si estos se coinciden con los esperados. Así, los resultados obtenidos pueden ser observados en la Figura 6.5.

Tabla 6.16: Valores esperados para el resumen semanal

Campo	Valor
Tiempo de visita promedio	1:34:32
Tiempo de llamadas promedio	0:00:00
Mensajes recibidos	4
Mensajes enviados	0
Duración promedio de ventanas sin interacción física	7:30:37
Duración promedio de ventanas sin interacción digital	8:47:16
Contraparte con más interacciones físicas	Primo Lenz (2 visitas)
Contraparte con más interacciones digitales	Primo Lenz (2 interacciones)
Día con más interacciones	Viernes (3 interacciones)
Cantidad de visitas	5
Cantidad de llamadas	0
Cantidad de mensajes	4

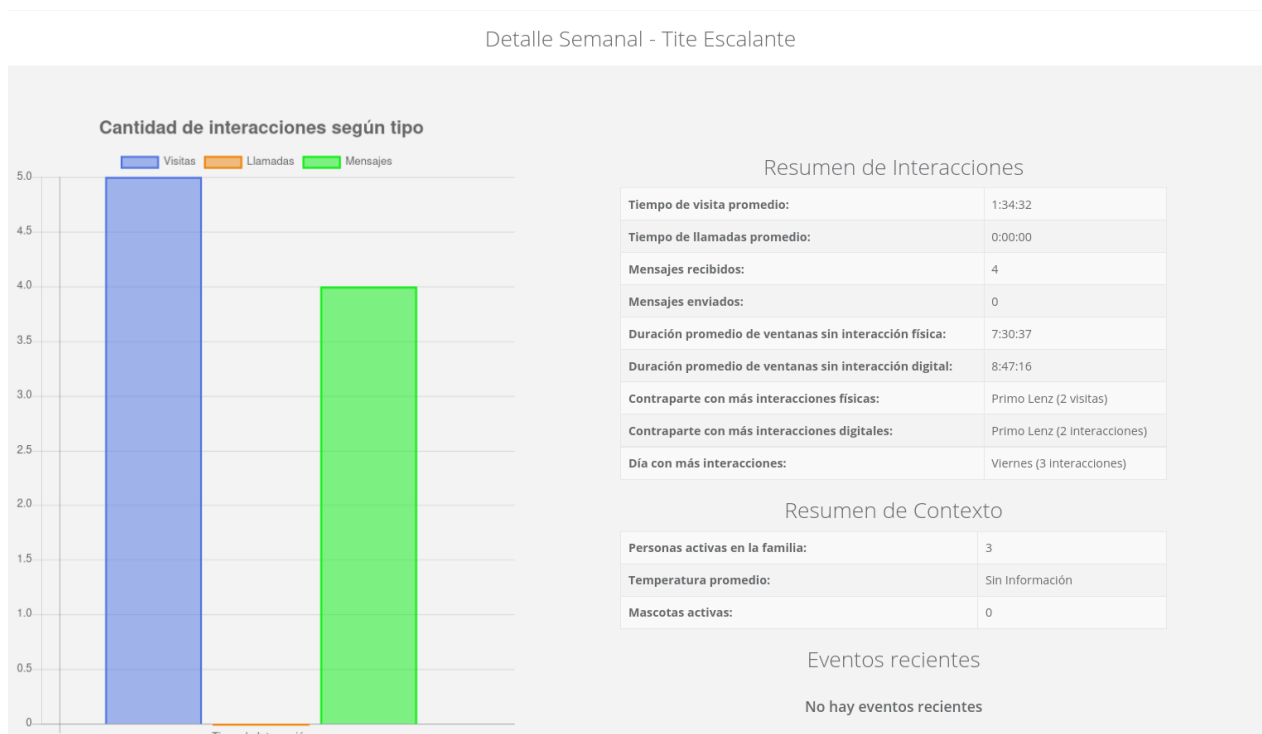


Figura 6.5: Resumen semanal generado para “Tite Escalante”

En dicha figura es posible ver que los datos mostrados por el sistema coinciden con los planteados en la tabla de valores esperados (Tabla 6.16), mostrando así la correctitud del sistema de procesamiento de datos encargado de generar el resumen semanal del dashboard.

6.4. Evaluación de utilidad

Una vez verificada la correctitud de los datos procesados por los distintos desarrollos implementados, se procedió a evaluar la utilidad de los mismos. Esta evaluación se realizó

con dos usuarios del sistema y tuvo por objetivo verificar el valor de la información exhibida por las mejoras implementadas durante este trabajo de título.

La evaluación de utilidad se realizó únicamente sobre los desarrollos correspondientes a la sección *dashboard* del sistema, donde se encuentran las gráficas implementadas (heatmap, boxplot y clustering) y el resumen semanal. La razón para enfocar la evaluación en esta parte del sistema (y no en las pantallas implementadas para la extensión del modelo de contexto, por ejemplo) es que estos elementos del sistema utilizan, procesan y muestran al usuario final información relativa a interacciones sociales de adultos mayores, que es justamente la información cuyo análisis se buscó apoyar con este trabajo de título.

6.4.1. Metodología de evaluación

Para realizar la evaluación de utilidad se reclutó a dos usuarios expertos en diseño de sistemas interactivos, los cuales realizan investigación en el dominio del cuidado informal de adultos mayores. La evaluación se llevó a cabo de manera remota mediante videoconferencia (vía Zoom) y de forma individual para cada evaluador. Con el respectivo permiso de cada evaluador, dichas videoconferencias fueron grabadas para poder ser analizadas posteriormente. Durante la evaluación de la plataforma, se le pidió a cada evaluador que compartiera su pantalla y se le asignaron tareas para realizar dentro del sistema. Las tareas definidas para la evaluación fueron las siguientes:

1. Ingresar al sitio con las credenciales asignadas.
2. Clickear la sección dashboard.
3. Generar el resumen semanal correspondiente a “Meme Alvarado”.
4. Identificar cuál fue su tipo de interacción más frecuente.
5. Identificar cuál es la contraparte con mayor cantidad de interacciones físicas, la contraparte con más interacciones digitales y el día con más interacciones.
6. Cambiar a la pestaña heatmap y generar una gráfica con los siguientes parámetros:
 - **Adulto Mayor:** “Meme Alvarado”
 - **Tipo de Interacción:** Física
 - **Fechas:** 24/08/2020 - 25/10/2020
7. Identificar los bloques que poseen un patrón de interacción fuerte (en el rango (75 %-100 %]).
8. Clickear el casillero correspondiente a la contraparte “Primo Valenzuela” en el bloque horario jueves 12:40-17:20.
9. Interpretar la gráfica mostrada por el sistema.
10. Cambiar a la pestaña boxplot y generar una gráfica con los siguientes parámetros:
 - **Adulto Mayor:** “Meme Alvarado”
 - **Contraparte:** Todos
 - **Tipo de Interacción:** Física
 - **Fechas:** 24/08/2020 - 25/10/2020
11. Identificar la cantidad de *outliers* en el bloque horario miércoles 12:40 - 17:20.
12. Clickear el *outlier* presente en dicho bloque.

13. Interpretar la información mostrada por el sistema.
14. Cambiar a la pestaña clustering y generar una gráfica con los siguientes parámetros:
 - **Adulto Mayor:** “Meme Alvarado”
 - **Contraparte:** Todos
 - **Tipo de Interacción:** Física
 - **Fechas:** 24/08/2020 - 25/10/2020
15. Clickear dos semanas correspondientes a un mismo cluster y comparar.
16. Clickear dos semanas correspondientes a clusters distintos y comparar.

Estas tareas se diseñaron con el objetivo de verificar si el sistema es capaz de responder posibles preguntas que puedan ser planteadas por sus usuarios. La evaluación se basó en el protocolo *thinking aloud* [22], por lo cual se le solicitó a los evaluadores que verbalizaran sus pensamientos mientras realizaban las tareas que se les iban asignando.

Para que los usuarios evaluadores pudieran acceder al sistema y realizar las tareas enunciadas anteriormente, se utilizó la herramienta `ngrok`³. Este software permite exponer un `localhost` mediante una url pública a través de túneles `http`, lo cual evita tener que realizar un *deploy* de la aplicación para que sea de público acceso, ahorrando así tiempo de trabajo.

6.4.2. Evaluador N°1

El evaluador N°1 realizó las tareas con el navegador web Google Chrome y en el sistema operativo Windows 10.

Las tareas en general fueron realizadas sin contratiempos. Sin embargo, hubo dos comentarios importantes a considerar durante la realización de las mismas. El primero surgió durante la tarea N°9, al pedirle al evaluador que interpretara la gráfica correspondiente al detalle del heatmap (Figura 4.9). Durante dicha tarea, el evaluador manifestó que “la gráfica no es fácil de interpretar así como así y los rótulos de los datos mapeados no ayudan mucho a su interpretación”. A pesar de ello, manifestó que los rótulos de los datos le parecían “técnicamente correctos” y que una vez que el usuario aprendiera el significado de la gráfica no le parecería un mayor problema.

La segunda observación que surgió, se dio durante la tarea N°12 (clickear un *outlier* en la gráfica de boxplot). Al clickear el *outlier*, el evaluador se preguntó si había ocurrido algo (debido a que no se percató de ningún cambio en la vista), ante lo cual se le indicó que hiciera scroll vertical para que pudiera ver la información desplegada por el sistema. Frente a esto, manifestó que “no hay nada que le diga al usuario que apareció más información en pantalla”.

A pesar de las dos observaciones realizadas por el usuario, este concluyó su evaluación mencionando que “la información exhibida por el sistema sin duda es útil”. Además, mencionó que a su juicio los desarrollos implementados representan “un avance en herramientas para interpretar los datos disponibles en la plataforma”, por lo cual cumplen su objetivo final.

³ngrok - Home Page

6.4.3. Evaluador N°2

El evaluador N°2 realizó la mayor parte de las tareas con el navegador Safari, mientras que las últimas 3 (tareas 14-16 correspondientes a la visualización de clustering) las realizó con el navegador Google Chrome. El sistema operativo utilizado por este evaluador fue macOS Big Sur.

El usuario completó las primeras 8 tareas sin mayores problemas. Sin embargo, al interpretar la gráfica correspondiente al detalle de un bloque horario del heatmap (Figura 4.9), el evaluador señala que “la curva que representa la media móvil no es evidentemente interpretable”. Respecto al heatmap, el evaluador también señaló que “el ritmo de visitas no es comparable a simple vista”, refiriéndose a que no es evidente diferenciar bloques horarios que, a pesar de que estén coloreados de igual color dentro de la gráfica, tengan distinto volumen de interacción (recordar que la escala seleccionada para el heatmap mapea la frecuencia de las interacciones).

Muy similar a la observación realizada por el evaluador N°1, el evaluador N°2 también mencionó que “la interacción al clicar un *outlier* en la gráfica de boxplot no es evidente”, debido a que el sistema muestra la información debajo de la gráfica y no implementa mecanismos para alertar al usuario de que dicha información ha sido desplegada. Además, comentó que podría ser de utilidad tener información con respecto al cuerpo de cada boxplot, es decir, que el sistema pudiera desplegar la información correspondiente a Q_1 , la mediana y Q_3 .

Con respecto a la gráfica de clustering, el evaluador señaló que esta le parecía de gran utilidad. Sin embargo comentó que “no es evidente qué representan los puntos en dicha gráfica”, haciendo alusión a que para alguien que no esté familiarizado con el sistema, no es intuitivo que los nodos de color naranja representen clusters y que los de color verde representen semanas dentro del período de estudio. Finalmente, a modo de comentario general, señaló que a pesar de que se pueden pulir algunas interacciones usuario-sistema, los desarrollos implementados le parecen un gran avance en materia de interpretar información relativa a las interacciones de los adultos mayores.

6.4.4. Discusión

Si bien ambos usuarios fueron capaces de completar todas las tareas asignadas sin mayores contratiempos, es evidente que ambos tuvieron reparos a la hora de ejecutar las tareas N°9 y N°12. Esto puede indicar que la información exhibida en la gráfica correspondiente al detalle de un bloque horario del heatmap (Figura 4.9) no es clara para usuarios que no están familiarizadas con la misma, y que la interacción del usuario con el sistema al clicar un *outlier* no es la óptima.

Con respecto a la observación del evaluador N°2, respecto a que “el ritmo de visitas no es comparable a simple vista”, es importante mencionar que el sistema implementa mecanismos para paliar esta deficiencia. Estos mecanismos corresponden al *tooltip* y a la gráfica de detalle de un bloque horario. Mediante estos dos elementos, el usuario tiene la posibilidad de visualizar el volumen de interacción entre las partes.

Es importante mencionar que los comentarios emitidos por ambos evaluadores no repre-

sentan fallas críticas del sistema, por lo que no evitan que se cumpla el objetivo de este trabajo de título. Sin embargo, dichos comentarios sí deben ser considerados como oportunidades de mejora del sistema, las cuales podrían ser eventualmente abordadas en un trabajo futuro.

Para terminar, es importante mencionar que ambos evaluadores valoraron el sistema implementado. Los dos evaluadores mencionaron que tanto las gráficas como el resumen semanal aportan información relevante para el usuario de la plataforma. Sin embargo, recalcaron que es necesario prestar atención al comportamiento del sistema una vez que este comience a operar en un ambiente de producción. Esto con la finalidad de estudiar posibles modificaciones o extensiones del mismo.

Capítulo 7

Conclusiones y trabajo futuro

Para un adulto mayor es muy importante mantener su independencia y autonomía a medida que envejece, ya que ello contribuye a mantener una buena salud mental y física. Este deseo de mantener su independencia se relaciona directamente con su voluntad de envejecer en casa, debido a que de esa forma pueden conservar en gran medida su independencia, a diferencia de escenarios donde el adulto mayor es trasladado a residencias para personas de la tercera edad o se muda a vivir con algún familiar cercano. Sin embargo, el hecho de envejecer en casa puede traer algunos peligros para el adulto mayor, como por ejemplo, la posibilidad de quedar socialmente aislado debido a que los miembros de su familia no pueden estar siempre a su disposición por si necesita algún tipo de ayuda. A esto se le suma el hecho de que los adultos mayores suelen tener problemas para relacionarse con la tecnología, lo cual puede dificultar su relación con otros miembros de la familia mediante redes sociales.

En esta línea, se hace fundamental que el adulto mayor tenga contacto frecuente con su círculo cercano (ya sea con familiares, amigos o vecinos). Esto ayuda a prevenir que la persona caiga en situaciones de aislamiento social. En este sentido, el sistema SocialConnector permite al usuario de la tercera edad interactuar de manera más fluida con su círculo cercano, agrupando los canales de comunicación digital más populares hoy en día en una sola aplicación. Además, permite al adulto mayor utilizar canales digitales de una forma más simple. También permite monitorear sus niveles de interacción social, gracias a que almacena las interacciones digitales que este tiene. Por otra parte, el sistema es también capaz de registrar las visitas físicas que recibe en su hogar (mediante el servicio Visitrack).

Durante este trabajo de título se desarrollaron herramientas para apoyar el análisis de la información de dichas interacciones, mejorando la herramienta actual en tres ámbitos principalmente: 1) la extensión del modelo de contexto de la aplicación, 2) la implementación de visualizaciones que permitieran explorar los datos en busca de información potencialmente relevante, y 3) la implementación de un dashboard que agrupa todas las visualizaciones implementadas. Además, este último genera un resumen de interacciones de la última semana para todos los adultos mayores registrados en la plataforma.

Mediante la extensión del modelo de contexto de la aplicación se le puede entregar a los investigadores, más y mejor información acerca del entorno del adulto mayor que está siendo

analizado. Además, se genera nueva información en base a las interacciones registradas por el sistema (ventanas de no interacción e interacciones semanales). Esta extensión brinda mayor contexto a los datos ya registrados por SocialConnector, entregando más herramientas a los investigadores para realizar estudios sobre los niveles de interacción social de los adultos mayores registrados en la plataforma.

Por otro lado, las visualizaciones implementadas son capaces de utilizar la información relativa a las interacciones y llevarla a contextos gráficos. A partir de ellas se puede generar conocimiento útil, como puede ser por ejemplo la búsqueda de patrones en las interacciones sociales de un adulto mayor. Además, es importante mencionar que el proceso de concepción de las visualizaciones implementadas se basó fuertemente en el trabajo colaborativo junto a expertos de dominio, en sentido que, la gran mayoría de las ideas de visualizaciones salieron de las reuniones semanales que tuvo el equipo de trabajo. Esto, además de dar la posibilidad de iterar los desarrollos realizados, permitió aumentar la utilidad de las mismas.

En cuanto al dashboard, este reúne en un solo lugar a todas las visualizaciones implementadas. Además, genera un resumen semanal para cada adulto mayor, lo cual puede darle al investigador una pincelada acerca del contexto social del sujeto en estudio.

El logro de estos tres puntos permite alcanzar el objetivo general definido en este trabajo de título. Como resultado se obtuvo una herramienta con mucho potencial para ser utilizada por investigadores de la materia, dándoles la posibilidad de disponer de mayor información de contexto para analizar la información almacenada en el sistema SocialConnector. La herramienta también permite analizar las interacciones de los adultos mayores desde la misma aplicación, evitando la descarga de dichos datos y la realización de un análisis manual de los mismos.

A pesar de que la correctitud de estos desarrollos haya sido validada con datos de prueba (no de producción), se obtuvieron resultados sumamente positivos de dicho proceso. En particular, no se encontraron indicios de que el sistema pueda cambiar su comportamiento frente a datos con otros valores, siempre y cuando los datos mantengan su estructura actual. En cuanto a la evaluación de utilidad realizada, los usuarios evaluadores valoraron positivamente los desarrollos implementados, y si bien se realizaron comentarios y se identificaron posibles puntos de mejora del sistema, el abordaje de estos no es crítico. Sin embargo, de ambas validaciones surge la inquietud acerca de como se comportarán los desarrollos una vez SocialConnector entre en régimen, por lo cual es recomendable monitorearlos una vez ocurra este hito.

Por otro lado, es interesante plantear algunos puntos de mejora del sistema. Uno de los principales es el uso de las APIs a la hora de procesar los datos necesarios para alimentar el modelo `DailyContext`. En este sentido, si en un futuro SocialConnector se monta en GoogleCloud por ejemplo, se podría utilizar directamente las APIs de Google para dicho fin, unificando así el origen de las APIs utilizadas, además de minimizar las dependencias del sistema (de tres proveedores de APIs, pasar a solo uno).

Como se mencionó anteriormente, la aplicación con la que se trabajó durante este trabajo de título no posee datos reales (de producción), por lo cual a pesar de que los desarrollos se comportaron de buena forma con los datos de prueba, puede que una vez que la aplicación

comience a operar con datos de producción haya que afinar algunos detalles. Uno de estos detalles, puede ser explorar otras formas de dividir el día (Figura 4.1), ya que no se tiene certeza de que los bloques actuales sean los que más se ajusten a los datos reales. Con miras a este posible ajuste, los bloques diarios están definidos como una variable y no “hardcodeados” a nivel del backend, para que su eventual modificación pueda realizarse sin la necesidad de hacer un análisis exhaustivo del código.

Otro punto que puede surgir del desarrollo en base a datos de prueba es la posible mejora de la escala de colores del heatmap. Es decir, actualmente no se tiene certeza de cómo se comportará esta escala con los datos de producción. Por ejemplo, puede ser que alguien nunca supere un cierto porcentaje de interacción, por lo que alguna parte de la escala definida podría llegar a no utilizarse. También podría pasar que haya una concentración de muchos usuarios en algún rango de color preestablecido, lo cual puede sugerir que se requiera refinar la escala de colores; es decir, agregarle “más pasos”.

Por otra parte, en un futuro puede ser interesante revisitar la gráfica de clustering implementada. Si bien esta gráfica cumple su objetivo y es funcional, aún hay detalles pendientes con respecto a su estética y a la capacidad del usuario para interactuar con ella. También, se puede experimentar variando el algoritmo con el cual se calcula la distancia entre las series de tiempo, como por ejemplo, la distancia euclidiana. Para realizar este cambio (en la medida de distancia), solo bastaría con reemplazar la matriz de distancia calculada.

Otro tema interesante de explorar puede ser expandir el análisis de datos a un nivel “más global”. El contar con un análisis “global” permitiría comparar los comportamientos de distintos adultos mayores. Es decir, comparar el comportamiento de distintas familias, a diferencia de las herramientas de análisis implementadas en esta memoria, las cuales apuntan a apoyar un análisis más local (nivel familia) de las interacciones. Un ejemplo de esto, puede ser el dar la posibilidad a los investigadores de realizar clustering sobre los adultos mayores, es decir, poder agrupar adultos mayores que posean patrones de interacción similares.

Para finalizar, y a modo de conclusión general, cabe mencionar que a pesar de que la expectativa de vida ha ido creciendo y la calidad de vida de las personas mayores mejorando, envejecer igual puede traer sus problemas. Algunos de estos problemas son intrínsecos al proceso de envejecimiento, por ejemplo, debido al desgaste del cuerpo y de sus diversos componentes. Otros problemas son producto del contexto social en el que se da este envejecimiento (bajas pensiones [41], discriminación por la edad [43], o aislamiento social como se ha visto durante este trabajo). Mientras la medicina y la ciencia pueden encargarse de alguno de estos problemas, es tarea de todos trabajar para poder paliar lo respectivo al ámbito social. En ese escenario, sistemas como SocialConnector pueden ser muy útiles para ayudar a cuidar la salud social de las personas de la tercera edad, sin necesariamente reducir su sensación de independencia.

Bibliografía

- [1] Jorge Aliaga Ana Callejas. *Así afecta la soledad y ansiedad a los adultos mayores*, 2020 (último acceso: 19 de julio, 2021). <https://www.uc.cl/noticias/asi-afecta-la-soledad-y-ansiedad-a-los-adultos-mayores/>.
- [2] AskPython. *Hierarchical Clustering with Python*, (último acceso: 22 de agosto, 2021). <https://www.askpython.com/python/examples/hierarchical-clustering>.
- [3] Australian Government - Department of Health. *Purposeful activity - hobbies*, 2019 (último acceso: 29 de diciembre, 2020). <https://headtohealth.gov.au/meaningful-life/purposeful-activity/hobbies>.
- [4] Tim Broady, Amy Chan, and Peter Caputi. Comparison of older and younger adults' attitudes towards and abilities with computers: Implications for training and learning. *British Journal of Educational Technology*, 41(3):473–485, 2010.
- [5] Kate Brush. *Data Visualization Definition*, 2020 (último acceso: 24 de julio, 2021). <https://searchbusinessanalytics.techtarget.com/definition/data-visualization>.
- [6] Homewatch CareGivers. *Living an Independent Life without Isolation*, (último acceso: 23 de julio, 2021). <https://www.homewatchcaregivers.com/blog/aging/living-an-independent-life-without-isolation/>.
- [7] Centers for Disease Control and Prevention. *Loneliness and Social Isolation Linked to Serious Health Conditions*, 2020. <https://www.cdc.gov/aging/publications/features/lonely-older-adults.html>.
- [8] Kendra Cherry. *The Color Psychology of Red*, 2020 (último acceso: 5 de julio, 2021). <https://www.verywellmind.com/the-color-psychology-of-red-2795821>.
- [9] Kendra Cherry. *The Color Psychology of Green*, 2021 (último acceso: 5 de julio, 2021). <https://www.verywellmind.com/color-psychology-green-2795817>.
- [10] Frans Coenen. Data mining: Past, present and future. *Knowledge Eng. Review*, 26:25–29, 03 2011.
- [11] Maria Cohut. *How is education linked to Alzheimer's progress?*, 2019 (último acceso: 25 de julio, 2021). <https://www.medicalnewstoday.com/articles/325452>.

- [12] Connect Health&Community. *Purposeful activity - hobbies*, último acceso: 29 de diciembre, 2020. <https://connecthealth.org.au/enews/pursuing-a-hobby-can-improve-your-mental-health/>.
- [13] Sara J. Czaja, Neil Charness, Arthur D. Fisk, Christopher Hertzog, Sankaran N. Nair, Wendy A. Rogers, and Joseph Sharit. Factors predicting the use of technology: Findings from the center for research and education on aging and technology enhancement (create). *Psychology and Aging*, 21(2):333–352, 2006.
- [14] Evolytics. *Top 5 Reasons Your Company Needs Tableau*, (último acceso: 24 de julio, 2021). <https://evolytics.com/blog/5-reasons-your-company-needs-tableau/>.
- [15] Alessandro Gabbiadini, Cristina Baldissarri, Federica Durante, Roberta Valtorta, Maria De Rosa, and Marcello Gallucci. Together apart: The mitigating role of digital communication technologies on negative affect during the covid-19 outbreak in italy. *Frontiers in Psychology*, 11, 10 2020.
- [16] Alonso Gaete, Francisco J Gutierrez, Sergio F Ochoa, Pablo Guerrero, and André Wyzykowski. Visitrack: A pervasive service for monitoring the social activity of older adults living at home. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 520–530. Springer, 2017.
- [17] Michael Galarnyk. *Understanding Boxplots*, 2018 (último acceso: 5 de julio, 2021). <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51>.
- [18] Francisco J. Gutierrez and Sergio F. Ochoa. It takes at least two to tango: Understanding the cooperative nature of elderly caregiving in latin america. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, page 1618–1630. Association for Computing Machinery, 2017.
- [19] Francisco J Gutierrez, Sergio F Ochoa, and Julita Vassileva. Mediating asymmetries in family communication: Supporting the inclusion of older adults. In *Universal Access in Human-Computer Interaction. Access to Today’s Technologies. UAHCI 2015.*, Lecture Notes in Computer Science. Springer, Cham, 2015.
- [20] Alexis Hope, Ted Schwaba, and Anne Marie Piper. Understanding digital and material social communications for older adults. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 3903–3912. Association for Computing Machinery, 2014.
- [21] Alboukadel Kassambara. *Agglomerative Hierarchical Clustering*, 2019 (último acceso: 27 de junio, 2021). <https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/#linkage>.
- [22] C. Lewis. *Using the “thinking Aloud” Method in Cognitive Interface Design*. Research report. IBM T.J. Watson Research Center.
- [23] Sara López. *Cómo diseñar dashboards*, 2020 (último acceso: 3 de julio, 2021). <https://www.hiberus.com/crecemos-contigo/como-disenar-dashboards/>.

- [24] Alan March. *La digitalización de la comunicación humana: alteraciones y cambios en la percepción*, 2012 (último acceso: 25 de julio, 2021). https://fido.palermo.edu/servicios_dyc/publicacionesdc/vista/detalle_articulo.php?id_libro=416&id_articulo=8721.
- [25] Saul McLeod. *What does a box plot tell you?*, 2019 (último acceso: 9 de julio, 2021). <https://www.simplypsychology.org/boxplots.html>.
- [26] Mental Health Foundation. *Pets and mental health*, último acceso: 29 de diciembre, 2020. <https://www.mentalhealth.org.uk/a-to-z/p/pets-and-mental-health>.
- [27] Amy Morin. *The Color Psychology of Blue*, 2020 (último acceso: 5 de julio, 2021). <https://www.verywellmind.com/the-color-psychology-of-blue-2795815>.
- [28] Diego Munoz, Francisco J Gutierrez, Sergio F Ochoa, and Nelson Baloian. Socialconnector: a ubiquitous system to ease the social interaction among family community members. *Comput. Syst. Sci. Eng*, 30(1):57–68, 2015.
- [29] Emiliana Pariente. *Brecha digital y tercera edad, ¿se están quedando fuera?*, 2020 (último acceso: 23 de julio, 2021). <https://www.latercera.com/paula/brecha-digital-y-tercera-edad-se-estan-quedando-fuera/>.
- [30] Alison Pearce. *Learning rewires the brain*, 2014 (último acceso: 25 de julio, 2021). <https://www.sciencenewsforstudents.org/article/learning-rewires-brain>.
- [31] Psychreg. *Why Is Having a Hobby Beneficial for Your Mental Health*, 2020 (último acceso: 29 de diciembre, 2020). <https://www.psychreg.org/hobby-mental-health/>.
- [32] Chotirat Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. 04 2004.
- [33] Felipe Rodriguez, Sergio F Ochoa, and Francisco J Gutierrez. Supporting asymmetric interaction in the age of social media. *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [34] Tea Romih. *Humans Are Visual Learners*, 2016 (último acceso: 25 de junio, 2021). <https://www.seyens.com/humans-are-visual-creatures/>.
- [35] Gonzalo Ríos. *Series de Tiempo*, 2008 (último acceso: 25 de julio, 2021). https://www.u-cursos.cl/ingenieria/2010/1/CC52A/1/material_docente/bajar?id_material=296003.
- [36] Alexander Seifert, Shelia Cotten, and Bo Xie. A double burden of exclusion? digital and social exclusion of older adults in times of covid-19. *The journals of gerontology. Series B, Psychological sciences and social sciences*, 07 2020.
- [37] Patrick J. Skerrett. *Pets can help their humans create friendships, find social support*, 2015 (último acceso: 29 de diciembre, 2020). <https://www.health.harvard.edu/blog/pets-can-help-their-humans-create-friendships-find-social-support-201505067981>.

- [38] Matthew Solan. *The introvert's guide to social engagement*, 2018 (último acceso: 25 de julio, 2021). <https://www.health.harvard.edu/blog/the-introverts-guide-to-social-engagement-2018111415353>.
- [39] Joe Tomaka, Sharon Thompson, and Rebecca Palacios. The relation of social isolation, loneliness, and social support to disease outcomes among the elderly. *Journal of aging and health*, 18:359–84, 07 2006.
- [40] Sebastián Toro. Modelo de captura automática de las interacciones sociales en una red familiar. *Memoria de Ingeniería Civil en Computación. DCC, FCFM, Universidad de Chile*, 2020.
- [41] TVN. *La gran lucha por sobrevivir a las bajas pensiones en Chile*, 2016 (último acceso: 29 de diciembre, 2020). <https://www.24horas.cl/economia/pensionesenchile/la-gran-lucha-por-sobrevivir-a-las-bajas-pensiones-en-chile-2088866>.
- [42] Alexandra Twin. *Data Mining*, 2020 (último acceso: 24 de julio, 2021). <https://www.investopedia.com/terms/d/datamining.asp>.
- [43] Age UK. *What is ageism?*, 2017 (último acceso: 5 de julio, 2021). <https://www.ageuk.org.uk/information-advice/work-learning/discrimination-rights/ageism/>.
- [44] Walden University. *Aging Well: Helping the Elderly Maintain Independence*, (último acceso: 19 de julio, 2021). <https://www.waldenu.edu/online-doctoral-programs/phd-in-human-services/resource/helping-the-elderly-maintain-independence>.
- [45] Eduardo Valenzuela. *Uso de fármacos en el adulto mayor*, último acceso: 29 de diciembre, 2020. <https://www.universidadviu.com/salud-social-definicion-determinantes-y-recomendaciones/>.
- [46] VIU. *Salud social: definición, determinantes y recomendaciones*, 2018 (último acceso: 26 de octubre, 2020). <https://www.universidadviu.com/salud-social-definicion-determinantes-y-recomendaciones/>.
- [47] Petr Volny, David Novak, and Pavel Zezula. Employing subsequence matching in audio data processing. 04 2012.
- [48] Branka Vuleta. *How Much Data Is Created Every Day?*, 2021 (último acceso: 25 de junio, 2021). <https://seedscientific.com/how-much-data-is-created-every-day/>.
- [49] Janine Wiles, Annette Leibing, Nancy Guberman, Jeanne Reeve, and Ruth Allen. The meaning of “aging in place” to older people. *The Gerontologist*, 52:357–66, 10 2011.

Anexos

A. Roles dentro de una familia

Dentro del sistema, los roles de los familiares están definidos en base a lo postulado por Gutiérrez et al. en [18]. Dichos roles se definen como sigue:

- **Asistente:** Esta es la persona que usualmente vive con el adulto mayor, por lo que es quien se encarga usualmente de proveer asistencia y cuidados inmediatos a éste. Este rol usualmente lo cumple la pareja del adulto mayor.
- **Monitor:** Esta persona no vive con el adulto mayor, pero está muy involucrada en el cuidado de éste. Los monitores se encargan de darle apoyo al adulto mayor, tanto en forma presencial como a distancia; por ejemplo, comprándole mercadería o medicamentos, acompañándolo a las consultas médicas y llamándolo por teléfono para ver cómo se siente (contención afectiva), entre otros. Este rol usualmente lo cumplen los hijos del adulto mayor.
- **Ayudante:** Esta persona apoya, de manera esporádica y puntual, el proceso de cuidado del adulto mayor. Los ayudantes son usualmente vecinos, amigos o nietos del adulto mayor, que pueden ayudarlo en alguna tarea específica, pero que no tienen un compromiso de cuidado permanente de esta persona o responsabilidad sobre ello.
- **Auto-excluido:** Corresponde a las personas que declaran no querer o poder asumir tareas en el cuidado del adulto mayor. Éstas suelen ser personas que se encuentran distanciadas afectivamente con el adulto mayor.

B. Criterios de unión de clusters

Existen varios criterios para la unión de clusters. En la Tabla 1 se presentan algunos de los métodos más usados, junto con una breve descripción y la fórmula de cada uno.

Tabla 1: Algunos de los métodos más utilizados unir clusters con algoritmos de clustering jerárquico [21]

Criterio	Descripción	Fórmula
Máximo o enlazamiento completo	La distancia entre dos clusters es definida como el máximo valor de todos los pares de distancia entre los elementos del cluster A y los elementos del cluster B. Tiende a producir clusters más compactos.	$\max\{d(a, b) : a \in A, b \in B\}$
Mínimo o enlazamiento simple	La distancia entre dos clusters es definida como el mínimo valor de todos los pares de distancia entre los elementos del cluster A y los elementos del cluster B. Tiende a producir clusters más largos o “menos densos”.	$\min\{d(a, b) : a \in A, b \in B\}$
Promedio o enlazamiento promedio	La distancia entre dos clusters es definida como la distancia promedio entre los elementos del cluster A y los elementos del cluster B.	$\frac{1}{ A B } \sum_{a \in A} \sum_{b \in B} d(a, b)$
Enlazamiento de centroides	La distancia entre dos clusters (A y B) es definida como la distancia entre sus centroides	$d(c_A, c_B)$

En cada fórmula, la función $d(x, y)$ representa la distancia entre dos puntos (generalmente euclídeana). Al final de cada iteración el algoritmo de clustering unirá aquellos clusters que tengan el mínimo valor respecto al criterio que se elija.

C. Código modelo de contexto

En este anexo, se exhibe el código modificado para la extensión del modelo de contexto. Se presentan las clases correspondientes a modelos de Django para las tablas UserProfile, Events, DailyContext, InteractionWindows, WeeklyInteractions y Pets.

C.1. UserProfile

```
1 class UserProfile(models.Model, ModelDiffMixin):
2     user = models.OneToOneField(User, on_delete=models.CASCADE,
3         related_name="profile")
4     first_name = models.CharField("Nombres", max_length=120, null=True,
5         blank=False)
6     last_name = models.CharField("Apellidos", max_length=120, null=True,
7         blank=False)
8     avatar = models.ImageField(upload_to='avatar/', null=True, blank=True)
9     gender = models.CharField("Genero", choices=GENDER, max_length=7,
10        null=True, blank=False)
11    birthday = BirthdayField('Cumpleaños', null=True, blank=True,
12        max_length=5)
13    phone = PhoneNumberField("Telefono", null=True, blank=False)
14    phone2 = PhoneNumberField("Telefono 2", null=True, blank=True,
15        default="")
16    address = models.TextField("Direccion", blank=True, null=True)
17    town = models.CharField("Comuna", blank=False, null=True,
18        max_length=50)
19    city = models.CharField("Ciudad", blank=False, null=True,
20        max_length=50)
21    country = models.CharField("Pais", blank=False, null=True,
22        max_length=50)
23    is_active = models.NullBooleanField("Activo?", default=True)
24    is_admin = models.NullBooleanField("Es Administrador?", default=False)
25    is_coordinator = models.NullBooleanField("Es coordinador?",
26        default=False)
27    is_reachable = models.NullBooleanField("Esta disponible?",
28        default=True)
29    is_grandpa = models.NullBooleanField("Es abuelito?", default=False)
30    personality = models.IntegerField("Personalidad", choices=PERSONALITY,
31        null=True, blank=True)
32    educational_level = models.IntegerField("Nivel Educativo",
33        choices=EDUCATIONAL, null=True, blank=True)
34    created_date = models.DateTimeField(auto_now_add=True, blank=True)
35    modified_date = models.DateTimeField(auto_now=True, blank=True)
36    age_group = models.IntegerField("Grupo Etareo", choices=AGE_GROUPS,
37        null=True, blank=False)
```

Listing 1: Código modificado del modelo UserProfile

C.2. Events

```
1 class Events(models.Model, ModelDiffMixin):
2     title = models.CharField('Titulo', max_length=45)
3     connotation = models.BooleanField('Connotacion', choices=CONNOTATION)
4     date = models.DateField('Fecha')
5     recurrent = models.BooleanField('Recurrente', choices=RECURRENT)
6     elderly = models.ForeignKey(UserProfile, on_delete=models.CASCADE,
7                               related_name="elderlyEvents")
8     created_date = models.DateTimeField(auto_now_add=True, blank=True)
9     modified_date = models.DateTimeField(auto_now=True, blank=True)
10
11     def __str__(self):
12         return self.title
13
14     class Meta:
15         verbose_name = 'Evento'
16         verbose_name_plural = 'Eventos'
```

Listing 2: Código del modelo Events

C.3. DailyContext

```
1 class DailyContext(models.Model, ModelDiffMixin):
2     date = models.DateField('Fecha')
3     average_temperature = models.FloatField('Temperatura Promedio')
4     rainfall = models.BooleanField('Precipitaciones')
5     season = models.IntegerField('Estacion del ano', choices=SEASONS)
6     holiday = models.BooleanField('Feriado', default=False)
7     country = models.CharField('Pais', max_length=60)
8     city = models.CharField('Ciudad', max_length=90)
9     town = models.CharField('Comuna', max_length=90)
10    light_time = models.FloatField('Tiempo de Luz (en segundos)')
11
12    def __str__(self):
13        return "Clima en {}, {}, {} el {}".format(self.town, self.city,
14        self.country, self.date)
15
16    class Meta:
17        verbose_name = 'Contexto Diario'
18        verbose_name_plural = 'Contextos Diarios'
```

Listing 3: Código del modelo DailyContext

C.4. InteractionWindows

```
1 class InteractionWindows(models.Model, ModelDiffMixin):
2     elderly = models.ForeignKey(UserProfile, on_delete=models.CASCADE,
3         related_name="elderlyInteractionWindows")
4     begin_time = models.DateTimeField('Hora de Inicio')
5     end_time = models.DateTimeField('Hora de Tormino')
6     duration = models.FloatField('Duracion')
7     interaction_type = models.BooleanField('Tipo de Interaccion')
8     with_support_personal = models.BooleanField('Con Personal de Apoyo')
9
10    class Meta:
11        verbose_name = 'Ventana sin Interaccion Social'
12        verbose_name_plural = 'Ventanas sin Interacciones Sociales'
```

Listing 4: Código del modelo InteractionWindows

C.5. WeeklyInteractions

```
1 class WeeklyInteractions(models.Model, ModelDiffMixin):
2     date = models.DateField('Fecha')
3     elderly = models.ForeignKey(UserProfile, on_delete=models.CASCADE,
4         related_name="elderlyWeeklyInteractions")
5     family_member = models.ForeignKey(ProfileFamily,
6         on_delete=models.CASCADE,
7         related_name="counterpartWeeklyInteractions")
8     interaction_count = models.IntegerField('Cantidad de Interacciones')
9     interaction_type = models.BooleanField('Tipo de Interaccion')
10    interaction_index = models.FloatField('Indice de Interaccion',
11        null=True, blank=True)
12    total_time = models.FloatField('Tiempo Total de Interaccion
13        (en segundos)', null=True, blank=True)
14
15    class Meta:
16        verbose_name = 'Interaccion Semanal'
17        verbose_name_plural = 'Interacciones Semanales'
```

Listing 5: Código del modelo WeeklyInteractions

C.6. Pets

```
1 class Pets(models.Model, ModelDiffMixin):
2     name = models.CharField('Nombre', max_length=45)
3     arrival_date = models.DateField('Fecha de Llegada')
4     death_date = models.DateField('Fecha de Defuncion', null=True,
5         blank=True)
6     elderly = models.ForeignKey(UserProfile, on_delete=models.CASCADE,
7 related_name="elderlyPets")
8     created_date = models.DateTimeField(auto_now_add=True, blank=True)
9     modified_date = models.DateTimeField(auto_now=True, blank=True)
10
11     def __str__(self):
12         return "{} - Mascota de {} {}".format(self.name, self.elderly.
13 first_name, self.elderly.last_name)
14
15     class Meta:
16         verbose_name = 'Mascota'
17         verbose_name_plural = 'Mascotas'
```

Listing 6: Código del modelo Pets

D. Detalle de la generación de registros InteractionWindows

D.1. Procesamiento de ventanas de no interacción físicas

En la Figura 1, se presenta un diagrama que muestra todos los pasos que sigue el sistema al procesar cada interacción física.

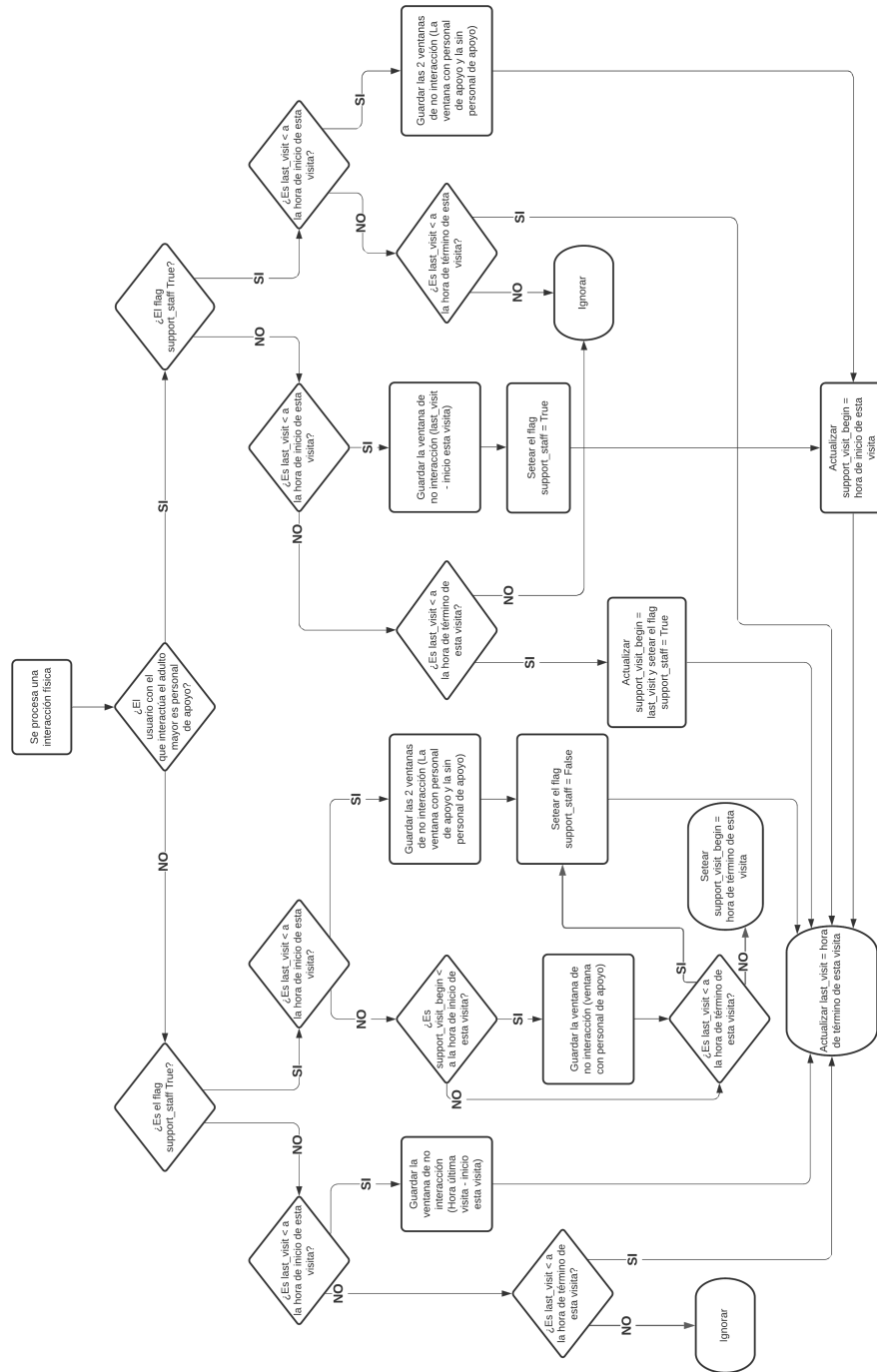


Figura 1: Diagrama del procesamiento de ventanas sin interacción física

D.2. Procesamiento de ventanas de no interacción digitales

En la Figura 2, se presenta un diagrama que muestra todos los pasos que sigue el sistema al procesar cada interacción digital. Es posible observar, que al ver una interacción de tipo mensaje, el sistema añade 20 minutos a su hora de término, a fin de cumplir con las especificaciones indicadas en la sección 3.3.5.

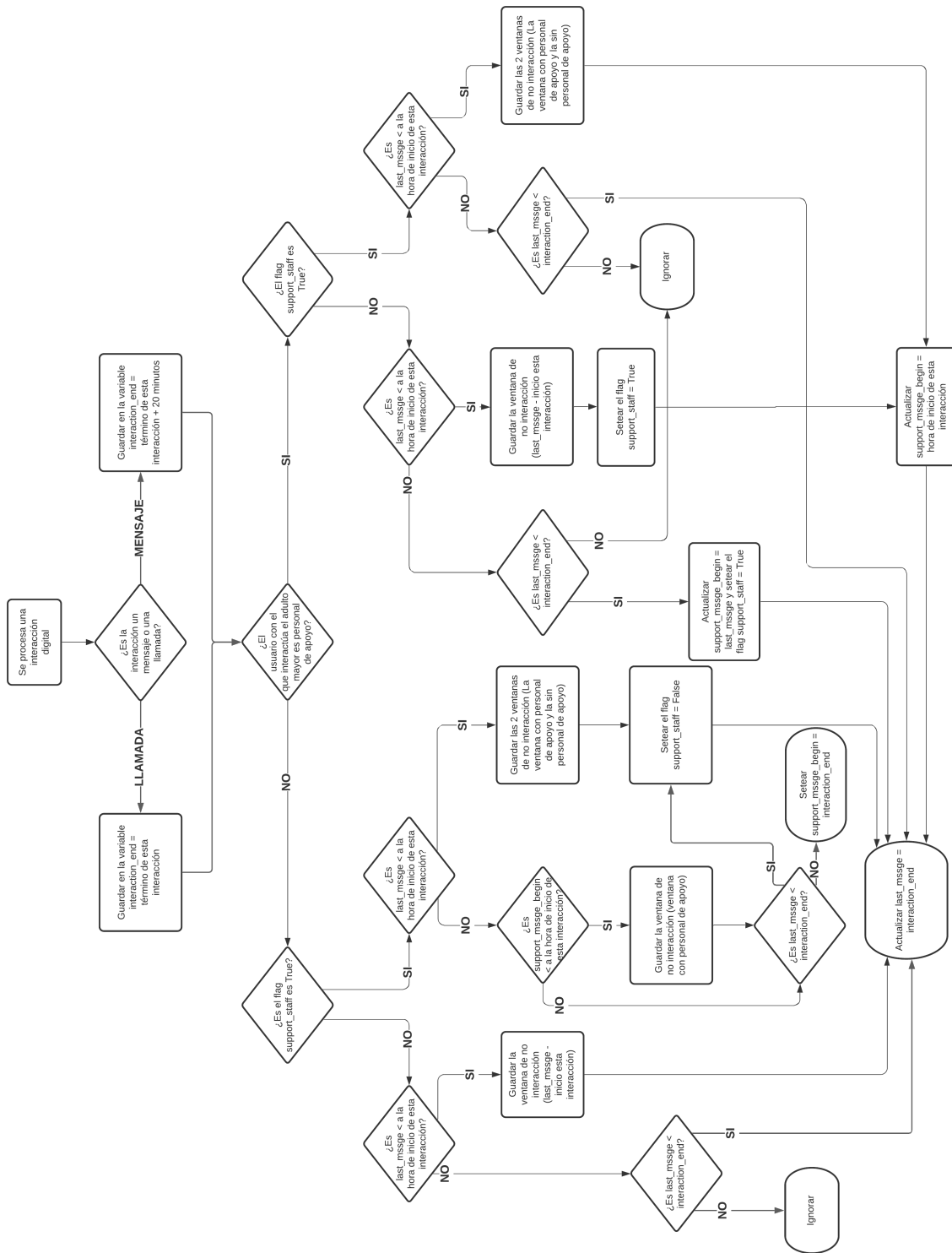


Figura 2: Diagrama del procesamiento de ventanas sin interacción digital

E. Otras visualizaciones exploradas

En este anexo se exponen dos gráficas que alcanzaron a ser implementadas tanto a nivel de backend como frontend. Sin embargo, estas no fueron incorporadas en el producto final de este trabajo de título por diversas razones.

E.1. Gráfica multilínea

Objetivo de la visualización: Esta visualización buscaba ayudar a observar cómo evoluciona la cantidad de interacciones entre una tupla adulto mayor - contraparte, un determinado día de la semana, durante un determinado período de tiempo.

Descripción: La gráfica multilínea fue concebida como un gráfico de tres líneas, donde cada una representa un bloque del día (Mañana, Tarde, Noche). En el caso de las interacciones físicas (visitas) y las interacciones digitales tipo llamadas, la visualización mapea la cantidad de horas que la contraparte interactúa con el adulto mayor al eje Y, mientras que en el caso de las interacciones digitales de tipo mensajes, el eje Y representa la cantidad de mensajes intercambiados entre las partes. Un ejemplo de la gráfica puede ser visto en la Figura 3.

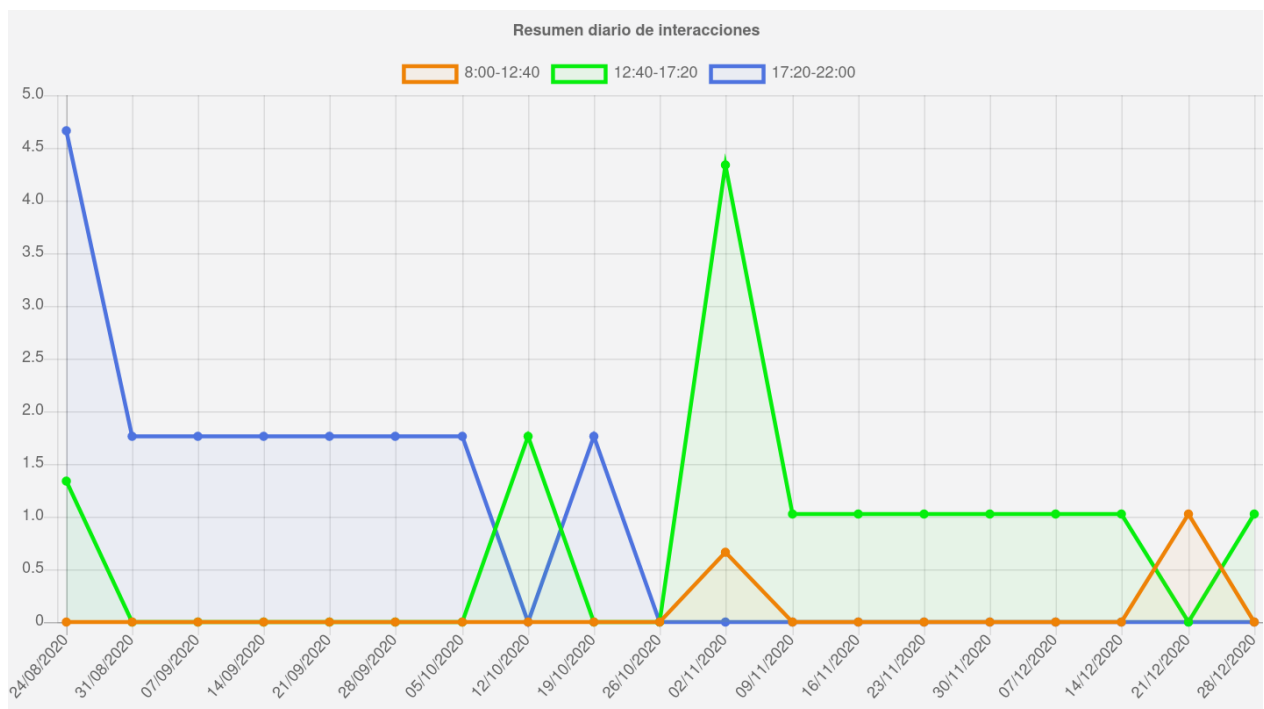


Figura 3: Gráfica multilínea de interacciones físicas

Para generar esta gráfica era necesario llenar un formulario muy parecido al presentado en la Figura 4.12, con la única diferencia que también era necesario especificar qué día de la semana era el que se quería graficar.

La idea de esta gráfica es que se pudiera observar la evolución de los volúmenes de interacción entre un par “adulto mayor - contraparte”, para un día de la semana en específico.

Razón de descarte: La gráfica se descartó principalmente por la complejidad de entender

la información entregada en dicha visualización. Tener tres líneas en el mismo gráfico podría ser potencialmente confuso para el usuario que la consultase. Además, para poder extraer información útil desde esta gráfica es necesario descubrir cierta información de antemano, es decir, para que un usuario pueda extraer información útil de la gráfica, es posible que tenga que saber cuáles son los días en los que existe un patrón de interacción o existe un volumen de interacciones interesante de estudiar.

A pesar de ser descartada, la idea general de esta visualización se reutilizó para construir la serie de tiempo que se le muestra al usuario al hacer click en las casillas del heatmap (Figura 4.9).

E.2. Serie de Tiempo con Autocorrelación

Objetivo de la visualización: Se buscaba que esta visualización ayudara a explorar la existencia de un patrón general de comportamiento (en cuanto a cantidad de interacciones) en las tuplas “adulto mayor - contraparte”, durante un período de tiempo definido.

Descripción: A diferencia del clustering, la gráfica de serie de tiempo con autocorrelación apuntaba a encontrar un patrón en la serie de tiempo completa, y no en una serie “quebrada en semanas”, como lo indicado en la subsección 4.3 (clustering).

Para esta visualización, se generaba una serie de tiempo que generaba un punto por cada día que hubiera dentro del período de análisis. Esta gráfica, tenía la particularidad de que su granularidad era diaria, y no basada en bloques horarios (es decir, el día no se dividía en 3 como el resto de las gráficas). Se hizo así para evitar generar una serie de tiempo con demasiados puntos, que puede ser difícil de entender para el usuario. A esto se le suma que el enfoque principal de esta gráfica fue explorar patrones a una granularidad diaria, más que de bloque horario. Es decir, se buscaba ver qué tan correlativa era la serie consigo misma, con X días de desfase, no con X bloques horarios de desfase.

El valor de cada punto de esta serie de tiempo corresponde a la cantidad de horas que la contraparte interactuó con el adulto mayor (caso interacciones físicas/digitales tipo llamada), o bien la cantidad de mensajes que intercambiaron las partes (caso interacciones digitales tipo mensajes).

Para seleccionar la información que utiliza esta gráfica, el formulario es el mismo que para el caso de la gráfica de boxplot Figura 4.12. Es decir, se debe elegir un adulto mayor, una contraparte, un tipo de interacción y un rango de fechas. En la Figura 4 se puede ver un ejemplo de la gráfica generada.

En esta Figura se puede observar que la mayor correlación de la serie consigo misma es cuando existen 14 días de desfase, con un coeficiente de correlación cercano a 0.41 (calculado en base a la correlación de Pearson).

Razón para descartarla: Durante las pruebas que se hicieron (generando gráficas en torno a distintos adultos mayores, rango de fechas y tipo de interacciones), la gráfica nunca mostró información que fuera claramente interpretable. En general, las correlaciones que mostraba no pasaban más allá de 0.6 y no mostraba patrones claros. En conversación con

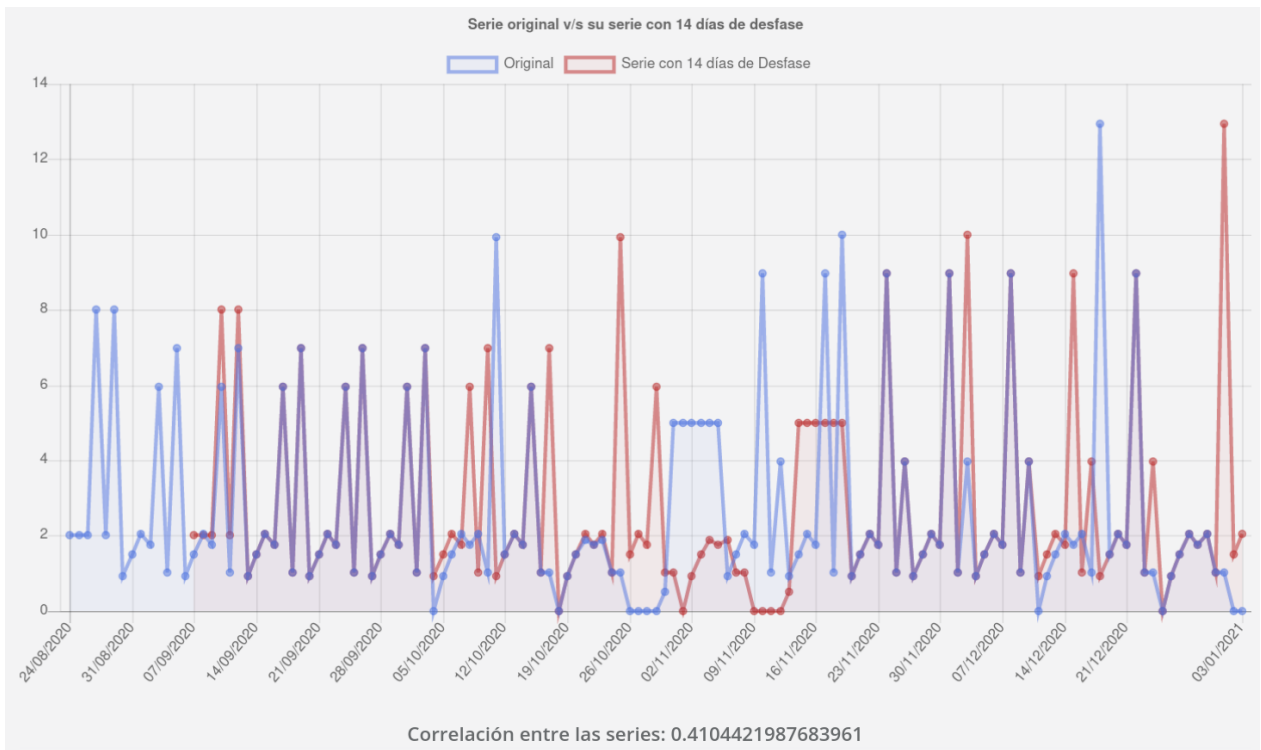


Figura 4: Gráfica de la serie de tiempo junto con su desfase de mayor correlación

expertos de dominio, se optó por remover esta gráfica debido a que se realizaron observaciones cuestionando la utilidad de la información entregada por esta gráfica en un régimen con datos reales (en producción).

A pesar de haber sido descartada, el código fuente que procesa la información para alimentar esta gráfica aún permanece en el backend. Esto permite que a futuro se pueda visitar dicha gráfica, para repensarla en pos entregar información relevante a través de ella.