



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

SISTEMA DE MONITOREO DE LA ACTIVIDAD SOCIAL DE ADULTOS MAYORES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ALONSO CRISTIÁN GAETE MORALES

PROFESOR GUÍA:
SERGIO FABIÁN OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
JOSE PINO URTUBIA
HUGO MORA RIQUELME

Este trabajo ha sido parcialmente financiado por el proyecto Fondecyt Nro 1191516

SANTIAGO DE CHILE
2020

Resumen

Los adultos mayores se enfrentan a varias de situaciones que impactan negativamente en su estado anímico, por ejemplo, enfermedades, la merma de sus capacidad físicas y mentales para realizar actividades. Por otra parte, el estilo de vida moderno de sus familiares deja poco tiempo para interactuar socialmente con ellos. Estas situaciones, entre otras cosas, hace a los adultos mayores más proclives a sufrir aislamiento social.

Este trabajo de memoria desarrolló una solución de software no invasiva, que permite monitorear la actividad social de los adultos mayores al interior de sus hogares. La información obtenida de este monitoreo se puede utilizar para inferir (de manera informal) el estado anímico de los abuelos e indicios de aislamiento social. Se escogió realizar el monitoreo dentro del hogar del adulto mayor, puesto que es el lugar de refugio de estas personas. Por otra parte, se decidió utilizar una Tablet PC como dispositivo de captura de la información base, dado que ya se cuenta con una aplicación de apoyo a la interacción social (llamada SocialConnector) que corre en este tipo de dispositivo.

El sistema desarrollado se llama Visitrack, y está compuesto por tres componentes principales: *Visitrack-Service*, *Visitrack-Server* y *Visitrack-Tagger*. *Visitrack-Service* se ejecuta en el Tablet PC, y es el encargado de detectar, recortar y enviar imágenes de caras a *Visitrack-Server*. *Visitrack-Tagger* es una aplicación de escritorio que permite el reconocimiento y el etiquetado manual de caras (particularmente de visitantes nuevos o no reconocidos por el software). *Visitrack-Server* es un componente que corre en un servidor en la nube, y recibe los inputs del *Visitrack-Service* y *Visitrack-Tagger*. Con esa información el sistema calcula las visitas ocurridas en el ambiente monitoreado.

Para evaluar la solución se dejó una Tablet PC funcionando dentro de una sala, y se simularon visitas a un adulto mayor al interior de ésta. La información de las visitas fue registrada en un documento por un observador, y luego se comparó esa información contra la que entregó el sistema. Como resultado se obtuvo que, en las condiciones en las que se pretende usar este sistema, la aplicación registró el 100% de las visitas. La duración de las visitas según el sistema, distó en menos de 10 minutos de la duración registrada. También se evaluó la usabilidad de *Visitrack-Tagger* en una segunda prueba, en esta se obtuvo que la aplicación posee muy buena usabilidad.

Agradecimientos

Quiero agradecer a mi profesor guía Sergio F. Ochoa por ser un gran apoyo y guía en el camino a concluir este trabajo de memoria.

También quiero agradecer a mi familia y amigos por apoyarme en el transcurso de toda mi vida universitaria.

Por último agradecer a mi pareja Paola Mena, por apoyarme en la vida diaria mientras trabajaba en finalizar este trabajo.

Este trabajo de memoria ha sido parcialmente financiado por el proyecto Fondecyt Nro 1191516.

Tabla de contenido

1	Introducción	3
1.1	Descripción del Problema a Abordar	3
1.2	Objetivos de la Memoria	4
1.3	Estructura del Documento	5
2	Marco Teórico	6
2.1	El Aislamiento Social de los Adultos Mayores	6
2.2	Detección de Personas en Objetos Multimediales	9
2.3	Reconocimiento de Identidad	10
2.4	Descripción del Software Anfitrión	11
3	Concepción de la Solución	12
3.1	Principales Requisitos a Considerar	12
3.2	Arquitectura de la Solución	13
3.3	Visitrack-Service	15
3.4	Visitrack-Server	19
3.5	Visitrack-Tagger	28
3.6	Modelo de datos	28
4	Implementación de la Solución	31
4.1	Algoritmo de Detección de Caras	31
4.2	Algoritmo Identificador de Visitas	32
4.3	Visitrack-Tagger	34
4.4	Integración del Servicio Visitrack al SocialConnector	38
5	Evaluación de la Solución	40
5.1	Evaluación del Visitrack	40
5.2	Evaluación del Visitrack-Tagger	43
6	Conclusiones y Trabajo a Futuro	48
	Bibliografía	51
	Anexo y Apéndices	54

1 Introducción

Los adultos mayores se ven enfrentados a una serie de situaciones adversas, propias de la etapa que viven, las cuales impactan negativamente en su estado anímico. Por ejemplo, las enfermedades físicas y mentales, o la merma de la capacidad para realizar diversas actividades. Con el paso de los años, también disminuye el tamaño de sus redes sociales, las cuales se circunscriben principalmente a los familiares [1].

Por otra parte, el estilo de vida de las familias modernas (es decir, del círculo familiar de los adultos mayores) deja poco espacio para interactuar con los abuelos. Por lo tanto, estos últimos pasan mucho tiempo realizando pocas interacciones sociales, lo cual también afecta negativamente su estado anímico, haciéndolos proclives a sufrir aislamiento social, cuadros de tristeza persistente, o inclusive depresión [2] [3].

Los estados anímicos negativos no sólo afectan la calidad de vida de las personas, sino también la forma en que éstas reaccionan a los medicamentos y a los tratamientos médicos. Hay diversos estudios que muestran que estos cuadros reducen el tiempo de vida de las personas; por lo tanto, detectarlos y tratarlos en forma temprana permite reducir su impacto en la vida de los adultos mayores [4]. Si tenemos en cuenta que la población mundial está envejeciendo en forma acelerada producto de los avances en medicina, es decir, el porcentaje de la población en que está en edad de adulto mayor es cada vez mayor, este problema se hace cada vez más grande a medida que pasa el tiempo.

Indudablemente sería muy útil contar con sensores capaces de monitorear el estado anímico y la actividad social de un adulto mayor, y que éstos informen a quienes monitorean o cuidan a los abuelos, especialmente si se detectan situaciones que requieren participación externa (tanto de profesionales de la salud, como miembros de su comunidad familiar). En esos casos es muy importante que el diagnóstico de los sensores sea certero, puesto que un diagnóstico incorrecto, y su consecuente solicitud de ayuda en favor del adulto mayor, haría perder tiempo a la persona que acude en apoyo del abuelo, y credibilidad al servicio de notificaciones asociado a dicho sensor.

Como se mencionó antes, no existe un sensor capaz de detectar en forma no invasiva este tipo de situaciones, por lo tanto, este trabajo de memoria busca desarrollar una solución no invasiva, que permita monitorear la actividad social de los adultos mayores al interior de sus hogares, como una forma de determinar su salud social. Complementariamente, en caso de identificarse situaciones de aislamiento social, el sistema debe enviar mensajes a algunos miembros de la red familiar solicitando su apoyo para revertir dicha situación.

Se escogió realizar el monitoreo dentro del hogar del adulto mayor, puesto que es el lugar de refugio de estas personas, particularmente cuando están física o emocionalmente mal, o bien cuando se sienten inseguras [5]. En otras palabras, si el adulto mayor voluntariamente

sale mucho de su hogar, es un síntoma bastante claro de que no sufre de un estado anímico y/o físico negativo.

Por otra parte, se decidió utilizar una Tablet PC como dispositivo de captura de la información base que permite determinar el estado anímico de los adultos mayores, dado que ya se cuenta con una aplicación de apoyo a la interacción social (llamada SocialConnector [6] [7]) que corre en este tipo de dispositivo. Dado que dicha aplicación ha tenido amplia aceptación por parte de los adultos mayores, se facilita considerablemente el proceso de experimentación y validación de la solución a desarrollar.

1.1 Descripción del Problema a Abordar

El estado anímico es una condición efímera de las personas, que puede cambiar muchas veces, y por múltiples razones, en un corto periodo de tiempo. Detectar emociones utilizando audio, imágenes o video es una tarea muy compleja que no cuenta aún con una solución precisa. Aunque hay mucho trabajo ya hecho al respecto por la comunidad científica del área, los estados anímicos negativos (por ejemplo, tristeza o aislamiento social) siguen sin poderse detectar con un alto grado de certeza. La situación se vuelve aún más desafiante cuando se busca realizar la detección de emociones utilizando como entrada información con baja resolución (calidad), por ejemplo, cuando la información es capturada por los periféricos estándares de una Tablet PC. Por esa razón, en este trabajo se buscó solo detectar la actividad social del adulto mayor, ya que este es un factor importante a considerar en la tarea de lograr predecir su estado anímico, y para ello se utilizó el video capturado por la cámara de la Tablet PC.

Para este trabajo, la calidad de la información está determinada por dos factores principalmente: 1) la resolución de la pieza de información analizada (por ejemplo, imágenes o videos) y 2) el contenido de la misma. Debido a que se busca tratar de intervenir lo menos posible el hogar de los adultos mayores, la captura de la información se realiza utilizando la cámara frontal de la Tablet PC que los usuarios tienen instaladas en sus casas. Dicho dispositivo simula ser un intercomunicador social entre el usuario y el resto de su comunidad familiar. De este modo, la calidad de la información capturada y analizada para detectar actividad social, está limitada por las capacidades propias de estos dispositivos, y a la situación misma que se dé al momento de tomar cada muestra.

Por otro lado, es esperable que las imágenes capturadas no sean las ideales para detectar la identidad de las personas (por ejemplo, no siempre se podrán capturar caras de las personas mirando de frente), puesto que el proceso de captura de imágenes a través de video no considera (a priori) si la persona está cerca o lejos del dispositivo, o si está mirándolo. Otro factor que usualmente afecta el reconocimiento de información es la baja luminosidad del ambiente, que hace que las muestras capturadas sean menos útiles para detectar personas y la identidad de éstas.

Bajo esas condiciones, el primer desafío radica en la detección puntual de la visita de una cierta persona en un instante dado, utilizando video. Esto requiere primero poder identificar a la

persona en base a información de entrada, aunque ésta esté en baja resolución, tal como se mencionó antes. Cuanto más alta sea la probabilidad de detección de la identidad de una persona, menor será el número de falsos positivos de la herramienta, al momento de determinar las visitas al hogar del adulto mayor.

Por otra parte, la identificación de la actividad social de un adulto mayor va más allá de la detección puntual de una visita en un instante dado. En primer lugar, hay que observar durante períodos suficientemente largos (por ej. dos o tres días), las visitas que recibe la persona monitoreada para evitar sacar conclusiones apresuradas. La historia de visitas recientes (por ejemplo, de las últimas 2 o 3 semanas) también debería considerarse en este análisis. Todas las visitas detectadas en ese período deben ser procesadas adecuadamente, para generar un dato resumen que sea efectivamente representativo del período observado. Es evidente la complejidad de este proceso de fusión de información, el cual involucra además definir un grado de confiabilidad de este dato resumen.

Actualmente existen diversas soluciones para detectar identidad utilizando principalmente imágenes y video. Sin embargo, la mayoría de ellas requiere información en alta resolución y un importante esfuerzo de entrenamiento del mecanismo de reconocimiento de personas. Se busca que este esfuerzo de entrenamiento sea lo más pequeño posible, pues la usabilidad y utilidad de la solución de monitoreo dependerá en parte de esto. Por otra parte, las propuestas que requieren poco entrenamiento tampoco resuelven el problema planteado, puesto que la complejidad del escenario antes descrito hace que usualmente se requiera utilizar más de un método para obtener un alto grado de certeza en la detección de identidad. Finalmente, tampoco está claro cuándo usar más de un método, y cómo fusionar la información resultante del reconocimiento. Por lo tanto, estas soluciones son potenciales insumos a evaluar durante el desarrollo del mecanismo de monitoreo de la actividad social de los adultos mayores.

1.2 Objetivos de la Memoria

El objetivo general de este trabajo de memoria es diseñar e implementar un sistema de monitoreo de la actividad social de un adulto mayor, al interior de sus hogares. El sistema debe permitir determinar, con alto de certeza, el nivel de actividad social de estas personas al interior de sus hogares (o lugar de monitoreo), entendiéndose esta actividad como la presencia o ausencia de la persona en el lugar y la cantidad de reuniones en la que participa en el lugar.

La aplicación considera el reconocimiento de la identidad de las personas en base al procesamiento de información de video principalmente. La solución obtenida fue diseñada como una extensión a la funcionalidad de la herramienta SocialConnector [6] [7], la cual actúa como un portarretrato digital, y a su vez como herramienta de interacción social para los adultos mayores.

Los objetivos específicos que se derivan de este objetivo general son los siguientes:

1. Diseñar una estrategia de detección de visitas al hogar de adultos mayores monitoreados, a través del procesamiento de video. Esta estrategia también permite

determinar automáticamente la identidad de las personas, siempre que sea factible. Este es el caso de visitantes habituales, es decir, de aquellos de los cuales ya se tiene registro.

2. Diseñar una heurística para determinar el nivel de actividad social de la persona monitoreada, a partir de la información registrada por el detector de visitas.
3. Integrar la solución al sistema SocialConnector; particularmente los algoritmos que la implementan.

1.3 Estructura del Documento

Este documento de memoria está organizado en seis capítulos. El capítulo 2 presenta el marco teórico de este trabajo. El capítulo 3 describe el resultado de la concepción de la solución, esto es, las definiciones estructurales asociadas al diseño del sistema desarrollado. El capítulo 4 presenta la implementación de la solución. El capítulo 5 describe la evaluación de la solución. El capítulo 6 presenta conclusiones y trabajo a futuro.

2 Marco Teórico

En esta sección se describe más en detalle el problema global de aislamiento social que sufren los adultos mayores, aportando evidencia de que dicho problema está presente también en Chile. Luego se presentan y discuten los principales trabajos reportados en el ámbito de monitoreo de la actividad social de adultos mayores. También se presentan algunos trabajos previos en el ámbito de identificación de personas e identidades de éstas, principalmente utilizando información de video. Finalmente, se describe brevemente el sistema SocialConnector [6] [7], el cual incorpora los servicios resultantes de este trabajo de memoria dentro de sus servicios.

2.1 El Aislamiento Social de los Adultos Mayores

La Organización Mundial de la Salud (OMS) ha reafirmado la importancia que tiene la convivencia social sobre el bienestar de las personas independientemente de su cultura [32]. Además, enfatiza que el concepto de salud está vinculado al estado de bienestar físico, mental y social de una persona, y no solamente a la ausencia de enfermedades o dolencias [31]. En este sentido, diversos estudios muestran que la dimensión social de la salud, particularmente la cantidad y calidad de relaciones interpersonales, afecta la salud física, mental, el comportamiento de las personas y su tasa de mortalidad [33] [34] [35]. Por lo tanto, mejorando la salud social de las personas se podría impactar positivamente su salud física, mental y su calidad de vida. En esta propuesta adherimos a la definición que considera a la salud social como la habilidad para mantener relaciones saludables con los amigos, familia, vecinos o compañeros de trabajo [36].

Por otra parte, se ha comprobado que a medida que la persona envejece, sus redes sociales tienden a reducirse y a focalizarse en los miembros de su comunidad familiar (principalmente hijos y nietos), y que la frecuencia de las interacciones sociales tiende a decrecer. Este empobrecimiento de las relaciones sociales impacta negativamente en su salud física y mental, y por lo tanto, en su calidad de vida y la de su familia. Según el Servicio Nacional del Adulto Mayor (SENAMA) del Gobierno de Chile: “el aislamiento y la inactividad producen una deficiente salud social que afecta nuestra salud física y psicológica. El peor enemigo de una persona mayor es la soledad y el aislamiento” [37]. Esto pone de manifiesto la necesidad de mejorar la salud social de los adultos mayores, como una estrategia para mejorar su calidad de vida, y así reducir el peso que estas personas representan para sus familias y los sistemas asistenciales de un país; especialmente para los sistemas de salud.

Si bien el problema de aislamiento social de los adultos mayores parece tener distintas causas, el consenso general apunta a la reducción de los espacios que estas personas tenían para socializar con otros. Hasta hace 20-30 años atrás las interacciones sociales se llevaban a cabo principalmente a través de reuniones cara-a-cara, y comunicación telefónica (Fig. 1.a). En sociedades colectivistas, los hijos adultos, especialmente las mujeres, aún dedican bastante tiempo a cuidar tanto de sus padres como de sus hijos [38] [39]. El cuidado de los padres es

parte de la obligación filial de los hijos, la cual tiene un gran protagonismo en la cultura latina [40].

En ese período, el cuidado simultáneo de padres e hijos era bastante factible puesto que muchas mujeres no trabajaban jornada completa fuera de sus hogares; situación que ha ido cambiando rápidamente. Hoy es poco probable que un hijo adulto (sin importar el género) no trabaje jornada completa, lo cual reduce considerablemente el tiempo para cuidar simultáneamente a sus hijos y a sus padres. Además, como muestra la literatura, cuidar a los hijos en Chile tiene prioridad por sobre el cuidado de los padres [40], lo cual reduce aún más el espacio de interacción social entre estas personas y los adultos mayores, afectando principalmente la salud social de estos últimos.

Este déficit de espacio para socializar ha fomentado el uso de tecnología para mediar las interacciones sociales entre los miembros de la comunidad familiar [41]. Si bien las interacciones cara-a-cara y vía teléfono entre adultos mayores y sus hijos se mantienen, su frecuencia, calidad y duración se ven reducidas; especialmente las interacciones cara-a-cara, que son las más apreciadas por los adultos mayores, y las que mayor impacto tienen en el bienestar emocional de las personas mayores (Fig. 1.b) [42] [20] [21].

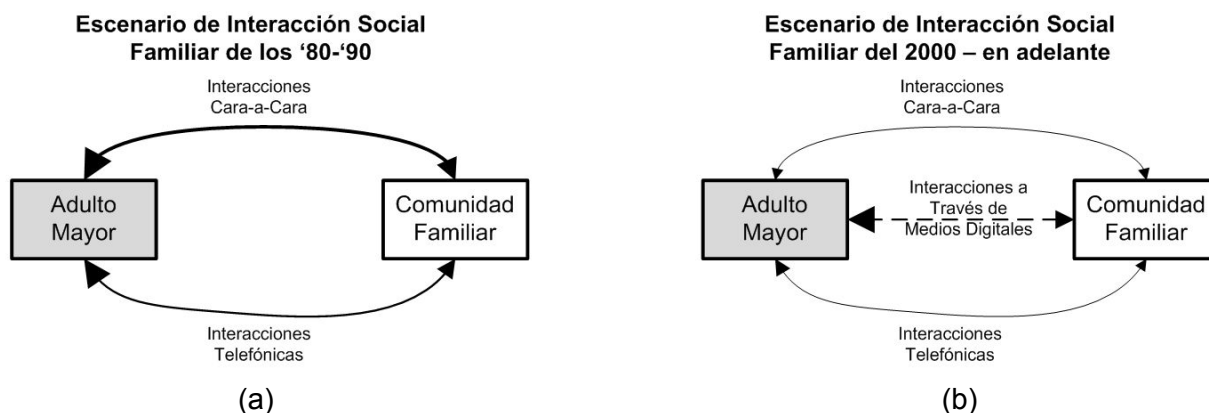


Figura 1. Evolución del Escenario de Interacción Social Familiar de los Adultos Mayores

La irrupción de la tecnología como medio de interacción social (por ejemplo, con redes sociales como Facebook e Instagram, sistemas de mensajería móvil como Whatsapp, sistemas de videollamadas por computador, etc.) ha afectado negativamente la integración de los adultos mayores, quienes generalmente se resisten reemplazar las tradicionales interacciones cara-a-cara, por este nuevo paradigma de socialización mediado por computador [22]. Sumado a esta reticencia, está la falta de destrezas cognitivas y motoras de estas personas para usar las nuevas tecnologías, que hace que la mayoría de ellos fallen en su adopción. Esto los aísla cada día más, los vuelve más sedentarios, dependientes, empobrece su calidad de vida, e incrementa la carga que estas personas representan para sus familias y para los sistemas de apoyo de un país. Si consideramos la velocidad de envejecimiento de la población y de evolución de la tecnología, se hace evidente que estamos frente a un problema de gran magnitud, que crece día a día y que requiere de una pronta atención.

Este problema está presente no sólo en Chile, sino en la mayor parte del mundo. Sin embargo, las soluciones a este problema tienen una fuerte influencia de la cultura local, lo cual hace que se requiera hacer investigación científico-tecnológica para determinar formas adecuadas de abordar el problema planteado, en el escenario Chileno [22] [23].

Para ilustrar este punto, consideremos las sociedades individualistas -como la Europea o la Estadounidense- en donde el problema planteado no está tan presente como en Chile, puesto que los adultos mayores son más independientes, más proclives a utilizar tecnología y menos a recibir ayuda [24] [25]. Además, sus hijos -aún sin perder el contacto con su familia de origen- están menos disponibles para hacerse cargo de sus padres en forma directa cuando ellos envejecen [26]. Por lo tanto, las soluciones para mejorar la salud social de los adultos mayores en dichos escenarios son menos requeridas que en Chile, y además persiguen otros objetivos que son relevantes particularmente para esas personas. Por lo tanto, esas soluciones son probablemente poco efectivas si se aplican al escenario chileno, debido al cambio de contexto del problema abordado.

En resumen, estamos frente a un problema caracterizado por aspectos de nuestra cultura, los cuales deben ser considerados tanto para entender el desafío, como para generar las soluciones que lo aborden. En el escenario chileno hay mucho trabajo realizado en pos de entender el proceso de envejecimiento y la diversidad de relaciones sociales del adulto mayor (por ejemplo, [27] [28]). Sin embargo, hay pocos estudios acerca de los mecanismos de apropiación de tecnología por parte de personas mayores, como un instrumento de mejora de su salud social. Particularmente, como una herramienta de ayuda al empoderamiento de este segmento de la población y su cambio de actitud en favor de un envejecimiento socialmente más activo. Por esa razón se requiere realizar investigación socio-técnica en este ámbito.

Tratando de contribuir a paliar la situación planteada, este proyecto busca definir mecanismos de apoyo a la interacción social mediada por tecnología, que respeten las capacidades y preferencias de los adultos mayores, así como sus diferentes niveles de apropiación tecnológica, con el fin de fomentar la inclusión social. Dichos mecanismos serán embebidos en una herramienta socio-técnica que fomentará el empoderamiento, la autonomía y la participación social activa de los adultos mayores, tanto dentro de su red familiar como de su entorno social (comunidades, redes asistenciales, vecinos y amigos).

Por otra parte, la adopción tecnológica, la inclusión social y el empoderamiento abren oportunidades para fomentar estilos de vida saludables en las personas mayores, tales como autocuidado, autoconfianza, actividad física, estimulación cognitiva y buenos hábitos de higiene, salud y alimentación. Como consecuencia, se busca mejorar la calidad de vida de los adultos mayores y la de sus familias.

Dada la similitud que existe entre los estilos de vida familiar en los distintos países de Latinoamérica, es de esperar que las soluciones que se generen en el marco de este proyecto, puedan también aplicarse a otras realidades dentro de la región.

2.2 Detección de Personas en Objetos Multimediales

Como se mencionó antes, la detección de personas es un desafío aún en curso, donde gran parte de la comunidad científica trabaja en resolverlo con distintos propósitos, y también con distintas restricciones. A continuación se presentan algunos de los trabajos relacionados más representativos en detección de personas utilizando imágenes, video y audio.

2.2.1 Detección de Personas a Través de Imágenes

En Viola et al. [8] se presenta el ‘Viola–Jones object detection framework’, un detector de objetos contenidos en imágenes, basado en la detección secuencial de características. La detección utiliza un conjunto de pruebas simples, que al ser realizadas en secuencia aumentan la certeza acerca de la probabilidad de que el objeto en cuestión esté presente. Con el entrenamiento adecuado, este sistema funciona bastante bien para detección de caras en múltiples condiciones de luz, de resolución de imagen y de tamaños de caras. También funciona bien frente a una rotación considerable de una cara, con respecto a la vista frontal. Si bien este sistema es muy útil para detectar caras en las imágenes usadas como input en este trabajo, sólo se limita a la detección, quedando el reconocimiento de identidad como un trabajo que debe ser abordado en forma separada.

2.2.2 Detección de Personas a Través de Videos

La detección de personas a través de videos se puede enfrentar como un caso puntual de la detección de personas en imágenes, cuando las imágenes utilizadas son frecuentes, consecutivas y de un mismo plano. Dado que el algoritmo presentado en Viola et al. [8] (‘Viola–Jones object detection framework’) es suficientemente eficiente, éste puede utilizarse para hacer detección en video y más aún en vídeo en tiempo real.

Por otro lado, considerando que un video puede entenderse como una secuencia de imágenes consecutivas tomadas en el mismo lugar, y a muy corta diferencia temporal, la información obtenida de una imagen puede ser usada para inferir información del resto de las imágenes cercanas en tiempo. Considerando esto, la detección de una persona en un video puede realizarse buscando entre todas las imágenes, aquella que tenga la mejor calidad para así tener más chances de realizar una correcta detección. También puede usarse la detección prolongada y consecutiva de una persona, como una forma de inferir presencia. Típicamente este método es el que provee mayor la probabilidad de detecciones correctas.

2.2.3 Detección de Personas a Través de Audio

El principal problema en la detección de voz en archivos de audio tiene relación con la cantidad de ruido ambiente registrado en el archivo o en la muestra en línea, si corresponde. Existen numerosos estudios que presentan métodos para reconocer voz en audios ruidosos [9] [10] [11], algunas estrategias se basan en umbrales de energía [12], detección de tonos [13] y en el análisis del espectro de frecuencias [14], entre otros. En Ramírez et al. [15] se presenta un acercamiento integral a los principales retos en detección de voz, y se muestran las principales soluciones y frameworks que se usan regularmente para este fin.

2.3 Reconocimiento de Identidad

A continuación se presentan y discuten los principales mecanismos reportados en la literatura para reconocimiento de identidad de las personas, utilizando imágenes, video y voz.

2.3.1 Reconocimiento de Identidad a Través de Imágenes

Existen múltiples métodos utilizados para el reconocimiento de identidad en imágenes, donde varios de ellos responden a una mecánica similar. Estos utilizan la clasificación en clases, esto se implementa a través del uso de un conjunto de imágenes de entrenamiento, clasificadas correctamente en clases particulares. Esta información se utiliza luego en el proceso de reconocimiento de personas en nuevas imágenes, a través de su clasificación en las clases ya definidas. Esta clasificación se realiza, extrayendo alguna característica de una imagen, propia de cada modelo, y comparando dicha característica contra las imágenes de entrenamiento. Este proceso lo realiza un clasificador. El sentido de la extracción de características es disminuir la información a procesar acerca de una imagen (sin perder la parte útil de ella), para que sea computacionalmente factible de comparar en tiempos razonables.

En [16] se describe al clasificador Eigenfaces, el cual utiliza análisis de las componentes principales (PCA) de la imagen para disminuir su dimensionalidad. La reducción dimensional se aplica tanto a las imágenes de entrenamiento como a las imágenes a reconocer. La clasificación se realiza buscando la imagen más cercana perteneciente a una clase, en comparación a la imagen a reconocer. Esta estrategia se ve muy afectada por fuentes externas, como la luminosidad o las sombras, ya que éstas producen fuertes diferencias en el resultado del PCA en imágenes de una misma persona.

En [17] se presenta el reconocimiento facial utilizando el método llamado histogramas de patrones binarios locales. En este caso se divide la imagen en celdas, y a su vez se calcula un número por cada píxel en la celda como resultado de una comparación de dicho píxel con sus píxeles vecinos. Luego dichos números se utilizan para generar un histograma en la celda a la que pertenecen sus respectivos píxeles. Estos histogramas tienen asociados la posición en la imagen de cada celda correspondiente. Estos histogramas se utilizan para la comparación y clasificación según cercanía.

2.3.2 Reconocimiento de Identidad a Través de Audio

Los trabajos de reconocimiento de personas utilizando el procesamiento de la voz siguen principalmente dos líneas: (1) las propuestas que intentan reconocer a personas cuando éstas pronuncian un texto conocido para el sistema, y (2) las que intentan reconocer a una persona que pronuncia un texto desconocido para el sistema. Estos dos enfoques se conocen como reconocimiento de voz dependiente del texto, y reconocimiento de voz independiente del texto respectivamente.

En [19] se presenta una visión panorámica de algunos métodos de reconocimiento de voz independiente del texto. Algunos métodos clásicos usados en el reconocimiento de identidad

procesando la voz son siguientes: cuantificación vectorial, modelo de mezcla Gaussiana y máquina de vector de apoyo. Estos métodos serán evaluados para ver cuál o cuáles de ellos serían los más apropiados para abordar el problema planteado.

2.4 Descripción del Software Anfitrión

En el trabajo de Muñoz y otros [6] [7] se presenta el diseño e implementación de la aplicación llamada SocialConnector, que permite a los adultos mayores comunicarse con sus familiares a través de redes sociales a pesar de no saber ocuparlas. Esta aplicación está implementada en Android, y entre otras cosas permite compartir fotografías, que son automáticamente bajadas desde las cuentas de Facebook de los familiares del adulto mayor. La aplicación también permite a estos últimos interactuar con los miembros de su familia a través de videoconferencia, o intercambiando mensajes de texto (que en realidad son emails). Este sistema es útil como fuente de datos (imágenes y grabaciones de voz) para alimentar el sistema de reconocimiento de actividad social a abordar en esta memoria.

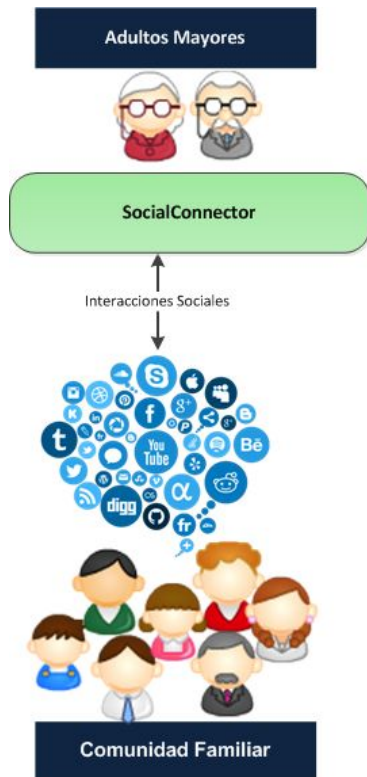


Figura 2. Estructura del escenario de interacción del SocialConnector



Figura 3. Interfaz principal del SocialConnector

3 Concepción de la Solución

En este capítulo se presentan las definiciones estructurales del sistema de monitoreo desarrollado.

3.1 Principales Requisitos a Considerar

A continuación se presentan los principales requisitos funcionales y de restricción de la solución:

Requisitos funcionales:

- La solución debe ser capaz de detectar la presencia de personas automáticamente en tiempo real a través de la detección de sus caras.
- La detección de caras debe ser un proceso capaz de funcionar de forma permanente en la aplicación. Por ejemplo, que no afecte severamente la capacidad de memoria o de almacenamiento del computador que ejecute este proceso.
- El sistema debe ser capaz de reconocer la identidad de las personas a través del reconocimiento de imágenes de sus caras, y este proceso debe realizarse de forma automática.
- La solución debe permitir a un usuario administrador, realizar la definición de identidades de personas por demanda. Por ejemplo, utilizando mecanismos como el tagging de los rostros capturados por la cámara del dispositivo.
- La solución debe permitir a un usuario administrador definir a una persona como “residente”, indicando así que esa persona es la que vive en la casa y el resto de las personas registradas serán visitas.
- La solución debe también permitir a un usuario administrador, realizar la identificación de personas a través del reconocimiento manual de las caras de los visitantes. Esto consiste en asignar la identidad a imágenes de caras que el sistema no haya logrado identificar de forma automática.
- La solución debe permitir a un usuario administrador revisar las imágenes de los visitantes reconocidos y no reconocidos por el computador.
- La solución debe ser capaz de retroalimentarse de las nuevas imágenes de caras ya identificadas, y de esa manera mejorar su capacidad de reconocimiento automático frente a nuevas imágenes de caras.

Requisitos de restricción:

- La solución debe utilizar la cámara frontal de un Tablet PC como fuente de captura de imágenes y videos, para luego realizar la detección y reconocimiento de caras.
- El Tablet PC debe permanecer en posición estable (es decir, mirando la misma zona de la casa) durante su uso.
- La solución debe soportar pérdidas momentáneas de conexión a internet en el Tablet PC, funcionando tanto de manera online como offline.

- El sistema tiene que manejar cuidadosamente y proteger la identidad de las personas identificadas como visitantes, y también aquellas no identificadas.

3.2 Arquitectura de la Solución

El sistema desarrollado en esta memoria se llama Visitrack. El principal servicio a brindar por este sistema es entregar información confiable acerca de la actividad social de los adultos mayores. Esta información está acotada a las interacciones presenciales ocurridas al interior del hogar del adulto mayor, particularmente en los espacios capaces de ser monitoreados por un Tablet PC. Típicamente dicho dispositivo está instalado en el living o en la cocina de la casa del adulto mayor; particularmente se busca que esté instalado en un lugar donde se lleven a cabo la mayor parte de las interacciones sociales presenciales de estas personas.

El sistema puede reconocer a los visitantes (por ejemplo, familiares, amigos y desconocidos), así como la frecuencia de estas visitas. Dado un periodo de monitoreo, el sistema puede entregar resúmenes y conclusiones respecto a la interacción social del adulto mayor.

Visitrack funciona integrado a la aplicación SocialConnector. A grandes rasgos el SocialConnector es un “punto de convergencia” entre distintos medios de comunicación (Gmail, Skype, Facebook), y a la vez un facilitador del uso de estos por parte de los adultos mayores. La Figura 4 muestra la arquitectura del escenario de trabajo donde opera tanto el SocialConnector, como el Visitrack.

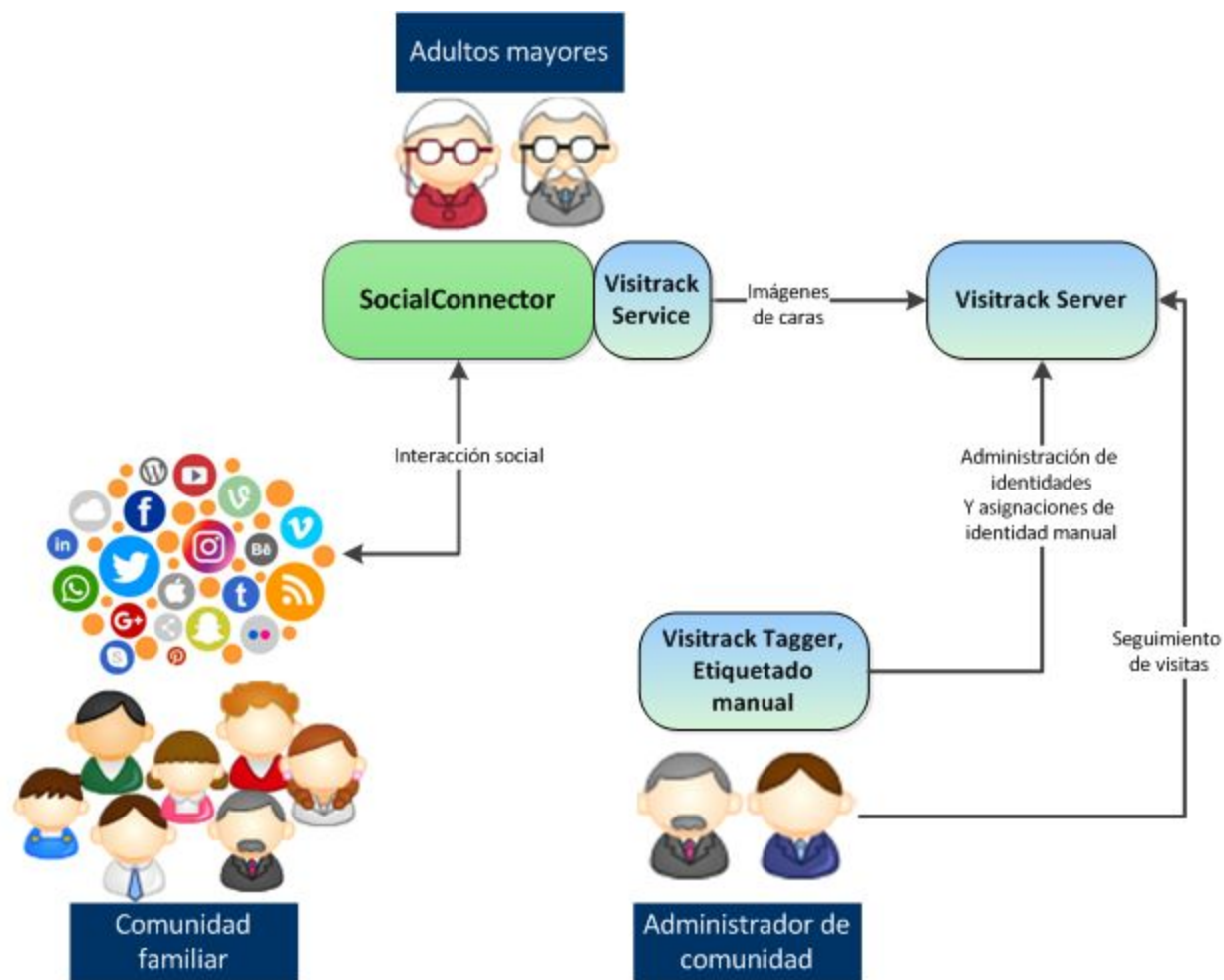


Figura 4. Arquitectura del Ambiente Operacional de SocialConnector

SocialConnector utiliza una Tablet PC para la interacción con los adultos mayores y este mismo es aprovechado por Visitrack para la recolección de fotografías. Visitrack se compone de 3 partes principales: *Visitrack-Service*, *Visitrack-Server* y *Visitrack-Tagger*. *Visitrack-Service* se ejecuta en el Tablet y es cliente de *Visitrack-Server*. Por su parte *Visitrack-Tagger* también es un cliente de *Visitrack-Server*, sin embargo, es una aplicación de escritorio que permite el reconocimiento y etiquetado manual de caras (visitantes nuevos o no reconocidos por el software).

La arquitectura del sistema usa acoplamiento por datos entre varios componentes del sistema; por ejemplo, entre el *Visitrack-Service* y *Visitrack-Server*, y entre todos los componentes de *Visitrack-Server*. Este desacoplamiento de la funcionalidad (o acoplado por datos) permite al sistema evolucionar sus componentes de forma autónoma. Particularmente, es posible hacer reingeniería del módulo de registro de visitas, sin necesidad de cambiar el código fuente del resto de los componentes de *Visitrack-Server*, lo cual es una gran ventaja para el sistema.

Visitrack, a través de su servicio llamado Visitrack-Service, usa la cámara frontal del Tablet para obtener imágenes para detectar personas, cuando éste detecta a una persona, captura una imagen de ella, recorta la porción de la imagen que corresponde a su cara y la sube a un servidor (repositorio de caras del sistema).

Cuando una imagen es enviada al servidor, ésta es procesada en etapas por diferentes módulos, que intentan reconocer la identidad de la persona en la foto correspondiente. Esto se realiza utilizando un modelo de detección de objetos entrenado previamente. Las caras que no logran ser identificadas de manera confiable, quedan disponibles para ser identificadas manualmente por administradores de la comunidad familiar (usualmente familiares de los adultos mayores), quienes a través de una aplicación de escritorio llamada Visitrack-Tagger rotulan a la gente que no identificó la aplicación. Estas rotulaciones retroalimentan la base de conocimiento que utiliza el sistema para realizar futuros reconocimientos de identidad.

Todas las fotografías a partir de las cuales se reconocen caras, dan lugar a un registro referenciado en el tiempo, que se almacena en un archivo de log. Usando esta información, cada cierto tiempo se determina el número y la duración de las visitas ocurridas en el hogar del adulto mayor, infiriendo así la actividad social realizada en la casa ese día. Ese mismo mecanismo sirve además para determinar el nivel en enclaustramiento (o encierro) tiene el adulto mayor, lo cual también hace parte del diagnóstico de la salud emocional de una persona.

Si se analiza esta información en períodos más largos de tiempo; por ejemplo, semanas o meses, se puede obtener una perspectiva mucho más representativa y precisa de la actividad social de estas personas.

La información sobre la actividad social puede ser revisada en un sitio web habilitado para este fin, también puede ser enviada periódicamente a algún miembro de la familia o cualquier persona que actúe como monitor del adulto mayor, para que éste a su vez pueda ocuparse en caso de existir baja actividad social. Una forma de abordar el aislamiento de los adultos mayores es, por ejemplo, persuadiendo a miembros de la familia a que realicen una visita a esa persona.

3.3 Visitrack-Service

Visitrack tiene un servicio llamado Visitrack-Service corriendo en el Tablet PC, el cual está constantemente buscando detectar la presencia de personas. El servicio usa la cámara frontal del dispositivo, toma una foto con esta cámara alrededor de 3 veces por segundo. En cada oportunidad se intenta detectar la presencia de caras en la foto, usando el algoritmo de detección de caras llamado "haarcascade", el cual está disponible en la librería OpenCV [29].

Cada vez que se detecta una cara, se intenta además detectar la presencia del par de ojos y la boca correspondiente, dentro de la cara detectada (Fig. 5). En caso de que exista presencia de ojos se revisa la diferencia de tamaño y de posición vertical entre ambos, si existe presencia de boca y de ojos, también se revisa la distancia que existe entre los ojos y la boca. La idea con esto es conseguir caras lo más completas posibles y de mejor calidad, buscando que no

tenham inclinación, a partir de las cuales se pueda reconocer la identidad de las personas siempre que sea factible. Tanto la detección de caras, ojos y bocas se realizan usando el algoritmo de detección haarcascade, pero con un set de entrenamiento específico para cada elemento a detectar, todos ellos disponibles en la librería OpenCV.

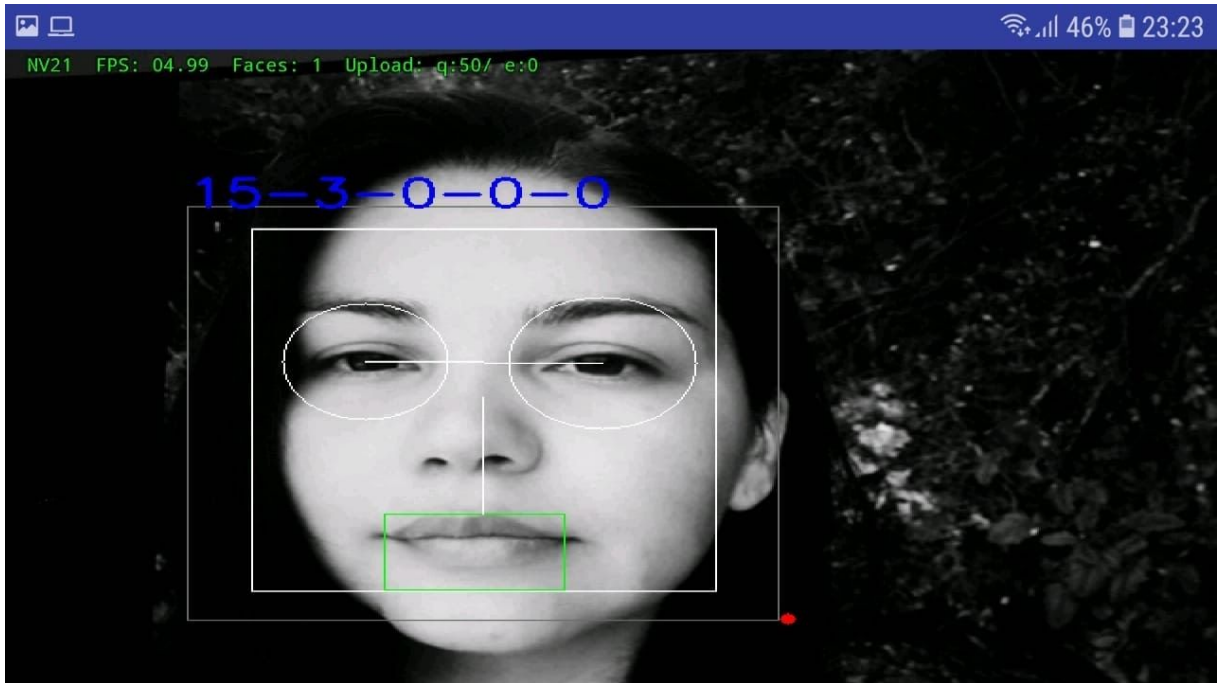


Figura 5. Ejemplo de una cara detectada y recortada por el sistema

La Figura 6 resume el proceso de captura de imágenes, así como de la obtención de las caras y su almacenamiento en la base de datos local.

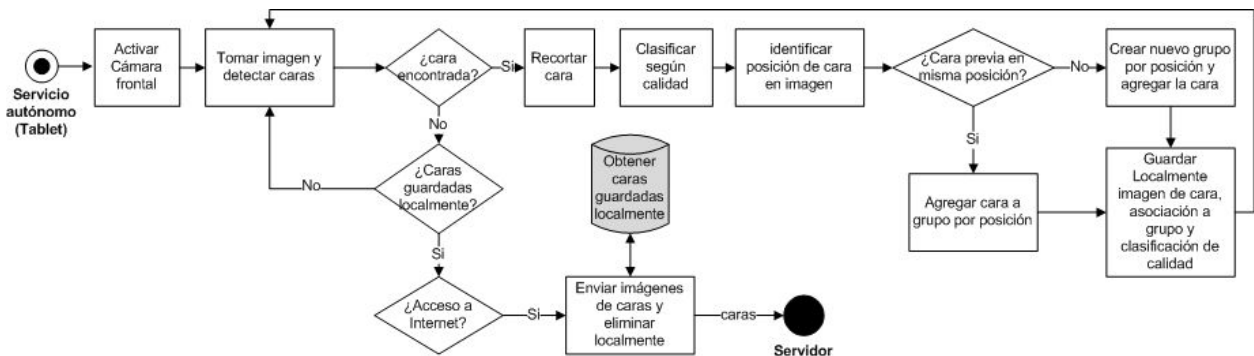


Figura 6. Obtención y almacenamiento de una cara realizado por el sistema.

Considerando los datos anteriores, las caras detectadas se clasifican en cinco categorías según su calidad:

- **Categoría 1:** Estas son las mejores caras para determinar la identidad de la persona, ya que se considera una cara de frente y de buena calidad. En este caso existe presencia de ambos ojos y boca, los ojos son de igual tamaño y están en la misma posición vertical. Además, la distancia de la boca a los ojos es suficiente para asegurar que no se trata de la cara inclinada hacia atrás o hacia adelante.
- **Categoría 2:** En este caso existe presencia de ambos ojos y boca, pero ocurre alguna de las siguientes 3 situaciones: los ojos no son de igual tamaño, los ojos no están en la misma posición vertical, o la distancia de la boca a los ojos no es suficiente para asegurar que no se trata de la cara inclinada hacia atrás o hacia adelante.
- **Categoría 3:** En este caso existe presencia de ambos ojos, pero no de la boca. Además, los ojos son de igual tamaño y están en la misma posición vertical.
- **Categoría 4:** En este caso existe presencia de ambos ojos, pero no de la boca, y los ojos no son de igual tamaño o no están en la misma posición vertical.
- **Categoría 5:** En este caso existe presencia de boca, pero no de los ojos.

La Figura 7 resume el proceso de clasificación de caras que lleva a cabo el sistema, considerando las cinco categorías antes mencionadas.

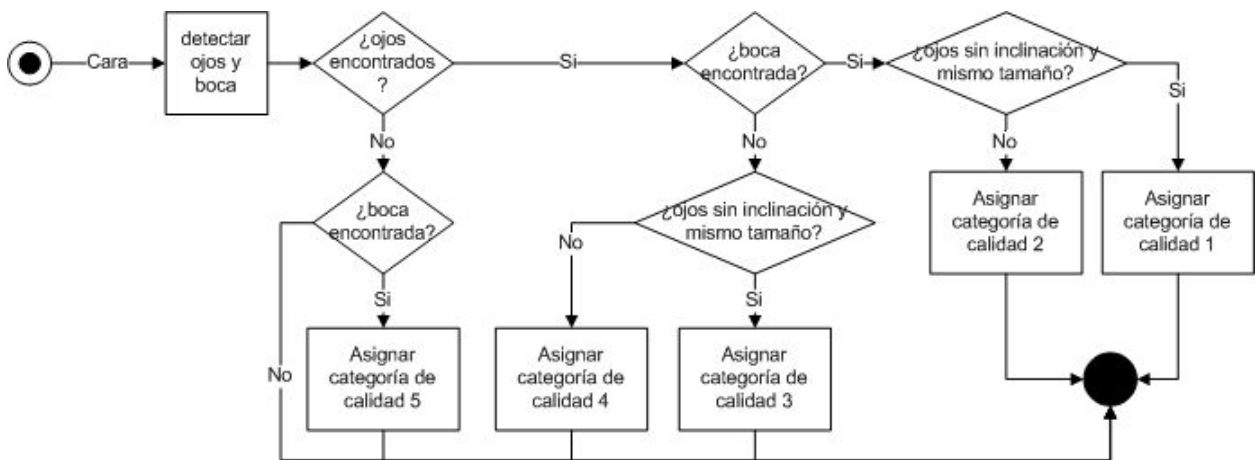


Figura 7. Proceso de clasificación de calidad de una imagen de cara capturada.

Las caras recortadas se agrupan por posición en el caso de que ocupen la misma zona de la imagen en detecciones sucesivas. Esto es porque se considera que se trata de la misma persona en ambas imágenes de cara. Si las caras comparten la posición en un 70% o más, se considera que ocupan la misma posición y se les asigna el mismo grupo.

Las caras en las cuales no se detectan ojos ni boca son descartadas, pues posiblemente se trate de errores en la detección de caras. El resto de caras, que sí quedan en alguna categoría,

se graban en memoria secundaria y se dejan en la bandeja de salida para ser enviadas al servidor.

El Visitrack-Service necesita un identificador del Tablet PC, pues pueden existir múltiples dispositivos funcionando en paralelo, cada uno en el hogar de un adulto mayor distinto. Estos dispositivos estarán enviando información al servidor, por lo que se requiere identificar a cada Tablet PC. Por defecto este identificador es el Android-Id del sistema Android del dispositivo.

El nombre del archivo que contiene la imagen de la cara recortada se ocupa para comunicar información de la cara. Éste se compone de la concatenación de los siguientes datos (separados por guión bajo):

1. Texto: "f"
2. Identificador del Tablet PC
3. Identificador del grupo por posición
4. Categoría de calidad de la imagen
5. Índice de la imagen dentro del grupo por posición

El identificador del grupo por posición se compone a su vez del Unix time del instante en que fue tomada la imagen original, y del índice del grupo dentro de la imagen, pues puede haber varios grupos simultáneos en la imagen (por ejemplo 3 grupos, uno por cada persona siendo detectada a la vez, en una reunión con 3 personas). Un ejemplo de nombre de archivo de imagen sería: "f_T1_1495109464388i0_2_1.jpg".

Por otra parte, existe un subprocesso encargado de enviar las caras al servidor. Cada 30 segundos se revisa si existen caras disponibles en la bandeja de salida, que pertenezcan a una de las cinco categorías definidas anteriormente. Si el envío es exitoso, se eliminan de la bandeja de salida y de la memoria secundaria por razones de privacidad y seguridad. Si por el contrario existe algún problema, las caras se mantienen en la bandeja de salida. Esto por ejemplo sirve en el caso de una desconexión a Internet, donde al momento en que se recupere la conexión, se enviarán todas las caras pendientes al servidor.

La Figura 8 muestra la interfaz de configuración del servicio en el Tablet PC. Esta interfaz permite definir VT-ID y API-URL. VT-ID es un nombre identificador del Tablet PC (como dispositivo) en particular. Esto es necesario considerando que el sistema soporta la existencia de múltiples Tablet PCs funcionando simultáneamente. API-URL es la URL donde se encuentra el servicio online de recepción de imágenes de caras, el cual luego dejará las imágenes disponibles para ser procesadas por el sistema en la nube. La interfaz también permite iniciar o terminar el funcionamiento del servicio en forma manual, así como ver la imagen de la cámara en tiempo real, entregando además feedback de sobre las detecciones de caras.

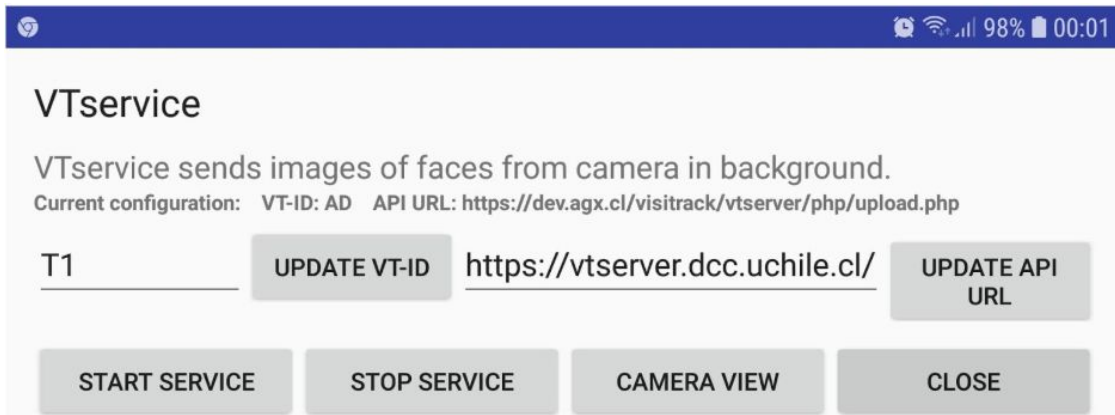


Figura 8. Interfaz de configuración del servicio VTService en el Tablet

3.4 Visitrack-Server

Visitrack-Server contiene un set de siete módulos corriendo en el servidor. Cada uno de ellos es independiente del resto, excepto por los datos en la base de datos que utilizan en común. La Figura 9 muestra la estructura de este componente y los principales módulos que conforman dicha solución. Luego, la Figura 10 resume el proceso general de detección de visitas usando imágenes de caras.

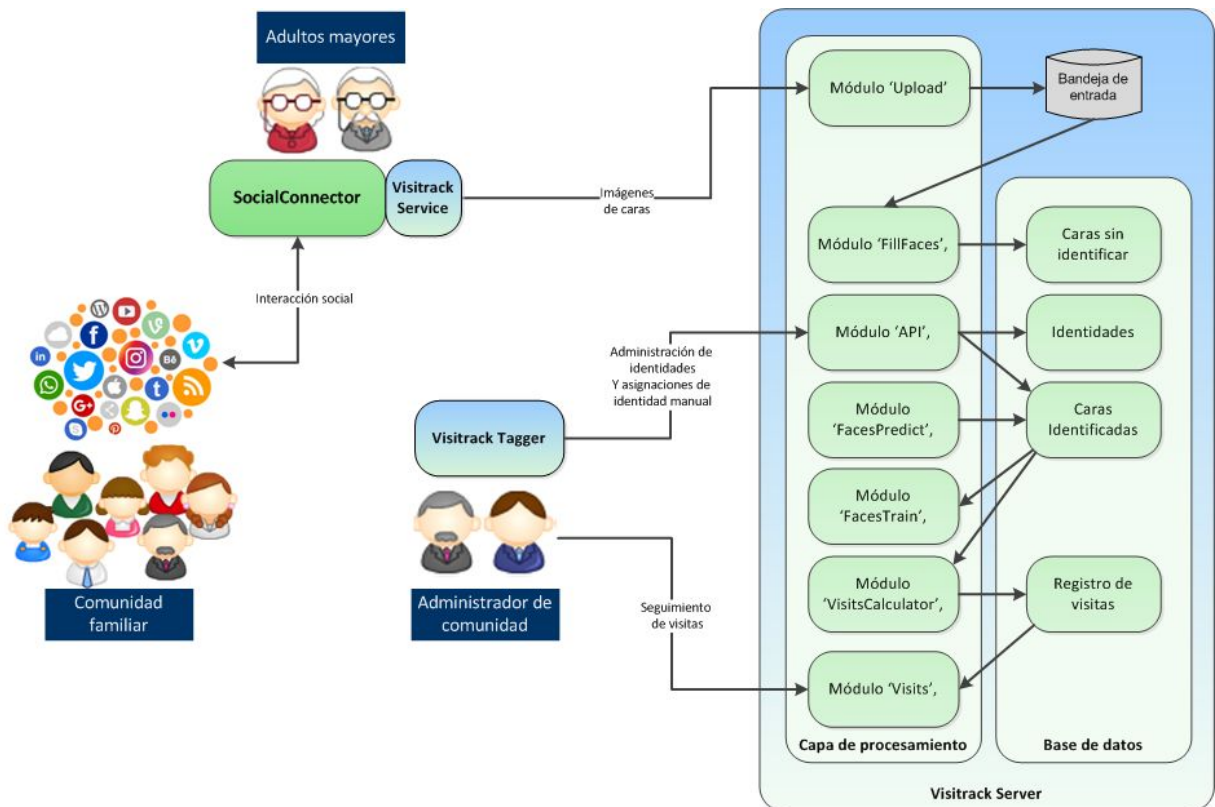


Figura 9. Diagrama de módulos de VTServer y sus interacciones

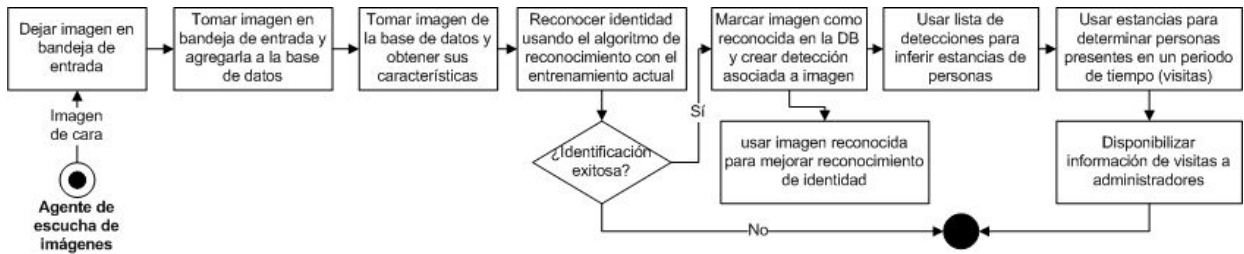


Figura 10. Proceso general de detección de visitas usando imágenes de caras

En las siguientes secciones se explican los distintos módulos mostrados en la Figura 9.

3.4.1 Upload

El módulo Upload está siempre esperando por el envío de imágenes de caras. Si es que recibe alguna, la deja en una carpeta que funciona como bandeja de entrada. La Figura 11 resume el proceso que lleva a cabo este módulo.

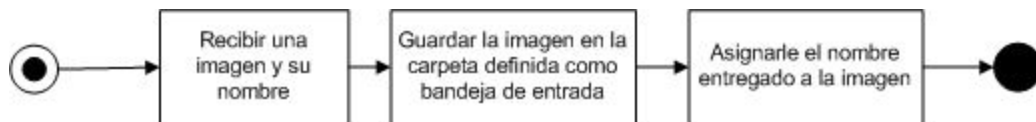


Figura 11. Proceso de upload.

3.4.2 Fill-Faces

Este módulo se encarga de tomar las caras desde la bandeja de entrada, para luego incluirlas en la base de datos del sistema (Mysql). Esto consiste en crear un registro nuevo en la tabla *faces*, por cada cara en la bandeja de entrada, considerando que cada imagen pertenece a un grupo específico (llamado grupo por posición), ya que son imágenes tomadas en forma continua en una misma área de la casa, y por lo tanto se asume que se trata de la misma persona. El módulo Fill-Faces también genera un registro nuevo en la tabla *face_groups* por cada uno de los grupos, por posición de las caras, en la bandeja de entrada. Además, asocia cada registro en la tabla *faces* con su correspondiente en la tabla *face_groups*. La identificación del grupo por posición de cada imagen de cara se obtiene analizando el nombre de la imagen, utilizando el formato explicado en la sección anterior.

Este proceso ocurre cada 15 minutos. La Figura 12 muestra el workflow que lleva a cabo del módulo Fill-Faces.

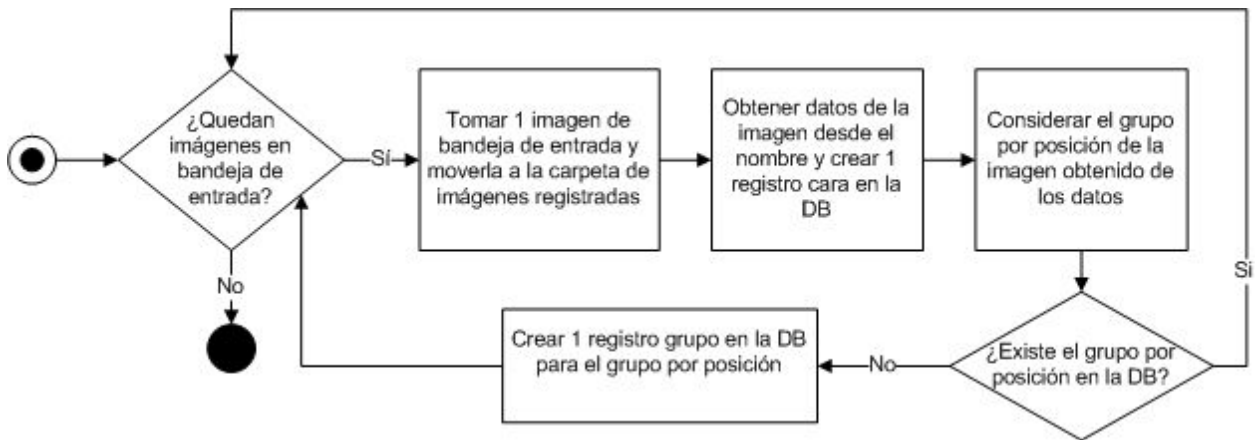


Figura 12. Workflow que lleva a cabo el módulo Fill-Faces.

Los datos ingresados en la tabla faces son los siguientes:

- **time:** tiempo en que se tomó la foto de la cara
- **source_id:** identificador del Tablet en el cual se tomó la foto de la cara
- **quality:** número identificador de la categoría de calidad de la foto
- **group_code:** identificador del grupo por posición de la cara.
- **url:** link hacia la imagen de la cara para utilizar por Visitrack-Tagger.

Los datos ingresados en la tabla face_groups son los siguientes:

- **time:** tiempo en que se tomó la primera foto del grupo
- **source_id:** identificador del Tablet en el cual se tomaron las fotos del grupo
- **quality:** número identificador de la categoría de calidad de la foto con mejor calidad del grupo
- **group_code:** identificador del grupo por posición de las caras.

3.4.3 Faces-Train

Este módulo se encarga de tomar todas las caras que han sido marcadas como reconocidas por el módulo Face-Predict o que hayan sido reconocidas manualmente en Visitrack-Tagger, luego utiliza LBPHFaceRecognizer de OpenCV [29], para crear un modelo de entrenamiento para utilizar para futuras detecciones automáticas, este se guarda en un archivo en memoria secundaria.

Este proceso se lleva a cabo cada 15 minutos. La Figura 13 muestra el workflow que lleva a cabo del módulo Faces-Train.

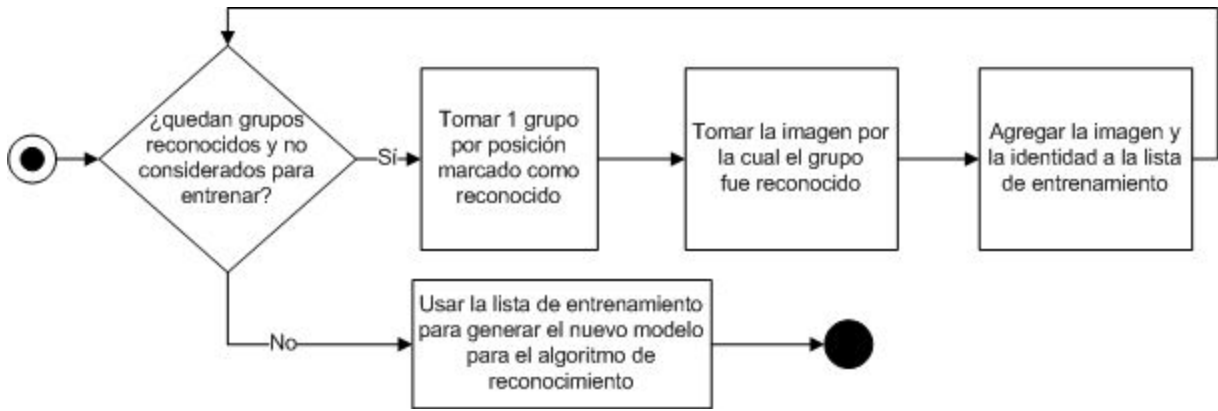


Figura 13. Workflow que lleva a cabo el módulo Faces-Train.

3.4.4 Faces-Predict

Este módulo se encarga de iterar sobre los elementos de la tabla `faces_groups` que no han sido identificados como reconocidos aún. Por cada uno de ellos, se vuelve a iterar sobre las caras que componen el grupo de posición en `faces_groups`, y éstas se ordenan desde las de mejor categoría de calidad a las de peor calidad. Luego se intentan reconocer automáticamente utilizando el algoritmo `LBPHFaceRecognizer` de OpenCV [29], entrenado con el modelo creado en Faces-Train. Si alguna cara es reconocida con una confianza suficientemente buena (bajo 100), se marca como reconocida, también se marca el elemento en `faces_groups` correspondiente para identificar al grupo como reconocido. Además, se crea un registro en la tabla “detections” para informar que existe un nuevo reconocimiento “persona-lugar-fecha”, usando el id de la persona detectada, el lugar donde está el Tablet PC, y la hora y día que se tomó la foto.

Este proceso ocurre cada 15 minutos. La Figura 14 muestra el workflow que lleva a cabo del módulo Faces-Predict.

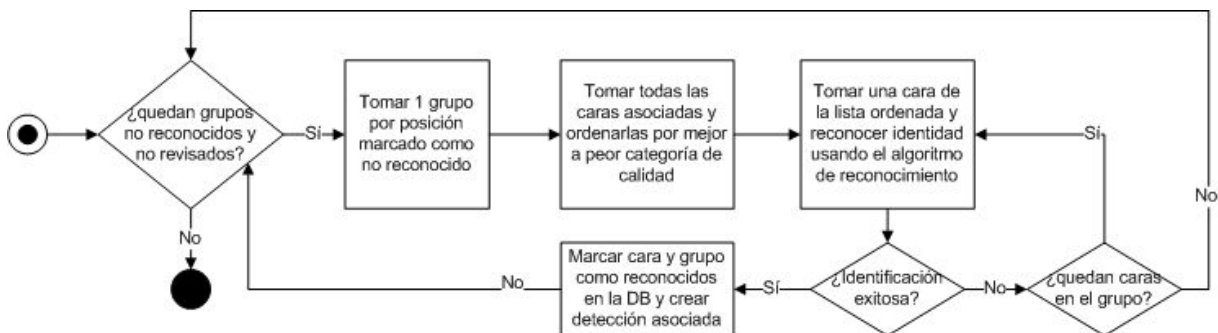


Figura 14. Workflow que lleva a cabo el módulo Faces-Predict.

3.4.5 Visits-Calculator

Este módulo se encarga de analizar la información en la tabla *detections*, que consiste en una lista ordenada cronológicamente de tríos “persona, lugar (Tablet PC) y fecha”, y determinar las visitas de personas al hogar del adulto mayor. Claramente, las visitas son las que se detectaron en el lugar donde está el Tablet PC, por lo que, el “lugar” en la tupla viene dado por la ubicación del Tablet que fue la fuente de las imágenes usadas para las detecciones. Determinar las visitas de forma precisa es un desafío para el sistema, dado que el Tablet PC sólo observa la parte de la casa, por lo tanto, es importante colocar el dispositivo en una posición estratégica, que tienda a maximizar las posibilidades de capturar el rostro de las visitas.

Para el cálculo de duración de visitas, el algoritmo de cálculo discretiza el tiempo, fraccionándolo en bloques de duración fija (por ejemplo, bloques de 5 minutos). Cada bloque es un intervalo de tiempo con una hora de inicio y otra de fin.

Por cada trío se identifica el bloque de tiempo cuyo intervalo contiene al tiempo de trío y se le asigna al bloque el id de persona del trío, este id de persona también es asignado al bloque de tiempo anterior y posterior del bloque mencionado anteriormente, por lo que cada detección resulta en el etiquetado de en 3 bloques consecutivos. Las etiquetas repetidas en un mismo bloque se descartan.

Como consecuencia del proceso anterior se obtiene una secuencia de bloques de tiempo, donde cada uno tiene asignado un conjunto de etiquetas de id de personas, el cual puede contener una o más etiquetas (todas distintas). En este último caso, se entiende que estuvo más de una persona a la vez en ese bloque de tiempo.

Luego se itera sobre los bloques de tiempo ordenados cronológicamente; en esta iteración se identifican rangos de visitas. Un rango de visita se define usando la secuencia de bloques consecutivos máxima (en cantidad de bloques) que contienen las mismas etiquetas en su conjunto de etiquetas. Este conjunto de etiquetas pasa a ser también el conjunto de etiquetas del rango.

El tiempo de inicio del rango se define con el tiempo de inicio del primer bloque de tiempo de la secuencia de bloques, y el tiempo de fin del rango se identifica con el tiempo del final del último bloque de la secuencia.

Para calcular los rangos de visitas se toma el primer bloque de la lista y se define como el bloque de inicio del rango a calcular. Luego se comprueba si el siguiente bloque contiene el mismo conjunto de etiquetas; de ser así, se incluye en el rango y se comprueba el siguiente bloque de la misma manera hasta que el conjunto de etiquetas sea distinto. En tal caso, el rango se da por finalizado y se da inicio a un nuevo rango comenzando con este último bloque. El proceso anterior se realiza hasta llegar al último bloque existente.

El resultado obtenido es una lista de rangos, y esto a su vez es una lista de intervalos de tiempo de diferentes duraciones que describen la duración de las visitas de las personas cuyas

etiquetas están en el conjunto de etiquetas del rango. Esta información se guarda en la tabla *visits*.

Este proceso ocurre cada 15 minutos. La Figura 15 muestra el workflow que lleva a cabo del módulo Visits-Calculator.

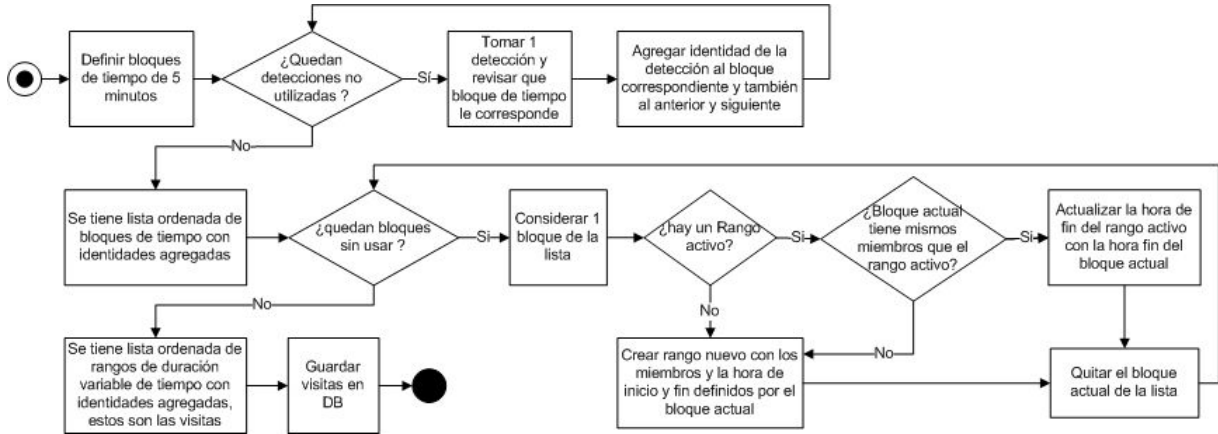


Figura 15. Workflow que lleva a cabo el módulo Visits-Calculator.

3.4.6 Visits

Este módulo es una vista web que permite revisar los resultados en la tabla *visits*, filtrando por persona y Tablet PC (o sea, fuente de datos). La Figura 16 muestra la interfaz de revisión de visitas.

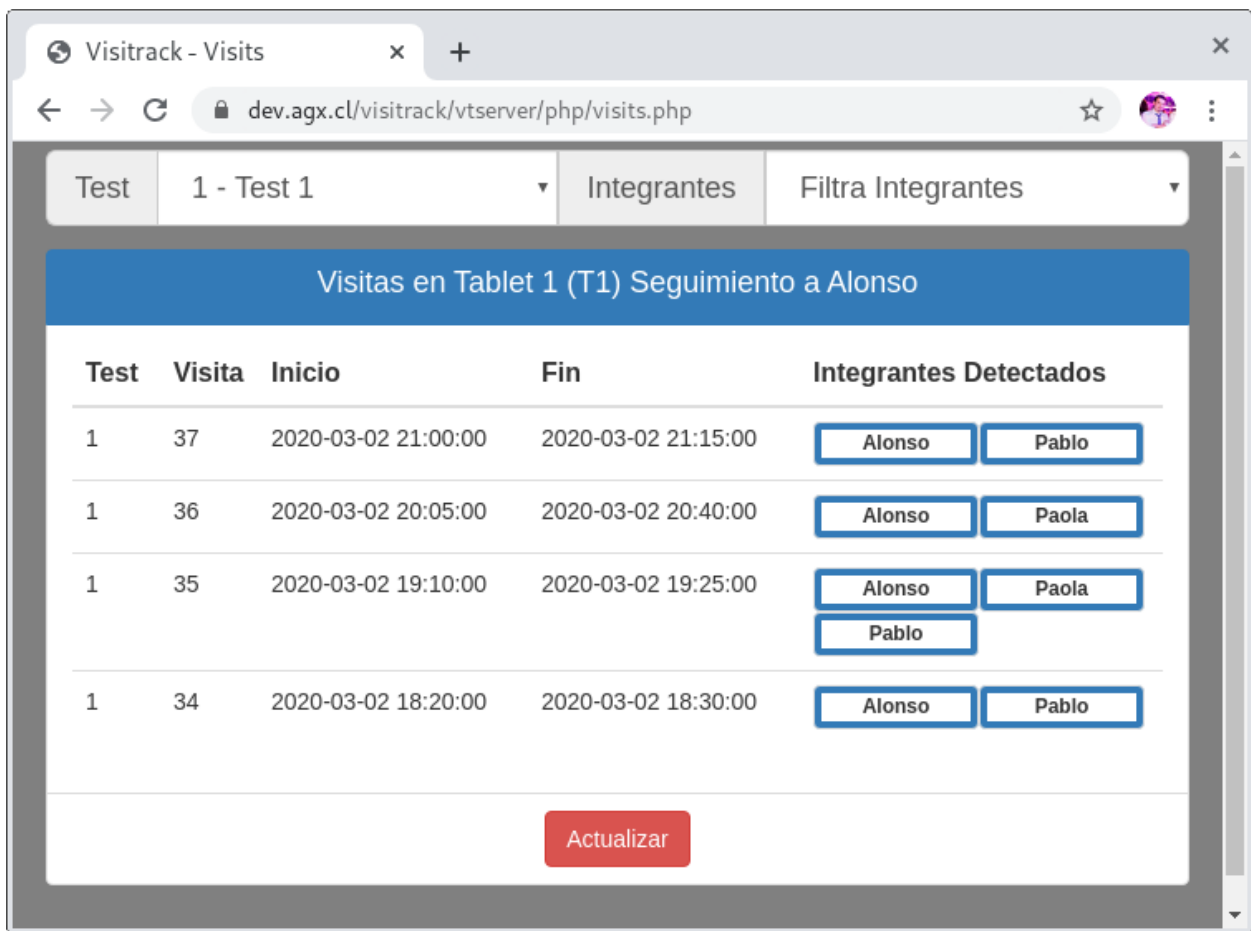


Figura 16. Interfaz de revisión de visitas

3.4.7 Application Programming Interface

Para que Visitrack-Tagger pueda interactuar con los datos del servidor se utiliza una pequeña API, creada como un servicio web desarrollado en PHP, la cual, en términos generales, tiene las siguientes funcionalidades:

- Entregar la lista de elementos (identidades), así como crear y eliminar elementos de la tabla identifications.
- Asignar y quitar identificaciones (etiquetas de identidad) en la tabla faces.
- Entregar la lista y eliminar elementos (caras) de la tabla faces.
- Disponibilizar a otros sistemas (fuera de este trabajo) los datos de visitas que ha calculado el sistema

Esta API está contenida en el archivo “api.php”, y recibe parámetros en la URL utilizando la forma: www.dominio.cl/api.php?parametro1=valor1¶metro2=valor2&... Para utilizar las funcionalidades de esta api, es necesario pasar como parámetro el nombre de la funcionalidad deseada (indicarlo en el parámetro “action”), y además agregar el resto de los parámetros propios de cada función, por ejemplo: www.dominio.cl/api.php?action=Identification_add&name=Juan%20Pérez

3.4.7.1 Funcionalidades relacionadas con Identidades

Estas funciones interactúan con la tabla “identifications” de la Base de datos. Esta tabla contiene todas las identidades conocidas por el sistema, las cuales sirven para ser usadas como tag de identidad en las imágenes de cara. Los atributos de esta tabla son “id” y “name”. Existen tres funciones en esta categoría, las cuales se explican a continuación:

- **Identification_add:** Recibe el parámetro “name” y lo usa para crear un registro en la tabla “identifications”, utilizando el parámetro “name” como nombre identificador de una identidad. En caso de que todo haya salido bien, retorna un string JSON con la siguiente forma:

```
{"code": 0, "data": []}
```

- **identification_remove:** Recibe el parámetro “identification_id” y lo busca en la tabla “identifications”. Si lo encuentra, entonces elimina el registro de dicha tabla. En caso de que todo haya salido bien, la función retorna un string JSON con la siguiente forma:

```
{"code": 0, "data": []}
```

- **identification_list:** Esta función no recibe parámetros, sino que se encarga de recuperar y devolver todos los registros en la tabla “identifications”. Para entregar el resultado esta función utiliza un string JSON con la siguiente forma:

```
{"data": [  
  {"id": "id 1", "name": "name 1"},  
  {"id": "id 2", "name": "name 2"},  
  ...  
  {"id": "id n", "name": "name n"}  
]}
```

3.4.7.2 Funcionalidades relacionadas con Caras

Estas funciones interactúan con la tabla “faces” de la base de datos. Esta tabla contiene información de todas las imágenes de caras enviadas al servidor. Existen dos funciones en esta categoría:

- **face_set_identity:** Ésta recibe el parámetro “identification_id”, que es un id de registro de la tabla “identifications”, y el parámetro “face_id_list”, que es una lista de ids de registros de la tabla “faces” separados por coma. Esta función “tagea” una foto con un identificador. Para ello asigna “identification_id” en el atributo “identification_id” de todos los registros de la tabla “faces”, cuyos ids estén en la lista “face_id_list”. En caso de que todo haya salido bien, retorna un string JSON con la siguiente forma:

```
{"code": 0, "data": []}
```

- **face_list:** Esta función recibe el parámetro “identification_id” (idem a la función anterior), y también recibe los parámetros “date_i” y “date_f” que representan una fecha inicial y una final correspondientemente. Estos últimos deben cumplir el formato 'AAAA-MM-DD HH:MM:SS'. La función *face_list* busca y devuelve todos los registros almacenados en la tabla “faces”, que han sido taggeados con la identificación “identification_id”, y cuya fecha de creación esté dentro del rango descrito por “date_i” y “date_f”. Para entregar el resultado esta función utiliza un string JSON con la siguiente forma:

```

{"data": [
  {"id": "id 1", "time": "time 1", "url": "url 1"},
  {"id": "id 2", "time": "time 2", "url": "url 1"},
  ...
  {"id": "id n", "time": "time n", "url": "url 1"},
]}

```

El atributo “URL” almacena la URL que permite ver o descargar la imagen del registro de la tabla “faces”.

3.4.7.2 Funcionalidades relacionadas con Visitas

Esta función interactúa con la tabla “visits” de la base de datos. Dicha tabla contiene información de todas las visitas calculadas por el sistema. Existe una función en esta categoría, la cual sólo puede ser ocupada por aplicaciones de terceros.

- **visits_list:** Esta función recibe el parámetro “start_date”, el cual debe tener la forma YYYY-MM-DD. Cualquier visita cuya fecha de inicio esté dentro de ese día, se listará en el resultado. Para entregar el resultado, esta función utiliza un string JSON con la siguiente forma:

```

{
  "code":"1",
  "data":[
    {
      "Id":"id 1",
      "start":"time start 1",
      "finish":"time finish 1",
      "participants": [identification 1-1, ... , identification 1-k, ...]
    },
    ...
    {
      "Id":"id n",
      "start":"time start n",
      "finish":"time finish n",
      "participants": [identification n-1, ... , identification n-k, ...]
    }
  ]
}

```

3.5 Visitrack-Tagger

Visitrack-Tagger es una aplicación de escritorio que sirve para que un administrador (típicamente, un miembro de la comunidad familiar) pueda definir, en forma manual, la identidad de la persona que aparece en una foto. Estas fotos corresponden a las caras que no pudieron ser reconocidas automáticamente por el sistema.

Este proceso siempre es necesario, por ejemplo, para las caras de personas que no se encuentran en la base de datos. Para agregar una nueva persona a la base de datos es necesario crear primero un identificador de identidad. Esto se realiza en esta misma aplicación, luego se puede asignar algunas caras a la identidad recién creada; esto es suficiente para que el sistema pueda a futuro reconocer automáticamente a esta persona. Esta aplicación también permite eliminar imágenes que erróneamente fueron detectadas (o asumidas) como caras por el Tablet PC. Para la interacción con los datos del servidor se utiliza la api web-service descrita en la sección anterior.

3.6 Modelo de datos

El modelo de datos relacional de Visitrack consta de siete entidades (Figura 17): *detections*, *face_groups*, *faces*, *identifications*, *sources*, *tests* y *visits*.

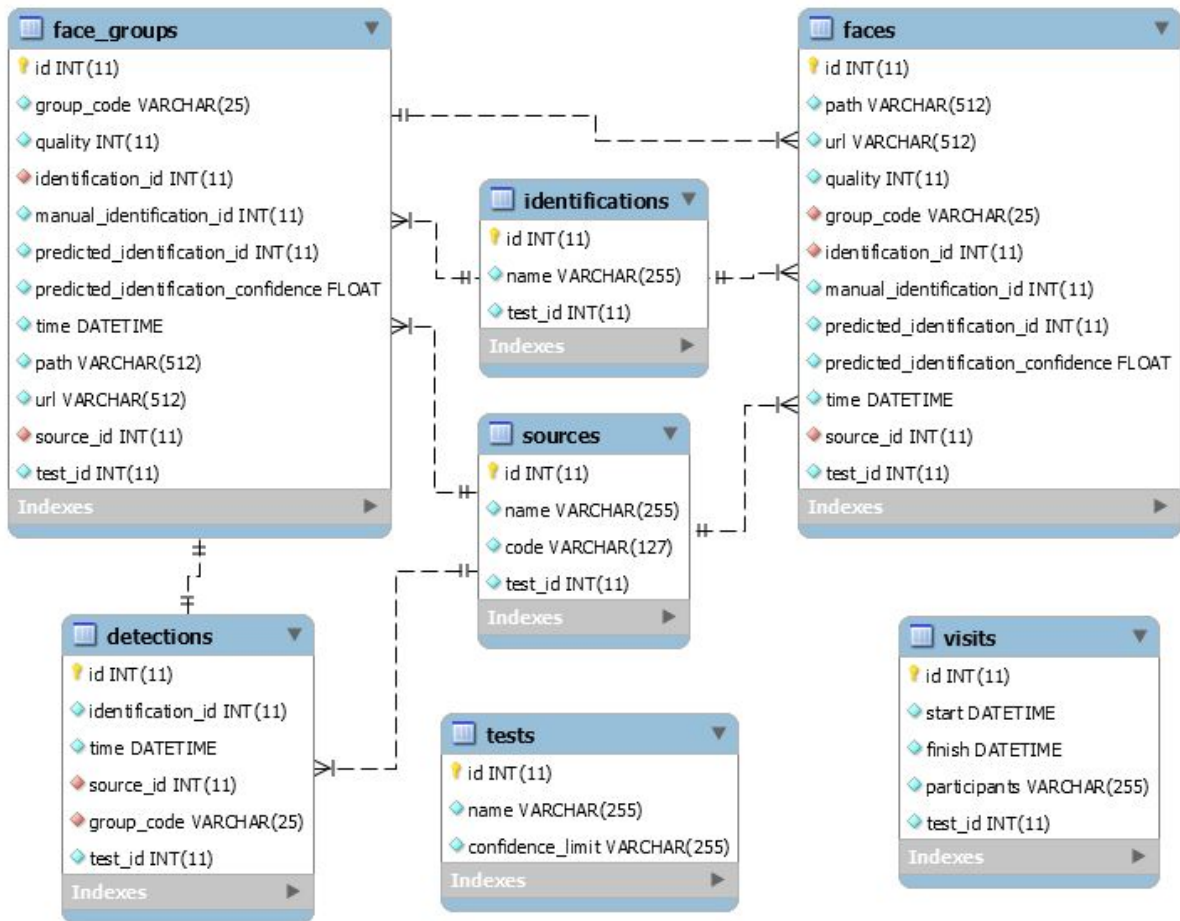


Figura 17. Modelo de datos del sistema

Descripción de los atributos de la entidad faces:

Path: ruta al archivo de imagen de la cara en el servidor.

url: link web al archivo de imagen de la cara para ser usado por Visitrak-Tagger.

quality: categoría de calidad de la cara.

group_code: identificador del “grupo por posición” al que pertenece la cara

identification_id: identidad de la cara aceptada como válida, obtenida ya sea automáticamente o manualmente.

manual_identification_id: identidad de la cara según el administrador usando Visitrak-Tagger (reconocimiento manual).

predicted_identification_id: identidad de la cara según el sistema (reconocimiento automático).

predicted_identification_confidence: nivel de confianza del reconocimiento de identidad automático.

time: fecha y hora en que se tomó la foto de la cara.

source_id: Tablet que tomó la foto de la cara.

Descripción de los atributos de la entidad face_groups:

group_code: identificador del grupo por posición al que pertenece el conjunto de caras.

quality: categoría de calidad de la cara con mejor calidad en el conjunto de caras.

identification_id: identidad de las caras en el conjunto de caras obtenido ya sea automáticamente o manualmente.

manual_identification_id: identidad de una cara del conjunto de caras según el administrador usando Visitrak-Tagger (reconocimiento manual).

predicted_identification_id: identidad de las caras en el conjunto de caras según el sistema (reconocimiento automático).

predicted_identification_confidence: nivel de confianza del reconocimiento de identidad de la cara usada para decidir la identidad automática del conjunto de caras.

time: fecha y hora en que se tomó la foto de la cara usada para decidir la identidad

path: la ruta al archivo de imagen de la cara usada para decidir la identidad del conjunto de caras en el servidor.

url: link web al archivo de imagen de la cara usada para decidir la identidad del conjunto de caras.

source_id: Tablet que tomó las fotos de las caras en el conjunto de caras.

Descripción de los atributos de la entidad detections:

identification_id: identidad de la persona.

time: fecha y hora en que se reconoció a la persona.

source_id: Tablet que identifica el lugar donde se reconoció a la persona.

group_code: identificador del “grupo por posición” al que pertenece la cara usada para decidir la identidad la persona.

Descripción de los atributos de la entidad identifications:

name: nombre de persona para identificar en el sistema.

Descripción de los atributos de la entidad sources:

name: nombre descriptivo del Tablet.

code: código para identificar el Tablet en Visitrack-Service.

Descripción de los atributos de la entidad visits:

start: fecha y hora en que comenzó la reunión.

finish: fecha y hora en que terminó la reunión.

participants: identificadores de los participantes de la reunión.

source_id: Tablet que identifica el lugar donde ocurrió la reunión.

Descripción de los atributos de la entidad tests:

name: nombre descriptivo del test.

confidence_limit: nivel de confianza límite para decidir si aceptar como válido el reconocimiento de identidad entregado por el sistema.

4 Implementación de la Solución

Para permitir la ejecución simultánea de múltiples pruebas se definió el concepto de “Test”. Un test es un conjunto de sources (Tablets) dispuestos en el mismo espacio. El cálculo de las visitas se realiza a nivel de test. Para esto se consideran las visitas reconocidas en todas las fuentes (sources) asociadas al test; así, entre más sources existan en el mismo test, mayor será la posibilidad de detectar a las personas que están en el espacio, debido a que se puede cubrir diferentes ángulos de visión. Es importante mencionar que cada source debe permanecer en un lugar fijo durante todo el test. Además, el entrenamiento de predicción de cada source se realiza con imágenes de la misma fuente, y no de las otras, pues el entrenamiento cruzado no funciona bien.

4.1 Algoritmo de Detección de Caras

Para la detección de caras se decidió utilizar el algoritmo Viola-Jones por varias razones. Primero por estar disponible en la librería OpenCV [29], también por su buena tasa de detección y por último por su bajo costo computacional, este último punto es muy valioso pensando en que este algoritmo debe correr en un Tablet PC. Este algoritmo se encuentra implementado en OpenCV con el nombre de CascadeClassifier, el cual es un clasificador de imágenes. OpenCV además provee distintos entrenamientos para este clasificador, de los cuales se usaron tres; para detectar caras, ojos y bocas. La Figura 18 muestra una representación gráfica del resultado del algoritmo de detección de caras, en el cual además de identificar una cara, se identifica también los ojos y la boca con el fin de determinar con mayor certeza qué se trata de un rostro.

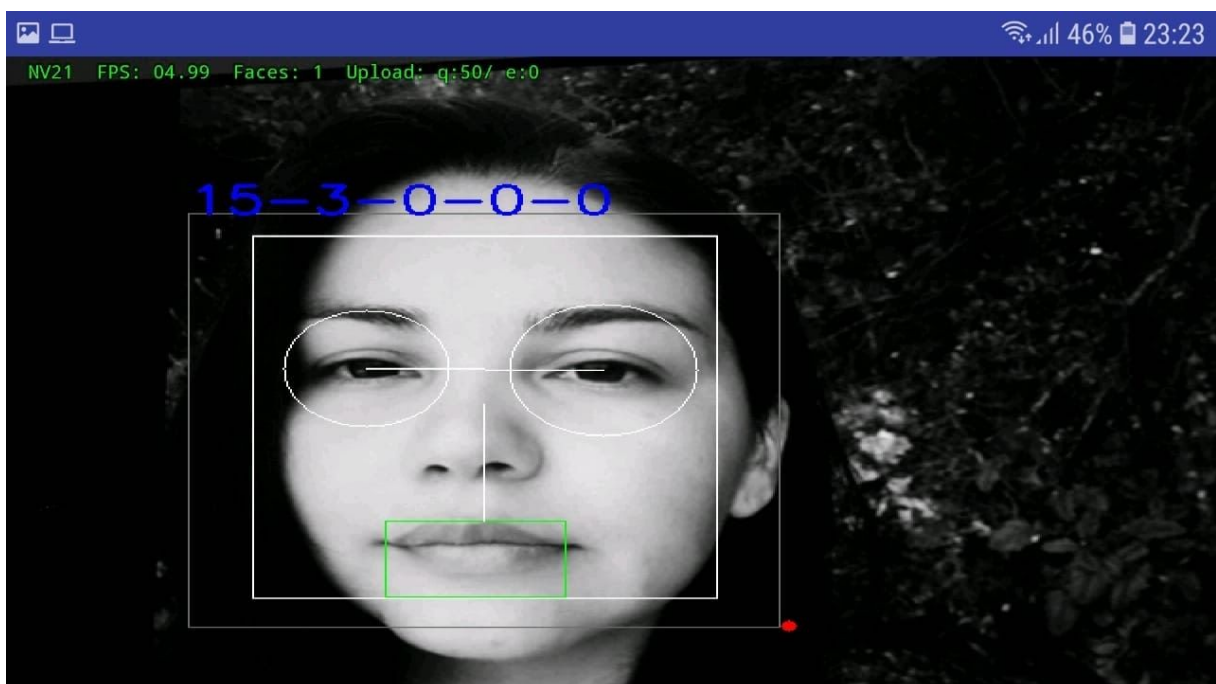


Figura 18. Visualización del resultado del algoritmo de detección de caras

4.2 Algoritmo Identificador de Visitas

Determinar las visitas que ocurrieron en un cierto intervalo de tiempo, es decir, lograr inferir desde qué y hasta qué hora estuvieron reunidas un conjunto de personas en un lugar determinado, es un gran desafío para el sistema. El desafío radica en que solo se tiene información puntual de presencia de personas, y a ésta la provee un Tablet PC que solo observa la parte de la casa en donde está ubicado el dispositivo. Por lo tanto, es importante situar el Tablet PC en una posición estratégica para mejorar las posibilidades de capturar a las visitas que ocurren en el hogar del adulto mayor.

A continuación se indican los pasos que sigue el sistema para intentar inferir estas visitas. Como información de entrada se tiene una lista de tuplas “persona-lugar-tiempo” que corresponden a datos confiables de presencia puntual de una persona en un lugar en un día y hora determinados, la figura 19 muestra un ejemplo.

```
Juan Pérez, living casa de Juan Pérez, 21-12-2019 14:44:05
David Pérez, living casa de Juan Pérez, 21-12-2019 15:05:23
Juan Pérez, living casa de Juan Pérez, 21-12-2019 15:12:18
José Pérez, living casa de Juan Pérez, 21-12-2019 15:12:27
...
Juan Pérez, living casa de Juan Pérez, 21-12-2019 18:12:78
David Pérez, living casa de Juan Pérez, 21-12-2019 18:22:78
```

Figura 19. Ejemplo de lista de tuplas input del algoritmo ordenadas cronológicamente

El siguiente paso consiste en discretizar todo el intervalo de tiempo a calcular, fraccionándolo en bloques de duración pequeña y fija (por ejemplo, de 5 minutos), así cada bloque es un pequeño intervalo de tiempo con una hora de inicio y una hora de fin, dando como resultado una lista ordenada de intervalos de tiempo pequeños (Figura 20).



Figura 20. Ejemplo de un día discretizado en bloques de tiempo de 5 minutos

Luego, la persona identificada en cada tupla del input, es asignada a su bloque de tiempo correspondiente. Esto corresponde a asignarla al bloque cuyo intervalo contiene al tiempo de la tupla a la que pertenece la persona; esta asignación se realiza etiquetando al bloque con el

identificador de la persona (Figura 21). La persona también es asignada al bloque de tiempo inmediatamente anterior y al inmediatamente posterior del bloque ya mencionado, por lo que por cada tupla en el input se realiza el etiquetado de tres bloques consecutivos. En el caso de que este proceso produzca una etiqueta repetida en un mismo bloque, dicha etiqueta se descarta.

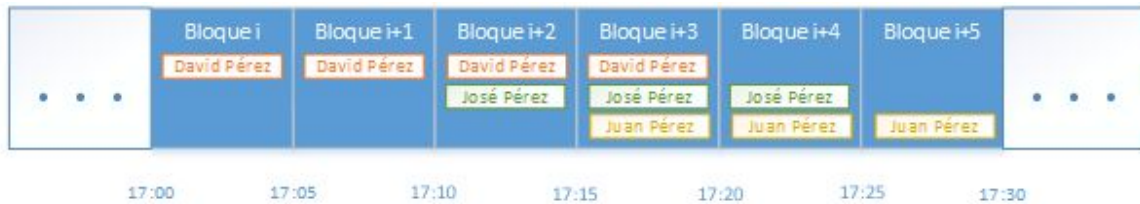


Figura 21. Ejemplo de bloques con etiquetas de personas

Del proceso anterior se obtiene una secuencia de bloques de tiempo, donde cada uno tiene asignado un conjunto de identificadores de personas, el cual puede contener una, más una o ninguna etiqueta. Esto quiere decir correspondientemente que estuvo una, más de una o ninguna persona en ese bloque de tiempo. Los bloques de tiempo sin etiquetas se descartan, el resto se mantiene sin perder el orden cronológico de los bloques.

Finalmente se calculan los llamados rangos de tiempo, que equivalen a los intervalos de tiempo de las visitas. Esto se realiza uniendo los bloques de tiempo consecutivos que contienen las mismas etiquetas (Figura 22). Así los rangos quedan formados como la concatenación de 1 o más bloques de tiempo consecutivos y mantienen un conjunto de etiquetas común. El tiempo de inicio del rango se define como el tiempo de inicio del primer bloque de tiempo usado en la secuencia de bloques que lo formó, y el tiempo de fin del rango se define como el tiempo del final del último bloque de la secuencia.



Figura 22. Ejemplo de unión de bloques en un rango

El resultado es una lista de rangos, y esto a su vez es una lista de intervalos de tiempo de diferentes duraciones que simulan la duración de las visitas, es decir, la duración de la reunión de las personas etiquetadas en el rango. La Figura 23 muestra un resumen de estos rangos.

```
...  
David Pérez, living casa Juan Pérez, 21-12-2019 17:00, 21-12-2019 17:10  
David Pérez;José Pérez;Juan Pérez, living casa Juan Pérez, 21-12-2019 17:10, 21-12-2019 17:25  
José Pérez, living casa Juan Pérez, 21-12-2019 17:25, 21-12-2019 17:30  
...
```

Figura 23. Ejemplo de una porción de la lista de visitas (output del algoritmo)

4.3 Visitrack-Tagger

La Figura 24 muestra la interfaz principal de Visitrack-Tagger. Arriba está el selector para escoger el Test en el cual se va a trabajar. Luego, se tiene 2 pestañas; la primera llamada “Etiquetar”, que contiene las imágenes nuevas que no se han taggeado manualmente ni se han reconocido en forma automática. En esta pestaña también aparece una lista de tags definidos para el test, junto con herramientas para definir nuevos tags o borrar alguno de los existentes.

En la segunda pestaña, llamada “Etiquetados”, aparecen las caras que ya han sido etiquetadas, sea de forma manual utilizando esta aplicación, o de forma automática a través del algoritmo predictivo.

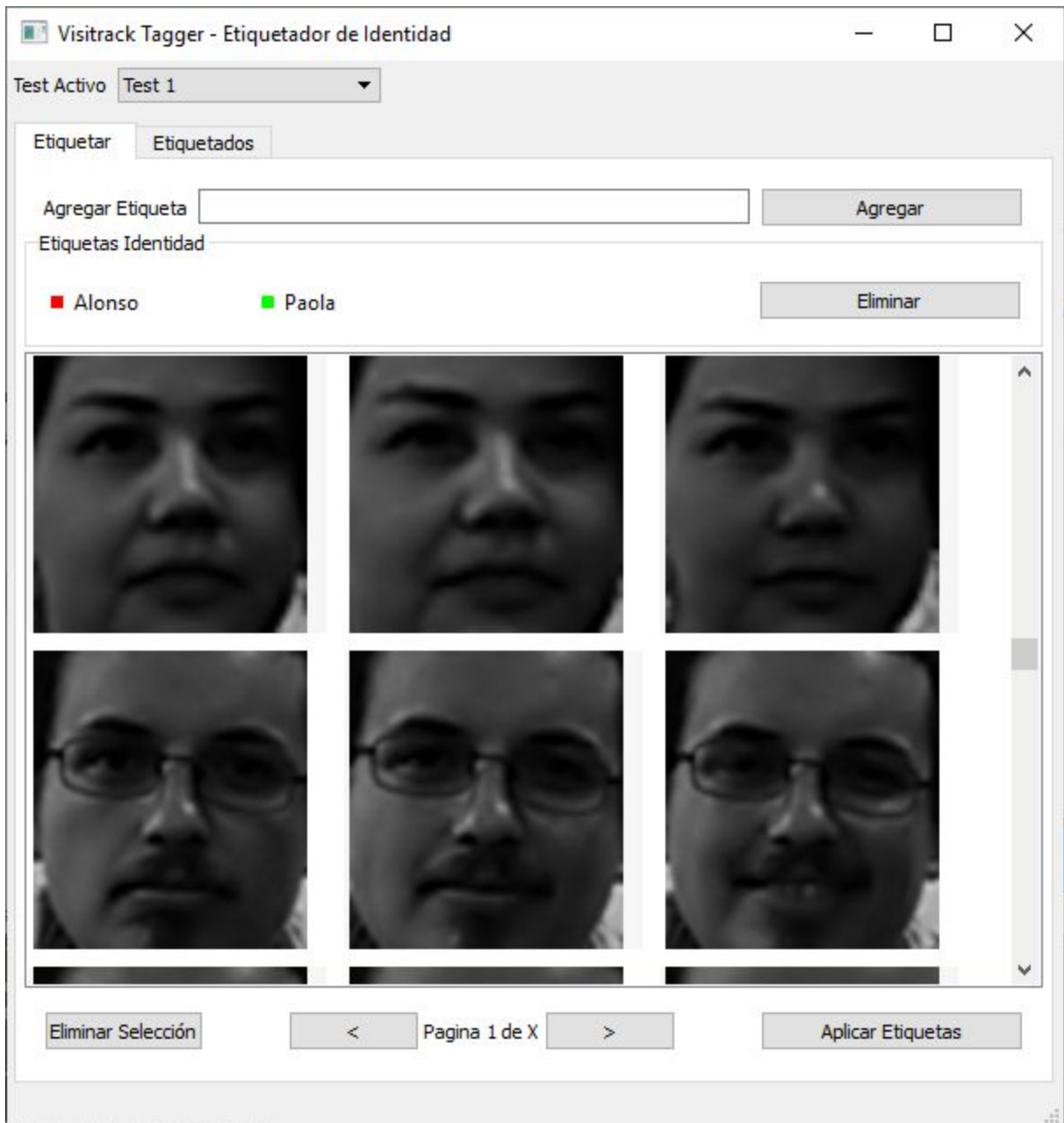


Figura 24. Interfaz principal del Visitrack-Tagger

Para etiquetar usando esta aplicación se debe seguir los siguientes pasos utilizando la interfaz de usuario del Visitrack-Tagger (Fig. 25):

1. Elegir el test correcto a trabajar.
2. Seleccionar la pestaña “Etiquetar”.
3. Si aún no existe la etiqueta, ésta se debe crear escribiendo un nombre para ella en el campo “agregar etiqueta”, y luego presionar el botón “Agregar” adyacente.
4. Seleccionar la etiqueta a usar.
5. Seleccionar una o más caras para etiquetar con la misma etiqueta.
6. Presionar el botón “aplicar etiquetas”.

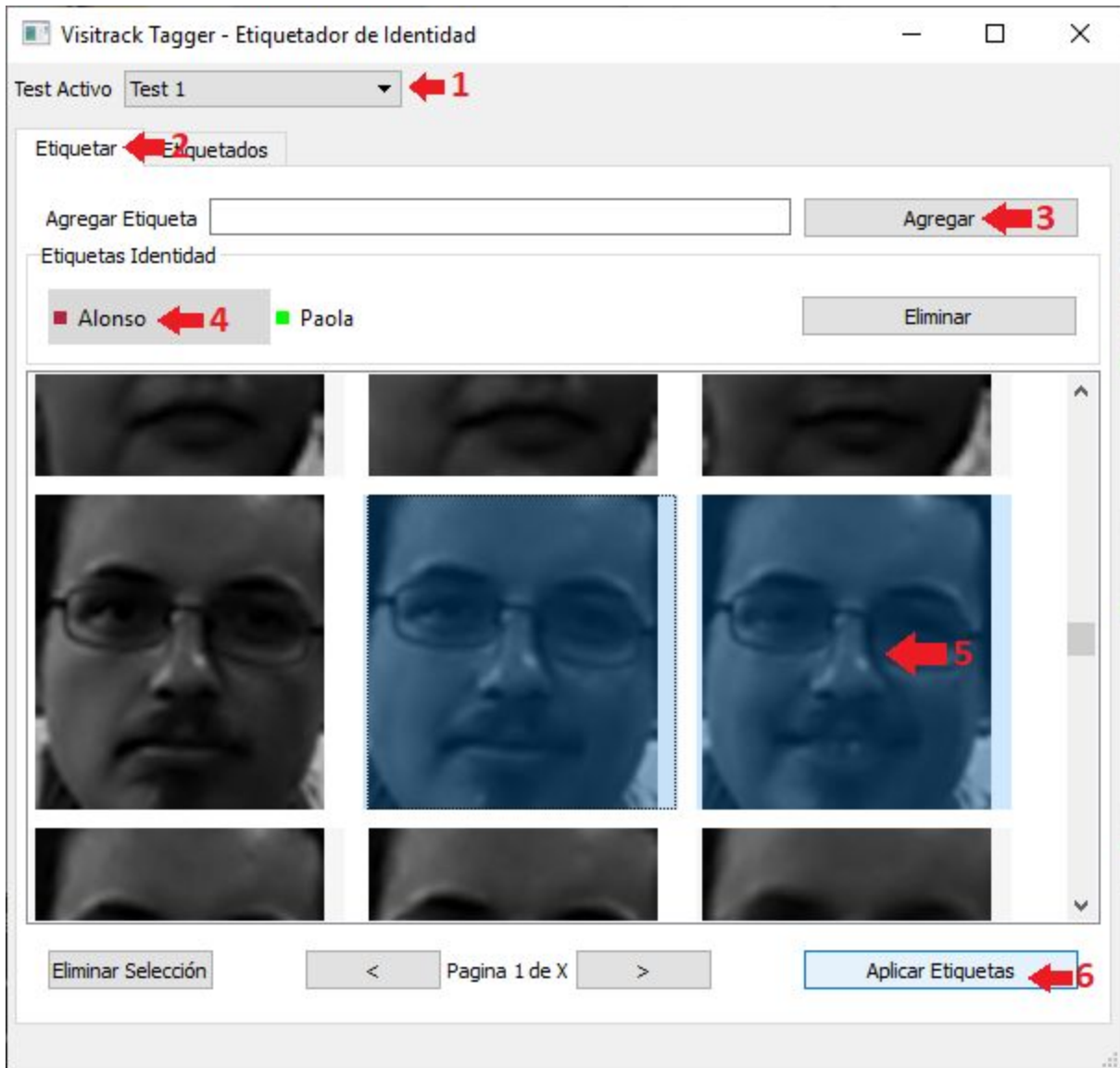


Figura 25. Interfaz de etiquetado

Luego de esto las imágenes etiquetadas desaparecen de esta pestaña, y se pueden encontrar en la pestaña “etiquetados”. En esta pestaña se puede seleccionar una etiqueta para filtrar por ella, y así ver todas las caras etiquetadas con la etiqueta seleccionada (Figura 26).

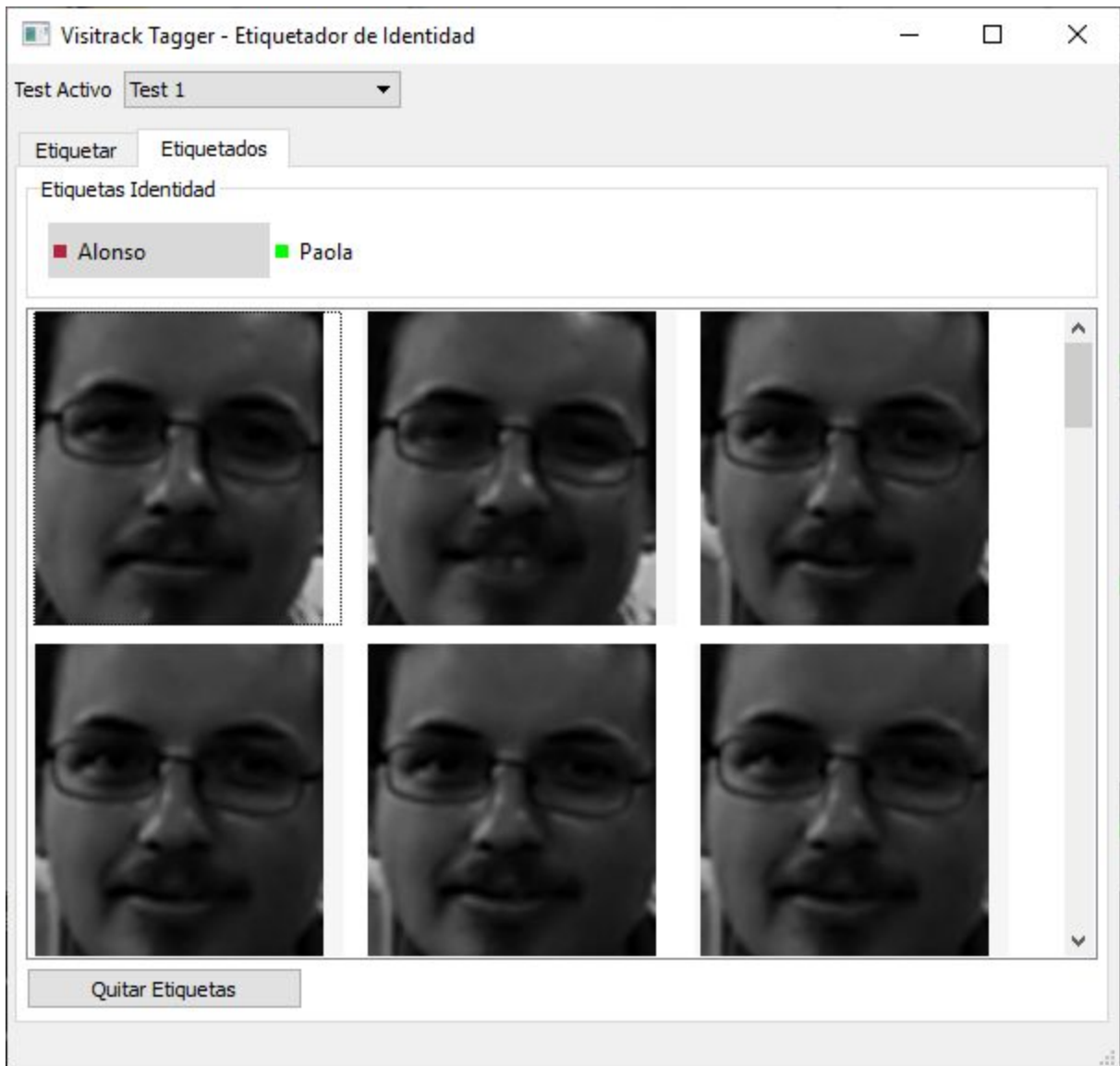


Figura 26. Imágenes asociadas a una cierta etiqueta

Si existe algún problema con una cara mal etiquetada, desde esta pestaña se puede quitar la etiqueta. Para esto se debe seleccionar la o las caras, y luego presionar “Quitar Etiquetas”.

Esta aplicación también permite eliminar imágenes donde el detector de caras falló, y detectó una cara donde no existe. Para esto se debe (Figura 27):

1. Seleccionar la pestaña “Etiquetar”.
2. Seleccionar una o más imágenes para eliminar.
3. Presionar “Eliminar Selección”.

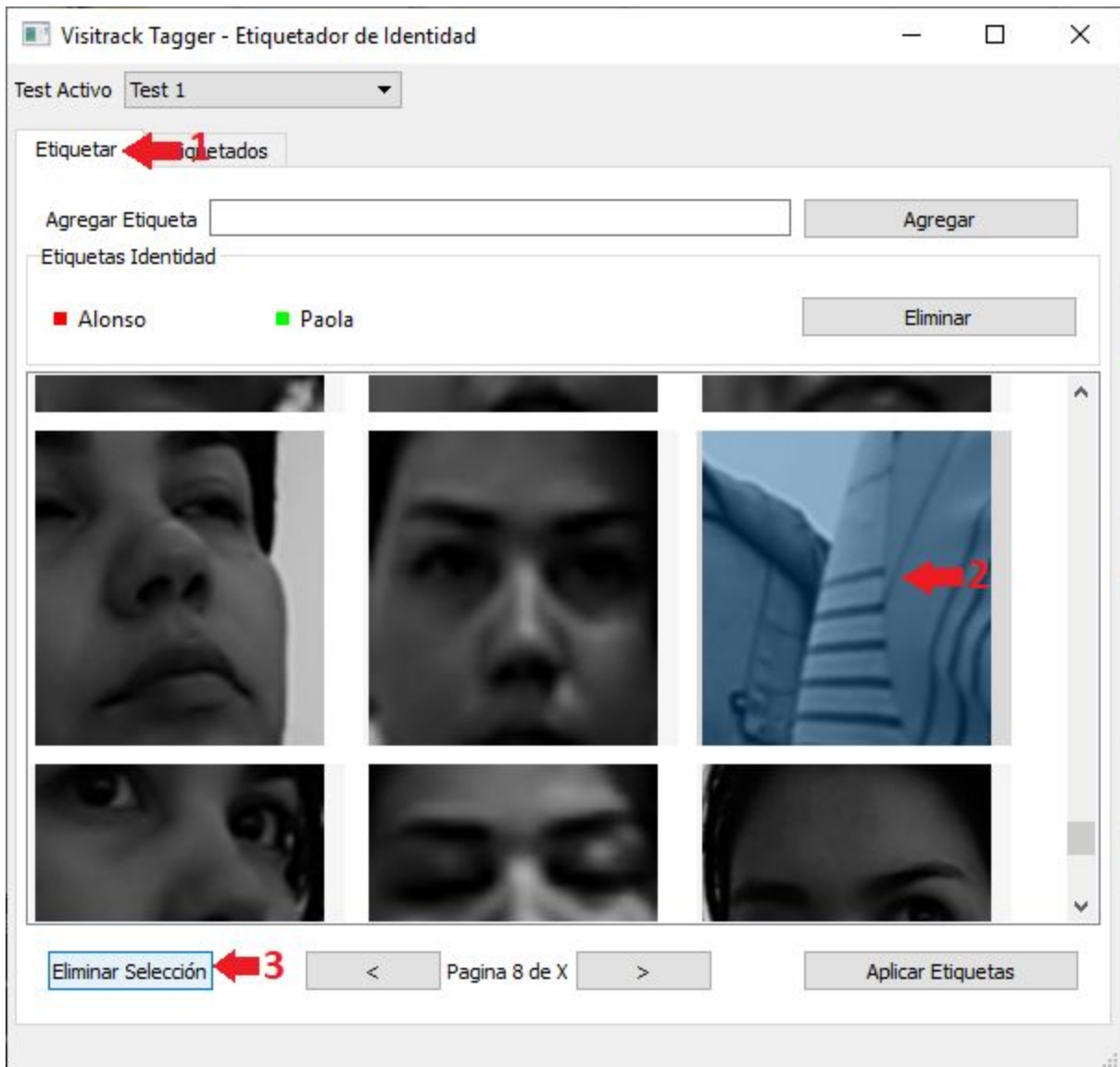


Figura 27. Eliminación de imágenes mal reconocidas por el sistema.

4.4 Integración del Servicio Visitrack al SocialConnector

La idea es que mientras el usuario utiliza el SocialConnector, al mismo tiempo el dispositivo esté sensando la actividad social a través del servicio Visitrack. Esto se logró de una manera muy simple y transparente; particularmente, al iniciar la aplicación SocialConnector, ésta revisa si el servicio de Visitrack está activo o no. Si no lo está, entonces lo levanta y así queda funcionando en background mientras el SocialConnector esté activo.

Técnicamente esto se pudo realizar de manera sencilla utilizando los Intents explícitos de Android, ya que el Visitrack service está definido como un servicio Android con el componente VTSERVICE definido público. Sólo un caso puntual requirió mayor cuidado; ese caso es cuando SocialConnector realiza una videollamada, pues ahí es necesario apagar momentáneamente el

servicio con el fin de liberar el uso de la cámara. Al terminar la videollamada el servicio se levanta otra vez.

5 Evaluación de la Solución

En este capítulo se explica la forma en que fueron evaluados el Visitrack y el Visitrack-Tagger, y los resultados obtenidos en cada una de estas evaluaciones.

5.1 Evaluación del Visitrack

A continuación se describe el procedimiento utilizado para determinar la efectividad del Visitrack, para identificar las visitas a un adulto mayor. Luego se presentan los resultados obtenidos.

5.1.1 Procedimiento de evaluación

Para evaluar la solución implementada en Visitrack se definió una lista de pruebas; todas ellas involucraron un escenario real, donde se dejó una Tablet PC con la aplicación funcionando dentro del living de una casa (supuestamente la de un adulto mayor). El Tablet PC estaba colocado en una posición fija, desde donde se podía detectar a los participantes de las visitas. En el lapso de algunas horas se simularon tres visitas. En cada una de éstas participaron las mismas dos personas. En paralelo se llevó un registro manual de cada visita, incluyendo las personas que participaron, y la hora de inicio y fin de cada una.

Para comenzar el proceso, se realizó una visita de entrenamiento (o visita 0):

- *Visita 0:* En esa visita, las imágenes tomadas para el entrenamiento del sistema fueron con la luz natural que entra a través de una ventana. Estas imágenes se usaron para entrenar el sistema en forma manual, usando Visitrack-Tagger. Se eligieron alrededor de 8 imágenes de cada persona, buscando la mayor variabilidad de ángulos posibles (posición de la cara de los participantes), y cambios de niveles de luz.

En algunas visitas se modificó una variable, para evaluar así distintas situaciones. A continuación se listan las visitas y sus respectivas variaciones:

- *Visita 1:* Se mantuvieron las condiciones de luz con las que se tomaron las imágenes de entrenamiento; esto es, imágenes con luz solar que entraba desde la ventana.
- *Visita 2:* Se cerró la cortina de la ventana para evitar la luz natural, y se usó luz artificial para iluminar el ambiente.

Al finalizar estas dos pruebas se revisaron las imágenes no reconocidas por el sistema, y se indicó la identidad de las personas usando el Visitrack-Tagger. Luego, se reentrenó el sistema con el nuevo conjunto de imágenes y se realizó la visita 3.

- *Visita 3:* Se realizó una visita que incluyó las condiciones de luz indicadas en los dos casos anteriores (ambas condiciones en la misma visita).

La ejecución de cada prueba involucró realizar varios pasos, los cuales se explican brevemente a continuación:

1. Dejar Visitrack-Server corriendo en la nube, a la espera de recibir imágenes de las caras para procesar.
2. Dejar Visitrack-Service corriendo en el Tablet PC que se está usando para las pruebas. El dispositivo debe quedar configurado para enviar datos al servidor Visitrack-Server, con nombre del Test "T1", y con el servicio Android corriendo y activo.
3. Ubicar el Tablet PC en el living de la casa, en una posición donde pueda detectar a las personas involucradas en la visita.
4. Se les pide a los dos participantes sentarse en el living para conversar por unos 5 minutos. Se utiliza Visitrack-Tagger para realizar el reconocimiento manual de las imágenes obtenidas. De esta forma, esas imágenes servirán de entrenamiento inicial (esto es la Visita 0).
5. Luego, se procede a realizar las reuniones (simuladas) ya definidas. Todas se llevaron a cabo en el mismo lugar, separadas por al menos 20 minutos entre cada una. Cada inicio y fin de visita se registró manualmente.
6. Finalmente, se revisó la interfaz web del Visitrack para ver cuáles fueron las visitas detectadas por el sistema, y la información asociada a cada una de ellas. Dicha información fue luego comparada con aquella registrada de forma manual.

5.1.2 Resultados obtenidos

A continuación se detallan los datos registrados por el sistema respecto a cada visita. En todos los casos participaron las mismas dos personas (Paola y Alonso):

Visita 1:

- Registro manual - Hora de inicio: 15:27, fin: 15:42
- Registro del sistema - Hora de inicio: 15:20, fin: 15:50
- Número de registros detections en base de datos de Alonso: 15, correctas: 100%
- Número de registros detections en base de datos de Paola: 20, correctas: 100%
- Número de registros detections totales en base de datos: 35, correctas: 100%

Visita 2:

- Registro manual - Hora de inicio: 16:32, fin: 16:54
- Registro sistema - Hora de inicio: 16:25, fin: 17:00
- Número de registros detections en base de datos de Alonso: 42, correctas: 71%
- Número de registros detections en base de datos de Paola: 26, correctas: 96%
- Número de registros detections totales en base de datos: 68, correctas: 81%

Visita 3:

- Registro manual - Hora de inicio: 17:41, fin: 17:48
- Registro sistema - Hora de inicio: 17:40, fin: 17:50
- Número de registros detections en base de datos de Alonso: 16, correctas: 100%
- Número de registros detections en base de datos de Paola: 12, correctas: 100%
- Número de registros detections totales en base de datos: 28, correctas: 100%



Figura 28. Diferencia de iluminación entre las imágenes de la visita 1 y la visita 2.

Todas las visitas realizadas fueron detectadas, y por cada visita existió un 100% de integrantes reconocidos. Sin embargo, la cantidad de reconocimientos de participantes varió ligeramente, dependiendo del nivel de luz y de entrenamiento del sistema. La detección de caras funcionó correctamente en todas las pruebas; esto es, la identificación de un rostro, aunque no necesariamente se sabe de quién se trata.

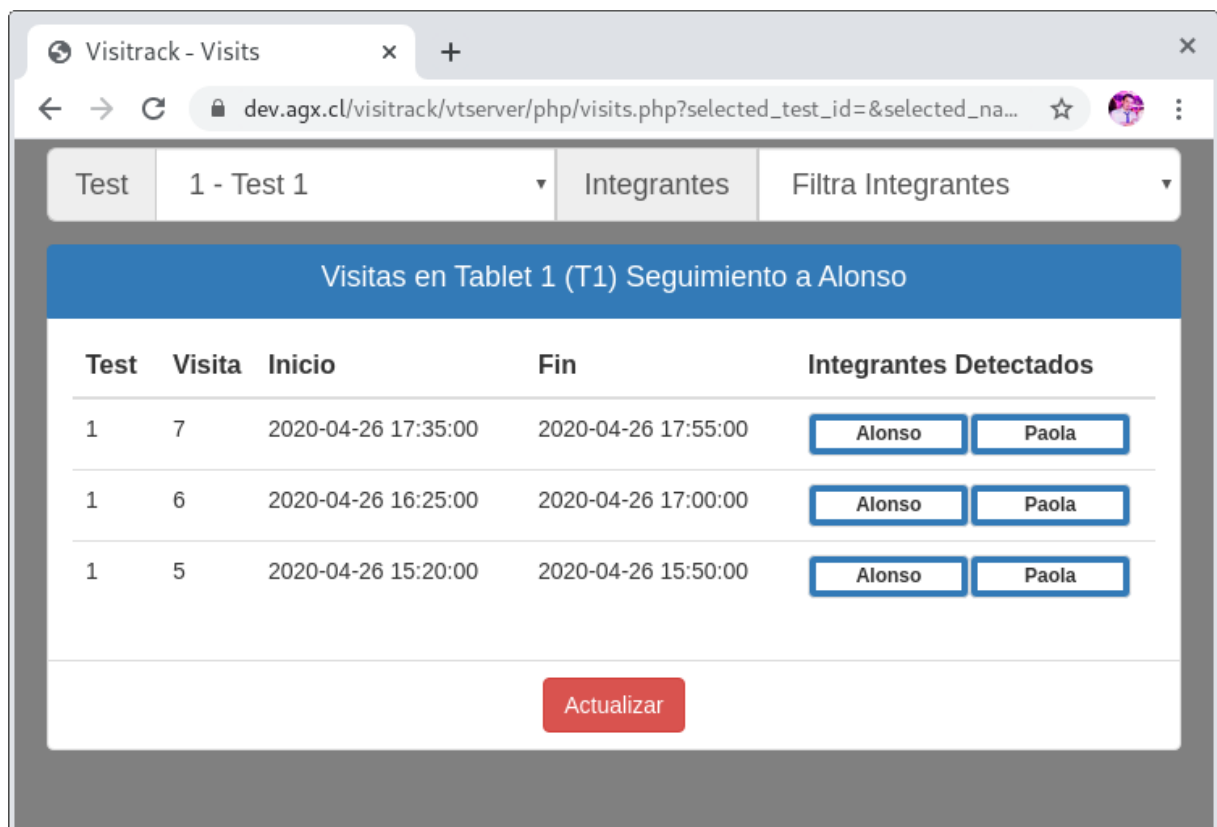
En la visita 1, el 100% de los integrantes fue reconocido, y de todas las detecciones de visitas (registros en tabla detections) realizadas el 100% fue correcto; esto se debió principalmente a que las condiciones de luz ambiente fueron ideales. Sin embargo, existe una diferencia en el tiempo de inicio y fin de las visitas entre lo registrado manualmente y el inferido por el sistema. Esto se debe a dos cosas principalmente; en primer lugar, el algoritmo de inferencia de visitas discretiza el tiempo a intervalos fijos (de 5 minutos). Esto impide reconocer visitas con una precisión mayor a 5 minutos.

Por otro lado, las personas que participan en las visitas no son detectadas en forma continua, sino sólo cuando miran de frente al Tablet PC (o apuntan su cara en esa dirección). Esta condición no ocurre con suficiente frecuencia como para no generar diferencias en los tiempos de inicio y fin de una visita. Sin embargo, estas situaciones sí ocurren con suficiente frecuencia como para que el sistema reconozca, al menos una vez, a cada uno de los participantes en una visita.

En la visita 2, el 100% de los integrantes fue reconocido a pesar de que las condiciones no eran ideales. De todas las detecciones de visitas realizadas, sólo un 81% de ellas fueron correctas, es decir, el 81% de las veces, la identidad de la imagen de cara fue reconocida

correctamente. En este caso, se obtuvo un conjunto de imágenes tomadas con luz artificial, mientras que el set de entrenamiento fueron imágenes tomadas con luz natural. Este resultado indica que, si bien el entrenamiento fue suficiente para reconocer a las personas de la visita a pesar de que el tipo de iluminación cambió, se ve una merma en la capacidad del sistema para reconocer personas puntualmente debido a este cambio en la iluminación del ambiente.

En la visita 3, el 100% de los integrantes fue reconocido, y de todas las detecciones de visitas realizadas, el 100% fue correcto. En este caso se utilizó Visitrack-Tagger para reentrenar el sistema en forma manual, usando las imágenes de personas no reconocidas de la visita 2. En esta prueba se observa la evidente mejora de la performance del reconocedor de rostros, debido al reentrenamiento de dicho mecanismo.



The screenshot shows a web browser window with the URL `dev.agx.cl/visitrack/vtserver/php/visits.php?selected_test_id=&selected_na...`. The interface includes a navigation bar with 'Test 1 - Test 1' and 'Integrantes' filters. The main content area is titled 'Visitas en Tablet 1 (T1) Seguimiento a Alonso' and contains a table with the following data:

Test	Visita	Inicio	Fin	Integrantes Detectados
1	7	2020-04-26 17:35:00	2020-04-26 17:55:00	Alonso Paola
1	6	2020-04-26 16:25:00	2020-04-26 17:00:00	Alonso Paola
1	5	2020-04-26 15:20:00	2020-04-26 15:50:00	Alonso Paola

Below the table is a red 'Actualizar' button.

Figura 29. Resultados de las pruebas mostrados a través de la interfaz web del Visitrack.

5.2 Evaluación del Visitrack-Tagger

A continuación se describe el procedimiento utilizado para determinar la usabilidad del Visitrack-Tagger, y los resultados obtenidos.

5.2.1 Procedimiento de evaluación

Para evaluar la usabilidad de la solución Visitrack-Tagger se definió una lista de pruebas, las cuales consisten a grandes rasgos en pedirle a un usuario que realice diferentes tareas. Estas tareas están asociadas al uso de los principales servicios que ofrece el Visitrack-Tagger. Una vez realizadas las tareas, y evaluado el resultado de las mismas, el usuario debe completar un formulario donde indica su percepción acerca de la usabilidad del sistema; para ello se utilizó una encuesta basada en SUS (System Usability Scale) [30]. Cinco personas participaron en esta evaluación.

Para iniciar las pruebas es necesario dejar instalado Visitrack-Tagger en el computador del usuario, y darle acceso a un set de imágenes previamente capturadas como resultado de una reunión de cinco personas (tres hombres y dos mujeres) usando Visitrack. Estas personas serán nombradas como (Fig. 30): María Paz, Paola, Alonso, Rodrigo, y Felipe.

Los evaluadores del sistema fueron María Paz, Paola, Pablo, Rodrigo, y Felipe. Cuatro de los cinco evaluadores (todos menos Pablo) fueron parte de la muestra de rostros que se solicitó taggear. Esto no es particularmente un problema, pues en un escenario real, los rostros a taggear por un usuario corresponden (con muy alta probabilidad) a familiares o personas cercanas al adulto mayor, y por lo tanto, son seguramente conocidas por el encargado (o sea, por quien coloca los tags a los rostros capturados por el sistema). Por lo tanto, la muestra de evaluadores escogida no escapa al escenario normal de uso de esta aplicación.

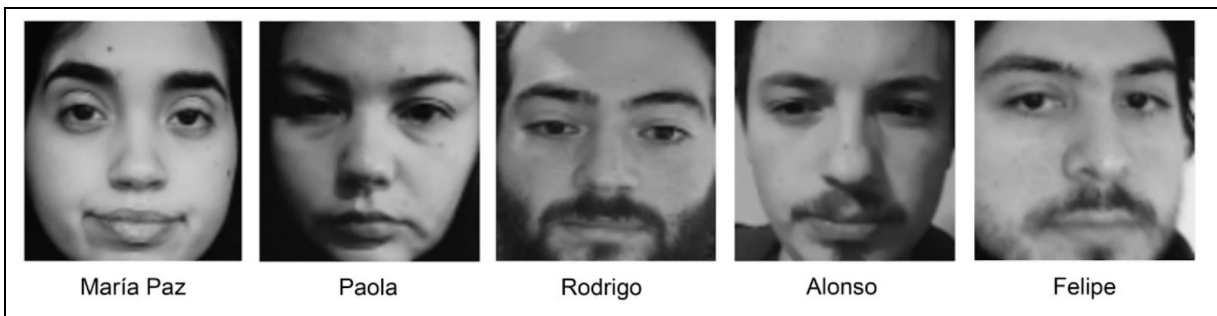


Figura 30. Las cinco personas presentadas para la evaluación del Visitrack-Tagger

Considerando lo antes dicho, a continuación se indica la lista de tareas asignada a los evaluadores:

1. Crear los cinco tags de identidad, con los nombres de los participantes en las visitas; es decir, María Paz, Paola, Alonso, Rodrigo, Felipe.
2. Asignar tags a distintas fotos de los visitantes, tomadas por el sistema:
 - a. Etiquetar como María Paz a 3 fotos de ella.
 - b. Etiquetar como Paola a 3 fotos de ella.
 - c. Etiquetar como Rodrigo a 3 fotos de él.
 - d. Etiquetar como Felipe a 3 fotos de él.
 - e. Etiquetar como Alonso a una foto de Alonso que está en la segunda página o en alguna posterior a esa. Las fotos con los rostros no reconocidos se muestran de forma paginada.
3. Asignar incorrectamente un tag (de forma intencional) y luego corregirlo.

- f. Etiquetar como Rodrigo a una foto de Alonso cualquiera.
 - g. Quitar la etiqueta de la foto anterior.
4. Borrar una foto.

Para cada una de estas pruebas se observó el desempeño del usuario evaluador, y se le brindó orientación sólo en el caso de que no lograra terminar una tarea por sí solo.

5.2.2 Instrumento de evaluación utilizado

Tal como se mencionó antes, para determinar el nivel de usabilidad de la aplicación Visitrack-Tagger, los evaluadores utilizaron el formulario SUS (System Usability Scale) [30], traducido al castellano. Este instrumento consta de 10 ítems (afirmaciones), que los evaluadores califican utilizando una escala Likert de 5 puntos (siendo 1 “totalmente en desacuerdo” y 5 “totalmente de acuerdo”). El cuestionario SUS utilizado incluyó los siguientes 10 ítems:

1. Creo que me gustaría utilizar esta app con frecuencia.
2. Encontré esta app innecesariamente compleja.
3. Pienso que esta app fue fácil de usar.
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta app.
5. Las funciones de esta app están bien integradas.
6. Creo que la app es muy inconsistente.
7. Imagino que la mayoría de la gente aprendería a usar esta app en forma muy rápida.
8. Encuentro que esta app es muy engorrosa de usar.
9. Me siento seguro al usar esta app.
10. Necesité aprender muchas cosas antes de aprender a usar esta app.

Promediando las respuestas de los participantes se obtiene una calificación final (o score), que determina el nivel de usabilidad de la aplicación. Normalmente el puntaje 68 indica si la usabilidad es aceptable o no, sobre 68 es aceptable y bajo 68 es deficiente. También, dependiendo del rango en el que se encuentre dicho score, se le puede asignar un adjetivo calificativo al nivel de usabilidad del software. La siguiente tabla muestra adjetivos calificativos en relación a diferentes score, según información presentada en [18].

Tabla 1. Relación entre adjetivos calificativos de usabilidad y score SUS [18].

Score SUS	Adjetivo calificativo
25	Muy mala
39	Mala
52	Aceptable
73	Buena
85	Muy buena
100	Excelente

5.2.3. Resultados obtenidos

Todos los evaluadores fueron capaces de completar las pruebas casi sin ayuda; hay que considerar que esta era la primera vez que ellos usaban el sistema. En general las personas necesitaron una explicación de alto nivel acerca del propósito de la aplicación, y un par de pruebas de uso del sistema para entender por sí solos cómo funcionaba éste. La siguiente Tabla muestra los resultados obtenidos en términos de usabilidad, según la opinión de los evaluadores.

Tabla 2. Resultados individuales en la encuesta SUS para Visitrack-Tagger

Evaluador	Score	Rango de usabilidad
María Paz	80	Buena - Muy buena
Pablo	90	Muy buena - Excelente
Paola	87,5	Muy buena - Excelente
Rodrigo	85	Buena - Muy buena
Felipe	87,5	Muy buena - Excelente

El promedio de los scores asignados por los participantes dio como resultado un valor de 86 sobre 100, que califica a la usabilidad del Visitrack-Tagger como “Muy buena” (queda en el rango “Muy buena - Excelente”, con una gran cercanía al score “Muy buena”). Sin embargo, durante las pruebas también se observaron aspectos de usabilidad que son susceptibles a mejora:

- A uno de los evaluadores se le hizo complicado borrar una etiqueta de identidad desde el tab “etiquetados”, ya que para realizar esa acción se necesita cambiar de tab.
- Dos evaluadores necesitaron instrucciones para revisar las fotos ya etiquetadas, pues solo presionaban el tab “etiquetados”, pero no elegían una etiqueta para ver.
- La semántica del botón “eliminar selección” no estaba clara. Un usuario pensó que servía para desmarcar la selección actual, en vez de eliminar las fotos seleccionadas.
- Un evaluador pensó que para aplicar las etiquetas podía seleccionar etiquetas y fotos en forma intercalada múltiples veces, y solo al final presionar “Aplicar Etiquetas”; esto con el fin de aplicar de una sola vez etiquetas distintas a varias fotos. El resultado fue que todas las fotos se asignaron sólo a la última etiqueta seleccionada.
- Cuando se realiza un etiquetado, un des-etiquetado, o una eliminación de imágenes, el único modo de saber si la acción ocurrió efectivamente, es observando si las imágenes seleccionadas desaparecieron de la lista de “imágenes a etiquetar”. Si el usuario no está atento a ese evento, la acción puede pasar desapercibida, por lo que no es una buena forma de dar a conocer que la acción ocurrió correctamente. En ese caso haría falta un recuadro de texto indicando que la acción ocurrió satisfactoriamente.

Es importante notar que la solución soporta selección múltiple de fotos. Esta funcionalidad se usa de la misma forma que se seleccionan múltiples archivos en una carpeta de Windows. Dicha capacidad del sistema no es visible a simple vista a través de la interfaz, por lo que si a un usuario que no se le ocurre explorar esta funcionalidad, no va a poder encontrarla. Sin embargo, en la práctica se observó que esta funcionalidad fue descubierta fácilmente por casi todos los usuarios.

A continuación se indican las mejoras propuestas por los evaluadores:

- Al etiquetar, hacer que el color de la selección de la cara sea igual al color de la etiqueta. Por ahora, el color que indica la selección realizada es siempre el mismo (azul, ver Figura 25), y eso puede confundir al usuario.
- Al agregar una etiqueta funcione presionar “Enter” y así no se necesite presionar el botón “Agregar”.
- Cambiar la glosa del botón “Eliminar Selección” a “Eliminar Fotos Seleccionadas”.

6 Conclusiones y Trabajo a Futuro

Esta memoria buscó desarrollar una solución no invasiva para monitorear la actividad social de los adultos mayores al interior de sus hogares. La motivación para esto nació de la observación de diferentes situaciones, que hacen que los adultos mayores sean proclives a sufrir aislamiento social. Para ello se propuso desarrollar un sistema que sea capaz de usar información de baja resolución (calidad), y que no sea muy invasivo.

El resultado fue el sistema *Visitrack*, que está compuesto de tres módulos o servicios principales: *Visitrack-Service*, *Visitrack-Server* y *Visitrack-Tagger*. *Visitrack-Service* se ejecuta en un Tablet PC que simula ser un portarretrato digital. Éste es el encargado de detectar, recortar y enviar imágenes de caras al *Visitrack-Server*. *Visitrack-Tagger* es una aplicación de escritorio que permite el reconocimiento y etiquetado manual de caras. *Visitrack-Server* corre en un servidor, y recibe el input del *Visitrack-Service* y del *Visitrack-Tagger*. Con esta información calcula las visitas ocurridas en el lugar donde se encuentra el Tablet PC, y las deja disponibles para ser revisadas por un usuario. La información de estas visitas también puede ser usada (por ejemplo, por un agente de software) para diagnosticar informalmente el nivel de actividad social de un adulto mayor, o para emitir notificaciones a los miembros de la comunidad familiar cuando sea requerido.

Para evaluar la solución desarrollada se realizó un set de pruebas. Éstas consistieron en dejar el Tablet PC funcionando dentro del living de una casa, tratando de identificar las reuniones que se realizaban al interior de la misma (simulando visitas). Estas visitas fueron registradas manualmente en un documento, y luego se comparó la información que entregó el sistema contra aquella registrada de forma manual. Como resultado se obtuvo que en las condiciones en las que se pretende usar este sistema, el 100% de visitas fueron detectadas, y la duración de ellas según el sistema, distó en menos de 10 minutos de la duración real.

Este trabajo demostró que es posible hacer seguimiento de la actividad social de un adulto mayor de forma no invasiva al interior de su hogar, utilizando un Tablet PC que simula ser un portarretrato digital. Además, los resultados mostraron que la performance del sistema mejora considerablemente en los casos en que las imágenes de entrenamiento son variadas en términos de las condiciones de luz y posición de las caras. Con relativamente pocas imágenes de caras (8 a 10 por persona), el sistema obtiene altas tasas de reconocimiento.

Otro punto importante es que el *Visitrack-Tagger* permite el taggeo fácil de rostros de los visitantes que no fueron reconocidos automáticamente. Por lo tanto, el sistema puede comenzar a operar con baja confiabilidad, pero en la medida que los usuarios administradores vayan taggeando caras de visitantes, y de esa manera se obtiene información adicional para reentrenar el sistema y volverlo más preciso. Esto permite no tener que enfrentarse a un esfuerzo importante para entrenar el sistema al momento del set up.

Para determinar el nivel de usabilidad de la aplicación Visitrack-Tagger, los evaluadores utilizaron el formulario SUS (System Usability Scale), esta evaluación califica a la usabilidad del Visitrack-Tagger como “Muy buena”. Es necesario considerar que esta prueba se realizó solo con 5 evaluadores, este número es bajo para sacar conclusiones generalizables, pero por alcance de la memoria no se extendió la prueba a más evaluadores.

Más allá de las pruebas reportadas en este documento, se hicieron algunas otras en donde se cambió la posición del Tablet PC, o la posición o tipo de la fuente de luz del ambiente monitoreado. Estos cambios impactaron negativamente en la performance del sistema, dado que las condiciones en las que éste fue entrenado distaban mucho de aquellas en las que se estaban monitoreando las visitas. Esas situaciones no deberían darse en la práctica (o sería raro que ocurrieran) ya que este sistema está pensado para ser instalado en un lugar fijo del hogar, y por lo tanto, debería enfrentarse a un set similar de condiciones lumínicas. Si bien este set contará con variantes debido a la hora del día o al clima imperante, las variantes se podrán apreciar en su totalidad en un cierto periodo de tiempo; periodo en el que se requerirá un uso intensivo de Visitrack-Tagger, con el fin de incluir al entrenamiento, las detecciones afectadas con todas las variantes lumínicas del set, y que el sistema no fue capaz de incluir en forma automática. Una vez cubierta la gama de variantes lumínicas, el sistema podrá funcionar de forma autónoma con muy baja necesidad de intervención humana.

Como parte del trabajo a futuro, se tiene lo siguiente:

- Actualmente el sistema está entregando una lista de visitas diarias, pero no está sacando ninguna conclusión a partir de esa información. Por lo tanto, se puede desarrollar un sistema que consuma esta información, la procese, y entregue alguna conclusión que permita tomar una acción orientada a mantener o mejorar la actividad social de adulto mayor.
- El sistema debería permitir funcionar con múltiples fuentes de imágenes (Tablet PCs) en forma simultánea desde una misma casa, para así mejorar la cantidad de detecciones y reconocimientos. Si bien esta característica ya está en parte soportada, falta algo fundamental para que funcione adecuadamente. Actualmente el sistema genera un solo set de entrenamiento por cada “Test”, es decir todos los Tablets asociados a un Test comparten el mismo set de caras de entrenamiento. En este sentido, el sistema debería generar un set de entrenamiento independiente por cada Tablet, ya que cada uno se enfrentará a condiciones lumínicas particulares.
- Sería bueno poder incluir a las personas detectadas, pero no reconocidas en el resultado de las visitas. El sistema actual solo muestra a las personas que fueron reconocidas de forma manual al menos una vez en la lista de visitas; es decir que exista al menos un tag manual que la identifique. En este sentido, el sistema debería ser capaz de incluir a las personas reconocidas, pero sin ningún tag manual. Esto viene de la mano con hacer que el sistema pueda reconocer personas, sin necesidad de que un humano genere un entrenamiento inicial. El reconocimiento de estos individuos involucra agregarle al sistema la capacidad de identificar al mismo rostro (a priori

desconocido) en diferentes momentos, logrando asegurar que se trata de la misma persona. A esta persona el sistema debería asignarle un tag identificador de forma interna, y luego dar la posibilidad de que un usuario le ponga un tag manual más adecuado; o sea, que indique el nombre de la persona u otra forma de reconocerla.

- Definir un grado (o nivel) de confiabilidad en la detección de una visita, utilizando como parámetro la cantidad de detecciones puntuales realizadas en la visita y la confiabilidad de esas detecciones.
- Por otra parte, para el diagnóstico del nivel de aislamiento social de los adultos mayores sería bueno considerar factores externos, como el clima o la época del año, pues por ejemplo, en períodos de lluvia o de frío es esperable que las visitas a los adultos mayores se reduzcan.

Bibliografía

- [1] Fernández, B., Soledad, M. Valenzuela, E., Chile y sus Mayores: Resultados Tercera Encuesta Nacional Calidad de Vida en la Vejez. Servicio Nacional del Adulto Mayor (SENAMA), (2013).
- [2] House, J. Social Isolation Kills, But How and Why? *Psychosomatic Medicine* 63, 273-274, (2001).
- [3] Karimi, A., Neustaedter, C. From high connectivity to social isolation: communication practices of older adults in the digital age. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, ACM, Seattle, Washington, USA, 127-130, (2012).
- [4] Windle, K., Francis, J., Coomber, C. Preventing loneliness and social isolation: interventions and outcomes. *SCIE Research briefing* 39, 1-15, (2011).
- [5] Diehl, M., Marsiske, M., Willis, S. Everyday competence and everyday problem solving in aging adults: Role of the physical and social context. In H.-W. Wahl, R. Scheidt, & P. Windley (Eds.). *Annual Review of Gerontology and Geriatrics: Vol. 23, Aging in context: Socio-physical environments*, New York, NY: Springer Publishing, 130-166, (2003).
- [6] Muñoz, D. Alineando los patrones de comunicación en miembros de una comunidad familiar. Tesis para optar al grado de Magíster en Ciencias, Mención Computación. Departamento de Ciencias de la Computación, Universidad de Chile (2014).
- [7] Muñoz, D., Gutierrez, F., Ochoa, S., Baloian, N. SocialConnector: A Ubiquitous System to Ease the Social Interaction Among Family Community Members. *Computer Systems Science & Engineering* 30(1) 57-68, (2015).
- [8] Viola, P., Jones, M. Rapid object detection using a boosted cascade of simple features. *Proc. of the Conference on Computer Vision and Pattern Recognition*, (2001).
- [9] Sohn, J., Kim, N.S., Sung, W. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, vol. 16(1) 1–3, (1999).
- [10] Cho, Y.D., Kondoz, A. Analysis and improvement of a statistical model-based voice activity detector. *IEEE Signal Processing Letters* 8(10) 276–278, (2001).
- [11] Gazor, S., Zhang, W. A soft voice activity detector based on a Laplacian-Gaussian model. *IEEE Trans. Speech Audio Processing* 11(5) 498–505, (2003).
- [12] Woo, K., Yang, T., Park, K., Lee, C. Robust voice activity detection algorithm for estimating noise spectrum. *Electronics Letters* 36(2) 180–181, (2000).
- [13] Chengalvarayan, R. Robust energy normalization using speech/non-speech discriminator for German connected digit recognition. *Proc. of EUROSPEECH'99*, (1999).
- [14] Marzinzik, M., Kollmeier, B. Speech pause detection for noise spectrum estimation by tracking power envelope dynamics. *IEEE Trans. Speech Audio Processing* 10(2) 109-118, (2002).
- [15] Ramírez, J., Górriz, J., Segura, J. Voice Activity Detection. *Fundamentals and Speech Recognition System Robustness*. In M. Grimm and K. Kroschel. *Robust Speech Recognition and Understanding*, 1-22, (2007).
- [16] Turk, M., Pentland, A. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3, 71–86, (1991).

- [17] Ahonen, T., Hadid, A., Pietikainen, M. Face Recognition with Local Binary Patterns. Proc. of Computer Vision - ECCV'04. LNCS 3021, 469–481, (2004).
- [18] Bangor, A., Kortum, P., Miller, J.: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, *Journal of Usability Studies* 4(3), 114-123. 2009.
- [19] Kinnunen, T., Li, H. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication* 52(1) 12–40, (2010).
- [20] Lindley, S.E., Harper, R., Sellen, A.: Desiring to be in touch in a changing communications landscape: attitudes of older adults. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*, 1693-1702, 2009.
- [21] Ministerio de Desarrollo Social. Gobierno de Chile: Chile y sus Mayores: Resultados de la Tercera Encuesta Nacional de Calidad de Vida en la Vejez (2013). URL: http://adultomayor.uc.cl/docs/Chile_y_sus_mayores_2013_Encuesta_de_Calidad_de_Vida.pdf. Última Visita: Agosto de 2020.
- [22] Gutierrez, F.J., Ochoa, S.F.: Mom, I do have a family!: attitudes, agreements, and expectations on the interaction with Chilean older adults. In: *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW'16)*, 1400-1409, 2016.
- [23] Gutierrez, F.J., Ochoa, S.F. It Takes at Least Two to Tango: Understanding the Cooperative Nature of Elderly Caregiving in Latin America. Proc. of the 20th ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW'17). Portland, OR, USA. February 25–March 1, 2017.
- [24] Nef, T., Ganea, R.L., Müri, R.M., Mosimann, U.P.: Social networking sites and older adults - a systematic review. *International Psychogeriatrics* 25(7):1041-1053, 2013.
- [25] Waycott, J., Vetere, F., Pedell, S., Kulik, L., Ozanne, E., Gruner, A., Downs, J.: Older adults as digital content producers. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*, 39-48, 2013.
- [26] Kennedy, T.L.M., Wellman, B.: The networked household. *Information, Communication & Society* 10(5):645-670, 2007.
- [27] Osorio-Parraguez, P.: Health and widowhood: meanings and experience of elderly women in Chile. *Health* 5(8):1272-1276, 2013.
- [28] Osorio-Parraguez, P., Seguel, A.G.: Social construction of dependence in elderly men in Chile. *Health* 6:998-1003, 2014.
- [29] OpenCV. OpenCV 4.2. URL: <https://opencv.org/releases/>. Última visita: 23 de Julio de 2020.
- [30] Brooke, J.: SUS: a "quick and dirty" usability scale. *Usability evaluation in industry* (P. W. Jordan, B. Thomas, B. A. Weerdmeester, I. L. McClelland):189-194, 1996
- [31] World Health Organization: Constitution of the World Health Organization, 2006. URL: http://www.who.int/governance/eb/who_constitution_en.pdf. Última Visita: Agosto de 2020.
- [32] World Health Organization Europe - Dr Lynne Friedli: Mental health, resilience and inequalities, 2009. Recuperado el 23 de Julio de 2020, desde: http://www.euro.who.int/__data/assets/pdf_file/0012/100821/E92227.pdf
- [33] House, J.S., Landis, K.R., Umberson, D.: Social relationships and health. *Science* 241(4865):540–545, 1988.
- [34] Hous01] House, J.S.: Social isolation kills, but how and why?. *Psychosomatic Medicine* 63(2):273-274, 2001.

- [35] Umberson, D., Montez, J.K.: Social relationships and health: a flashpoint for health policy. *Journal of Health and Social Behavior* 51:S54-S66, 2010.
- [36] Alidoust, S., Holden, G., Bosman, C.: Urban environment and social health of the elderly: a critical discussion on physical, social and policy environments. *Athens Journal of Health* 1(3):169-180, 2014.
- [37] Servicio Nacional del Adulto Mayor (SENAMA). Gobierno de Chile: Participación y Vejez. Recuperado el 23 de Julio de 2020, desde: <http://www.senama.cl/EnvejecimientoAct.html>
- [38] González, F., Massad, C., Lavanderos, F., Albala, C., Sánchez, H., Fuentes, A., Lera, L., Cea, X., Salas, F., Bravo, D., Castillo, E., Quijada, S.: Estudio Nacional de la Dependencia en las Personas Mayores. Servicio Nacional del Adulto Mayor (SENAMA), Instituto de Nutrición y Tecnología de los Alimentos (INTA), y Centro de Microdatos - Universidad de Chile, 2010. Recuperado el 23 de Julio de 2020, desde: http://www.senama.cl/filesapp/Estudio_dependencia.pdf
- [39] Tao, H., McRoy, S.: Caring for and keeping the elderly in their homes. *Chinese Nursing Research* 2:31-34, 2015.
- [40] Barros, C., Fernández, M.B., Herrera, M.S.: Nociones sobre la familia y las interrelaciones entre sus miembros de parte de los mayores de 45 años. *Psicoperspectivas* 13(1):121-130, 2014.
- [41] Neustaedter, C., Harrison, S., Sellen, A. (Eds.). *Connecting Families: The Impact of New Communication Technologies on Domestic Life*. Springer-Verlag, London, 2013.
- [42] Lindley, S.E., Harper, R., Sellen, A.: Designing for elders: exploring the complexity of relationships in later life. In: *Proceedings of the British HCI Group Annual Conference on HCI (BCS-HCI'08)*, 77-86, 2008.

Anexo y Apéndices

Anexo A

A continuación se presenta la guía entregada a cada participante durante la evaluación de usabilidad de la aplicación Visitrack-Tagger.

Evaluación del Visitrack-Tagger

Visitrack-Tagger es una aplicación de escritorio que sirve para que un administrador pueda definir, en forma manual, la identidad de la persona que aparece en una foto.

Instalación de app

- Para iniciar las pruebas es necesario dejar instalado Visitrack-Tagger en el computador del usuario, para esto debe descargar este archivo y descomprimirlo: <https://tags.cd/vti.zip>
- Luego instalar los requisitos del software y abrir App.py con python3, para windows <https://python-pasa-impaciencia.blogspot.com/2017/02/instalar-python-paso-paso.html>
- En la app ya funcionando se debe seleccionar "Test activo": "Test 1" en el selector de arriba

Tareas a evaluación

Estas cuatro personas son presentadas para dadas por conocidas:



Se pide realizar las siguientes tareas, por cada una de estas tareas se observará el desempeño del usuario, y se le brindará orientación sólo en el caso de que no logre terminar una tarea por sí solo.

Debe avisar al finalizar cada una de las 4 tareas para revisar la correcta ejecución.

1. Crear los 5 tags de identidad, con los nombres de los participantes en las visitas: es decir, Maria Paz, Paola, Rodrigo, Alonso, Felipe.
2. Asignar tags a distintas fotos de los visitantes tomadas por el sistema:
 - a. Etiquetar como Maria Paz a 3 fotos de ella.
 - b. Etiquetar como Paola a 3 fotos de ella.
 - c. Etiquetar como Rodrigo a 3 fotos de él.
 - d. Etiquetar como Felipe a 3 fotos de él.
 - e. Etiquetar como Alonso a una foto de Alonso que está en la segunda página o en alguna posterior a esa. Las fotos con los rostros no reconocidos se muestran de forma paginada.
3. Asignar incorrectamente un tag (de forma intencional) y luego corregirlo.
 - f. Etiquetar como Rodrigo a una foto de Alonso cualquiera.
 - g. Cambiar la etiqueta de la foto anterior
4. Borrar una foto.

Encuesta

Al terminar se debe contestar esta encuesta: <https://forms.gle/foLmou6GtoAMuArK6>

Anexo B

A continuación se presenta un extracto de encuesta SUS entregada a cada participante en la prueba de usabilidad del Visitrack-Tagger.

SUS para Vistrack Tagger

System Usability Scale para Visitrack Tagger

Instrucciones: para cada una de las siguientes afirmaciones, marque una casilla que mejor describa sus reacciones frente a la app Visitrack Tagger.

***Obligatorio**

Nombre de encuestado *

Tu respuesta _____

1. Creo que me gustaría utilizar esta app con frecuencia. *

1 2 3 4 5

Muy en desacuerdo Muy de acuerdo

2. Encontré esta app innecesariamente compleja. *

1 2 3 4 5

Muy en desacuerdo Muy de acuerdo

3. Pienso que esta app fue fácil de usar. *

1 2 3 4 5

Muy en desacuerdo Muy de acuerdo

Anexo C

A continuación se presenta la estructura de la base de datos que utiliza el Visitrack-Server.

```
-----  
-- Dump Database: visitrk_db  
-----  
  
CREATE TABLE `detections` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `identification_id` int(11) NOT NULL,  
  `time` datetime NOT NULL,  
  `source_id` int(11) NOT NULL,  
  `group_code` varchar(25) COLLATE utf8_spanish2_ci NOT NULL,  
  `test_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)  
  
CREATE TABLE `face_groups` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `group_code` varchar(25) COLLATE utf8_spanish2_ci NOT NULL,  
  `quality` int(11) NOT NULL,  
  `identification_id` int(11) NOT NULL DEFAULT '-1',  
  `manual_identification_id` int(11) NOT NULL DEFAULT '-1',  
  `predicted_identification_id` int(11) NOT NULL DEFAULT '-1',  
  `predicted_identification_confidence` float NOT NULL DEFAULT '-1',  
  `time` datetime NOT NULL,  
  `path` varchar(512) COLLATE utf8_spanish2_ci NOT NULL,  
  `url` varchar(512) COLLATE utf8_spanish2_ci NOT NULL,  
  `source_id` int(11) NOT NULL,  
  `test_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)  
  
CREATE TABLE `faces` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `path` varchar(512) COLLATE utf8_spanish2_ci NOT NULL,  
  `url` varchar(512) COLLATE utf8_spanish2_ci NOT NULL,  
  `quality` int(11) NOT NULL,  
  `group_code` varchar(25) COLLATE utf8_spanish2_ci NOT NULL,  
  `identification_id` int(11) NOT NULL DEFAULT '-1',  
  `manual_identification_id` int(11) NOT NULL DEFAULT '-1',  
  `predicted_identification_id` int(11) NOT NULL DEFAULT '-1',  
  `predicted_identification_confidence` float NOT NULL DEFAULT '-1',  
  `time` datetime NOT NULL,  
  `source_id` int(11) NOT NULL,  
  `test_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)  
  
CREATE TABLE `identifications` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_spanish2_ci NOT NULL,  
  `test_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)  
  
CREATE TABLE `sources` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_spanish2_ci NOT NULL,
```

```
`code` varchar(127) COLLATE utf8_spanish2_ci NOT NULL,  
`test_id` int(11) NOT NULL,  
PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `tests` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_spanish2_ci NOT NULL,  
  `confidence_limit` varchar(255) COLLATE utf8_spanish2_ci NOT NULL  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `visits` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `start` datetime NOT NULL,  
  `finish` datetime NOT NULL,  
  `participants` varchar(255) COLLATE utf8_spanish2_ci NOT NULL,  
  `source_id` int(11) NOT NULL,  
  `test_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```