



UNIVERSIDAD DE CHILE

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

# SISTEMA DE CENSADO DEL ESTADO ANÍMICO

---

ENFOCADO A PERSONAS DE LA TERCERA EDAD

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACION

KEVIN ANTONIO GUERRA ROMERO

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

PATRICIO POBLETE OLIVARES

ANDRES FARIAS RIQUELME

SANTIAGO DE CHILE

2015

## Resumen

La vida moderna y la evolución de la sociedad hacen que las personas dispongan cada vez de menos tiempo para el ocio y fomentar las relaciones sociales con sus pares. Esta falta de tiempo también nos hace perder lazos, incluso con nuestros seres más queridos, llegando al punto de no tener el tiempo ni siquiera para visitarlos. Esto es particularmente crítico para los adultos mayores, que son los que más resienten este empobrecimiento del vínculo social, especialmente entre ellos y sus familiares (hijos, nietos, etc.).

Esta situación produce aislamiento, y afecta negativamente el estado anímico de esas personas. Producto de la edad, es muy probable que los adultos mayores deban sobrellevar tratamientos médicos de diversa índole. Lamentablemente, los estados anímicos negativos (por ejemplo, depresión) reducen el impacto de fármacos y tratamientos en estas personas. El simple hecho de tener poco tiempo para visitarlos y darles el empujón anímico, afecta las emociones de estas personas. Por lo tanto, es importante que los familiares del adulto mayor sepan cuándo éste pasa por un cuadro de emociones depresivas, para así intentar de corregir o paliar dicha situación de la manera más rápida posible.

En este trabajo de memoria se diseñó e implementó un sistema que permite monitoreo del estado anímico de adultos mayores, cuando estos se encuentran al interior de sus hogares. El sistema permite a un usuario monitor, realizar un seguimiento de la persona monitoreada. Además, el sistema informa al monitor, mediante mensajes de texto (emails), cuando el usuario monitoreado está pasando por un cuadro de emociones negativas.

El sistema de monitoreo utiliza una tablet PC para tomar registros de audio e imágenes al interior del hogar de estas personas, y en base a esa información, determinar el estado anímico de las personas monitoreadas. Los procesos de reconocimiento de emociones usando audio e imágenes trabajan juntos, de manera de obtener los mejores resultados posibles. En el marco de esta memoria también se desarrolló el sistema de envío automático de alertas, el cual es capaz de notificar a los usuarios de la aplicación, cuando alguno de sus familiares monitoreados necesita apoyo anímico.

Si bien es cierto que es bastante complicado el trabajar con las características inherentes a la biometría de una persona para detectar sus emociones, pudimos completar el desarrollo de la aplicación propuesta. Sin embargo, no nos fue posible realizar pruebas en vivo con usuarios reales, por restricciones de tiempo. Por lo tanto, y aunque la aplicación esta funcional, aun se debe medir su impacto real en el escenario de aplicación previsto.

## Agradecimientos

Agradezco primero que todo a mi madre, quien me ha apoyado en todo momento, desde que decidí estudiar Ingeniería, hasta el día de hoy en el que estoy completando mi trabajo de memoria. Ella ha sido mi principal sustento, y de no ser por ella hoy en día no estaría contando la misma historia. Agradezco además a toda mi familia, que si bien es cierto no están en el día a día, cada vez que los he necesitado han estado para mí.

Agradezco de sobremanera a mi polola, es ella la que desde hace 3 años ha tenido que aguantarme cada vez que me encuentro colapsado con las tareas, ya sean universitarias o del trabajo. Es ella la que ha estado ahí para ayudarme en lo que ha podido, aun cuando la computación no se su fuerte, y le agradezco además el que me haya permitido ser parte de la vida de la personita que hoy llamo hija, la cual ha sido en muchos momentos mi razón de seguir, mi razón de continuar esforzándome.

Agradezco además a cada uno de mis amigos, por estar ahí cada vez que los necesité, por ayudarme a no estresarme semana a semana, les agradezco simplemente por ser quienes son.

En lo que respecta a este documento, le agradezco a Víctor Rojas y a Diego Muñoz por resolver con tan buena disponibilidad cada una de mis dudas con respecto a sus desarrollos.

Le agradezco al profesor Pablo Guerrero, por haberme entregado la información necesaria para que el sistema final de reconocimiento de emociones mediante imágenes, pudiese ser desarrollado a cabalidad.

Por último y por ningún motivo menos importante, le agradezco a mi profesor guía Sergio Ochoa, me faltarían páginas para agradecerle por cada uno de los aspectos técnicos en los que me ayudó para terminar la memoria, pero en realidad más que nada quisiera agradecerle por confiar siempre en mí.

Este trabajo de memoria ha sido parcialmente financiado por el proyecto Fondecyt Nro. 1120207.

Bonus Track: Un agradecimiento especial a todos los mantenedores de esos foros ucranianos, en los cuales fue posible encontrar soluciones para completar el desarrollo de este trabajo.

## Tabla de Contenido

1.	Introducción .....	1
1.1.	Justificación de la propuesta .....	2
1.2.	Objetivos de la memoria .....	4
1.3.	Resumen de la solución propuesta .....	5
2.	Trabajos relacionados .....	8
2.1.	Detección de emociones utilizando el procesamiento de voz .....	8
2.2.	Detección de emociones utilizando el procesamiento de imágenes.....	9
2.3.	Revisión de los insumos para la detección de emociones .....	10
2.4.	Tecnología Escogidas para la Implementación de la Solución .....	11
3.	Diseño de la solución.....	13
3.1.	Consideraciones generales.....	13
3.2.	Diseño de subsistemas .....	13
3.3.	Requisitos iniciales de la solución .....	16
3.4.	Diseño arquitectónico de la solución .....	20
4.	Implementación de los subsistemas .....	22
4.1.	Sistema de captura y transmisión de datos .....	22
4.2.	Sistema de reconocimiento de identidad .....	23
4.3.	Sistema de reconocimiento de emociones .....	33
4.4.	Subsistema de visualización de datos .....	36
4.5.	Subsistema de alertas .....	37
4.6.	Subsistema de detección de actividad .....	39
4.7.	Subsistema de administración de usuarios.....	40
4.8.	Flujo final del proceso realizado.....	42
4.9.	Tabla Resumen de Librerías .....	44
5.	Evaluación preliminar de la solución.....	45
6.	Conclusiones y trabajo a futuro .....	47
6.1.	Dificultades encontradas.....	47
6.2.	Conclusiones de la memoria .....	47
6.3.	Trabajo a futuro.....	48
7.	Bibliografía .....	50

## Índice de figuras

FIGURA 1 PROTOTIPO DEL SISTEMA.....	3
FIGURA 2 CASOS DE USO DEL USUARIO.....	17
FIGURA 3 CASOS DE USO DEL SISTEMA.....	17
FIGURA 4 DIAGRAMA DE FLUJO “REVISAR HISTORIAL” .....	18
FIGURA 5 DIAGRAMA DE FLUJO “RECOPILACIÓN DE DATOS” .....	19
FIGURA 6 ESQUEMA BÁSICO DE LA ARQUITECTURA DE LA SOLUCIÓN .....	20
FIGURA 7 ARQUITECTURA FÍSICA DE LA SOLUCIÓN.....	21
FIGURA 8 APLICACIÓN DE ALGORITMO EIGENFACES, SOBRE NUESTRO SET DE DATOS.....	26
FIGURA 9 APLICACIÓN DEL ALGORITMO FISHERFACES, SOBRE NUESTRO SET DE DATOS .....	27
FIGURA 10 APLICACIÓN DEL ALGORITMO LOCAL BINARY PATTERNS HISTOGRAMS, SOBRE NUESTRO SET DE DATOS .....	27
FIGURA 11 RECONSTRUCCIONES FACIALES CON PREDICTOR FISHERFACES.....	28
FIGURA 12 FORMATO DE ARCHIVO CSV .....	33
FIGURA 13 INTERFACES DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS.....	37
FIGURA 14 EMAIL DE EJEMPLO DE EMOCIONES NEGATIVAS DETECTADAS.....	39

## 1. Introducción

Durante los últimos años, el ritmo acelerado de vida moderna nos ha llevado a sufrir cada vez más trastornos asociados a nuestros estados de ánimo, por ejemplo depresiones, trastornos afectivos, bipolaridad, etc. En el caso de las personas de la tercera edad, está científicamente comprobado que estos trastornos de ánimo pueden producir enfermedades concretas, o bien hacer mucho más complejo el tratamiento de las enfermedades que ellos ya poseen [Hou01, Kar12]. Entrando más en detalle, si pensamos en la forma en la que viven los ancianos, estos generalmente están expuestos a la soledad, melancolía, y poca interacción social. Todos estos factores los hacen más propensos a contraer trastornos asociados a su estado de ánimo.

Por otro lado, las personas de la tercera edad son también las usualmente más frecuentemente afectadas por enfermedades de largo aliento o crónicas (por ejemplo, problemas motores, Alzheimer, diabetes, etc.). Si mezclamos los factores antes mencionados probablemente llegaríamos a la conclusión de que la gente de la tercera edad suele poseer enfermedades de largo aliento, cuyo tratamiento y el nivel de impacto de éste se ve afectado por el estado anímico de estos adultos mayores.

Diversos estudios han demostrado también que el incremento de la relaciones sociales ayuda a mejorar el estado de ánimo de la gente, e impacta positivamente en los tratamientos médicos que ello siguen [Con13, New09]. Lamentablemente hoy en día la mayor parte de la gente utiliza tecnología para sociabilizar, más específicamente aplicaciones computacionales, como email, redes sociales o mensajería instantánea. Esto limita considerablemente la eventual participación de los adultos mayores en este tipo de interacción social, puesto que la mayoría de ellos no sabe cómo utilizar la tecnología o son reticentes a hacerlo [Cza06].

La vida moderna hace que los familiares jóvenes de estos adultos mayores tengan cada vez menos tiempo para realizar visitas cara-a-cara, y por lo tanto puede costarles mucho identificar que un adulto mayor miembro de su familia está pasando por algún cuadro de depresión o estados de ánimo negativos. Por ende, puede ser muy difícil que los jóvenes (y también los adultos) reaccionen ante estas situaciones, más por desconocimiento que por otra cosa.

El problema que se desea abordar en este trabajo de memoria es la incapacidad de los miembros de una comunidad familiar, para detectar oportunamente los estados de ánimo negativos de sus adultos mayores, y de esa manera tener la chance de tomar medidas paliativas. La idea es utilizar una

solución computacional (actualmente en estado de prototipo) para detectar el estado de ánimo de los adultos mayores, utilizando dicha solución como sensor ambiental. Dicho sensor es instalado en el hogar de los adultos mayores y está conectado a Internet.

Cuando el sensor detecta síntomas de estados de ánimo negativos en los adultos mayores que están siendo monitoreados, éste disparará alarmas o avisos a la familia o a quienes corresponda, a fin de que estos puedan apoyar al adulto mayor que se encuentra en necesidad.

### **1.1. Justificación de la propuesta**

Esta memoria desarrolló un software que permite avisar, de manera certera, los cambios de ánimo de una persona de la tercera edad, la cual se encuentra monitoreada. Estos avisos son enviados a uno de sus familiares o personal médico que monitoree a esa persona. Particularmente, el sistema notificará cuando la emoción detectada supere un cierto umbral de emoción negativa (por ejemplo, tristeza prolongada o escenarios de angustia prolongada), y por lo tanto, que se requiera asistencia familiar o bien profesional.

En estricto rigor el software desarrollado podría servir para monitorear estados anímicos de un amplio espectro de personas, sin embargo su diseño actual ha sido evaluado principalmente con adultos mayores.

Para llevar a cabo la tarea planteada, esta memoria integra tres software existentes, los cuales han sido adaptados para hacerlos interoperar a nivel de datos, y así brindar un servicio de monitoreo integral y preciso de adultos mayores. Los productos de software en cuestión son los siguientes:

- Una red social ad hoc para personas de la tercera edad [Muñ13a,b].
- Un software de reconocimiento de estado anímico a través del procesamiento de la voz [Roj14].
- Un software de reconocimiento de estado anímico a través del reconocimiento facial (en base a componentes desarrollados por el profesor Pablo Guerrero).

La idea general de la aplicación desarrollada consistió en tomar la red social para personas de la tercera edad [Muñ13a,b], y utilizarla como conducto para alimentar con información a las otras dos aplicaciones. Éstas últimas son las encargadas de detectar las emociones utilizando el audio y el video capturado a través de una tablet PC, la cual se encuentra instalada en la casa de los adultos mayores actuando como sensor.



**FIGURA 1 Prototipo del sistema**

Mediante las aplicaciones de censado, se realiza el monitoreo del estado anímico de los adultos mayores que viven en esa casa, y se generarán las notificaciones respectivas, las cuales viajarán a través de la red social hacia los participantes involucrados en ésta (o sea, hacia los miembros de la familia de dichas personas). Además del sistema de notificaciones, se desarrolló también un sistema de monitoreo de la persona censada, de manera que algún familiar de la misma pueda tener acceso en línea y de manera visual a la información del estado anímico del adulto mayor.

Es importante destacar que los tres software antes mencionados fueron desarrollados por separado, sin prever la interoperación de los datos y servicios manejados por dichos componentes. Por lo tanto, esta memoria consiste principalmente en ajustar esos componentes de software para que funcionen apropiadamente en el contexto definido, y además automatizar el proceso antes mencionado para que no requiera de intervención humana. Esto último implicó la integración de los componentes en una única herramienta, permitiendo la interoperabilidad a nivel de datos y servicios entre los componentes antes mencionados. Es importante aclarar que el trabajo realizado también permite a otras soluciones de software, utilizar los servicios y datos provistos por esta solución.

Finalmente, se realizó un trabajo extra el cual consiste en el desarrollo del sistema de alertas y notificaciones, el cual permite notificar a los familiares en caso de que haya algún cambio de ánimo abrupto de la persona monitoreada.

La gran dificultad de este trabajo de memoria, radica en el ajuste e integración de los sistemas antes mencionados, pues como se mencionó antes,



estos fueron desarrollados en forma independiente unos de otros, y además escritos por distintas personas. Por lo tanto, para poder ajustarlos e integrarlos, debemos comprender a cabalidad todo el funcionamiento de los mismos. Junto con esto, se debe crear un vínculo de comunicación entre ellos.

Además, el desarrollo de estos componentes siguió distintos métodos, estilos y paradigmas de construcción; por lo tanto, para poder comprender cada sistema se debe intentar comprender la lógica con la que éstos fueron diseñados e implementados.

Otra dificultad a la cual nos vimos enfrentados, es a la capacidad de prueba de la aplicación, tanto a medida que se va desarrollando la solución propuesta, como cuando el sistema estuvo terminado. Para poder probar la aplicación necesitaremos reproducir de alguna manera los cambios en el estado de ánimo de las persona, lo cual involucra la utilización de datos que son difíciles de conseguir. Podríamos entender esta dificultad como el desafío que produce el utilizar un sistema biométrico; es decir, las personas no interactúan con este sistema sólo realizando “clicks”, sino que más bien los aspectos de la persona misma son los que se están relacionando con el sistema. Esto es lo que a mi parecer hace muy interesante a esta memoria; es decir, el hecho de poder trabajar aún más afondo con la información del usuario, en este caso en particular, con sus estados de ánimo.

## **1.2. Objetivos de la memoria**

El objetivo general de este trabajo de memoria es desarrollar un sistema de monitoreo del estado anímico de personas de la tercera edad, usando procesamiento de voz y de imágenes. El sistema tiene la capacidad de notificar a los familiares de la persona monitoreada (o a sus cuidadores) acerca de los distintos cambios de ánimo del adulto mayor. Dependiendo del tipo de cambio de ánimo y de la duración de la emoción detectada, podría ser necesario apoyo externo para dicha persona. Estos cambios de ánimo puedan ser dañinos para la salud del adulto mayor, por lo tanto deberían ser notificados a quien corresponda. Los objetivos específicos que se desprenden del objetivo general son los siguientes:

- Automatizar el sistema de reconocimiento de emociones por voz.
- Desarrollar el sistema de reconocimiento de emociones mediante imágenes.
- Incluir en el sistema “Red social para personas de la tercera edad” los mecanismos necesarios para la captura de datos.
- Generar un sistema funcional que permita la visualización de la información resumen resultante (de los estados anímicos).

- Generar un sistema de alertas para los miembros de una comunidad familiar.
- Probar la solución desarrollada, en un conjunto acotado de personas.

### 1.3. Resumen de la solución propuesta

Primero que nada debemos identificar los servicios y datos que necesitamos para obtener una solución que realmente resuelva nuestro problema. Dicha solución necesita poseer al menos 4 subsistemas importantes. Además de eso, se presenta como necesidad un 5º subsistema, el cual no es completamente necesario, pero ayudaría a mejorar la solución final. Por lo tanto, los subsistemas a ajustar e integrar son los 4 primeros, mientras que al último hay que desarrollarlo desde cero:

- Subsistema de reconocimiento de identidad.
- Subsistema de reconocimiento de emociones.
- Subsistema de visualización de datos.
- Subsistema de emisión de alertas.
- Subsistema de reconocimiento de actividad.

A continuación se describen brevemente cada uno de ellos.

**Sistema de reconocimiento de identidad:** Nuestro software tiene como finalidad el monitorear los estados de ánimo de una persona en particular. Para esto, un dispositivo estará realizando tomas (registros) auditivas y visuales de las personas monitoreadas. En un comienzo la aplicación estará alojada en una tablet. Aun cuando éste dispositivo se encuentre físicamente cerca de la persona monitoreada, no tenemos la seguridad de que las grabaciones realizadas por el dispositivo, pertenezcan al adulto mayor que deseamos monitorear. Por ejemplo, es muy probable que en ciertos momentos del día el sujeto monitoreado reciba alguna visita; de ser así, nuestras grabaciones contendrían tomas de dos personas diferentes. Por esto se hace necesario el tener un sistema que pueda distinguir cuáles grabaciones corresponden a la persona que estamos monitoreando, y cuáles no. Por lo tanto, se hace completamente necesario poseer un sistema que pueda identificar la identidad de la persona monitoreada. Así podemos asegurar que los datos procesados serán sólo de la persona que nos interesa monitorear, y que no estarán contaminados con datos del ambiente de los cuales no tenemos control.

**Sistema de reconocimiento de emociones:** Es claro que si estamos intentando medir el estado anímico de una persona, necesitaremos un

sistema que se encargue de reconocer las emociones que está sobrellevando el adulto mayor. Poniendo esto un poco más en concreto, lo que realmente queremos detectar son situaciones en donde una persona de la tercera edad se encuentra frente a un cuadro de “emociones negativas”, que pueden influir de mala manera en sus tratamientos médicos o en su diario vivir. Para esto, la emoción más influyente que deseamos aislar y reconocer es la tristeza, por ende este sistema debe estar completamente enfocado a reconocer esta emoción en las grabaciones (ya sean audios o imágenes).

**Sistema de visualización de datos:** Nuestra aplicación estará monitoreando día a día el estado de ánimo de una persona; por lo tanto, esto generará una gran cantidad de datos para ser procesados. La idea es entregar al usuario de la aplicación (por ejemplo, un familiar de la persona de la tercera edad), todos los datos obtenidos de manera que él mismo pueda realizar un monitoreo de esa persona. Este sistema tiene que enfocarse en entregar la información de una manera entendible, para que el usuario pueda utilizarla.

**Sistema de alertas:** Tal como se mencionó anteriormente en este trabajo, nuestro software debe tener la capacidad de enviar alertas a los familiares de las personas monitoreadas, utilizando diversos mecanismos, por ejemplo emails. Estas alertas serán enviadas cuando la persona monitoreada pase por un cuadro amplio de tristeza. Es muy importante que las alertas sean enviadas sólo cuando se logren alcanzar niveles altos de certeza sobre si el usuario está pasando por un cuadro depresivo. Esto se debe a que si el sistema envía una alerta al usuario, se espera que se genere una reacción por parte de éste último. Esta reacción puede ser un llamado telefónico, una visita al adulto mayor o lo que el usuario estime conveniente. Sea cual sea la acción que éste decida tomar, requerirá tiempo del usuario el cual puede ser muy valioso. Por lo tanto, debemos mantener los falsos positivos al mínimo (ojalá en cero) para que nuestro software no se convierta en un instrumento de molestia.

**Sistema de reconocimiento de actividad:** En términos de prioridades, este sistema está al final de nuestra lista de trabajos a incorporar. Esto se debe a que el software que debemos desarrollar, podría funcionar de buena manera sin la necesidad de este subsistema. Es por eso que, por el momento, este componente de software es considerado como “opcional”.

La idea de este sistema es poder reconocer si es que hay actividad humana en las cercanías del dispositivo que realiza la toma de datos. De haberla, comenzaríamos a registrar imágenes y audios para ser procesadas en el futuro. Este sistema no es completamente necesario, debido a que podría definirse una frecuencia de muestreo y realizar la toma

de datos cada cierto tiempo. Esto genera una mayor cantidad de datos, lo cual produce también mucho más procesamiento. En cambio, si realizamos un sistema de reconocimiento de actividad, se reduce en gran medida el procesamiento, y por ende la carga en el servidor.

Un punto importante es que este sistema de reconocimiento debe poseer dos ramas distintas: una es la rama auditiva, en la cual el sistema debiese poder diferenciar entre ruido y voz humana, y la otra rama es visual, con lo cual el sistema debería poder “notar” cuando alguien se encuentra frente al dispositivo de grabación.

Para poder decir que el proyecto está terminado, se deben desarrollar estos cinco subsistemas e interconectarlos de manera que un usuario pueda monitorear a alguna persona de su interés.

## 2. Trabajos relacionados

A continuación se presentan los frameworks más conocidos y soluciones ad hoc para realizar la detección del estado anímico de las personas, utilizando procesamiento de audio y de imágenes.

### 2.1. Detección de emociones utilizando el procesamiento de voz

Existen diversos productos de software para extraer las características de un archivo de audio. Estos productos fueron revisados en profundidad en la realización del software “*Monitoreo del estado anímico de los adultos mayores mediante el procesamiento de la voz*” [Roj14]. Es por esto que en este apartado sólo nos dedicaremos a mencionar las opciones que fueron barajadas en el momento de la realización de dicho software.

**HTK3**<sup>1</sup> es un software cuyo propósito está enfocado en la transcripción de audio a texto, realizando el proceso completo. Sin embargo, éste da muy pocas libertades para utilizar otras herramientas de la aplicación en conjunto con dicho software.

**Yaafe**<sup>2</sup> es un software diseñado especialmente para la extracción de características de cualquier tipo de audio, ya sea música, voz, etc. La aplicación está implementada en C++, y los resultados son generados en formato CSV. Su uso es sencillo, amigable al usuario y se puede interconectar con algunos programas como Matlab y IDEs como por ejemplo para Phyton.

**OpenSMILE**<sup>3</sup> es un conjunto de herramientas especializadas en la extracción de características sonoras, y se usa principalmente para realizar procesamiento de voz. Esta librería permite la creación de descriptores de audio, específicos para el reconocimiento de emociones. Además, estos descriptores pueden ser entregados en distintos formatos, entre ellos archivos para procesar con máquinas de soporte vectorial (SVM).

Además de esto, buscando en la Web por alguna solución ad hoc que realizara el reconocimiento de emociones mediante el procesamiento de voz, nos encontramos con **Emospeech**<sup>4</sup>. Este es un software que fue diseñado para

---

<sup>1</sup> <http://htk.eng.cam.ac.uk/>

<sup>2</sup> <http://yaafe.sourceforge.net/>

<sup>3</sup> <http://www.audeering.com/research/opensmile> OpenSmile es un desarrollo de la empresa Audering Intelligent Audio Engineering

<sup>4</sup> <http://emospeech.net/>

mejorar los servicios de un call center. El objetivo de este producto es reconocer las emociones de la persona a la cual se está llamando, para ver qué tan satisfactoria o insatisfactoria fue la atención entregada por el call center. Con esto se podría realizar con mayor facilidad el control de calidad de cada uno de los trabajadores de estos centros. Esta solución realiza exactamente el proceso que se necesita obtener en esta memoria, pero lamentablemente es pagada, y lo que busca esta memoria es desarrollar una solución open source.

## 2.2. Detección de emociones utilizando el procesamiento de imágenes

En la actualidad existen diversos productos, tanto comerciales como de acceso gratuito, los cuales permiten detectar emociones utilizando videos o imágenes. Primero que todo revisaremos algunas soluciones ad hoc que, entre otras cosas, pueden realizar lo que nosotros necesitamos.

**Nviso**<sup>5</sup>: Este es un programa comercial, el cual permite el reconocimiento de emociones en tiempo real. Para esto el software se basa en un sistema de reconocimiento facial, mediante una representación en 3 dimensiones de las imágenes. Utilizando sus servicios se pueden detectar y medir 7 emociones distintas, entre ellas la tristeza, felicidad, enojo, etc. Esto se realiza mediante el seguimiento de 43 músculos faciales distintos.

**Noldus**<sup>6</sup>: Este es otro programa comercial, el cual en esencia nos permite hacer el tracking de comportamientos humanos, tal como el anterior software. Noldus nos permite detectar 7 emociones en tiempo real, mediante la categorización de imágenes. La razón por la cual decidimos mencionarlo, aun cuando realiza funciones similares a las de Nviso, es porque este software nos ofrece además una medida de comportamientos animales.

**Auduno Clmtrackr**<sup>7</sup>: Esta es una solución de código abierto, la cual permite la detección de emociones en tiempo real. Esta solución está desarrollada en Javascript y puede ser incluida en cualquier sistema Web. Esta librería nos permite detectar entre 4 emociones distintas; particularmente, alegría, tristeza, enojo y sorpresa.

**OpenCV**<sup>8</sup> (Open Source Computer Vision Library): Esta es una librería de código abierto de visión computacional y aprendizaje de máquina (computer vision and machine learning). La librería fue diseñada

---

<sup>5</sup> <http://www.nviso.ch/>

<sup>6</sup> <http://www.noldus.com/>

<sup>7</sup> <https://github.com/auduno/clmtrackr> software de código abierto desarrollado por Audun Mathias Oygard

<sup>8</sup> <http://opencv.org/>

pensando siempre en la eficiencia del sistema, y enfocándose en las aplicaciones en tiempo real. Es por esto que la librería en si fue escrita en C/C++ optimizado utilizando estrategias de multiprocesadores.

OpenCV contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión computacional, como por ejemplo el reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica. Pero en realidad en este trabajo estamos particularmente interesados en la librería **FaceRecognizer**<sup>9</sup>. Dicha librería contiene todas las funcionalidades necesarias para realizar el trabajo de reconocimiento facial, lo cual al ser extendido de manera correcta, puede también realizar el reconocimiento de emociones.

Si bien existe una importante cantidad de software para reconocimiento de emociones utilizando audio y también video, la tasa de error de estos no es despreciable, por lo tanto se requiere combinar al menos un par de ellos para tratar de reducir al mínimo los falsos positivos.

### 2.3. Revisión de los insumos para la detección de emociones

Como hemos dicho anteriormente, para la realización de esta memoria necesitamos usar al menos tres productos de software distintos: (1) el de detección de emociones mediante el reconocimiento de la voz, (2) el de detección de emociones mediante reconocimiento facial, y por último (3) la red social para personas de la tercera edad, en este apartado detallaremos el estado actual de dichos software. A continuación se revisa el trabajo realizado en cada una de estas líneas, por parte de los respectivos autores.

**Detección de emociones mediante la voz:** Se desarrollaron una serie de scripts, los cuales pueden determinar de buena manera, si dentro de una grabación de audio se detecta la tristeza como emoción predominante. Inicialmente los scripts no estaban conectados y corrían de manera local. Además de esto, se recibió el esqueleto de una aplicación móvil que buscaba mostrar, a los usuarios monitores, los datos recopilados a través de dichos scripts.

**Detección de emociones mediante imágenes:** Respecto a este producto, en realidad no hay un software utilizable de detección de emociones mediante imágenes, pero lo que si hay son componentes desarrollados por el Prof. Pablo Guerrero (y sus alumnos), los cuales fueron utilizados como input para realizar esta tarea. El Prof. Guerrero

---

<sup>9</sup> <http://docs.opencv.org/trunk/modules/contrib/doc/facerec/> librería de reconocimiento facial dentro de OpenCV

está vinculado directamente con este proyecto, por lo tanto nos hemos apoyado en él para lograr este objetivo particular.

**Red social para personas de la tercera edad:** Esta es una aplicación móvil completamente funcional, permite intercomunicar a personas de la tercera con personas que puedan ser cercanas a ellas de una manera más fácil. Esta red social está completamente diseñada e implementada para ser usada por personas de edad avanzada, por lo que las interfaces son simples de entender y usar. En concreto la aplicación permite realizar video llamadas utilizando Skype, posee también una herramienta de intercambio de mensajes asíncronos (tipo email y Facebook). Además, se pueden dejar publicaciones para personas de la red social, y se pueden ver las fotos de la gente que tienes como contactos.

#### 2.4. Tecnología Escogidas para la Implementación de la Solución

Este trabajo conlleva la utilización de una gran cantidad de tecnologías distintas. Algunas de ellas son decisiones ya tomadas, por lo tanto sólo las comentaremos; otras en cambio, son decisiones tomadas completamente en el transcurso de esta memoria.

Primero que todo, nuestra aplicación tiene que funcionar interconectada con a la aplicación llamada *SocialConnector* [Muñ13a]. Esta aplicación, de tipo social, funciona sobre Android. Por lo tanto, nuestra aplicación debe usar Android como sistema de toma de datos de los usuarios monitoreados. En realidad con Android es bastante fácil captar tanto imágenes como audios, por lo tanto esta decisión si bien fue tomada con anterioridad, es realmente útil para los propósitos de esta memoria.

Uno de los sistemas que debemos integrar es el software “*Monitoreo del estado anímico de los adultos mayores mediante el procesamiento de la voz*” [Roj14]. Este sistema ya estaba creado al momento de realizar nuestra memoria, por lo tanto las tecnologías usadas en éste no son decisión nuestra. Algunas de las tecnologías utilizadas en este software son, por ejemplo, OpenSMILE para la recopilación de información desde los archivos de audio, Python como lenguaje de programación base, y Sox para el manejo de archivos de audio.

Por otro lado, para que nuestro proyecto funcione necesitamos enviar las imágenes y audios desde la tablet PC hacia algún servidor. Este servidor fue creado también dentro de la memoria antes mencionada [Roj14], y fue escrito en PHP utilizando el Framework Yii. Es por esto que adoptamos esta configuración para el resto del trabajo de memoria. Además, este servidor está conectado a una base de datos MySQL, en la cual incluiremos todos los datos necesarios para el correcto funcionamiento de la aplicación.



Por último, hemos decido utilizar OpenCV con C++, tanto para el reconocimiento de identidad, como para el reconocimiento de estados anímicos. Tal como lo dijimos anteriormente, OpenCV posee una serie de librerías importantes las cuales tienen optimizadas gran parte de las tareas que necesitamos realizar para construir estos dos procesos. La decisión de utilizar C++ pasa netamente por la compatibilidad que tiene con OpenCV, por lo tanto nos resultó bastante fácil preparar el ambiente para comenzar a trabajar utilizando estas dos tecnologías.

### 3. Diseño de la solución

Antes de empezar a definir por completo el diseño de nuestra solución, tenemos que entender que el realizar este trabajo conlleva un proceso largo, y con esto algunos aspectos de diseño cambiaron en el transcurso de la memoria, debido a los problemas encontrados en el camino. Con esto quiero decir que lo aquí presentado es el resultado de un proceso evolutivo, realizado en función de los resultados de las pruebas que se fueron realizando.

#### 3.1. Consideraciones generales

Existen dos consideraciones generales las cuales condicionaron por completo ciertos aspectos del diseño final. Estas consideraciones son las siguientes:

- La aplicación debe ser desarrollada en una Tablet con sistema operativo Android. Este primer punto es una condición importante en el proyecto y se da más que nada porque el software “*Red Social para Integración de Personas de la Tercera Edad*” está desarrollado en Android, y su diseño de interfaces está pensado para el uso en una Tablet. El software desarrollado en esta memoria debe estar embebido en dicha solución, por lo tanto debe ser compatible tanto con dicho sistema operativo, como con la red social antes mencionada.
- El desarrollo realizado en el lado del servidor utiliza PHP como lenguaje de programación. Esto se debe principalmente a que este proyecto se apoya en trabajos anteriores, los cuales están en su mayoría realizados en PHP. Es por esto que nos resultó mucho más fácil llegar a buen puerto utilizando el mismo lenguaje.

#### 3.2. Diseño de subsistemas

En este apartado nos centraremos en el diseño de cada uno de los subsistemas definidos en la sección “Solución Propuesta”.

**Sistema de reconocimiento de emociones:** En este proyecto utilizamos dos sistemas distintos de reconocimiento de emociones; uno de ellos funciona utilizando como input grabaciones de audio y el otro funciona utilizando imágenes tomadas a través de la Webcam de la tablet PC.

**Reconocimiento de emociones mediante audio:** Para realizar este proceso utilizaremos el software desarrollado en la memoria titulada “*Monitoreo del estado anímico de los adultos mayores mediante el procesamiento de la voz*” [Roj14]. Este sistema permite detectar algunas

emociones dentro de pistas de audio, en particular nos permite saber si es que se encuentra la tristeza como emoción predominante dentro de una pista de audio, además de entregarnos el porcentaje de certeza que se tiene respecto de la emoción detectada.

Este software es en realidad un conjunto de scripts separados, los cuales permiten realizar el proceso de detección en forma batch. El autor de este software recomendó realizar este procesamiento del lado del servidor, por ende esto funciona de la siguiente manera:

- El dispositivo captura los audios de la persona que estamos monitoreando, utilizando el micrófono embebido en la tablet PC.
- Estos audios son enviados al servidor para su procesamiento.
- El servidor realiza el reconocimiento de emociones y envía los resultados al tablet PC.

Para que todo esto funcione, se necesita de un procedimiento que nos permita automatizar y conectar cada uno de los scripts que permiten el reconocimiento del audio.

***Reconocimiento de emociones mediante imágenes:*** La verdad es que no tenemos a nuestra disposición un software de reconocimiento de emociones utilizando información visual (es decir, imágenes). Por lo tanto, tendremos que desarrollar uno ad hoc a nuestras necesidades. Para esto utilizaremos la librería OpenCV, la cual puede funcionar dentro de Android y provee ya algunas funciones interesante para el procesamiento de imágenes.

Si bien es cierto que en este punto no se han realizado muchos avances, hay un tema importante a considerar; éste es la cantidad de datos que se deberán enviar desde el dispositivo sensor (tablet PC) hacia el servidor. Si enviamos imágenes o streaming de video completo, probablemente el proceso de envío tardaría mucho tiempo, por lo cual debemos realizar una serie de pruebas para ver que procesamiento se puede hacer dentro del dispositivo móvil. De esa manera, no se recarga tanto la conexión con el servidor. Es decir, debemos generar un proceso en el cual el envío de datos sea el mínimo posible. Esto lo podemos lograr realizando algunas operaciones de reconocimiento facial en la Tablet misma.

Para el reconocimiento de emociones usamos una combinación de los dos procedimientos anteriores. La idea es tomar tanto audios como imágenes en un momento determinado, y luego analizar mediante ambos procesos dichos recursos. Con esto si ambos procesos muestran que la persona pasa por un cuadro de tristeza, tenemos mayor porcentaje de certeza que si utilizáramos sólo uno. Aun cuando la idea sea utilizar ambos procesos de detección durante la mayor parte del tiempo, existirán

ocasiones donde debemos utilizar sólo uno de ellos. Para que esto resulte bien, debemos establecer un cierto grado de certeza, con el cual nos sintamos cómodos asegurando que la persona efectivamente está triste.

**Sistema de reconocimiento de identidad:** Dado que nuestro software utilizará dos procesos distintos para el reconocimiento de emociones, debemos tener también dos sistemas de reconocimiento de identidad; uno que nos permita autenticar a una persona mediante su voz, y otro que nos permita hacerlo utilizando reconocimiento facial. En la práctica, el software que usaremos para el reconocimiento de voz no nos permite reconocer la identidad de la persona con certeza. En cambio utilizando el software recomendado para reconocimiento facial, si es posible reconocer la identidad de una persona con una confianza muy alta. Esto nos genera 3 opciones distintas a considerar:

- Utilizar sólo el sistema de reconocimiento facial, y sólo tomar datos cuando la persona se encuentre en frente del dispositivo. Esta es la solución más fácil de realizar, debido a que no requiere trabajo extra. Obviamente al utilizar esta opción perderíamos instancias en las cuales podríamos detectar algún patrón extraño en la persona monitoreada.
- Extender el software de reconocimiento auditivo que poseemos, para que éste pueda realizar de buena manera el reconocimiento de identidad. Esta opción nos brinda la posibilidad de obtener grabaciones de audio, aun cuando la persona no se encuentre frente al dispositivo. Sin embargo, el optar por esta solución implica agregar unas cuantas horas de desarrollo adicional al proyecto final.
- Obtener un software que nos permita realizar el reconocimiento de identidad mediante la vía auditiva. Con “obtener” me refiero a comprar algún software que nos permita realizar este proceso. De no ser posible la compra, se podrían evaluar también algunas soluciones Open Source.

La decisión final acerca de qué opción utilizar fue tomada a medida que avanzaba el proyecto, y ésta fue el utilizar la identificación mediante el reconocimiento facial.

**Sistema de visualización de datos:** Para realizar el sistema de visualización de datos se completó el desarrollo de una aplicación móvil, en Android, la cual funciona aparte de la red social para adultos mayores. Como parte de un trabajo previo fue recibido el diseño completo de este sistema, e incluso la base necesaria para su implementación. La idea de esta aplicación es conectar el sistema ya desarrollado con el servidor, de manera de poder obtener los datos necesarios para hacer que el

desarrollo existente sea funcional. Las interfaces del prototipo recibido se encuentran validadas con usuarios.

**Sistema de alertas:** El sistema de alertas pasa completamente por el servidor; es decir, cuando el servidor detecte que la persona pasa por un cuadro depresivo, éste mismo enviará una alerta a los contactos del adulto mayor. Esta alerta será un email simple mostrándole al usuario que la persona monitoreada se encuentra triste en este preciso momento, y que dicha emoción ha estado presente por un período considerable de tiempo. Tal como lo dijimos antes, es muy importante tener la certeza que se está mandando una alerta correcta, para que no molestar al usuario con alertas “falsas”.

**Sistema de reconocimiento de actividad:** En el marco de esta memoria no se alcanzó a terminar este subsistema por un tema de tiempo. Sin embargo, se intentará incluirlo para la presentación de defensa de la memoria.

El desarrollo o no de este último subsistema dependerá del tiempo con que se disponga antes de realizar la defensa de la memoria.

### 3.3. Requisitos iniciales de la solución

En esta sección mostraremos algunos de los principales requisitos y procesos involucrados en la solución propuesta. Dentro de nuestra aplicación consideraremos dos tipos de usuarios distintos.

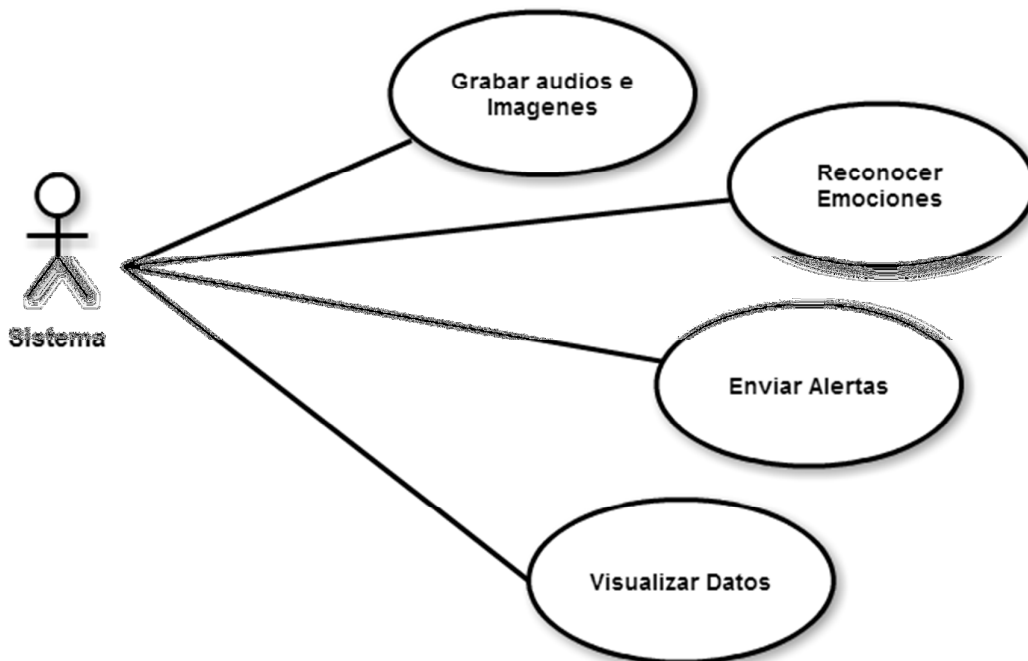
- **Usuario monitor:** El usuario monitor corresponde a aquel que desea realizar un monitoreo de otra persona, con el fin de obtener información relevante a los estados de ánimo de la esta última.
- **Usuario monitoreado:** El usuario monitoreado corresponde a la persona, sobre la cual se está ejecutando un monitoreo. Dicho usuario es el que debe encontrarse en las cercanías de la tablet PC para poder realizar la toma de muestras.

La Figura 2 muestra los casos de uso asociados a un “usuario monitor”, el cual mantiene monitoreado a un adulto mayor (es decir, a un “usuario monitoreado”).



**FIGURA 2 Casos de uso del usuario**

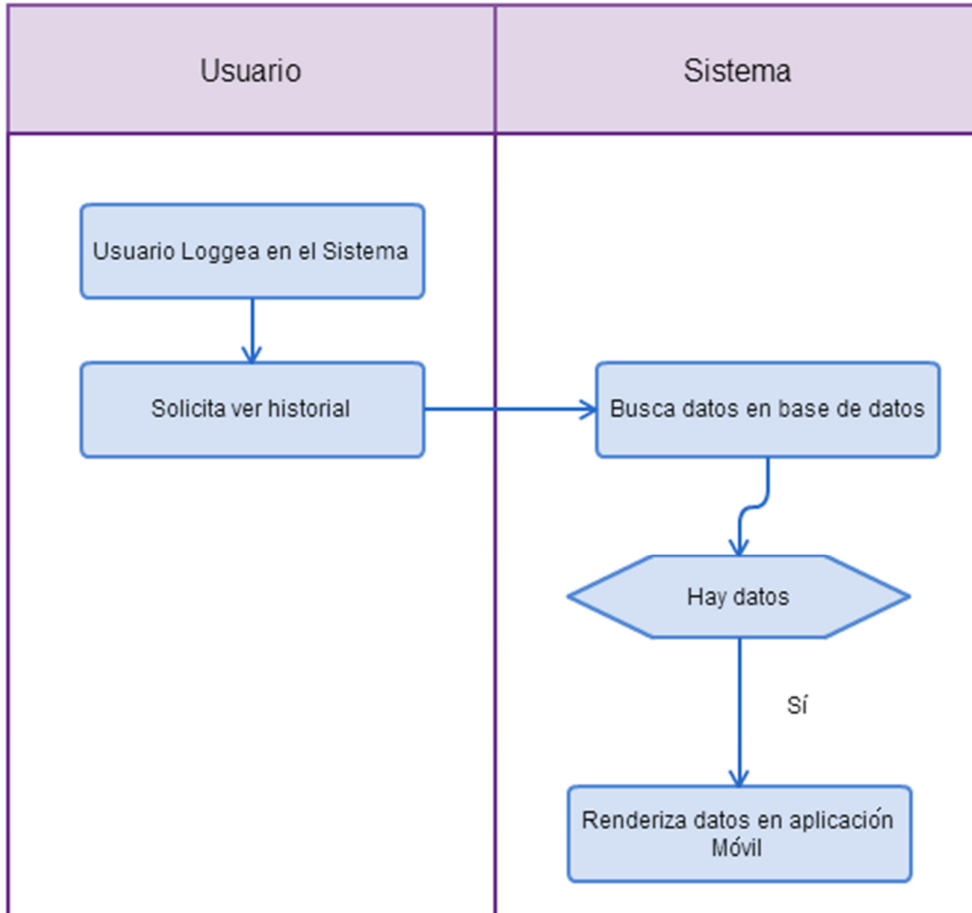
La Figura 3 muestra los casos de uso asociados al “usuario monitoreado”. En este caso, el sistema posee una gran carga de casos de uso y poca interacción con los usuarios monitoreados. Básicamente las funciones que se realizan son la captura y reconocimiento de la información de audio e imágenes a través de la tablet Pc (con la ayuda del servidor).



**FIGURA 3 Casos de uso del Sistema**

La Figura 4 muestra el flujo que ocurre cuando un usuario cualquiera, el cual mantiene monitoreada a otra persona, ingresa a la aplicación para revisar el historial de estados de ánimo de esta última.

*Diagramas de Flujo*



**FIGURA 4 Diagrama de flujo “revisar historial”**

La Figura 5 muestra el flujo necesario para que el sistema pueda recopilar datos de una persona monitoreada. Este diagrama considera dos condiciones distintas de inicio, la primera sólo se dará si se termina por desarrollar el sistema de reconocimiento de actividad. Si este sistema no fuera completado, entonces se eliminarían los primeros pasos del flujo.

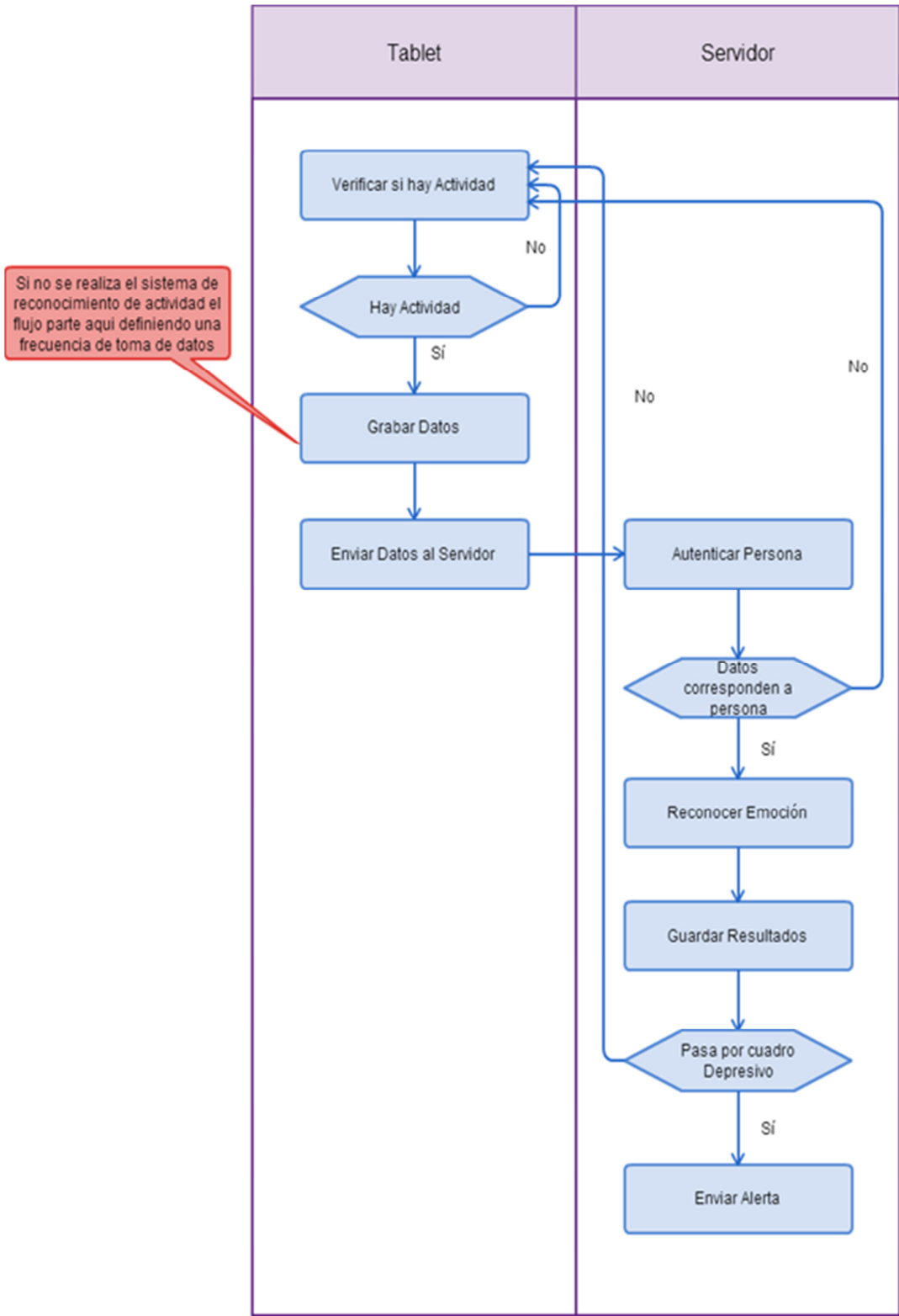


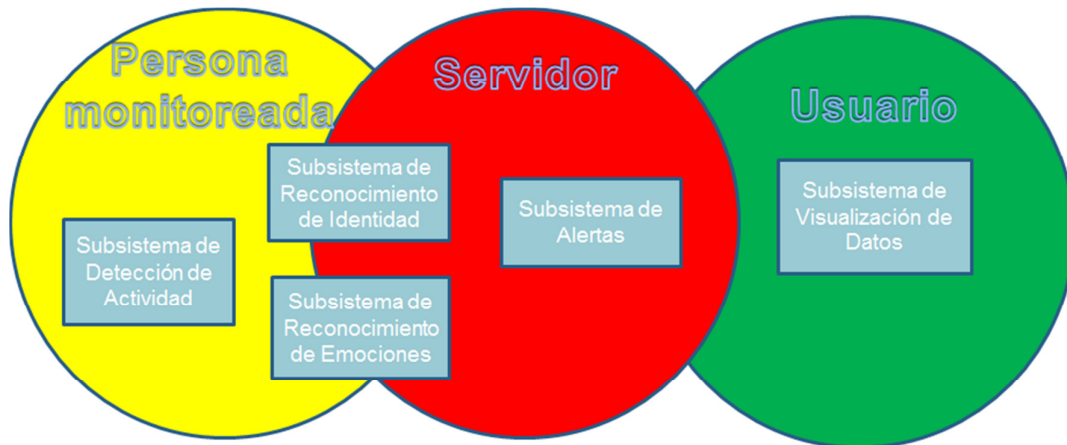
FIGURA 5 Diagrama de flujo “recopilación de datos”



### 3.4. Diseño arquitectónico de la solución

En esta sección mostraremos un par de diagramas de diseño estructural, los cuales ayudan a entender de mejor manera el proyecto completo. La figura 6 muestra dónde están funcionando cada uno de los distintos subsistemas de este proyecto.

*Diagrama de Subsistemas*



**FIGURA 6 Esquema básico de la arquitectura de la solución**

La Figura 7 muestra la arquitectura física sobre la cual se implementó el sistema propuesto. Nuestra arquitectura se compone de 3 partes fundamentales, la primera corresponde al dispositivo de interacción del usuario monitoreado con el sistema, el cual en este caso corresponde a una tablet PC. En segundo lugar se tiene un servidor, el cual será el responsable de alojar cada uno de los fragmentos de código que se necesitan para el funcionamiento de la aplicación. El sistema interactúa directamente con una base de datos relacional, la cual almacena todos los datos recopilados por la aplicación.

Por último, tenemos el dispositivo de interacción del usuario monitor, en el cual se puede revisar el historial de estados de ánimo del usuario monitoreado. Como se mencionó antes, esto corresponde a una aplicación Android desde su celular.

Arquitectura Física



FIGURA 7 Arquitectura física de la solución

## 4. Implementación de los subsistemas

En esta sección de la memoria detallaremos la construcción de cada uno de los subsistemas que forman parte de la solución. Además de esto, mostraremos en qué forma los subsistemas se encontrarán relacionados; es decir, cómo estos trabajan juntos para lograr entregar el servicio comprometido.

Antes de comenzar a explicar la construcción de cada uno de los subsistemas, nos centraremos un poco en un proceso importante del software, la cual no está considerado como un subsistema en sí, pero en realidad es completamente necesario para poder alimentar a cada uno de los subsistemas. Este proceso es la captura y transmisión de datos, el cual se explica a continuación.

### 4.1. Sistema de captura y transmisión de datos

Antes empezar a definir sistemas de reconocimiento de identidad o emociones, se hace imperativo tener un sistema que capture y envíe la información al servidor, ya sea que esta esté asociada a imágenes o capturas de audio. Dado que el proceso es muy similar para audios y para imágenes, se explicará para éste para el segundo tipo de datos.

Como se mencionó antes, nuestro sistema debe estar embebido dentro del sistema red social para personas de la tercera [Muñ13a], es por esto que nuestra toma de imágenes debe ser realizada en background. Es decir, no podemos traer la cámara al frente cada vez que deseemos tomar una fotografía, lo cual hace que el sistema de toma de imágenes posea una complejidad extra.

El primer problema que enfrentamos a la hora de capturar imágenes es que la cámara en Android necesita una superficie para poder realizar la captura. Cuando decimos “superficies” nos referimos a un “pedazo de pantalla” sobre el cual tomar las imágenes.

La primera aproximación para intentar resolver este problema, fue darle un pequeño pedacito de pantalla a la cámara, donde este pequeño pedacito no pudiera ser visto en ningún momento por el usuario de la aplicación. Esta aproximación no trajo muy buenos resultados debido a que se generaban muchos errores, y por lo tanto no podíamos asegurar que la cámara funcionara cada vez que fuera necesario; es por esto que se decidió probar con una aproximación distinta.

Desde la versión 11 de la API de Android tenemos a disposición un componente llamado *surfaceTexture*, el cual entre otras cosas funciona como

una superficie que puede ser utilizado por la cámara para realizar la toma de fotografías. Con esto se resuelve el problema planteado, pero debemos agregar la condición de que el software debe trabajar en dispositivos que cuenten con (al menos) la versión 11 de dicha API. Considerando que en la versión actual de la API es la 21, exigir contar con al menos la API 11, no es particularmente restrictivo.

Por último el tener la cámara tomando imágenes en background, significa que necesitamos que la cámara esté disponible para ser usada cada vez que queramos tomar una imagen. Es por esto que necesitamos poner una condición para saber que tenemos acceso a la cámara en el momento en particular en el que deseamos analizar a la persona. Esto en realidad es un problema bastante grande, debido a que si se está realizando una video llamada mediante Skype, la cámara estará siendo utilizada por esta aplicación, con lo cual en el único momento en el que podemos estar completamente seguros de que la persona se encuentra frente al dispositivo, el cual debería ser el mejor momento para medir las emociones por las cuales el usuario monitoreado está pasando, no podremos realizar mediciones debido a que en particular no podremos tomar las imágenes necesarias para el análisis.

Por otro lado, la función final de nuestra aplicación es poder avisar a un familiar de la persona monitoreada, que ésta se encuentra pasando por un cuadro depresivo, para que el usuario monitor tome algún tipo de acción al respecto. En el momento en que las personas se encuentran realizando una video llamada, ambos usuarios de la aplicación se están comunicando, de manera que es muy probable que si el usuario monitoreado pasa por un cuadro depresivo, su familiar pueda notarlo mediante la misma conversación.

En el caso de que la persona está pasando por un cuadro depresivo, aun cuando su familiar no se dé cuenta, el mismo hecho de tener esa video llamada puede subirle el ánimo. Es por esto que si bien es cierto este sería el momento más apropiado para tomar muestras (capturas de datos) analizables, perder la capacidad de hacerlo tampoco inhabilita por completo el correcto funcionamiento del software, debido a las razones anteriormente mencionadas.

#### **4.2. Sistema de reconocimiento de identidad**

En un principio, al pensar en el software final a obtener en esta memoria, el sistema de reconocimiento de identidad fue uno de los principales problemas que se debía abordar. Sabíamos que era necesario tenerlo listo, pero no sabíamos qué complejidad podría involucrar su desarrollo. Es por esto que se decidió empezar por este subsistema, sin saber en realidad si este mismo

sistema sería el que nos abriría las puertas para la realización del sistema de reconocimiento de emociones mediante imágenes.

La primera gran decisión tomada para la realización de este subsistema fue dónde debería estar alojado el procesamiento que éste realiza para reconocer una emoción. En un comienzo la idea era realizar la mayor parte del procesamiento dentro de la Tablet, y enviar los resultados finales al servidor. Todo esto, con el fin de recargar lo menos posible el procesamiento en el dispositivo móvil.

Para lograr realizar esto, se comenzó trabajando con OpenCV para Android. Si bien es cierto que dentro de la Tablet podíamos usar gran parte de las librerías que OpenCV tiene para ofrecer, no contábamos con las librerías que se deseaban utilizar para lograr el reconocimiento de identidad. Por lo tanto, se decidió que la Tablet sólo se utilizaría para tomar imágenes y enviarlas hacia el servidor. En este último es donde se realizaría el procesamiento completo de dichos datos, para poder reconocer la identidad de la persona que se encuentre en las imágenes.

Como mencionamos anteriormente el procesamiento se realizaría del lado del servidor, debido a que OpenCV tenía ciertas librerías que no funcionaban en Android. La librería en particular de la que estamos hablando es *FaceRecognizer*. Esta es una librería diseñada completamente para la realización de distintas tareas de reconocimiento facial. Esta librería dentro de algunas funcionalidades extra, nos ofrece tres algoritmos o predictores distintos, que nos ayudan a realizar el reconocimiento de identidad.

Para que alguno de estos algoritmos pueda predecir la identidad de una persona en una imagen, se debe trabajar en un procedimiento base, conocido como entrenamiento. En este proceso se le “enseña” al predictor a reconocer a una persona, es decir, se le entregan los rasgos característicos de dicha persona, para que a futuro éste pueda decidir si una imagen corresponde o no a él/ella.

Para la realización de este proceso debemos entregarle al algoritmo una base de datos de imágenes de la persona que deseamos reconocer, es decir debemos entrenar al algoritmo con un set de imágenes de la persona en cuestión.

#### *4.2.1. Evaluación de los algoritmos de identificación de personas*

Los tres algoritmos de identificación de personas que fueron considerados son los siguientes:

- **EigenFaces:** Obtiene los componentes principales (eigenfaces o autovectores) de la matriz de covarianza, formada por las imágenes de la base de datos y la imagen de entrada. En base a esto, se comparan las distancias entre el vector de la imagen original, y el resto. Si el resultado es menor que un umbral previamente fijado, habremos reconocido la cara positivamente. Este algoritmo tiene como ventajas un tiempo de cómputo pequeño, y muy buenos resultados en condiciones óptimas. Por otro lado, este algoritmo posee malos resultados en condiciones adversas, ya sea cuando hay cambios en la iluminación del ambiente, posiciones y orientación de las imágenes, o incluso pocas imágenes en la fase de entrenamiento.
- **FisherFaces:** Derivado de Eigenfaces, este algoritmo suaviza el problema de cambios de posturas, deformaciones de la cara, o problemas de luz, por lo que representa una ventaja respecto a EigenFaces. Este algoritmo utiliza LDA (Análisis Discriminante Lineal), cuya finalidad es describir las diferencias entre  $g$  grupos de objetos, sobre los que se observan  $p$  variables. Es por esto que se comporta mejor que EigenFaces cuando el número de poses es menor.
- **Local Binary Patterns Histograms:** Dado que una imagen puede ser caracterizada con un vector de  $n$  dimensiones, este algoritmo tiene como finalidad extraer una cierta cantidad de información de las imágenes, convirtiéndola en un vector de  $m$  dimensiones ( $m < n$ ) que preserva la información importante de la persona, y con esto se puede realizar su catalogación correcta (es decir, su identificación).

Realizamos un pequeño set de pruebas para ver cómo funcionan estos tres algoritmos en la práctica. Para esto usamos una base de datos de 3 personas y entrenamos a los algoritmos con 9 imágenes de cada persona. Esta cantidad de imágenes de entrenamiento, según los proponentes de dichos algoritmos, debería ser suficiente para realizar una buena predicción.

Luego de eso, se realizaron una serie de predicciones para ver cuál de los tres algoritmos obtenía un mejor resultado. Es importante destacar que se realizaron pruebas previas, en las cuales pudimos obtener un conjunto de imágenes lo suficientemente buenas como para entrenar al predictor. Es decir, estas imágenes contienen, entre ellas, imágenes de cuerpo completo, con una intensidad de luz similar, y diferentes posiciones de la cámara. Esto se hace para hacer funcionar de mejor manera a los predictores.

En las siguientes figuras mostraremos los resultados que obtenemos al utilizar cada uno de los algoritmos sobre la base de datos explicada en el párrafo anterior. Para esto utilizamos 42 imágenes de las distintas personas.

Las figura 8, 9 y 10 nos muestran los resultados obtenidos al utilizar los predictores *EigenFaces*, *FisherFaces* y *Local Binary Patterns Histograms* respectivamente, sobre la muestra de datos antes mencionada. Cada línea representa el procesamiento realizado para una de las 42 imágenes. La primera parte de la línea muestra la clase predicha de la imagen, la segunda parte nos muestra la clase a la cual realmente pertenecía la imagen, y la tercera parte nos muestra la confianza con la que el algoritmo entrega el resultado.

```

Predicted class = 0 / Actual class = 0 confidence= 65002.5
Predicted class = 0 / Actual class = 0 confidence= 28199.6
Predicted class = 0 / Actual class = 0 confidence= 48319.1
Predicted class = 0 / Actual class = 0 confidence= 52733.8
Predicted class = 0 / Actual class = 0 confidence= 51665
Predicted class = 0 / Actual class = 0 confidence= 46352.2
Predicted class = 0 / Actual class = 0 confidence= 44383.2
Predicted class = 0 / Actual class = 0 confidence= 45099.8
Predicted class = 0 / Actual class = 0 confidence= 56385
Predicted class = 0 / Actual class = 0 confidence= 40970.8
Predicted class = 0 / Actual class = 0 confidence= 40410.5
Predicted class = 0 / Actual class = 0 confidence= 50474.7
Predicted class = 0 / Actual class = 0 confidence= 60608.5
Predicted class = 0 / Actual class = 0 confidence= 70862.6
Predicted class = 0 / Actual class = 0 confidence= 60502.3
Predicted class = 0 / Actual class = 0 confidence= 66125.3
Predicted class = 0 / Actual class = 0 confidence= 65087.8
Predicted class = 0 / Actual class = 0 confidence= 80879.3
Predicted class = 0 / Actual class = 0 confidence= 56051.9
Predicted class = 1 / Actual class = 1 confidence= 37655.7
Predicted class = 0 / Actual class = 1 confidence= 59854.1
Predicted class = 0 / Actual class = 1 confidence= 65569
Predicted class = 0 / Actual class = 1 confidence= 51593.7
Predicted class = 0 / Actual class = 1 confidence= 72994.2
Predicted class = 0 / Actual class = 1 confidence= 73410.9
Predicted class = 1 / Actual class = 1 confidence= 72108.3
Predicted class = 1 / Actual class = 1 confidence= 50979.7
Predicted class = 0 / Actual class = 1 confidence= 59727.9
Predicted class = 1 / Actual class = 1 confidence= 45533
Predicted class = 1 / Actual class = 1 confidence= 49256.1
Predicted class = 2 / Actual class = 2 confidence= 61074.7
Predicted class = 2 / Actual class = 2 confidence= 48832.2
Predicted class = 2 / Actual class = 2 confidence= 46864.1
Predicted class = 2 / Actual class = 2 confidence= 45372.9
Predicted class = 2 / Actual class = 2 confidence= 33393.4
Predicted class = 2 / Actual class = 2 confidence= 34939.4
Predicted class = 2 / Actual class = 2 confidence= 29074.4
Predicted class = 2 / Actual class = 2 confidence= 39041.4
Predicted class = 2 / Actual class = 2 confidence= 82831.5
Predicted class = 2 / Actual class = 2 confidence= 66397.4
Predicted class = 0 / Actual class = 2 confidence= 74083.9
Predicted class = 2 / Actual class = 2 confidence= 62251.5

```

**FIGURA 8** Aplicación de algoritmo *EigenFaces*, sobre nuestro set de datos

```

Predicted class = 0 / Actual class = 0 confidence= 14660.3
Predicted class = 0 / Actual class = 0 confidence= 8799.72
Predicted class = 0 / Actual class = 0 confidence= 11111.5
Predicted class = 0 / Actual class = 0 confidence= 10193.3
Predicted class = 0 / Actual class = 0 confidence= 11058
Predicted class = 0 / Actual class = 0 confidence= 9939.52
Predicted class = 0 / Actual class = 0 confidence= 11934.6
Predicted class = 0 / Actual class = 0 confidence= 12049.4
Predicted class = 0 / Actual class = 0 confidence= 322.802
Predicted class = 0 / Actual class = 0 confidence= 2299.78
Predicted class = 0 / Actual class = 0 confidence= 5599.84
Predicted class = 0 / Actual class = 0 confidence= 11158
Predicted class = 0 / Actual class = 0 confidence= 8228.89
Predicted class = 0 / Actual class = 0 confidence= 17434.2
Predicted class = 0 / Actual class = 0 confidence= 12992.7
Predicted class = 0 / Actual class = 0 confidence= 10259.9
Predicted class = 0 / Actual class = 0 confidence= 12666.8
Predicted class = 0 / Actual class = 0 confidence= 16168.5
Predicted class = 0 / Actual class = 0 confidence= 12826.6
Predicted class = 1 / Actual class = 1 confidence= 13229.6
Predicted class = 1 / Actual class = 1 confidence= 13241.2
Predicted class = 1 / Actual class = 1 confidence= 9768.17
Predicted class = 1 / Actual class = 1 confidence= 7985.52
Predicted class = 1 / Actual class = 1 confidence= 9014.29
Predicted class = 1 / Actual class = 1 confidence= 11478.7
Predicted class = 1 / Actual class = 1 confidence= 14627.4
Predicted class = 1 / Actual class = 1 confidence= 3458.87
Predicted class = 1 / Actual class = 1 confidence= 15510.6
Predicted class = 1 / Actual class = 1 confidence= 7623.81
Predicted class = 1 / Actual class = 1 confidence= 16031.8
Predicted class = 2 / Actual class = 2 confidence= 2568.67
Predicted class = 2 / Actual class = 2 confidence= 861.701
Predicted class = 2 / Actual class = 2 confidence= 763.388
Predicted class = 2 / Actual class = 2 confidence= 5146.95
Predicted class = 2 / Actual class = 2 confidence= 1970.04
Predicted class = 2 / Actual class = 2 confidence= 5335.27
Predicted class = 2 / Actual class = 2 confidence= 2192.73
Predicted class = 2 / Actual class = 2 confidence= 2728.66
Predicted class = 2 / Actual class = 2 confidence= 1357.67
Predicted class = 1 / Actual class = 2 confidence= 21255.4
Predicted class = 0 / Actual class = 2 confidence= 11499.6
Predicted class = 2 / Actual class = 2 confidence= 15103.9

```

FIGURA 9 Aplicación del algoritmo FisherFaces, sobre nuestro set de datos

```

Predicted class = 0 / Actual class = 0 confidence= 7.03293
Predicted class = 0 / Actual class = 0 confidence= 5.23141
Predicted class = 0 / Actual class = 0 confidence= 8.35803
Predicted class = 0 / Actual class = 0 confidence= 10.7973
Predicted class = 0 / Actual class = 0 confidence= 13.7344
Predicted class = 0 / Actual class = 0 confidence= 6.24267
Predicted class = 0 / Actual class = 0 confidence= 7.77387
Predicted class = 0 / Actual class = 0 confidence= 9.72804
Predicted class = 2 / Actual class = 0 confidence= 24.2353
Predicted class = 0 / Actual class = 0 confidence= 16.1204
Predicted class = 0 / Actual class = 0 confidence= 9.61397
Predicted class = 2 / Actual class = 0 confidence= 50.9972
Predicted class = 1 / Actual class = 0 confidence= 44.7983
Predicted class = 2 / Actual class = 0 confidence= 83.6139
Predicted class = 1 / Actual class = 0 confidence= 69.2518
Predicted class = 1 / Actual class = 0 confidence= 31.6434
Predicted class = 2 / Actual class = 0 confidence= 19.4772
Predicted class = 0 / Actual class = 0 confidence= 15.7149
Predicted class = 0 / Actual class = 0 confidence= 17.3935
Predicted class = 1 / Actual class = 1 confidence= 49.261
Predicted class = 1 / Actual class = 1 confidence= 10.5311
Predicted class = 1 / Actual class = 1 confidence= 15.4415
Predicted class = 0 / Actual class = 1 confidence= 13.5894
Predicted class = 1 / Actual class = 1 confidence= 22.9945
Predicted class = 0 / Actual class = 1 confidence= 74.7958
Predicted class = 1 / Actual class = 1 confidence= 11.4522
Predicted class = 2 / Actual class = 1 confidence= 29.5567
Predicted class = 2 / Actual class = 1 confidence= 44.7317
Predicted class = 1 / Actual class = 1 confidence= 15.2755
Predicted class = 1 / Actual class = 1 confidence= 17.2265
Predicted class = 2 / Actual class = 2 confidence= 7.48535
Predicted class = 2 / Actual class = 2 confidence= 6.72618
Predicted class = 2 / Actual class = 2 confidence= 3.48609
Predicted class = 2 / Actual class = 2 confidence= 3.5596
Predicted class = 2 / Actual class = 2 confidence= 2.38586
Predicted class = 2 / Actual class = 2 confidence= 3.65852
Predicted class = 2 / Actual class = 2 confidence= 3.18838
Predicted class = 2 / Actual class = 2 confidence= 3.03153
Predicted class = 0 / Actual class = 2 confidence= 13.6187
Predicted class = 2 / Actual class = 2 confidence= 12.5339
Predicted class = 1 / Actual class = 2 confidence= 22.1432
Predicted class = 0 / Actual class = 2 confidence= 12.5446

```

FIGURA 10 Aplicación del algoritmo Local Binary Patterns Histograms, sobre nuestro set de datos



Luego de apreciar los resultados mostrados anteriormente, tomamos la decisión de utilizar el modelo de FisherFaces, debido a que éste es el que mejor resultados obtiene, al menos en nuestra muestra, la cual se acerca bastante a la realidad en la cual funcionará el sistema final cuando esté en producción. Además de esto, los tiempos de cómputo son razonables.

Por último, mostramos en la Figura 11 unas reconstrucciones faciales creadas con el predictor FisherFaces, en las cuales se pueden apreciar claramente los rasgos más notorios de la cara de una de las personas usadas en nuestro set de datos.



**FIGURA 11 Reconstrucciones faciales con predictor FisherFaces**

Por lo tanto, tenemos un sistema que nos permite identificar a una persona mediante imágenes, las cuales pueden ser tomadas por la misma Tablet. Aun así, luego de tomar la decisión de utilizar el predictor FisherFaces, con el cual obteníamos muy buenos resultados, dichos resultados no son perfectos. Por ende, se realizó un pre-procesamiento que nos permite obtener aún mejores resultados.

Dicho pre-procesamiento consiste en detectar el punto exacto donde se encuentran los ojos en una imagen, y realizar un corte de la foto de manera de obtener siempre la cara de la persona. Con esto nos evitamos estar tomando factores que sólo ensucian el reconocimiento de la imagen, como lo son la ropa del individuo, o cualquier factor externo presente en la imagen.

#### *4.2.2. Proceso de detección de caras en una imagen*

Como dijimos antes, se decidió realizar un pre-procesamiento sobre cada una de las imágenes que serían analizadas luego por nuestro sistema de reconocimiento de identidad. Dicho procesamiento consiste en la detección del punto exacto de la fotografía en el que se encuentra la cara de una persona, para posteriormente realizar un corte de la imagen de manera de tomar sólo los rasgos faciales de la misma. Dichos rasgos son lo que realmente nos importan a la hora de hablar de reconocimiento de identidad.

Está claro que dependiendo de la distancia entre la persona y el dispositivo que estará tomando las fotografías, el tamaño de su cara variará. Por lo tanto, si simplemente realizáramos el recorte de la imagen, restringiendo su tamaño netamente a la cara de la persona, tendríamos muchas imágenes de distintos tamaños enviadas al servidor.

Lo anterior es realmente un problema debido que una de las restricciones importantes para obtener buenos resultados usando la librería FaceRecognizer, es que todas las imágenes utilizadas deben poseer el mismo tamaño. Es por esto que se decidió además realizar un cambio de tamaño a cada una de las fotografías, con el fin de obtener un set de imágenes que pudiesen ser analizadas con la librería antes mencionada. El proceso de escalado de las imágenes se realiza en forma automática en la solución desarrollada.

Para realizar la detección de caras en nuestras fotografías, se utilizaron lo que llamamos “haar classifier” o en español clasificador de Haar. El clasificador de Haar fue desarrollado por Viola y Jones, como una versión mejorada del algoritmo “Adaboost” [Yof01]. Este clasificador utiliza árboles de decisión con entrenamiento supervisado.

El entrenamiento se realiza determinando una serie de características basadas en sumas y restas de los niveles de intensidad de la imagen. Basándonos en estas características locales podemos obtener un detector de objetos robusto. También se denominan estos clasificadores mediante el nombre de cascada, ya que el resultado del clasificador es el fruto de varios clasificadores más simples o etapas. El candidato a objeto dentro de la imagen a procesar debe superar todas las etapas para ser aceptado.

AdaBoost es un algoritmo que se emplea para encontrar reglas de clasificación altamente precisas (fuertes), combinando un número elevado de reglas de baja precisión (débiles). Es decir, se trata de una combinación lineal de clasificadores débiles para obtener clasificadores fuertes.

OpenCV pone a nuestra disposición una gran cantidad de clasificadores de cascada, los cuales nos permiten encontrar diferentes objetos dentro de nuestras imágenes. En particular, estamos interesados en los clasificadores que nos permiten detectar la posición de la cara de una persona. Luego de realizar una serie de pruebas con los distintos clasificadores, encontramos que con el clasificador de OpenCV llamado “haarcascade\_frontalface\_alt” se obtenían buenos resultados. Este clasificador, como el nombre lo dice, intenta encontrar una cara que esté en una posición frontal dentro de nuestra imagen.

Dentro de los clasificadores evaluados se encuentran los que buscaban en la imagen la posición de los ojos (“haarcascade\_eye”, “haarcascade\_mcs\_righteye”, y “haarcascade\_mcs\_leftt-eye”), además de un clasificador que buscaba una cara en la posición de perfil (“haarcascade\_profileface”).

Un punto importante a destacar es que la librería FaceRecognizer está diseñada para reconocer identidad sobre una base de datos de un grupo de personas. Es decir, teniendo un algoritmo entrenado con las imágenes de dichas personas, al alimentar el algoritmo con una nueva imagen, éste podrá decir a cuál de las personas dentro de la base de datos corresponde. Ponemos énfasis en esta parte, debido a que éste no es realmente el comportamiento que nosotros necesitamos. De hecho, lo que necesitamos es que dada una base de datos de imágenes de una persona, el algoritmo pueda decirnos al entregarle una nueva imagen, si ésta corresponde o no a dicha persona.

Para poder lograr el comportamiento antes descrito, decidimos utilizar una base de datos que tuviera las imágenes de la persona a ser monitoreada, junto con otra categoría de imágenes a las que catalogamos vacías. De ahora en adelante llamaremos a este modelo *persona-vacío*. En la categoría de imágenes vacías simplemente tenemos dos imágenes, una de ellas completamente gris y la otra completamente negra. La razón de esto es que luego de hacer pruebas con imágenes, las cuales no poseían una cara, éstas al ser pasadas por el algoritmo de reconocimiento facial, terminaban como imágenes completamente grises o negras. Es por eso que dichas imágenes tienen una gran probabilidad de caer en nuestra categoría de imágenes vacías, y con esto al menos en este tipo de imágenes, definir de buena manera si la persona monitoreada se encuentra o no en la imagen que estamos procesando.

Además de lo anterior, existe la posibilidad de que la persona que deseamos monitorear no se encuentre sola en el lugar donde se pondrá el dispositivo, o no sea la única que haga uso del mismo. Es por esto que existe la posibilidad de que el algoritmo deba distinguir entre dos personas distintas, teniendo a sólo una de ellas en la base de datos que está utilizando para el

modelo. Esto es bastante distinto a distinguir entre la persona monitoreada y el vacío, debido particularmente a que para las dos personas se podrá encontrar una cara en la fotografía. Esto hace que las fotos de las dos personas, en la mayoría de los casos sean reconocidas como la de la persona que está siendo monitoreada, lo cual es bastante lógico, debido a que aunque le estemos pasando una imagen de una persona que no se encuentra en la base de datos, esta persona de seguro tiene más rasgos en común con otro ser humano, que con la catalogación de vacío que estamos utilizando.

Para resolver el problema anterior FaceRecognizer nos ofrece una característica simple, la cual nos será de gran ayuda. Para cada una de las predicciones que el modelo realiza, es posible obtener la confianza con la cual llega a obtener dicha predicción. Esta confianza se presenta a través de un número decimal (es decir, en código está representado como un decimal de doble precisión), el cual dependiendo de su valor, nos puede decir qué tan “correcta” es la predicción que está realizando, siendo el 0,0 el valor que representa el 100% de confianza en la predicción.

La verdad es que la librería no tiene una documentación acerca de los valores de confianza predeterminados, sin embargo pudimos descubrir el valor correspondiente, realizando una prueba simple. Es dicha prueba, cada una de las imágenes que intentamos predecir era parte del conjunto de imágenes utilizadas para entrenar el algoritmo. Para cada una de estas imágenes el modelo estaba 100% seguro de a qué categoría pertenecían. Utilizando los valores de confianza entregados por FaceRecognizer, pudimos darnos cuenta que en la mayoría de los casos que el algoritmo fallaba en una predicción utilizando el modelo persona-vacío, el valor de confianza superaba los 10.000. De hecho, para los casos en los cuales se presentaban imágenes de personas que no estaban en el modelo, el valor de confianza solía sobrepasar los 20.000.

Teniendo esto en cuenta pudimos generar las condiciones suficientes para que el software reconociera identidad, donde la estrategia se basa en aplicar una simple conjetura. Esta conjetura es que para que una persona sea reconocida por nuestro software, es decir para poder asegurar que la persona que se encuentra en la imagen que estamos procesando es la persona con la cual el modelo fue entrenada, se deben cumplir las siguientes dos condiciones:

1. El predictor obtenido con el modelo debe darnos como output que la imagen procesada corresponde a la de la persona que se encuentra en la base de datos.
2. La confianza que el predictor nos asegure para el procesamiento de la imagen no puede sobrepasar el valor 10.000.

De cumplirse las dos condiciones anteriores podemos asegurar que la persona procesada en la imagen, es la persona que estamos monitoreando.

Los valores que estamos utilizando en los niveles de confianza son los que logramos obtener luego de realizar una serie de pruebas empíricas. En ningún caso representan un límite teórico o de referencia para cualquier escenario, fundamentalmente porque no tenemos ningún tipo de referencia bibliográfica acerca de cómo se calcula este nivel de confianza. Si bien es cierto que en nuestra primera versión de este software fijamos el nivel de confianza en los 10.000, esta condición es un tanto relajada y podríamos permitirnos disminuir dicho valor, sacrificando precisión. Esto significaría que algunas imágenes que sí corresponden a la persona en monitoreada, podrían no ser aceptadas por el software. Sin embargo, también significaría que la probabilidad de producir un falso positivo, también disminuiría. La única forma de adaptar de buena manera este coeficiente será con el uso real de la aplicación.

Anteriormente se había mencionado la posibilidad de tener un sistema de reconocimiento de identidad mediante la voz, si bien es cierto esto podría haber ayudado mucho a la performance final del software desarrollado, se decidió no realizarlo al menos para esta primera versión del software, esto debido simplemente al tiempo de desarrollo que esto implicaba y a los buenos resultados obtenidos utilizando el procesamiento de imágenes. Debido a esto, tuvimos que agregar algunas condiciones, las cuales afectan directamente el desempeño final de la aplicación. La condición más importante es el momento en el que se tomarán las imágenes. Debido a que la única forma de reconocer identidad será mediante el procesamiento de las mismas, esto nos obliga a que cada vez que queremos tomar una muestra auditiva, debemos además tomar una imagen para poder autenticar a la persona monitoreada. Debido a esta condición, perderemos algunos momentos en los cuales se podrían tomar muestras.

El caso más simple que se nos viene a la mente es cuando la persona está hablando en las cercanías del dispositivo que toma las muestras, pero no se encuentra directamente en frente del mismo, debido a que la imagen tomada en ese momento estará "vacía", aun cuando la persona que está hablando es la monitoreada, pero la forma de funcionamiento del software la autenticará como si no lo fuese. Si bien es cierto que esta es una condición bastante fuerte, será aceptada para esta primera versión del software.

Este subsistema se presenta por medio de un ejecutable desarrollado en C++, de nombre *facial\_reconition.exe*. Éste recibe dos parámetros por la línea de comandos. El primero contiene la información de la base de datos con la cual se entrenará el algoritmo, y el segundo parámetro es la imagen que se desea procesar sobre dicha base de datos.

El primer parámetro de este ejecutable debe ser entregado en forma de un archivo de formato .CSV, el cual simplemente contiene una serie de líneas en las cuales se encuentran las direcciones de las imágenes utilizadas en la base de datos, además de un identificador final el cual nos dice a qué categoría pertenecen cada una de las imágenes. A continuación se muestra un extracto (como ejemplo) del formato que dicho archivo debe poseer:

```
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180107.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180115.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180119.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180123.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180153.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180208.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180215.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20141118_180216.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20150227_152824.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20150227_152827.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20150227_152830.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\Kevin\IMG_20150227_152835.jpg;0
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\vacio/gris.jpg;1
C:\Users\Kevin\Desktop\ReconocimientoFacial\imagenes\vacio/negro.jpg;1
```

**FIGURA 12 Formato de archivo csv**

Si bien es cierto la cantidad de imágenes utilizadas para crear una base de datos para los modelos antes mencionados es arbitraria, los proponentes recomiendan usar al menos 9 imágenes para poder crear una base de datos, en la cual se puedan reconocer de buena manera los rasgos faciales de la persona en cuestión. En la mayoría de los ejemplos utilizados para realizar pruebas en esta memoria se utilizaron 12 imágenes para entrenar al modelo.

Por último, la construcción de los archivos .CSV podría ser algo tediosa de hacerse a mano, es por esto que además se decidió utilizar un script en Python, el cual se encarga de crear automáticamente el listado de todos los archivos contenidos en un directorio determinado, el cual le es entregado como parámetro al script. Dicho script lleva por nombre *create\_csv.py*.

### 4.3. Sistema de reconocimiento de emociones

Tal y como lo hemos señalado en reiteradas ocasiones, dentro de este documento existen dos procesos que nos permitirán realizar el reconocimiento

de emociones. El primero de ellos es mediante audios y el segundo mediante imágenes. A continuación se describe cada uno de ellos:

#### *4.3.1. Sistema de reconocimiento de emociones mediante el procesamiento de audios*

Si bien es cierto el sistema que reconoce emociones mediante audios, dicha solución no fue generada en esta memoria, sino que será más bien fue ajustada y adaptada a partir de la solución propuesta en [Roj14]. Esta tarea fue necesaria de realizar, pues debíamos asegurarnos de que ese software estuviera completamente funcional e integrado con los demás subsistemas del software final. La versión que se tiene hoy en día de este sistema realiza el siguiente proceso:

1. Define un umbral de volumen para la Tablet. Si el sonido sobrepasa dicho umbral, entonces el programa graba un audio.
2. El audio grabado en el paso anterior se envía al servidor.
3. El servidor recibe y guarda el audio.
4. El servidor genera las entradas necesarias en la base de datos, para realizar el correspondiente procesamiento del audio.

Luego de realizar estos pasos, se debe llevar a cabo el procesamiento de la información. Para esto existe un script que revisa la base de datos, y luego realiza el procesamiento necesario para las entradas que en ella se encuentren. Por lo tanto, el paso final es generar una tarea programada, la cual estará utilizando este script cada cierto tiempo dentro del servidor, de manera que los audios que se hayan recibido sean procesados apropiadamente y a tiempo.

#### *4.3.2. Sistema de reconocimiento de emociones mediante el procesamiento de imágenes*

Anteriormente dijimos que el sistema de reconocimiento de identidad nos abriría las puertas para realizar el reconocimiento de emociones mediante imágenes. Esto se debe a que los predictores que estamos usando nos permiten también realizar el reconocimiento de emociones. Lo único que necesitamos es una base de datos de emociones la cual nos permita definir las emociones por las que está pasando una persona.

Para realizar el reconocimiento de emociones es muy importante tener funcional el pre-procesamiento del que hablamos en la sección anterior. Esto es porque para poder reconocer emociones, necesitamos obtener la mayor cantidad posible de información útil de la imagen. La información útil (cuando hablamos de reconocimiento de emociones) viene dada, en su mayor parte, por las facciones y expresiones de la cara.

Luego de revisar varias bases de datos de imágenes, encontramos dos que se podrían ajustar a nuestras necesidades, debido a que la emoción que prevalece en cada imagen, está rotulada y clasificada. Las dos bases de datos evaluadas son:

- **Yale Face Database** [Bel97]: Esta base de datos contiene 165 imágenes de 15 individuos distintos. Las imágenes están en escalas de grises y almacenadas en formato GIF. Dentro de las imágenes podemos encontrar 11 clasificaciones distintas para cada usuario. Cada clasificación representa una expresión facial o configuración de luz. Estas clasificaciones son las siguientes: luz en el centro, con anteojos, feliz, luz a la izquierda, sin anteojos, normal, luz a la derecha, triste, somnoliento, sorpresa y guiñando el ojo.
- **The Japanese Female Facial Expression (JAFFE) Database:** Esta base de datos contiene 213 imágenes de 7 expresiones faciales distintas, posadas por 10 personas japonesas; todas ellas de sexo femenino. Las categorías de las imágenes son las siguientes: miedo, sorpresa, normal, feliz, tristeza, enojo y disgusto.

Para el desarrollo de nuestro subsistema decidimos utilizar la base de datos de Yale, debido principalmente a que dentro de ésta los individuos fotografiados tienen géneros, edades y etnias distintas. Por lo tanto, los resultados que se obtengan con esta base de datos serán más representativos, respecto a lo que podemos esperar cuando se pruebe el sistema en un escenario real.

Dentro de esta base de datos existen algunas categorías que no aportan información respecto a las emociones de una persona. Es por esto que tomaremos un subconjunto de categorías para formar una base de datos que resuelva nuestras necesidades. Las categorías que tomaremos serán las siguientes: feliz, normal, triste, sorprendido y somnoliento. Si bien es cierto que la única emoción que realmente queremos aislar es la tristeza, el poseer un conjunto aún más grande de estados posibles nos ayudará a reducir la cantidad de falsos positivos.

Tal como lo dijimos anteriormente, el funcionamiento de nuestro sistema de detección de emociones mediante imágenes, es muy similar al de detección de identidad. Esto básicamente por que se utiliza el mismo algoritmo, pero con un fin diferente. Aun así, debemos señalar que existe una gran diferencia con respecto a la manera de tomar la base de datos para alimentar el algoritmo. En el sistema de reconocimiento de identidad entrenamos al algoritmo cada vez que tenemos que analizar una muestra. Esto se desarrolló de esa manera



debido a que en el futuro las imágenes que se utilizarán como base para una persona, podrían cambiar con el tiempo. En cambio la base de datos que utilizamos para el reconocimiento de emociones se encuentra fija, por lo tanto, sólo entrenaremos al algoritmo una vez y dejaremos guardada esta información de entrenamiento para los futuros usos de la aplicación.

Dicho lo anterior, este subsistema está compuesto por dos programas distintos: *train\_predictor.exe* y *predict\_emotion.exe*.

- **Train\_predictor:** Este programa realiza el entrenamiento de un predictor con una base de datos de imágenes, y almacena esta información en un archivo, para luego utilizar dicho predictor en el procesamiento de las imágenes.
- **Predict\_emotion:** Con este programa se realiza la predicción de emociones sobre las imágenes de los usuarios monitoreados. Para ello simplemente carga el archivo generado con el ejecutable anterior, y en base a ese archivo se genera una predicción.

#### 4.4. Subsistema de visualización de datos

Como lo mencionamos anteriormente, ya existe un armazón (estructura) de este subsistema, el cual fue desarrollado en Android. Esta estructura posee interfaces que fueron desarrolladas y validadas en el trabajo de memoria de Víctor Rojas [Roj14]. Por lo tanto, en esta memoria lo que debemos hacer es crear la funcionalidad que permita llenar dichas interfaces con la información correcta. Esta información será extraída de la base de nuestro sistema, donde lo más relevante es el historial de emociones de la persona monitoreada.

A medida que esta memoria fue avanzando, algunas de las opciones e interfaces la aplicación recibida, dejaron de ser útiles. Esto fue debido a las diferentes decisiones de diseño que debieron tomarse en el marco de esta memoria.

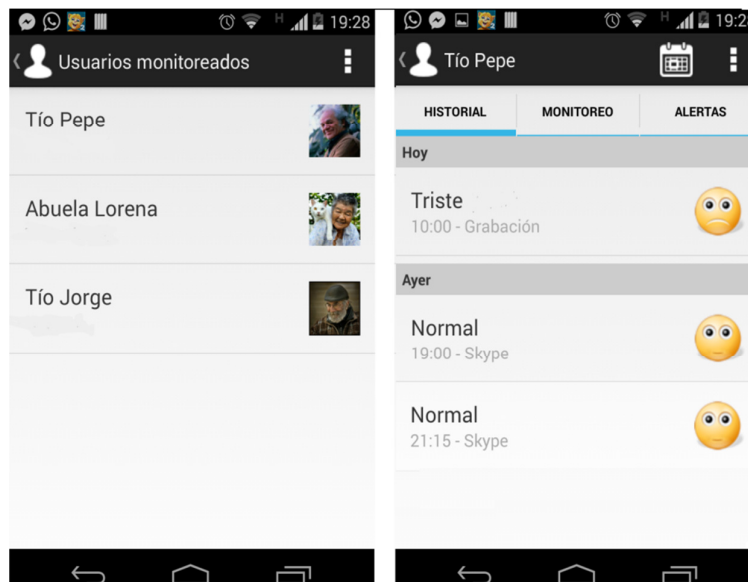
Sin embargo, en esta primera versión de la aplicación, dichas opciones no fueron eliminadas de la aplicación, debido a que este desarrollo es una primera versión, y es posible que en futuras evaluaciones descubramos que dichas interfaces vuelvan a tener a sentido. Por otro lado, estas interfaces están diseñadas para presentar la información de diversas maneras, las cuales no será abordado en esta memoria, por ser parte del trabajo realizado por otra persona.

Por lo tanto, sólo describiremos la funcionalidad desarrollada para apoyar a las interfaces básicas, las cuales nos permiten revisar las emociones de la

personas monitoreadas. Además, también se brinda un historial en forma de lista, el cual nos presenta los distintos estados que la persona monitoreada ha pasado en el día. Debido a que estaremos censando emociones la mayor cantidad de veces posible, acotaremos esta lista a los últimos 3 días.

Para proveer este servicio simplemente le pedimos la información necesaria al servidor, mediante requests HTTP, desde la aplicación en Android hacia nuestro servidor. En el servidor existen dos archivos en PHP; el primero de ellos es *usuarios\_monitoreados.php*, el cual dado el id de un usuario nos devolverá una lista de los usuarios que el mismo está monitoreando. El segundo archivo es *historial\_emociones.php*, el cual recibe el id de un usuario monitoreado y nos devuelve el historial de los últimos 3 días para dicha persona. Para realizar la comunicación entre el servidor y la aplicación de Android usamos objetos en formato Json, los cuales pueden ser recibidos y procesados por la aplicación desarrollada.

La figura 13 nos muestra la interfaces finales que se desarrollaron en esta memoria. Estas dos interfaces están completamente funcionales y se alimentan directamente desde la base de datos de la aplicación.



**FIGURA 13 Interfaces de la aplicación de visualización de datos**

#### 4.5. Subsistema de alertas

Para el desarrollo del subsistema de alertas lo más importante a considerar es el cuándo nos realmente debemos enviar una alerta. Esto es debido a que dicha alerta debería producir una acción de respuesta por parte

del usuario receptor, y por ende consumir parte de su tiempo, el cual podría ser escaso.

Para esto hemos decidido mandar alertas sólo una vez al día. Con eso nos aseguramos de tener la información de un día completo, para poder asegurar que la persona realmente se encontraba por un estado depresivo. La cantidad de veces que serán procesadas las emociones de una persona en el día, es un número variable. Es por esto que para el correcto funcionamiento de nuestro sistema de alarmas, trabajaremos en base a porcentajes. El algoritmo de envío de alertas funciona de la siguiente manera:

- Toma todas las grabaciones del día para uno de los usuarios monitoreados.
- De dichas grabaciones cuenta la cantidad de veces que el resultado de la emoción presente en las grabaciones es “ira”, “tristeza” o “miedo”.
- Calcula el porcentaje de grabaciones que contienen las tres características antes mencionadas, sobre el total de elementos que hay en las grabaciones.
- Si el porcentaje supera el 75% de las grabaciones, enviamos un email a la persona “a cargo” de dicho usuario (o sea, al monitor), para indicarle que la persona que está monitoreando pasa por un cuadro de emociones depresivas.

El algoritmo antes mencionado, nos sirve para quitar posibles errores de cálculo en algunos de nuestros subsistemas, dado que la decisión de enviar o no el correo pasa por el análisis del total de las grabaciones. Esto significa que la persona tiene que estar la mayor parte del día pasando por cuadros depresivos, para que efectivamente se envíe una alerta.

Por otro lado, si alguno de nuestros algoritmos presentaran algún error de cálculo, es muy poco probable que este error se presente durante todo el día y sobre una gran cantidad de grabaciones. Es por esto que este algoritmo nos asegura tener un gran porcentaje de certeza antes enviar un correo al usuario.

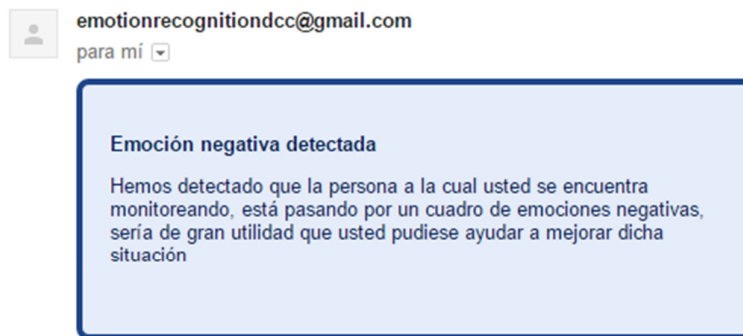
Este algoritmo está implementado por completo en una función la PHP, la cual está alojada en el servidor de nuestra aplicación. Dicha aplicación también se encarga del envío de los emails (alertas) correspondientes. Esto lo hace repitiendo aquel algoritmo para cada uno de los usuarios de nuestra aplicación. Tal como lo dijimos anteriormente, este proceso de envío de alertas se lleva a cabo una vez al día, para no molestar a los usuarios y para poder contar con la mayor cantidad posible de grabaciones antes de decidir si una potencial alerta debe o no ser enviada. Este proceso se lleva a cabo al finalizar el día, específicamente a las 23:00 hrs (horario chileno). Con esto no aseguramos de

no tener tanto tráfico con la base de datos, además de tomar la mayor cantidad de muestras posibles.

Para poder proveer el servicio deseado, al menos el poder realizar el procedimiento todos los días a las 23:00 hrs., necesitaremos utilizar tareas programadas. Es decir, hacer que nuestro sistema operativo inicie una tarea en aquel horario, la cual ejecute la aplicación que se encargará de realizar el envío de emails.

La mayor parte de los servicios programados en las memorias que esta memoria utiliza como base fueron desarrollados en un ambiente de Windows, con lo cual decidimos seguir trabajando bajo ese ambiente, en vez de migrar a cualquier otro. Es por esto que para poder realizar una tarea programada, debemos utilizar el programador de tareas de Windows.

La figura 14 nos muestra un ejemplo del email que le llegaría a un usuario, si alguna de las personas monitoreadas por él/ellas estuviese pasando por un cuadro de emociones depresivas.



**FIGURA 14 Email de ejemplo de emociones negativas detectadas**

#### **4.6. Subsistema de detección de actividad**

Por temas de tiempo se decidió no diseñar un sistema completo de detección de actividad. En vez de eso se decidió utilizar un algoritmo que busca grabar la mayor parte del tiempo que le sea posible. De esa manera, la solución recolecta la mayor cantidad de datos posibles para analizar la emoción por la que está pasando una persona en su día-a-día.

De todas maneras, sí existe un mecanismo básico de detección de actividad dentro de nuestra solución. Este mecanismo detecta un umbral de sonido antes de realizar cualquier tipo de grabación; es decir, verifica que exista algún tipo de sonido en las cercanías del dispositivo antes de realizar la toma de muestras.

Como se mencionó antes, cada vez que se realiza una grabación auditiva, se realiza también una toma de imagen. Por lo tanto, ambos pasos se realizan sí y sólo sí el dispositivo detecta sonidos. Sin embargo, este mecanismo básico no se puede rotular realmente como un sistema de detección de actividad; al menos no de actividad humana, debido a que sólo estamos midiendo un umbral de sonido. Es por eso, que incluso con ruidos fuertes o música a un volumen normal, se puede sobrepasar dicho umbral, y por lo tanto el dispositivo podría realizar grabaciones.

Con lo anterior el software estará tratando de captar datos en todo momento; es decir, intentará realizar grabaciones día y noche. Es por esto que simplemente basta con que el ruido ambiente supere dicho umbral, para que el dispositivo intente realizar una toma de datos. La aplicación tomada como base contaba con una opción (sólo a nivel de interfaz) que le permitía, al usuario monitor, señalar un horario de grabación para una persona monitoreada. Esto en realidad podría llegar a ser muy útil en desarrollos futuros, porque es probable que el familiar de la persona monitoreada, es quien más sabe en qué horarios la persona se encontrará en su casa. Más aun, podría saber cuáles son los horarios en que la misma interactúa con la Tablet. Esto nos permitiría inferir de mejor manera los períodos de monitoreo más apropiados, y también aumentar la certeza acerca de que las mediciones pertenezcan realmente a la persona que deseamos monitorear.

Si bien la implementación actual consideró los 5 subsistemas antes mencionados, existe un subsistema adicional que debió ser implementado por un tema de completitud. Dicho subsistema fue el de administración de usuarios, el cual se explica en la próxima sección.

#### **4.7. Subsistema de administración de usuarios**

En realidad en las aplicaciones que usamos como base para desarrollar esta solución, no hay un módulo administrador de usuarios que nos permita llevar en forma integral y a cabalidad todas las funciones que nuestra aplicación debiese cumplir. Es por esto que debemos desarrollar un sistema en el cual se puedan compatibilizar cada una de las distintas partes de la aplicación.

En un comienzo, es decir, antes de comenzar a desarrollar esta memoria, se tenía una base de datos que entre sus tablas podíamos encontrar las tablas *users* y *monitored\_users*, las cuales se hacían cargo de la información de los distintos usuarios monitores y usuarios monitoreados respectivamente. Como se mencionó antes, llamamos usuario monitor a toda persona que desee monitorear a otra.

Además de estas dos tablas existía la tabla *monitorings*, la cual permitía ligar a un usuario con alguno de los usuarios que él deseaba monitorear. Para la creación de dichos usuarios se recurre a una aplicación Web, la cual está implementada utilizando en el framework Yii. En la actual implementación de este subsistema, una persona crearse un usuario (que asume el rol de monitor), y luego crear un usuario monitoreado. Se decidió hacerlo así para maximizar la flexibilidad en la administración de usuarios, y minimizar la burocracia.

Por otro lado, la aplicación red social para personas de la tercera edad, no posee un sistema de usuarios que sea manejado por la aplicación, sino más bien, utiliza los servicios de Google (Gmail), Facebook y Skype para autenticar a una persona. Por lo tanto, no existe un inicio de sesión real dentro de esta aplicación. Esto genera una gran dificultad para encontrar una manera de relacionar los usuarios de la esta aplicación, con los usuarios que estamos creando mediante la aplicación web, los cuales son utilizados para realizar todo el procesamiento posterior a la toma de datos. Es decir, la toma de datos es realizada en la Tablet PC, y está alojada en la red social. Luego el procesamiento de datos se realiza mediante scripts alojados en el servidor, los cuales son alimentados mediante la base de datos que da origen a la aplicación web. Es por esto que se hace tan importante realizar una relación entre dichos subsistemas de administración de usuarios.

Partiremos por indicar algunas condiciones que se deben dar para la realización de nuestro subsistema de usuarios.

- Dentro de la aplicación SocialConnector [Muñ13a], se deberían realizar capturas de audio e imágenes de la persona monitoreada. Luego, dichas grabaciones e imágenes deben ser enviadas al servidor.
- En el envío al servidor, se debe dejar registro de que la información que se está enviando, pertenece a alguno de los usuarios monitoreados en el sistema.
- El servidor debe procesar la información, y luego de ser necesario mandar un email a uno de los usuarios de la aplicación, es decir a algún familiar de nuestro usuario monitoreado.
- Por último, para un usuario de la aplicación, el servidor debe poder mostrar la información relacionada a cada uno de sus usuarios monitoreados.

Para cumplir con cada una de las condiciones presentadas, cada vez que un usuario monitor desee crear un usuario monitoreado, éste debe asegurarse de usar como *username* el mismo nombre de usuario que dicha persona posee en el Facebook. De esa manera, mediante el análisis del nombre de usuario

podemos establecer los matchings correspondientes entre los diferentes subsistemas. Utilizando esto, desde la Tablet se estarán enviando al servidor archivos, los cuales en su nombre poseerán el identificador único del usuario monitoreado. Así, a la hora de realizar el procesamiento cada uno de los archivos, nos aseguramos de que éstos pertenezcan al usuario correspondiente. Con esto, como cada usuario monitoreado estará relacionado directamente con un usuario de la aplicación, podremos fácilmente decidir si enviar emails, o decidir qué información mostrar a cada usuario monitor a través del servicio de visualización de datos.

#### **4.8. Flujo final del proceso realizado**

En este apartado explicaremos el flujo final que realiza el software, luego de todos los rediseños y ajustes realizados. Es decir, se explica cuáles son los pasos que se llevan a cabo, desde que el software toma los datos hasta que estos son presentados al usuario monitor (o que una alerta es disparada).

1. Nuestra aplicación dentro de la Tablet PC detecta los niveles de sonido a los cuales está expuesta, si dichos niveles sobrepasan el umbral señalado ésta realiza una captura de imagen y comienza con la grabación de sonido. Esto se realiza siempre y cuando la cámara del dispositivo esté disponible para ser usada.
2. La aplicación en la Tablet PC envía los archivos recientemente tomados hacia el servidor, dichos archivos poseen como parte de sus nombres, los identificadores necesarios para la correcta asignación de estados a los distintos usuarios.
3. El servidor al recibir los archivos genera una entrada en la base de datos, esta entrada será posteriormente consumida para realizar el procesamiento.
4. Dentro del servidor existe un script corriendo minuto a minuto, el cual se encarga de tomar cada una de las entradas en la base de datos y realizar el procesamiento correspondiente.
5. Un usuario puede ingresar en la aplicación de visualización de datos para revisar el historial de cada uno de sus usuarios monitoreados.
6. Por último, a las 23:00 hrs. de cada día, el sistema correrá un script el cual se dedicará a revisar qué porcentaje de emociones negativas tuvo cada usuario durante el último período. Si este porcentaje supera el 75% de las grabaciones, se enviará un email al usuario monitor correspondiente (puede ser más de uno), indicando esta situación.

De todo lo anterior existe un punto del cual no hemos hablado en esta memoria, y ese es el script que corre minuto a minuto dentro del servidor. Este script es el que se encarga de realizar el procesamiento completo de cada una de las entradas en el servidor. Este script funciona de la siguiente manera:

- Revisa la base de datos, en busca de todas las entradas que necesitan ser procesadas.
- Para cada una de las entradas, primero que nada toma la imagen asociada a la misma y verifica que en esa imagen se encuentre la persona que se desea monitorear. Es decir, en este paso se está realizando el proceso de autenticación (reconocimiento de identidad).
- Si la autenticación de identidad, resulta negativa se termina el proceso y se realiza una actualización de la base de datos de manera de dejar de manifiesto que no se logró autenticar a la persona.
- Si la autenticación resultó ser positiva, se continúa hacia el sistema de reconocimiento de emociones. Aquí se realiza tanto la detección de emociones mediante la grabación, como también mediante imágenes, utilizando la misma imagen con la que la persona ya fue autenticada.
- Luego de realizar el proceso de reconocimiento de emociones, se actualiza la base de datos dejando todos los datos necesarios para que estos puedan ser utilizados, tanto por el sistema de visualización de datos, como por el sistema de emisión de alertas.

Es importante destacar que el sistema de reconocimiento de emociones mediante las grabaciones de audio, fue completamente diseñado en la memoria del alumno Víctor Rojas [Roj14]. De manera que en esta memoria sólo se le hicieron los ajustes necesarios, para que dicho sistema se acomodara a los distintos cambios que se realizaron a la base de datos, los cuales eran necesarios para el correcto funcionamiento de los demás subsistemas.



#### 4.9. Tabla Resumen de Librerías

En este apartado presentamos una pequeña tabla con los Software más utilizados dentro de nuestra aplicación, esta tabla es presentada a modo de resumen centralizado de las librerías utilizadas.

Sistema	Descripción del Software	Comentarios adicionales
Sistema de Captura de Datos	PHP como lenguaje del servidor del sistema	Para realizar el desarrollo se utilizó la suite de sistemas XAMPP para Windows
	Mysql como servidor de base de datos	
	Framework Yii para desarrollo de página web	
Sistema de captura de Datos	Para el sistema de captura de datos se utiliza una Tablet PC con sistema Operativo Android	
Sistema de Reconocimiento de emociones	OpenCV para realiza reconocimiento de emociones mediante imágenes	Estas dos librerías deben ser compiladas para el sistema en el cual se está trabajando. En la solución final de este trabajo de memoria se incluyen los compilados de estas librerías para un sistema Windows de 64 bits
	Sox como sistema de conversión entre distintos formatos de audio	
	OpenSmile como sistema para extracción de características sonoras	
	Python como lenguaje de desarrollo de scripts de reconocimiento de audio	
	Bash como lenguaje de unión entre los distintos scripts	

## 5. Evaluación preliminar de la solución

En esta sección se reporta las pruebas realizadas para determinar el porcentaje de reconocimiento que nuestros algoritmos de detección de emociones. Primero que nada evaluamos el sistema de reconocimiento de emociones mediante imágenes, realizando 2 pruebas distintas. La primera prueba consistió en utilizar las bases de datos que habíamos considerado para nuestra solución inicial. Para esto utilizamos la base de datos de YALE [Bel97], como modelo para predecir cada una de las imágenes contenidas en la base de datos JAFFE. Los resultados de esta prueba son en realidad poco alentadores. Al finalizar la prueba y medir un total de 122 imágenes obtuvimos sólo 38 correctas, con lo cual nuestro porcentaje de reconocimiento es de un 31%.

En la prueba anterior existen muchos factores que pueden determinar el mal funcionamiento del algoritmo, en particular destacamos el hecho de que cada una de las imágenes dentro de la base de datos JAFFE, corresponde a mujeres japonesas. Éstas poseen rasgos completamente distintos a los que se encuentran en las imágenes de la base de datos de YALE, lo cual influye negativamente en los resultados del algoritmo diseñado.

Para la segunda prueba decidimos utilizar sólo la base de datos de YALE, para verificar si al no tener tantas diferencias étnicas, el algoritmo podía obtener mejores resultados. Para esta prueba separamos el set de imágenes en dos subsets. El primer subset correspondió a las imágenes de 10 de los 15 sujetos de prueba que se encuentran en la base de datos. Las imágenes de los cinco sujetos restantes (segundo subset), se dejaron como imágenes a procesar por el algoritmo. El resultado de esta prueba fue considerablemente mejor. Sobre un total de 25 predicciones, el algoritmo acertó en 16 de ellas, obteniendo así un 64% de asertividad.

Aunque esta evaluación es preliminar, los resultados obtenidos en estos experimentos nos llevan a la conclusión de que el algoritmo es muy dependiente de las personas que están siendo procesadas. Es por esto que para obtener los porcentajes reales de reconocimiento, se deben realizar pruebas con la población que realmente utilizará la aplicación.

Lamentablemente, nos fue imposible realizar este tipo de experimentos en el marco de la memoria, debido a que generar una muestra representativa de los usuarios finales, conlleva a enfrentar importantes dificultades, tanto en término de la cantidad de personas necesarias para alimentar la base de datos, como de la coordinación con la captura de la información correspondiente (es decir, las fotos de ellas).

Con respecto al reconocimiento de emociones mediante registros de audio, no fueron generadas pruebas dentro de esta memoria. Aun así, basándonos en los resultados de pruebas realizadas en el trabajo de Víctor Rojas [Roj14], podemos decir que los porcentajes esperados de reconocimiento de emociones, están cercanos al 70%. Esto se cumple al menos cuando el registro de audio involucra una conversación entre 2 personas.

Estos resultados son preliminares, y por lo tanto se requiere aún bastante trabajo para determinar exactamente cuándo el sistema tiene certeza acerca de la emoción detectada. Determinar esto es parte del trabajo a futuro que se pretende llevar a cabo en esta iniciativa.

## 6. Conclusiones y trabajo a futuro

Antes que nada es importante destacar algunas de las dificultades que se debieron enfrentar durante el desarrollo de este proyecto. Luego de eso, se presentan las conclusiones y el trabajo a futuro.

### 6.1. Dificultades encontradas

La primera y más grande dificultad que tuvimos que enfrentar en la realización del proyecto, es la cantidad de tecnologías distintas que se debieron manejar para poder realizar el proyecto final. Con esto nos referimos a que el proyecto tiene tantas directrices distintas, que fue bastante complicado poner todo a funcionar. De hecho se gastó bastante tiempo de desarrollo simplemente haciendo funcionar, de manera correcta, cada uno de las piezas de software consideradas en nuestra aplicación. En particular el software de reconocimiento de emociones mediante grabaciones auditivas fue especialmente problemático.

Dicho software utiliza diversos programas para poder llevar a cabo la tarea final de reconocer dichas emociones, puesto que la tarea es sumamente compleja. Algunos estos programas, al ser descargados debieron ser compilados para poder correr dentro del ambiente de desarrollo, y luego de eso ponerlos a trabajar de manera conjunta.

Otra gran dificultad fue la necesidad de comprender a cabalidad los dos sistemas desarrollados por mis compañeros de universidad (Víctor Rojas y Diego Muñoz). Si bien es cierto ambas implementaciones cumplen con los estándares que se esperarían, el comprender el estilo de programación de cada una de estas personas, representó una dificultad adicional para la realización de nuestro proyecto.

### 6.2. Conclusiones de la memoria

Dentro de nuestro proyecto de memoria logramos cumplir con casi todos los objetivos específicos que nos planteamos en un comienzo. Con respecto al primer objetivo planteado “*automatizar el sistema de reconocimiento de emociones por voz*”, está completamente cumplido. El sistema implementado se mantiene continuamente enviando grabaciones desde un dispositivo móvil hacia el servidor, el cual está realizando el procesamiento de cada uno de los audios de manera autónoma.

Con respecto al segundo objetivo planteado “*desarrollar el sistema de reconocimiento de emociones mediante imágenes*” también está completado,

considerando de que la emoción que se busca detectar es tristeza. Si se quisieran detectar otras emociones, claramente habría que extender la solución.

Con respecto al tercer objetivo planteado *“incluir en el sistema red social para personas de la tercera edad (SocialConnector), los mecanismos necesarios para la captura de datos”*, también podemos decir que está cumplido. De hecho se realizaron las correcciones necesarias al sistema de usuarios para poder enviar de manera correcta los archivos desde la tablet PC, hacia el servidor.

Con respecto al cuarto objetivo propuesto *“generar un sistema funcional que permita la visualización de la información resumen resultante (de los estados anímicos)”*, podemos decir que se logró generar un sistema básico, el cual puede presentar los datos de una manera simple y legible para el usuario. Dicho sistema está desarrollado en base a las interfaces validadas en el trabajo de Víctor Rojas [Roj14].

Con respecto al quinto objetivo propuesto *“Generar un sistema de alertas para los miembros de una comunidad familiar”*, se realizó un sistema que permite alertar los estados de ánimo negativos de una persona monitoreada, a la persona que decidió monitorearlo. Estas alertas se realizan mediante el envío de un email a la cuenta indicada por el usuario monitor, durante el proceso de configuración.

Con respecto al último objetivo propuesto *“Probar la solución desarrollada, en un conjunto acotado de personas”*, este objetivo no fue cumplido, y la razón fue un tema de tiempo. El proceso de desarrollo llevó más de lo previsto, y el proceso de evaluación con los usuarios finales de la aplicación era complejo y requería bastante tiempo adicional.

Además de los objetivos que fueron concebidos en el diseño inicial de la solución, se desarrolló un subsistema de reconocimiento de identidad de las personas, mediante el procesamiento de imágenes. Este subsistema funciona como único mecanismo reconocimiento de identidad para la solución implementada.

### **6.3. Trabajo a futuro**

El principal trabajo que queda pendiente es el desarrollo de un sistema de reconocimiento de identidad, mediante grabaciones de audio. Esto nos permitiría realizar una separación completa de los sistemas de reconocimiento de emociones; es decir, separar el reconocimiento de emociones mediante una vía auditiva, del reconocimiento que emplea imágenes. Así es posible tener una

mayor cantidad de muestras diarias, debido a que no sería necesario que la cámara estuviera disponible para tomar una muestra auditiva.

En particular, si se desarrollara esto, se podrían tomar muestras aun cuando el sistema esté realizando una video conferencia vía Skype, lo cual sería un comportamiento altamente deseado.

Otro punto importante en el que se podría mejorar, sería en el subsistema de visualización de datos. Como se mencionó en este informe, se completó la funcionalidad requerida por el subsistema de visualización de datos. Aun así, dentro de estas interfaces de dicho subsistema existían algunas formas distintas de visualización de datos, las cuales no fueron tratadas y podrían dar al usuario más feedback acerca del estado de ánimos de sus familiares.

El siguiente punto en el cual se debería trabajar fue tomado desde una memoria anterior a ésta, pero tiene la suficiente importancia como para ser considerado también en esta memoria. Este punto se refiere a confeccionar una política de privacidad, que asegure a los usuarios de la aplicación, que cada una de sus grabaciones e imágenes serán utilizadas sólo para efectos de nuestra aplicación, y sólo se extraerá la información de identidad y emociones de las muestras.

Como se mencionó anteriormente, existe una aplicación web que es usada como servidor, pero que además se encarga de la creación de usuarios. En el estado actual, la aplicación no es muy usable para un usuario final, por lo tanto es un excelente punto de mejora.

## 7. Bibliografía

- [Cor13] R. Cornejo, M. Tentori, J. Favela, Enriching in-person encounters through social media: A study on family connectedness for the elderly, *International Journal of Human-Computer Studies*, (2013).
- [Cza06] S.J. Czaja, N. Charness, A.D. Fisk, C. Hertzog, S.N. Nair, W.A. Rogers, J. Sharit, Factors predicting the use of technology: findings from the Center for Research and Education on Aging and Technology Enhancement (CREATE), *Psychology & Aging*, 21 (2006) 333-352.
- [Hou01] J.S. House, Social Isolation Kills, But How and Why?, *Psychosomatic Medicine*, 63 (2001) 273-274.
- [Kar12] A. Karimi, C. Neustaedter, 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, (2012), pp. 127-130.
- [Muñ13a] D. Muñoz, F. Gutierrez, S.F. Ochoa, N. Baloian. Enhancing Social Interaction Between Older Adults and Their Families, *Proc. of the 5th International Work-conference on Ambient Assisted Living (IWAAL'13)*, Guanacaste, Costa Rica. December (2013), LNCS 8277, pp. 47-54.
- [Muñ13b] D. Muñoz, R. Cornejo, S.F. Ochoa, J. Favela, F. Gutierrez, M. Tentori. Aligning intergenerational communication patterns and rhythms in the age of social media, *Proceedings of ChileCHI'13*, ACM Press. Temuco, Chile. November 11–15, (2013).
- [New09] N.E. Newall, J.G. Chipperfield, R.A. Clifton, R.P. Perry, A.U. Swift, J.C. Ruthig, Causal beliefs, social participation, and loneliness among older adults: A longitudinal study. *Journal of Social and Personal Relationships*, 26 (2009) 273-290.
- [Roj14] V. Rojas. Monitoreo del estado anímico de adultos mayores, mediante el procesamiento de la voz. *Memoria de Ingeniería Civil en Computación*, DCC, Universidad de Chile. (2014).
- [Yof01] Yoav Freund. An adaptive version of the boost by majority algorithm (2001).
- [Bel97] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1997), pp. 711-720.