Instituto Tecnológico de Hermosillo

División de Estudios de Posgrado e Investigación

Diseño e implementación de una aplicación móvil para el desarrollo de habilidades sociales en los niños mediante el reconocimiento del lenguaje corporal básico del rostro utilizando visión por computadora

T E S I S

Presentada por:
Ing. Ramón Omar Parra Guerrero

Como requisito parcial para
obtener el grado de

Maestro en Ciencias de la Computación

Director de tesis:
M.C. Ana Luisa Millán Castro

Hermosillo, Sonora, México Agosto de 2018

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo

SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/278/18.
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS.

23 de Agosto de 2018

C. RAMÓN OMAR PARRA GUERRERO,
P R E S E N T E.

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre "***DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA EL DESARROLLO DE HABILIDADES SOCIALES EN LOS NIÑOS MEDIANTE EL RECONOCIMIENTO DEL LENGUAJE CORPORAL BÁSICO DEL ROSTRO UTILIZANDO VISIÓN POR COMPUTADORA***", que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

A T E N T A M E N T E

M.C. ANA LUISA MILLAN CASTRO
DIRECTORA

M.G. MARCELA PATRICIA VÁZQUEZ VALENZUELA
CO-DIRECTORA

M.C. CÉSAR ENRIQUE ROSE GÓMEZ
SECRETARIO

M.C. SONIA REGINA MENESES MENDOZA
VOCAL

M.C.O. ROSA IRENE SANCHEZ FERMÍN
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

S.E.P.

INSTITUTO TECNOLÓGICO DE HERMOSILLO
DIVISIÓN DE ESTUDIOS DE POSGRADO

RISF/momv*

Av. Tecnológico S/N Col. El Sahuaro, C.P. 83170
Hermosillo, Sonora. Tel. (662) 2-606500 Ext. 136
e-mail: depi_hermosillo@tecnm.mx
www.tecnm.mx
www.ith.mx

Resumen

Anteriormente conocido como el Síndrome de Asperger, el Trastorno de Comunicación Social es un trastorno del neurodesarrollo que se caracteriza por las deficiencias existentes en la comunicación. Más específicamente, las personas diagnosticadas con este trastorno tienen dificultad con la pragmática del lenguaje, las reglas no habladas y sutiles del lenguaje.

Hoy en día para poder ayudar a las personas, especialmente a niños, diagnosticados con dicho trastorno a mejorar sus habilidades sociales es a través de terapias. Una de estas terapias consiste en ayudarlos a reconocer las emociones que se encuentran detrás de un rostro. Es decir, les enseñan que, si una persona está sonriendo, debe ser que dicha persona está feliz. Sin embargo, esta terapia suele ser muy personalizaad y sólo se aplica en instituciones especializadas.

Este trabajo de tesis presenta el desarrollo de una aplicación móvil que trabaja como una terapia asistida por computadora para niños de 8 a 12 años con el propósito de mejorar sus habilidades sociales. Esta aplicación se implementa a través de una serie de ejercicios que hacen uso de técnicas avanzadas de visión por computadora para poder reconocer y clasificar las expresiones faciales. Para poder completar los ejercicios, el usuario tendrá que realizar distintas expresiones faciales para que la cámara frontal del dispositivo móvil la pueda capturar.

Abstract

Previously known as the Asperger's Syndrome, the Social Communication Disorder is a neurodevelopment disorder characterized by deficits in social communication. Specifically, the area where people diagnosed with this disorder have more difficulties is with pragmatics, the subtle and no-spoken rules of the language.

Nowadays in order to help people, specially young children, diagnosed with said disorder to improve their social skills is through the use of therapies. One of these therapies consists to help them recognize emotions behind a facial expression. That is, to teach them that if a person is smiling it must be because that person is currently feeling happy. However, this therapy is usually extremely personalized and it's only imparted on specialized institutions.

This thesis presents the development of a mobile application that works as a computer assisted therapy for children between the ages of 8 and 12 to help them improve their social skills. This will be implemented through a series of exercises through the use of advanced computer vision techniques to recognize and classify facial expressions. In order to complete the exercises, the user will have to make different facial expressions in order to be captured by the frontal camera of the mobile device.

Agradecimientos

A mi directora de tesis

M.C. Ana Luisa Millán Castro, por el apoyo y la confianza brindada estos últimos años, por la buena comunicación, las correcciones y comentarios del presente trabajo y del proyecto. Muchas gracias por poder contar siempre con usted cuando lo necesitaba.

A los sinodales

M.C. César Enrique Rose Gómez y M.C. Sonia Regina Meneses Mendoza, por haber aceptado ser revisores del presente trabajo y por todo el tiempo que invirtieron en leerlo y comentarlo.

A mis compañeros

Por toda la ayuda e irrepetibles momentos que me brindaron durante los últimos dos años.

Al colegio EDIA

M.C. Marcela Patricia Vázquez Valenzuela, por todo el apoyo y la valiosa orientación que tuvimos al trabajar con los maestros y alumnos del colegio EDIA.

A Conacyt

Por la beca con número 446275, otorgada durante dos años para realizar los estudios de maestría.

Dedicatoria

Dedico esta tesis a mis padres, Ramón y Patricia, quienes me han entregado todo su apoyo, confianza y cariño a lo largo de toda mi vida. Sin ellos no sería quien soy actualmente.

A mis hermanas, Patty y Dafne, por siempre estar presentes y darme ánimos cuando lo necesitaba.

A toda mi familia, por su constante apoyo que me han dado siempre.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Una de las características más importantes que tenemos como seres humanos es la de poder comunicarnos: ser capaces de hablar y discutir con otra persona, poder escribir y leer distintas ideas, decir con una mirada cómo es que nos sentimos, o simplemente sonreír porque algo nos resultó divertido. Sin embargo, existen muchas personas a las que no les resulta natural poder expresar sus sentimientos, y muchas de ellas, si no es que la mayoría, viven aisladas e incomprendidas por el resto de la sociedad. A estas personas se les clasifica con el trastorno de comunicación social (TCS), anteriormente conocido como el síndrome de Asperger. Para comprender de qué trata este trastorno, primero tenemos que tener claro en dónde es que se encuentra.

El trastorno de comunicación social se ubica dentro de los trastornos del neurodesarrollo. Este tipo de trastornos son alteraciones asociados al desarrollo de las funciones del sistema nervioso central. Tales alteraciones afectan las habilidades que tiene una persona para recibir, procesar, almacenar y responder la información [1]. Según el Manual Diagnóstico y Estadístico de los Trastornos Mentales, Quinta Edición (*Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition*, DSM-V): "los trastornos se manifiestan normalmente de manera precoz en el desarrollo, a menudo antes de que el niño empiece la escuela primaria, y se caracterizan por un déficit del desarrollo que produce deficiencias del funcionamiento personal, social, académico u ocupacional. El rango del déficit del desarrollo varía desde limitaciones muy específicas del aprendizaje o del control de las funciones ejecutivas hasta deficiencias globales de las habilidades sociales o de la inteligencia" [2].

Como podemos ver, este trastorno se caracteriza principalmente por las deficiencias que se tienen en el lenguaje del individuo que lo padece, esto es, en el habla y en la comunicación. Con el habla nos referimos específicamente a la articulación, fluidez y voz; y con la comunicación nos referimos a todo el comportamiento verbal (pragmática y semántica) y no verbal de la persona (lenguaje corporal y contexto). A su vez, se distinguen entre 5 tipos diferentes de trastornos que se encuentran dentro del trastorno de comunicación social: el trastorno del lenguaje, el trastorno fonológico, el trastorno de la fluidez de inicio en la infancia (tartamudeo), el trastorno de comunicación social (pragmático), y el trastorno de la comunicación no especificado [2].

El problema con el que más se enfrentan aquellos niños diagnosticados con el trastorno de comunicación social es con el de comunicación pragmática, las reglas no habladas y sutiles del lenguaje hablado. Los niños que padecen este trastorno no comprenden la dirección que toma una conversación, monopolizan las conversaciones o las interrumpen [3]. En México se estima que uno de cada 300 niños se encuentra dentro del espectro de este trastorno, y cada año se registran más de 6 mil casos nuevos [4].

El presente trabajo explica el diseño y la implementación de una aplicación móvil que ayuda a mejorar el reconocimiento y la reproducción de expresiones faciales en niños de 8 a 12 años que hayan sido diagnosticados con este trastorno. Para realizar esta tarea, se usaron técnicas de procesamiento de imágenes, visión por computadora y aprendizaje automático para reconocer, catalogar y calificar las expresiones faciales que haga un niño en tiempo real. De esta forma, el niño podrá practicar y le ayudará a reconocer las expresiones faciales que hace él mismo y las demás personas.

1.1. Antecedentes

La tarea principal del sistema es el reconocimiento y la clasificación de las expresiones faciales del usuario en tiempo real. Realizamos esta tarea a través de la cámara frontal del dispositivo móvil para capturar y analizar el rostro del usuario. Lo anterior puede lograrse a través de diferentes técnicas: por una parte, está la visión por computadora tradicional, que se enfoca en extraer características de una imagen para formular una manera de representar y poder clasificar. La otra, es a través de técnicas de aprendizaje automático, más concretamente, redes neuronales.

La visión por computadora no es un tema nuevo, ya que el estudio y práctica de este eje de investigación se ha ido desarrollando desde los años 60. Empezando por los trabajos de Larry Roberts, quien por muchos, es considerado el padre de la visión por computadora. En su tesis de doctorado, Roberts discute la posibilidad de extraer información geométrica de bloques de tres dimensiones a partir de imágenes de dos dimensiones con diferentes perspectivas [5].

La visión por computadora tiene dos objetivos diferentes, el biológico y el de ingeniería. El objetivo que se trata de cumplir desde el punto de vista biológico es el de crear un modelo computacional que se asemeje a la visión humana. Por otra parte, desde el punto de vista de la ingeniería, su objetivo es el de construir un sistema autónomo [5].

A pesar de las dificultades que se ha tenido, se ha generado suficiente conocimiento para poder desarrollar librerías de visión por computadora, tales como OpenCV o el *Computer Vision System Toolbox* de MATLAB, que ponen a nuestra disposición poderosas herramientas matemáticas para manipular, procesar e interpretar imágenes usando la computadora. Asimismo, en los últimos años se ha incorporado el uso de tecnologías y técnicas del área de aprendizaje automático (*machine learning*), tales como las redes neuronales y el uso del aprendizaje profundo (*deep learning*), para mejorar aún más los sistemas de visión por computadora.

Si a esto se le suma la tendencia que existe hoy en día de contar con un dispositivo móvil, donde las personas tienen en casi todo momento una pequeña computadora con una cámara, se abre un sinfín de posibilidades para proyectos que desarrolladores e investigadores pueden explorar. Entre ellos se encuentran proyectos de detección y reconocimiento de rostros, escáneres de documentos y fotografías, aplicaciones que usan la realidad aumentada, traductores en tiempo real, entre otros.

Recordando que el mayor problema con el que lidian las personas diagnosticadas con trastorno de comunicación social es con el de comunicación pragmática, el lenguaje no hablado, se puede ver entonces cómo la visión por computadora llega a ser herramienta óptima para poder implementar una aplicación que pueda ayudar a estas personas.

1.2. Planteamiento

Teniendo en cuenta los antecedentes, la presente investigación pretende atender la siguiente problemática: ¿Cómo implementar un sistema de reconocimiento del lenguaje corporal básico del rostro, que ayude a los niños con trastorno de comunicación social a trabajar su habilidad, para distinguir emociones al imitar las diferentes expresiones indicadas?

Lo anterior nos plantea las preguntas de investigación:

- ¿Cuáles emociones básicas debe de reconocer el sistema?
- ¿Cómo diseñar una herramienta de software que ayude a los niños con trastorno de comunicación social?
- ¿Cómo funciona el reconocimiento de rostros?
- ¿Cómo implementar un sistema que considere el reconocimiento corporal básico del rostro?

1.3. Objetivos

Para lograr el desarrollo del sistema que propone este trabajo se han planteado los siguientes objetivos.

1.3.1. Objetivo general

Diseñar e implementar una aplicación móvil que mediante el reconocimiento básico del rostro, ayude a los niños con trastorno de comunicación social a trabajar su habilidad para distinguir emociones, imitando las diferentes expresiones indicadas, utilizando visión por computadora.

1.3.2. Objetivos particulares

- Estudio exhaustivo de los diferentes aspectos que inciden en el tema, específicamente en el trastorno de comunicación social.

- Determinar las emociones básicas para que las identifique el sistema a desarrollar.
- Reconocer el lenguaje corporal básico del rostro a través de visión por computadora.
- Analizar las características de los dispositivos móviles a utilizar. Implementar un sistema inteligente para el desarrollo de habilidades sociales.
- Analizar las características y funcionalidades de diferentes librerías de visión por computadora y aprendizaje automático en dispositivos móviles.

1.4. Justificación

Una de las formas en las que se ayuda a mejorar la calidad de vida de las personas con diferentes trastornos es a través del software. Existen empresas, organizaciones e individuos que han desarrollado diferentes sistemas que los ayudan en distintas áreas de la vida como la formación de hábitos, recordatorios, creación de rutinas, reconocimiento de situaciones sociales, reconocimiento y seguimiento de emociones, entre otros [6].

La idea de desarrollar un sistema como el propuesto surge al observar el auge que existe hoy en día con los dispositivos móviles y la falta de una aplicación en el mercado que ayude a resolver la problemática planteada. Específicamente, no existe una aplicación que utilice el reconocimiento de gestos para ayudar a las personas con trastorno de comunicación social.

1.5. Alcances y delimitaciones

Este proyecto de tesis se realizó en un periodo de 2 años, y se desarrolló en el laboratorio de sistemas inteligentes del Tecnológico Nacional de México / Instituto Tecnológico de Hermosillo. El alcance que se propone con el proyecto es que pueda ser utilizado por cualquier niño de 8 a 12 años que haya sido diagnosticado con trastorno de comunicación social.

Los estudios se realizaron para el Centro Regional de Formación Docente e Investigación Educativa del Estado de Sonora (CRFDIES), y como caso de estudio se consideró el colegio Educación Dinámica Actualizada (EDIA).

1.6. Metodología

Para desarrollar este proyecto de tesis, la metodología que se utilizó consistió en tres diferentes fases: el estudio del estado del arte, el análisis y diseño, la implementación y pruebas de la aplicación móvil.

A continuación, se describirán cada una de las fases y se explicará en qué consisten.

1.6.1. Estudio del estado del arte

En la primer fase, se encuentra el estudio del estado del arte. Aquí se tiene como objetivo afianzar los conocimientos con los que se trabajarán en esta tesis, saber qué investigaciones se han hecho anteriormente relacionadas al tema, conocer a los autores líderes en sus áreas, y saber cuáles son los sistemas de vanguardia que existen dentro del tema de interés.

Los temas que se presentarán en esta sección son: el trastorno de comunicación social, el cómputo afectivo, la visión por computadora, y por último, el aprendizaje automático.

1.6.2. Análisis y diseño de la aplicación móvil

En la segunda fase, el análisis y el diseño, se utilizarán herramientas de modelado de la ingeniería de software para describir con más detalle qué es lo que hará el sistema y cómo lo hará.

Para realizar este modelado, la presente tesis se enfocará en los resultados de las investigaciones sobre del trastorno de comunicación social, ya que habrá que basarse en cómo el niño va a interactuar con la aplicación, cuáles son las emociones que el sistema podrá detectar, y qué tipo de retroalimentación se podrá generar.

1.6.3. Implementación y pruebas

Como objetivo final, la tesis consiguió terminar una aplicación móvil que se puso en práctica en el colegio EDIA. Para su implementación se utilizó la visión por computadora, el reconocimiento y procesamiento de imágenes, además de una red neuronal para realizar la clasificación de las expresiones faciales. Para las pruebas, se aplicó un cuestionario basado en el modelo de aceptación tecnológica para probar si tuvo éxito o no la aplicación.

1.7. Organización

Esta tesis se desarrolla en cuatro diferentes capítulos: el estado del arte, el análisis y diseño de la aplicación móvil, la implementación de la misma y el análisis de los resultados.

En el capítulo dos, dedicado al estado del arte, se desarrollan los temas de investigación que se mencionaron anteriormente: el trastorno de comunicación social, el cómputo afectivo, la visión por computadora, y el aprendizaje automático. Además de que se hace referencia a otros trabajos relacionados con esta investigación.

En el capítulo tres, el análisis y diseño del sistema, se hace uso de las diferentes herramientas de modelado de ingeniería de software con el fin de poder integrar lo que se investigó acerca del trastorno de comunicación social. Estas herramientas incluyen el análisis de requerimientos del sistema, modelado a través de diagramas UML, patrones de diseño, entre otros ejemplos.

En el cuarto capítulo, se explica a detalle la implementación del sistema en una aplicación móvil; además de incluir las pruebas de funcionalidad hechas al sistema.

Finalmente, el capítulo cinco presenta el análisis de resultados, y en el capítulo seis, las conclusiones obtenidas de la presente investigación y sugiere algunos trabajos a futuro. También se expone a detalle cuál fue el procedimiento para realizar el reconocimiento de rostros y su análisis.

Capítulo 2

Estado del arte

Dentro de este capítulo se definirán los conceptos básicos que sirven como fundamento de este trabajo de tesis. Estas definiciones provienen del estado del arte de cada uno de los temas con los que se trata. Antes de profundizar en cada uno de ellos, se hablará un poco acerca de cuáles son las instituciones con las que se está trabajando para hacer realidad este proyecto: el CRFDIES y el instituto EDIA.

Más adelante, se verá a qué se refiere con el trastorno de comunicación social. Aquí se definirán cuáles son las características de las personas que padecen de este síndrome, cómo es que se diagnostica, de qué manera influye el lenguaje, y cómo es que estas personas perciben el mundo que los rodea. Se tomarán dos definiciones principalmente, una dada de forma oficial, a través del ya mencionado Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-V), y la otra dada por uno de los psicólogos líderes del área de este trastorno, Tony Attwood.

El siguiente tema abordará el cómputo afectivo. Un área de extrema importancia para este proyecto de tesis, porque puede verse como el enlace entre la psicología y la computación. En esta sección se explicará cómo fue que surgió y se mencionarán algunos casos de uso que tiene.

Una vez que se haya definido qué es el trastorno de comunicación social y cómo es que se puede ayudar a partir del cómputo afectivo, se empezará a definir los conceptos más técnicos del proyecto. El primero se trata de la visión por computadora, donde se aclarará algunos conceptos básicos como la definición de lo que es la visión humana y la visión artificial, cómo es que las computadoras pueden capturar e interpretar las imá-

genes, y cuáles son los métodos y algoritmos principales para obtener las características de una imagen.

Después, se hablará acerca del aprendizaje automático. Este es un tema muy relacionado con la visión por computadora, ya que muchos métodos del aprendizaje de máquinas pueden usarse para interpretar y clasificar imágenes de un sistema de visión.

Para finalizar, se listarán y explicarán otros trabajos relacionados al que se está presentando. Estos trabajos son los antecedentes a este proyecto, y muchas de sus ideas se utilizaron para mejorar lo que se está desarrollando.

2.1. Instituciones relacionadas

Este proyecto se realiza para dos instituciones: el Centro Regional de Formación Docente e Investigación Educativa del Estado de Sonora, y el Colegio EDIA. En esta sección se explicará el objetivo de cada institución.

2.1.1. CRFDIES

El Centro Regional de Formación Docente e Investigación Educativa del Estado de Sonora es un organismo descentralizado de la Administración Pública Estatal con personalidad jurídica, patrimonio propio y autonomía técnica y académica para decidir sobre su oferta educativa, y demás servicios académicos y sectorizados a la Secretaria de Educación y Cultura. Este organismo privilegia la investigación básica y aplicada, particularmente la referida a los procesos de formación de docentes. Contempla además el uso intensivo de las tecnologías de la información y la comunicación para potenciar el cumplimiento de sus funciones sustantivas: docencia, investigación,vinculación y comunicación.

Establece una vinculación orgánica entre instituciones de educación superior relacionadas con la formación docente y la investigación educativa. El Centro Regional es el primero de los cinco centros que surgen a nivel nacional con el propósito de desarrollar investigación educativa y fortalecer la formación de profesionales de la educación.

El Centro Regional de Formación Docente e Investigación Educativa del Estado

de Sonora tiene por objeto fortalecer la calidad de la formación inicial y continua de los maestros, mediante el desarrollo de programas pertinentes a los sistemas educativos nacionales y estatales, la generación de líneas de investigación y aplicación del conocimiento, así como la construcción de modelos de innovación e intervención que incidan favorablemente en las prácticas educativas [7].

Misión

La misión del CRFDIES es ser un centro de formación de profesionales de la educación altamente especializados, capaces de mejorar su entorno a través de proyectos de intervención e investigación que contribuyan a elevar la calidad educativa de la región [7].

Visión

El Centro Regional de Formación Docente e Investigación Educativa del Estado de Sonora es reconocido a nivel nacional e internacional como institución de educación superior innovadora y de excelencia en la investigación educativa, vinculación y formación de profesionales de la educación, que contribuyen a la generación y divulgación del conocimiento en la región noroeste de México [7]. El logo de esta institución puede verse en la figura 2.1.

Figura 2.1: Logo del CRFDIES

2.1.2. Colegio EDIA

Colegio EDIA fue fundado en el año de 1980, originalmente con el nombre de "Instituto Psicopedagógico", surge por la gran necesidad que existía de una escuela que atendiera a niños, que sin presentar una discapacidad, su rendimiento escolar era bajo. Después de tres años, el Colegio registró un crecimiento muy significativo, y a partir de septiembre de 1994 pasó a ser "Centro pedagógico". A los veinte años de su fundación, se reestructuró y pasó a ser Colegio EDIA, que significa Educación Dinámica Actualizada.

Actualmente, el colegio se encuentra consolidado, y abarca los niveles de: primaria, secundaria y preparatoria. Cuenta con planes muy definidos de crecimiento y mejoramiento, además de un notable reconocimiento de la sociedad.

Objetivos

El Colegio EDIA tiene los siguientes objetivos [8]:

1. Brindar una educación de calidad que responda a las necesidades específicas del estudiante.
2. Mantener Protocolos de Intervención Psicopedagógica a la vanguardia, homologados en todos los niveles.
3. Incrementar los niveles de eficacia en los procesos de gestión de la calidad.
4. Incrementar la matrícula en nivel secundaria.
5. Elevar el nivel de profesionalización de nuestro personal.
6. Incrementar los niveles de aprovechamiento escolar.

Misión y visión

El Colegio EDIA es una institución que forma estudiantes competentes a través de la atención centrada en las necesidades del alumno, en un ambiente de profesionalización docente y bajo un diseño pedagógico que promueve el aprendizaje de la comunidad educativa. En el colegio EDIA [8]:

1. El aprendizaje se centra en el alumno para el desarrollo de competencias.
2. Ofrecer opciones de vanguardia para la intervención psicopedagógica del Trastorno por Déficit de la Atención con o sin Hiperactividad.
3. Impulsar el perfeccionamiento docente en la comunidad escolar.
4. Promover la investigación y difusión de aspectos psicopedagógicos y normativos para la atención a la diversidad.
5. Ser una institución líder en la gestión de la calidad.
6. Compartir el conocimiento con la comunidad.
7. Contribuir al cuidado del medio ambiente fomentando una conciencia ecológica comunitaria. Su logo puede verse en la figura 2.2.

Figura 2.2: Logo del colegio EDIA

2.2. El trastorno de comunicación social

El trastorno de comunicación social, anteriormente conocido como el Síndrome de Asperger, se caracteriza por una llegada tardía de la madurez social y del razonamiento social. Hans Asperger fue el primero que describió este desorden autístico en los años 40 [9]. Al mismo tiempo, y sin saber acerca de los estudios de Hans Asperger, Leo Kanner describió lo que era el "espectro autista". No obstante, no fue sino hasta los años 80 que se comenzó a utilizar el término de "Síndrome de Asperger", gracias a un artículo escrito por Lorna Wing, una reconocida psiquiatra británica en el año de 1981. Este artículo describió 34 casos de niños y adultos con autismo, dentro de los cuales se incluía el Síndrome de Asperger [10].

Los individuos que padecen de este trastorno tienen dificultades al interrelacionarse con otras personas, se les hace difícil desarrollar una empatía con aquellos que los rodean. Por ejemplo, los niños diagnosticados con el trastorno de comunicación social generalmente tienen dificultad al hacer amigos, ya que tienen el inconveniente de no poder comunicarse adecuadamente por no poder controlar sus emociones.

Formalmente, el trastorno de comunicación social se encuentra definido en el Manual Diagnóstico y Estadístico de los Trastornos Mentales como un trastorno de comunicación. Dentro de los trastornos de comunicación, podemos encontrar deficiencias en diferentes áreas de la comunicación, tales como en: el lenguaje, el habla y la comunicación. En este manual, cuando se refiere a la comunicación, sugiere todo comportamiento verbal o no verbal que influye en las ideas o actitudes del individuo [2].

Otras características propias de mencionar acerca de las personas con el trastorno de comunicación social está sus habilidades con el lenguaje. A pesar de tener la complicación de no poder comunicarse con los demás, se caracterizan también por la forma poco usual que le dan. Estas habilidades incluyen el uso de un vocabulario y una sintaxis avanzada para su edad, y poseer una tendencia a ser pretenciosos al momento de

hablar. También, poseen una fascinación intensa por temas muy específicos, y a su vez, se les dificulta prestar atención en las clases. Ocupan de una forma diferente de enseñanza para que puedan aprender, y por consecuencia también necesitan de ayuda con sus habilidades de auto-ayuda y de organización. Por último, hay que mencionar que estas personas poseen de cierta falta de coordinación al andar, y muy poca tolerancia a sonidos específicos, algunas fragancias, texturas o al contacto humano [9].

El resultado del trastorno es variable, algunos niños mejoran sustancialmente con el tiempo y en otros, sus dificultades persisten hasta la edad adulta. Las deficiencias tempranas en la pragmática pueden causar alteraciones duraderas en las relaciones y comportamientos sociales [2].

2.2.1. Diagnosis

Como la pragmática depende del progreso adecuado para el desarrollo del habla y lenguaje, el diagnóstico del trastorno empieza desde los 4 años. A esta edad, la mayoría de los niños deberían de tener las suficientes capacidades del habla y lenguaje para poder identificar alguna deficiencia [2]. En ocasiones, el diagnóstico correcto de este trastorno resulta algo complicado, ya que los niños que lo padecen no poseen características físicas que indiquen que son diferentes a los demás niños. Incluso, muchas de sus características en su personalidad pueden generar expectativas altas con respecto a su conocimiento social.

Según indica Attwood en [9], existen varios caminos hacia un diagnóstico del trastorno de comunicación social sobre un niño. Uno de estos es el diagnosis de autismo a inicios de la infancia y su progreso hacia un autismo altamente funcional más adelante; otro de estos caminos es a través de diagnosis previos por otro tipo de desorden, algunos de estos pueden ser el diagnosis de un desorden de atención, lenguaje, movimiento, humor, habilidad para aprender o incluso un desorden en el apetito; por último, tampoco se debe de descartar el reconocimiento del trastorno por uno de los maestros cuando el niño comienza la escuela primaria.

Las señales de la existencia del trastorno de comunicación social en una persona empiezan a ser aparentes durante la adolescencia, cuando las expectativas sociales y académicas se vuelven más complejas. Como consecuencia, se pueden desarrollar problemas y conflictos con los padres, maestros y autoridades escolares [9].

Cuando una escuela, o terapeuta reciben una persona que se ha identificado con el trastorno de comunicación social, el siguiente paso está en asignarles un cuestionario que respalde la decisión de llevar a la persona a un especialista. Sin embargo, existen varios cuestionarios, de los cuales aún no existe una "primera elección". Los siguientes cuestionarios son los que se aplican generalmente a niños, según se describe en [9]:

- **ASAS**: Australian Scale for Asperger's Syndrome (por Garnett y Attwood en 1998).
- **ASDI**: Asperger Syndrome Diagnostic Interview (por Gillberg *et al*. en 2001).
- **ASDS**: Asperger Syndrome Diagnostic Scale (por Myles, Bock y Simpson en 2001).
- **ASSQ**: Autism Spectrum Screening Questionnaire (por Ehlers, Gillberg y Wing en 1999).
- **CAST**: Childhood Asperger Syndrome Test (por Scott *et al.* en 2002, y William *et al.* en 2005).
- **GADS**: Gilliam Asperger Disorder Scale (por Gilliam en 2002).
- **KADI**: Krug Asperger's Disorder Index (por Krug y Arick en 2002).

Para darnos una idea del contenido que pueden incluir estos cuestionarios, se muestra en la tabla 2.1. Cabe resaltar que aquí, una de las características de las personas con trastorno de comunicación social es la de "expresiones faciales limitadas".

Tabla 2.1 Criterio diagnóstico del trastorno de comunicación social (Síndrome de Asperger) de Gillberg.

Característica	**Descripción**
1. Discapacidad social (egocentrismo extremo) (*debe tener por lo menos dos*)	Dificultades para interactuar con otras personas, indiferencia en hacer contactos, dificultades al interpretar señales sociales.
2. Interés limitado (*por lo menos uno*)	Exclusión de otras actividades, adherencia repetitiva, aprende de memoria en lugar de razonar el significado.
3. Necesidad compulsiva por introducir rutinas o intereses (*por lo menos uno*)	Donde afectan al individuo en todos los aspectos de la vida, donde afectan a las demás personas.

Característica	Descripción
4. Peculiaridades del habla y el lenguaje	Desarrollo retrasado del habla; lenguaje expresivo superficialmente perfecto; lenguaje formal y pedante; prosodia extraña, características particulares de la voz; falta de capacidad para la comprensión, incluye la mala interpretación de significados literales o implícitos.
5. Problemas de comunicación no verbales (*por lo menos uno*)	Uso limitado de gestos, lenguaje corporal torpe o cohibido, expresiones faciales limitadas, expresiones faciales inapropiadas, mirada peculiar o rígida.
6. Torpeza motora	Mal desempeño en pruebas del neurodesarrollo.

El trastorno de comunicación en algunos niños y adultos es relativamente fácil de diagnosticar, un médico clínico puede sospechar de un diagnosis positivo en cuestión de minutos. Aunque también existen pacientes un poco más difíciles, como es el caso de algunos adultos que demuestran habilidades cognitivas considerables, debido a que algunas de sus características podrían camuflar sus dificultades. Igualmente, también es más difícil diagnosticar a una niña o a una mujer con este trastorno, ya que existen menos probabilidades de que ellas lo padezcan [9]. Los médicos clínicos suelen guiarse por los criterios del Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-V) para realizar sus diagnosis. Esta guía se muestra en la tabla 2.2.

Tabla 2.2 Criterios diagnósticos del Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-V) para el trastorno de comunicación social

Criterio diagnósticos 315.39 (F80.89)
A. Dificultades persistentes en el uso social de la comunicación verbal y no verbal.
B. Las deficiencias causan limitaciones funcionales en la comunicación eficaz, la participación social, las relaciones sociales, los logros académicos o el desempeño laboral, ya sea individualmente o en combinación.

Criterio diagnósticos 315.39 (F80.89)

C. Los síntomas comienzan en las primeras fases del periodo de desarrollo (pero las deficiencias pueden no manifestarse totalmente hasta que la necesidad de comunicación supera las capacidades limitadas).
D. Los síntomas no se pueden atribuir a otra afección médica o neurológica, ni a la baja capacidad en los dominios de morfología y gramática, y no se explican mejor por un trastorno del espectro autista, discapacidad intelectual (trastorno del desarrollo intelectual), retraso global del desempeño u otro trastorno mental.

Adicionalmente, este manual incluye un diagnóstico diferencial donde explica 5 diferentes puntos que separan al trastorno de comunicación social de los demás trastornos [2]:

- **Variaciones normales del lenguaje:** Al evaluar se debe de tener en cuenta los aspectos regionales, sociales y culturales del lenguaje. A veces esta evaluación puede ser difícil si se realiza antes de los cuatro años de edad.
- **Audición u otra deficiencia sensorial:** Se debe tomar en cuenta que no existan deficiencias al escuchar como causa primaria de la dificultad del lenguaje.
- **Discapacidad intelectual (trastorno del desarrollo intelectual):** Los retrasos del lenguaje muchas veces son causa de una discapacidad intelectual, por lo que hay que descartar primero esto antes de diagnosticar a una persona con el trastorno de comunicación social.
- **Trastornos neurológicos:** Los trastornos del lenguaje pueden ser producidos por trastornos neurológicos como la epilepsia.
- **Regresión del lenguaje:** Las regresiones del lenguaje pueden ser ocasionadas por afecciones neurológicas como las epilepsias.

Existen muchas ventajas en tener un diagnosis correcto del trastorno de comunicación social en un niño, ya que de esta forma se puede prevenir o reducir los efectos de algunas estrategias compensatorias o de ajuste, también puede eliminar preocupaciones acerca de otros diagnósticos y además, se puede reconocer que se tienen dificultades genuinas con experiencias que otras personas consideran fáciles o placenteras. Como resultado, una vez que se realiza el diagnóstico, existen más probabilidades que el niño pueda ser entendido, y por consecuente, ser más respetado por los demás.

2.2.2. Lenguaje

El lenguaje es particularmente importante cuando se habla del trastorno de comunicación social, ya que uno de las características que Hans Asperger describió es la del uso inusual del mismo [9].

Para entender lo crucial que es el lenguaje en el diagnosis y tratado del trastorno de comunicación social, primero tenemos que definir a qué nos referimos por lenguaje. El lenguaje se puede describir a través de la ontología. La ontología ofrece una explicación diferente a lo que es el fenómeno humano: nos dice que el lenguaje es una acción. Ya que los seres humanos somos seres lingüísticos, y vivimos completamente dentro del lenguaje, podemos decir que nos transforma en seres éticos, reflexivos y capaces de cuestionarnos. Basándonos en lo que nos dice la ontología, podemos decir que la razón es un tipo de experiencia humana que se deriva del lenguaje [11].

Teniendo en cuenta que los seres humanos estamos sumergidos en nuestra totalidad en un mundo regido por el lenguaje, en la mayoría del tiempo nos encontramos en un estado de transparencia. Esta transparencia, es la actividad no reflexiva, no pensante y no deliberativa, con un umbral mínimo de conciencia. En este estado, no sólo no estamos pensando en lo que no hacemos, tampoco estamos en un mundo que se rige por la relación sujeto-objeto. De aquí se puede tomar el concepto del quiebre. Un quiebre es cuando el fluir en la transparencia se ve interrumpido. Todo quiebre modifica el espacio de lo posible y transforma nuestro juicio sobre lo que nos cabe esperar. Con esto podemos decir que todo problema es siempre función de la interpretación que lo sustenta y desde la cual se califica como problema [11].

Teniendo una vez claro qué es el lenguaje, la transparencia y cómo es que ésta puede quebrantarse al momento en que una situación hace que cobremos conciencia sobre lo que nos rodea, podemos ver por qué las características del lenguaje que tienen aquellos diagnosticados con el trastorno de comunicación social pueden verse como problemáticas por aquellos que no lo padecen.

2.2.3. Expresiones y emociones

Las personas con el trastorno de comunicación social se consideran extremadamente inteligentes, gracias a su capacidad de absorber la información que les interesa

y a su peculiar forma de expresarse. A pesar de lo anterior, estas personas sufren de una falta de madurez indiscutible cuando se trata de las emociones. A las personas que tienen este trastorno, el Manual Diagnóstico y Estadístico de los Trastornos Mentales les atribuye la "falta de reciprocidad socio-emocional", y recordando un poco el criterio de Christopher Gillber, que se encuentra en la tabla 2.1, uno de los puntos menciona el "uso de expresiones faciales limitadas" y "el uso de expresiones faciales inapropiadas" [9].

Al platicar con los adultos con este trastorno, ellos pueden describir y comprender las emociones de una forma racional. Es decir, pueden explicar qué es y cuándo es que una persona debe de sentirla, pero no experimentarla. Cabe aclarar que existen ciertas emociones que son más difíciles que una persona adulta con el trastorno de comunicación social describa y comprenda, tales como el amor [9].

En su libro "*The Complete Guide to Asperger's Syndrome*" [9], Tony Attwood menciona que las personas con el trastorno de comunicación social pueden establecer una escala propia (el cual la maneja como un termómetro) para medir las respuestas emocionales. Este tipo de escala es muy útil al momento de establecer e incorporar la educación de las emociones cuando se trabaja con ellos.

2.3. Cómputo afectivo

El cómputo afectivo es una rama de estudio de las ciencias de la computación, la psicología y la ciencia cognitiva que se enfoca en sensar el estado emocional del usuario. Este sensado se puede realizar de varias formas, entre ellas se encuentran: micrófonos, cámaras, y otros tipos de sensores. Tiene como fin hacer que la computadora responda de forma apropiada a las emociones del usuario. Esta área nació con el artículo publicado en 1995 por Rosalind Picard, llamado *Affective Computing*, el cual es el resultado del estudio de la habilidad para simular la empatía [12].

En su artículo, Picard argumenta que las emociones son un tema rodeado por estigma en el mundo científico, ya que los principios científicos deben de derivarse del pensamiento racional, argumentos lógicos, hipótesis que puedan ser probadas y experimentos repetibles. Sin embargo, no por esto tenemos que perder de vista el propósito que se tiene al construir sistemas inteligentes: si queremos hacer que un dispositivo "piense"

de la misma forma que lo hacen los humanos, no podemos dejar atrás los sentimientos. Quizá la prueba más famosa para saber si una máquina puede "pensar" de la misma forma en que lo hace una persona, es la prueba de Turing. Para que la computadora pueda pasar la prueba de Turing, es necesario que sus respuestas sean indistinguibles a las mismas que daría un humano. Aquí podemos ver lo importante que es el rol de las emociones en el pensamiento, aunque éste sea artificial.

En este mismo artículo se describen diferentes escenarios donde se define la importancia que tiene el cómputo afectivo en el desarrollo de sistemas inteligentes además de proponer ciertos modelos. El más relevante para nuestro proyecto es el escenario del profesor de piano, donde Picard propone la siguiente situación: imagínese que está sentado con un profesor de piano virtual, el cual no sólo lee el movimiento de las manos, ritmo y fraseo; sino que también el estado emocional en que se encuentra el usuario. Este sistema no sólo interpreta la expresión musical, también las expresiones faciales y quizá otros cambios físicos que correspondan a los sentimientos. Gracias al reconocimiento de emociones, el profesor de piano en la computadora puede evaluar si se está progresando correctamente y si el usuario está satisfecho con el progreso. A partir de los datos anteriores, el sistema podría preguntarle al usuario si sigue con el interés de continuar practicando, o incluso podría convencerlo en intentar ejercicios más difíciles para tener un mejor progreso.

En el ejemplo anterior podemos ver cómo se puede tomar un tema de interés ya sea emocional como la música, o algo no emocional como la ciencia, y hacer que el sistema de enseñanza trate de maximizar el placer y el interés del usuario mientras minimiza la aflicción [12].

2.4. Visión por computadora

La función principal de la visión, ya sea biológica o artificial, es la de reconocer y localizar objetos que se encuentran en el medio ambiente [13]. La visión por computadora es como se define como: "la ciencia de programar a una computadora para procesar y entender las imágenes y videos que se capturan, en otras palabras hacer que la computadora vea" [14]. Esta es el área que se encarga de estudiar los procesos de reconocimiento y localización de los objetos a través de imágenes, para así entenderlos y construir sistemas computacionales con capacidades similares a la visión biológica

[13]. Para cumplir con su objetivo, es necesario poder extraer las diferentes características que se encuentran dentro de las imágenes utilizando diferentes métodos y técnicas en forma de algoritmos computacionales.

La extracción de características de una imagen puede lograrse con mayor facilidad si primero se optimiza la imagen, a este proceso se le conoce como pre-procesamiento. La visión por computadora se ayuda del área de procesamiento de imágenes para mejorar la calidad de éstas, mediante los diversos algoritmos existentes para filtrar imágenes. La visión por computadora se ha aplicado exitosamente en otras aplicaciones prácticas, entre estas podemos mencionar las siguientes: robótica móvil, vehículos autónomos, manufactura, interpretación de imágenes aéreas y de satélite, análisis e interpretación de imágenes médicas, entre otros [13].

2.4.1. Captura de imágenes

Para la adquisición de imagen se requiere de un dispositivo físico sensible a una determinada banda del espectro electromagnético. Existen varios dispositivos capaces de capturar imágenes, entre los más comunes se encuentran [13]:

- Cámaras fotográficas
- Cámaras de televisión
- Escáneres
- Sensores de rango
- Sensores de ultrasonido
- Rayos X
- Imágenes de tomografía
- Imágenes de resonancia magnética

2.4.2. La representación de una imagen digital

Las computadoras tienen una forma especial de representar las imágenes que nosotros, como usuarios, no podemos apreciar a simple vista. Dentro de un sistema de cómputo las imágenes no son mas que un conjunto de valores que representan cierta información de una escena. A su vez, a este conjunto de valores que existe en la me-

moria de una computadora, podemos darle la representación de una matriz de 2 ó más dimensiones [15]. Un ejemplo de esta representación puede verse en la figura 2.3.

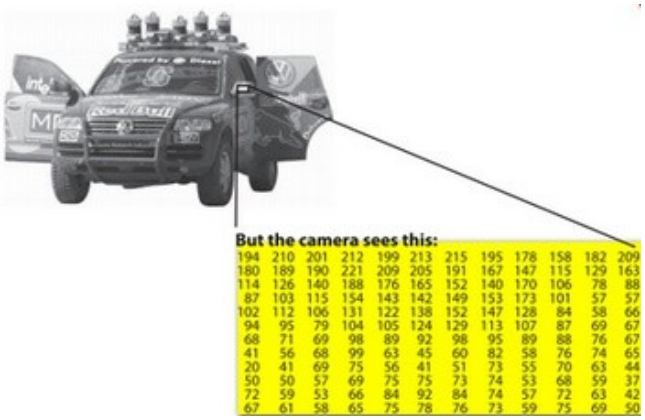

Figura 2.3: Representación matricial de una imagen. Fuente: adaptado de [16]

Gracias a esta representación matricial de las imágenes, se pueden implementar varias técnicas usadas en el álgebra lineal dentro los dispositivos de cómputo. Tal y como se pueden manipular las matrices, se pueden manipular las imágenes. Esto ha dado como resultado que se aproveche software que maneje matrices como MATLAB o librerías como NumPy de Python para complementar el procesamiento.

Las primeras dos dimensiones que tiene la matriz representan las coordenadas x y y de una imagen, y cada uno de los puntos de esta red representa un píxel de la imagen. Cabe aclarar que, en el procesamiento de imágenes las coordenadas de las imágenes son un poco diferentes a las que se acostumbra: ya que el punto de origen está en la esquina superior izquierda, como puede verse en la figura 2.4. Estas matrices no están limitadas únicamente a dos dimensiones, éstas suelen tener una tercera dimensión la cual define el color. El color puede definirse como un fenómeno que está relacionado a la forma en que un ojo humano reacciona a diferentes longitudes de onda encontradas en el espectro visible [13]. La sensación del color viene de la sensibilidad de tres tipos de sensores neuroquímicos en la retina [17].

Existen formas de codificar los colores dentro de la matriz que representa una imagen, a esta representación se le conoce como el espacio de color. Estos espacios de color son creados por razones prácticas, y son una forma de organizar los colores que perciben los humanos a través de combinaciones de tres principales longitudes de onda que percibimos [17].

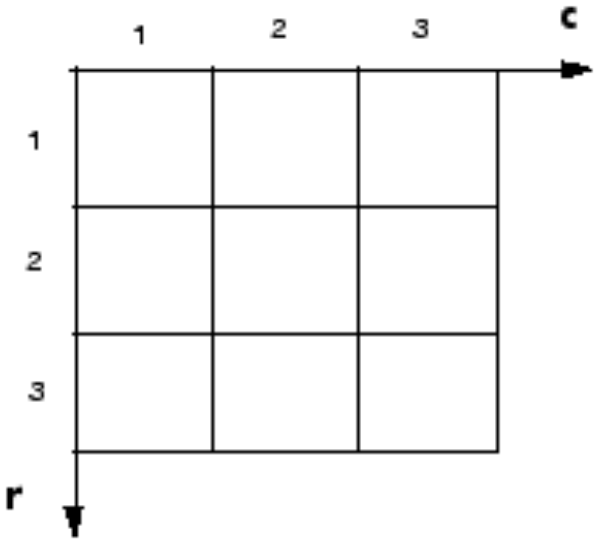

Figura 2.4: Coordenadas en una imagen. Fuente: adaptado de [16]

Generalmente, el espacio de color puede representarse como una tercera dimensión en la matriz de una imagen digital [13]. El espacio de color más común es el RGB (rojo, verde, y azul). Los espacios de color en las imágenes digitales pueden convertirse a otros tipos para poder procesar la imagen de otras formas. A continuación se mencionarán los espacios de color más comunes en el procesamiento de imágenes:

- **RGB:** Formalmente, este espacio de color lineal está compuesto por las longitudes de onda primarias: 645.16nm para R (rojo), 526.32nm para G (verde), y 444.44nm para B (azul). Informalmente, se usan aquellos colores principales con los que cuente el monitor [18].
- **HLS:** Se basa en las coordenadas polares en tres dimensiones. Donde el vértice superior equivale al blanco y el inferior al negro; el eje vertical la brillantez (L), el horizontal la saturación (S) y el ángulo de la proyección el croma (H) [13].
- **HSI:** Se considera el más aproximado a la percepción humana. Aquí, los colores se codifican a través de tres componentes: la intensidad o brillantez (I), el croma (H), y la saturación (S) [13].
- **CMY(K):** Este espacio de color está pensado para pigmentos. A diferencia de aquellos espacios de color que funcionan a través de la suma de luces, este espacio funciona con la substracción de pigmentos. Sus colores primarios son el cían (C), magenta (M) y amarillo (Y), además del negro (K) [17].

Gracias a esta dimensión adicional se pueden hacer conversiones entre un espacios de color y otro para poder hacer operaciones más complejas. Por ejemplo, si se quiere cambiar un color que existe en una imagen por otro, es más fácil hacer la conversión a HLS y cambiar la saturación, en lugar de evaluar píxel por píxel dentro del espacio de color RGB.

Figura 2.5: La foto “Lenna” convertida a otros espacios de color

2.4.3. Procesamiento y normalización de imágenes

La normalización tiene como objetivo el de preparar una imagen para que luego sea usada por algoritmos de aprendizaje automático. Este tipo de preparación, o mejoramiento, de imágenes no tiene un método predefinido, ya que los pasos para mejorarla dependen del problema al que se esté enfrentando. Sin embargo, existen operaciones comunes que los desarrolladores pueden combinar para tener los resultados deseados.

Una de estas operaciones es la normalización de la intensidad. Tiene como objetivo nivelar la saturación en las imágenes en blanco y negro, para que las condiciones externas como la iluminación o el ruido tengan menos efecto y puedan verse las imágenes más claramente.

Algunas técnicas para normalizar las imágenes son:

- Normalizar de forma numérica la intensidad a un intervalo estándar
- La ecualización del histograma

Figura 2.6: Proceso de la ecualización de histograma en la foto “Lenna”

2.4.4. Reconocimiento de expresiones faciales

El reconocimiento facial es un tema de interés dentro de la visión computacional, el cual básicamente se basa en saber si dentro de una imagen existe o no una cara humana. Esta técnica se usa en la actualidad en varios sistemas, como son: aquellos que lidian con seguridad (para el registro o reconocimiento de usuarios), en los sistemas embebidos que incluyen las cámaras modernas (para enfocarse en el rostro), entre otros. Un ejemplo de detección facial utilizando Cascadas Haar (*Haar Cascades*) puede verse en la figura 2.7.

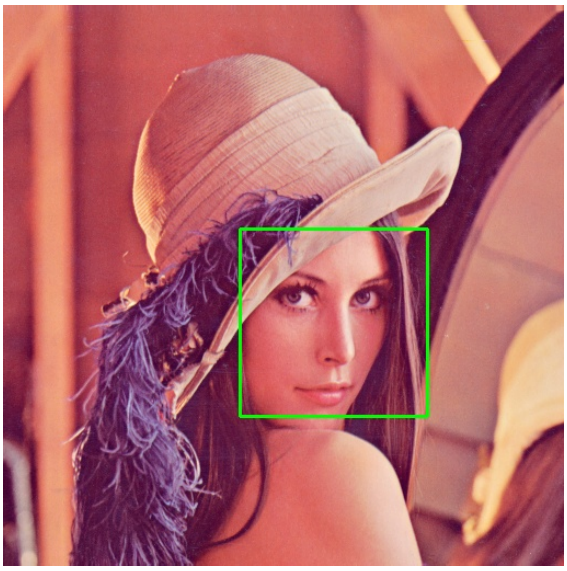

Figura 2.7: Detección facial en “Lenna” usando Cascadas Haar

No obstante, ya no sólo basta con reconocer qué es una cara y qué no, el problema

al que nos estamos enfrentando es saber cuál es la expresión que se encuentra haciendo.

El proceso del reconocimiento de expresiones en el rostro está definido en [19] de la siguiente manera:

1. Localizar la cara en una escena (a este paso se le conoce como detección facial).
2. Extraer las características faciales de la región donde se encuentra la cara (por ejemplo, se detecta la forma de los componen faciales o describir la textura de la piel que se encuentra en la cara, entre otros. Este paso se le conoce como extracción de las características faciales).
3. Analizar el movimiento de las características faciales y/o los cambios de apariencia para clasificarla en categorías predeterminadas como una cara que demuestre felicidad, tristeza o enojo (a este paso se le conoce como interpretación de las expresiones faciales).

2.4.5. Arquitectura para un sistema de visión

Para concluir con el tema de visión, podemos definir la estructura de un sistema de visión por computadora. Esta estructura generalmente está compuesta por cinco etapas [13]:

- **Adquisición de la imagen**: se encarga de la visualización, formación de la imagen y la digitalización.
- **Pre-procesamiento de la imagen**: la imagen se normaliza y filtra para poder usarla en pasos posteriores.
- **Extracción de características locales**: Se aplican algoritmos para obtener características como las esquinas o texturas de la imagen.
- **Segmentación**: Se separa la imagen en contornos y regiones.
- **Clasificación e interpretación**: Se crean descripciones simbólicas de los primitivos de las imágenes o de los objetos que existe en la escena.

2.5. Redes neuronales

Como se pudo ver en la sección anterior, las tareas que trata de resolver la visión por computadora son muy complejas. Especialmente si consideramos que, mientras los

humanos podemos reconocer sin dificultad una serie de números o el rostro de una persona, tratar de enseñarle lo mismo a una computadora a través de un programa convencional parece un problema imposible. Las computadoras pueden verse limitadas debido a que éstas sólo pueden trabajar a partir de bits y pixeles [20].

Una forma de resolver las tareas más complejas de este campo, es a través de las redes neuronales. Las redes neuronales son modelos computacionales inspirados en la biología del cerebro. Esta solución proviene de una pregunta clave para la visión por computadora: ¿qué tipo de representación interna permitiría a un sistema de visión artificial clasificar objetos independientemente de la pose, escala, iluminicación o ruido? [21].

Como humanos, somos capaces de procesar información visual gracias a la corteza visual encontrada en nuestro cerebro. La corteza visual primaria, también llamada V1, tiene alrededor de 140 millones de neuronas cada una con billones de conexiones entre ellas [22]. Una representación interna, generalmente se produce a través de un extractor de características hecho manualmente. Este extractor se hace a partir de las técnicas que exploramos en la sección anterior. Después de adquirir las características de un objeto que se encuentra en una imagen, en forma de un vector de características, se alimenta a un clasificador entrenable [21].

Durante décadas, las redes neuronales artificiales se consideraron útiles sólo para unos cuantos problemas relacionados al aprendizaje automático, y no para resolver problemas del mundo real. Aún así, gracias a los avances que se ha tenido en la última década, tales como la creación de las redes neuronales profundas (*deep neural networks*) y el avance que se ha tenido en el hardware, éstas se consideran como el estado del arte para resolver problemas relacionados con imágenes, video y voz [20]. A continuación se explicará en qué consisten las redes neuronales.

2.5.1. El perceptrón

El perceptron es la base de las redes neuronales, representa de forma matemática una neurona artificial. Los perceptrones fueron desarrollados entre los años 50 y 60 por un científico llamado Frank Rosenblatt, inspirado por los trabajos de Warrenn McCulloch y Walter Pitts [22].

El funcionamiento del perceptrón puede resumirse en que toma distintos valores binarios de entrada y produce sólo un valor binario de salida. Para calcular esta salida, Rosenblatt propuso un sistema de pesos (*weights*), los cuales representan la importancia que tienen las entradas y las salidas [22]. O vista desde la forma biológica, el peso define qué tan fuerte se encuentra conectada una neurona con otra. Otra forma de ver el funcionamiento del perceptrón, es como un dispositivo que toma decisiones a partir de las entradas que se le asignan y los pesos de sus conexiones [20]. El modelo puede verse en la figura 2.8.

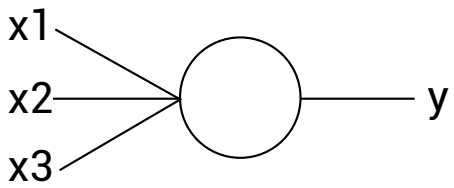

Figura 2.8: Perceptrón simple

La salida de la neurona se determina si la suma de los pesos es superior o inferior a un valor umbral (*bias*), un número real que tienen las neuronas como parámetro. El umbral puede verse como una medida de saber qué tan fácil es que la neurona se active o no. Matemáticamente, la salida se representa de la siguiente manera:

$$\text{salida} = \begin{cases} 0, \text{ si } w \cdot x + b \leq 0 \\ 1, \text{ si } w \cdot x + b > 0 \end{cases}$$

Un perceptrón aprende al modificar sus pesos a una serie de valores determinados. Estos valores se obtienen a partir de cientos de ejemplos etiquetados con cierta entrada y su valor de salida esperado, los cuales modifican los pesos del perceptrón gradualmente hasta encontrar los indicados.

2.5.2. Redes neuronales estándar

Una red neuronal estándar consiste en distintos procesadores simples o perceptrones conectados entre sí, y cada una produce una secuencia de activaciones con un valor real [23]. Está compuesta por una capa de entrada, una capa escondida y una capa de salida, un ejemplo puede verse en la figura 2.9.

Las redes neuronales pueden aprender de dos formas: supervisadas y no supervisadas. Para este trabajo de tesis, nos enfocaremos en el aprendizaje supervisado.

Usando el aprendizaje supervisado, las redes neuronales utilizan una gran cantidad de datos como ejemplos para después inferir de forma automática las reglas, y poder clasificar nuevas muestras [22].

2.5.3. Redes neuronales profundas

El aprendizaje profundo (*deep learning*) se refiere a una familia de algoritmos que extienden las redes neuronales tradicionales, los cuales consisten en agregar un cierto número de capas, llamadas capas escondidas, a los perceptrones [20]. La diferencia entre una red neuronal estándar y una profunda puede verse en la imagen 2.9.

Las redes neuronales profundas resultaron ser óptimas para la clasificación de imágenes. Ya que como vimos anteriormente, si queremos clasificar una, se tiene que representar las características a través de números [20], y como todas las características de la imagen se encuentran codificadas a través de pixeles e intensidades, podemos usar esos valores como entrada para que la red aprenda a través de métodos de aprendizaje automático.

La desventaja que tienen este tipo de redes, es que ocupan muchos recursos para poder entrenarse, incluso cuando éstas lo hacen en hardware especializado como tarjetas gráficas [20].

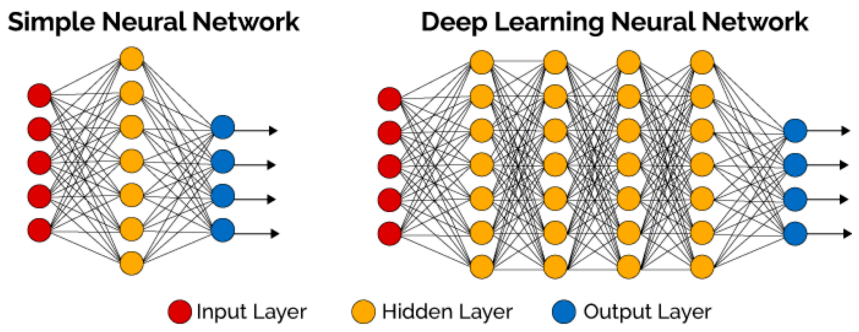

Figura 2.9: Comparación de una red neuronal estándar y una profunda. Fuente: adaptado de [24]

2.5.4. Redes neuronales convolucionales

En el campo de visión, se han hecho grandes avances gracias al desarrollo del aprendizaje profundo. Sin embargo, el mayor avance se dio gracias a las redes convolucionales. Como se discutió anteriormente, para tener un buen sistema de visión artificial necesitamos tener una representación interna, y para la visión (ya sea artificial o natural) las representaciones jerárquicas son buenas [21].

En los sistemas de visión artificial, los pixeles pueden agruparse en esquinas, que a su vez se agrupan en patrones, estos patrones en partes, las partes en objetos y los objetos en escenas. Esto sugiere que para reconocer, se requiere que las arquitecturas sean de capas múltiples entrenables apiladas una sobre la otra [21]. Sin embargo, ¿cómo es la arquitectura de una red neuronal convolucional?

Las redes neuronales convolucionales son arquitecturas entrenables y requieren múltples etapas. Cada etapa se compone por tres capas [21]:

- **Capa de banco de filtros**: es un arreglo de tres dimensiones, donde dos de éstas consisten en crear un mapa de características. Su salida también es un mapa. Cada filtro detecta una característica particular en cada parte de la entrada.
- **Capa de no linealidad**: esta capa consiste en hacer una operación con el mapa de características. En las redes convolucionales tradicionales, se suele aplicar la función tanh. Sin embargo, en implementaciones recientes se usan funciones más complejas, tales como el sigmoide rectificado.
- **Capa de agrupación de características**: esta capa trata a cada mapa de forma separada, y su operación consiste en reducir su resolución para que pase a otra etapa.

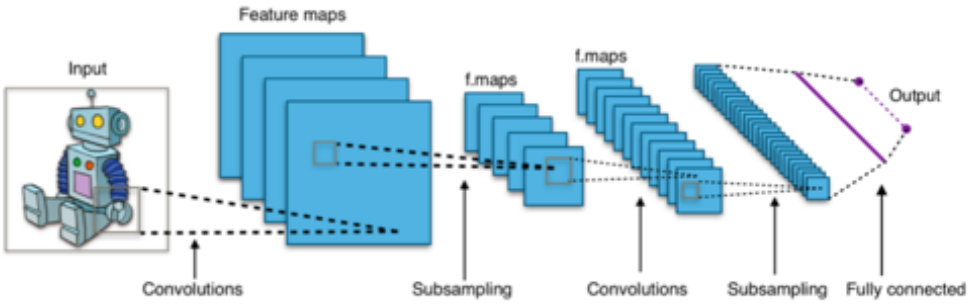

Figura 2.10: Arquitectura de una red neuronal convolucional. Fuente: adaptada de [25]

2.6. Trabajos relacionados

El primer trabajo relacionado, llamado "*Evaluation of a new computer intervention to teach people with autism or Asperger*", trata de un programa de cómputo que le enseña a los niños con desórdenes del espectro autista y Asperger, a reconocer y predecir las emociones en las demás personas [26]. Aquí se mencionan dos trabajos previos de software para enseñar el manejo de emociones: el primero es un trabajo de Swettenham en 1996, donde trata de ver si se le podía enseñar a los niños a través de un test de Sally-Anne por computadora; el segundo se trata de *Gaining Face*, un programa diseñado para ayudar a los niños a aprender expresiones faciales. El software que se desarrolló para este trabajo tiene el nombre de *Emotion Trainer*.

Los autores de *Emotion Trainer* argumentan que un niño desde unos cuantos días de nacido, puede imitar expresiones faciales. A los dos meses de edad, el niño puede percibir y responder a señales emocionales. A los tres años, el infante puede reconocer entre caras felices, enojadas y asustadas, además de diferenciar las causas y consecuencias de estas emociones. Esto hace que *Emotion Trainer* base sus funcionalidades en estas emociones [26].

Emotion Trainer es un programa de cómputo multimedia que tiene cinco secciones diferentes: la primera sección muestra fotografías de expresiones faciales y le pregunta al usuario si la persona se encuentra feliz, triste, contenta o asustada; la segunda sección muestra fotografías con leyendas que ponen a la persona en una situación que puede desencadenar una emoción; en la tercera se le presenta al usuario una imagen de lo que una persona quiere (una expectativa) y lo que le dan (la realidad), para que así el usuario elija la emoción que piensa que tendría esa persona; la sección cuatro es similar a la segunda, pero trata de estados mentales en lugar de físicos; por último, la quinta está diseñada para enseñar ausencia de objetos y situaciones no deseados [26].

Además, en este proyecto se demostró lo útil que puede resultar un programa de cómputo para intervenciones psicológicas, ya que significa que por diseño, el material puede repetirse indefinidamente y sin fatiga, a su vez, la computadora elimina muchos de los aspectos sociales que pudieran incomodar a muchos de los niños con Asperger y autismo [26].

Un segundo trabajo relacionado es el de "*Using Assistive Technology to Teach Emotion Recognition to Students With Asperger Syndrome*", aquí el trabajo trata igual-

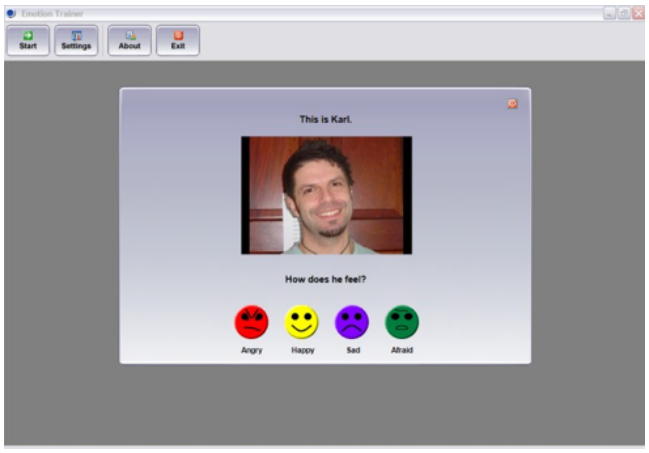

Figura 2.11: Emotion Trainer

mente sobre el problema que muchas de las personas que se encuentran dentro del espectro autista tienen al reconocer emociones en sí mismos y en las demás personas. Los autores proponen el uso de tecnología asistida, a través de un software llamado *Mind Reading: The Interactive Guide to Emotions,* para enseñar el reconocimiento de emociones [27].

Esta "intervención asistida por tecnología", como lo llaman los autores, se probó en niños de 8 a 11 años de edad durante diez semanas. Se llevó a prueba en la escuela o en la casa de cada uno de los niños durante diez semanas. Los resultados indicaron que, tras la intervención, los participantes mejoraron en el reconocimiento de expresiones faciales y voz, además de reconocimiento más complejos que no se incluían en el software [27].

Capítulo 3

Análisis y diseño

En este capítulo se describe el análisis y diseño del sistema. El análisis del sistema se hará a través de diagramas de contexto, y el diseño se realizará usando diagramas hechos en el lenguaje unificado de modelado (*Unified Modeling Language*, UML). Los diagramas de contexto tienen como propósito representar a través de flujos de datos las diferentes interacciones que se tienen entre el sistema y diferentes agentes externos. Por otra parte, el UML es un lenguaje que describe la arquitectura de sistemas basados en software. Esta descripción se hace a través de diferentes vistas [28], las cuales permiten enfocarse en diferentes aspectos y problemas a resolver del sistema. Estos enfoques pueden ser desde la perspectiva de: usuarios finales, ingenieros de sistema, líderes de proyecto, entre otros.

Después de presentar los distintos diagramas de contexto, se explicarán los diagramas de casos de uso para definir los requerimientos, y los diagramas de actividades, clases, componentes y despliegue para definir el diseño. Adicionalmente, se verá el diseño de la base de datos que ocupa el sistema para registrar datos usando diagramas de entidad relación y se presentará el diseño que se eligió para el algoritmo de clasificación.

Para concluir con el capítulo, se presentará la arquitectura propuesta del sistema en su totalidad.

3.1. Análisis del sistema

3.1.1. Diagrama de contexto nivel 0

En la figura 3.1 se muestra el diagrama de contexto nivel 0. Aquí se describe la interacción básica que hay entre el niño (usuario), el dispositivo móvil con la cámara, y la red neuronal que clasifica las expresiones del niño.

La entrada al sistema consiste en un cuadro (*frame*) capturado por la cámara del dispositivo móvil. El sistema se encargará de realizar el procesamiento adecuado sobre el cuadro para que éste pueda ser usado como entrada al algoritmo de clasificación. Este algoritmo retornará la categoría en la que fue clasificado y el dispositivo móvil se la presentará al usuario.

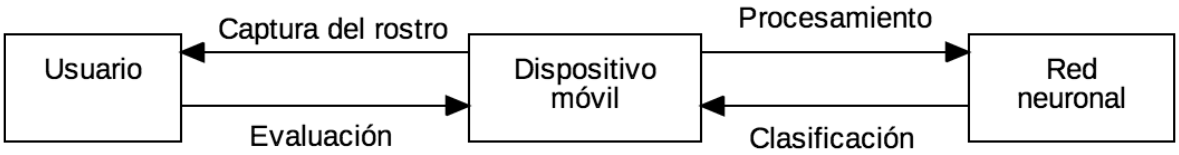

Figura 3.1: Diagrama de contexto nivel 0

3.1.2. Diagrama de contexto nivel 1

El diagrama anterior se expande como se ve en la figura 3.2. Donde se le añaden los procesos de: selección, ejercicios, y evaluación. Además de hacer referencia a una base de datos. En este diagrama, se inicia con la selección de un ejercicio por parte del usuario. Una vez que el usuario se encuentra realizando el ejercicio, se empezará la captura de cuadros desde la cámara del dispositivo y se procesarán para que el algoritmo clasificador pueda hacer su trabajo. Una vez que se tenga la evaluación del algoritmo, una base de datos se encargará de guardar la información relevante para hacer reportes en un futuro, además de que se le presentará al usuario si está realizando correctamente el ejercicio o no.

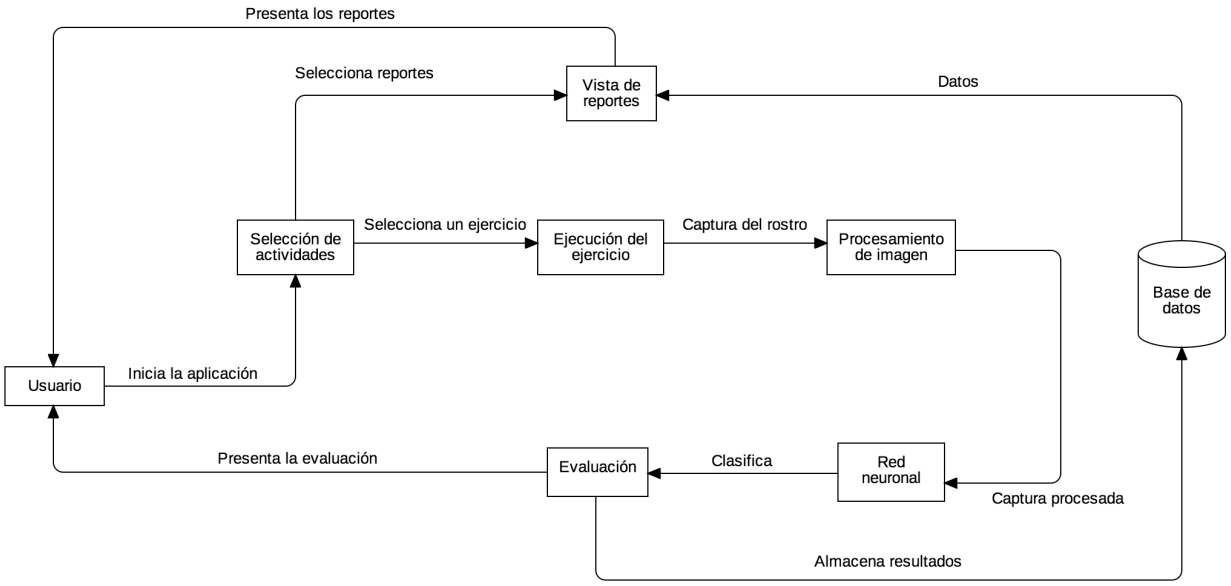

Figura 3.2: Diagrama de contexto nivel 1

3.1.3. Diagrama de casos de uso

El diagrama de casos de uso es una de las vistas que se proponen en el modelo de 4+1 vistas y explica cómo es que los actores interactúan con los elementos el sistema. En este proyecto existen dos actores: el niño (usuario) y el psicólogo. Este diagrama puede verse en la figura 3.3.

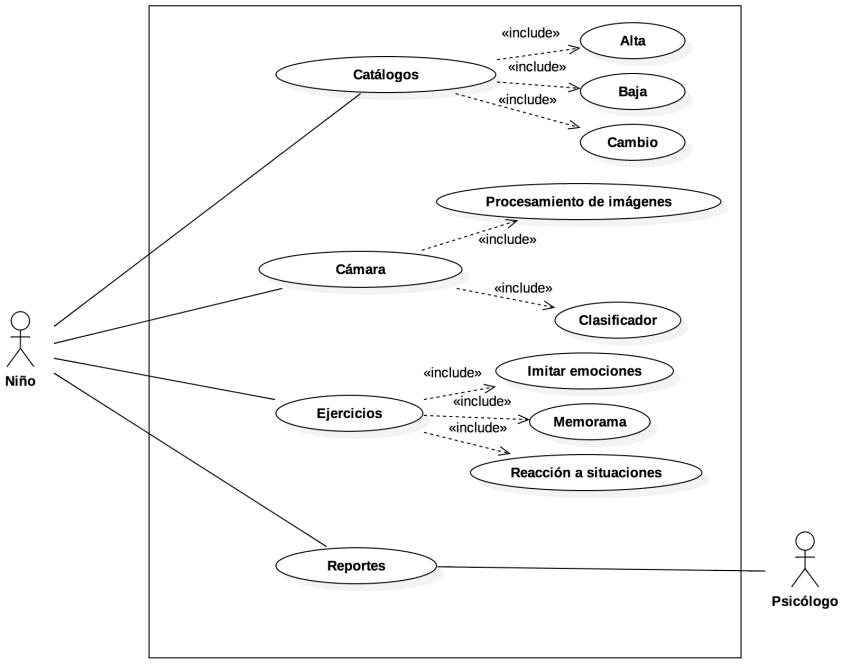

Figura 3.3: Diagrama de casos de uso

El niño es el actor principal en el sistema, ya que es el que tiene contacto directo con la aplicación móvil. Hará uso directo de la cámara, podrá entrar a distintos ejercicios, y ver el reporte de actividades, también hará uso indirecto de los catálogos, ya que al terminar ejercicios se guardarán los resultados automáticamente. El psicólogo tendrá la opción de ver los reportes. Los casos de uso se describen en las tablas 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, y 3.12.

Tabla 3.1 Caso de uso: Catálogos

Caso de uso:	Catálogos
Actores:	Usuario
Descripción:	Dependiendo de la acción del usuario, dará de alta, baja o realizará un cambio en un registro de la base de datos de la aplicación.
Flujo normal de eventos:	**1.** El usuario realiza una acción, como terminar un ejercicio o realizar una expresión correcta. **2.** El sistema realiza la acción indicada (alta, baja o cambio).

Caso de uso:	Catálogos
Flujo alternativo de eventos:	**1.** El usuario no termina el ejercicio o no realiza la expresión facial correcta.

Tabla 3.2 Caso de uso: Alta de catálogos

Caso de uso:	Catálogos: Alta
Actores:	Usuario
Descripción:	El sistema dará de alta un registro en la base de datos cuando el usuario termine un ejercicio o realiza la expresión facial que el sistema le indica.
Flujo normal de eventos:	**1.** El usuario termina un ejercicio o realiza la expresión facial que el sistema le indica. **2.** El sistema realiza el alta del evento.
Flujo alternativo de eventos:	**1.** El usuario no termina el ejercicio o no realiza la expresión facial correcta.

Tabla 3.3 Caso de uso: Baja de catálogos

Caso de uso:	Catálogos: Baja
Actores:	Usuario
Descripción:	El sistema dará de baja uno o más registros si el usuario le indica que desea eliminar su historial de actividades.
Flujo normal de eventos:	**1.** El usuario selecciona la opción de borrar el historial de actividades. **2.** El sistema dará de baja todos los registros que se encuentren en la base de datos.
Flujo alternativo de eventos:	**1.** El usuario no selecciona la opción de borrar el historial de actividades.

Tabla 3.4 Caso de uso: Cambio de catálogos

Caso de uso:	Catálogos: Cambio
Actores:	Usuario
Descripción:	El sistema realizará un cambio en algún registro de la base de datos si es que versiones futuras del sistema lo requieren.
Flujo normal de eventos:	**1.** Después de una actualización del sistema, se realizarán los cambios necesarios a la base de datos.
Flujo alternativo de eventos:	**1.** No existe una actualización del sistema.

Tabla 3.5 Caso de uso: Cámara

Caso de uso:	Cámara
Actores:	Usuario
Descripción:	El módulo de cámara deberá mostrar el rostro del usuario y enviar los cuadros (o *frames*) para que sean procesados y clasificados.
Flujo normal de eventos:	**1.** El usuario entra a una actividad de la aplicación donde se use la cámara. **2.** La cámara se activa. **3.** Comienza una vista previa y se envía los cuadros capturados para que sean procesados y clasificados.
Flujo alternativo de eventos:	**2.** Si el dispositivo del usuario no cuenta con una cámara o las librerías adecuadas, se mostrará un diálogo de error y no se podrá usar.

Tabla 3.6 Caso de uso: Cámara: Procesamiento de imágenes

Caso de uso:	Cámara: Procesamiento de imágenes
Actores:	Usuario
Descripción:	Los cuadros capturados por la cámara serán procesados para encontrar rostros, cambiar el espacio de color y normalizar la imagen.

Caso de uso:	Cámara: Procesamiento de imágenes
Flujo normal de eventos:	**1.** Se recibe un cuadro de la cámara. **2.** Se busca si existen rostros en el cuadro. **3.** En caso de que existan, la imagen será recortada y se normalizará para que pueda usarse en el algoritmo de clasificación.
Flujo alternativo de eventos:	**3.** En caso de que no existan rostros, el cuadro se ignorará.

Tabla 3.7 Caso de uso: Cámara: Clasificación

Caso de uso:	Cámara: Clasificación
Actores:	Usuario
Descripción:	Los cuadros procesados se introducirán en el algoritmo de clasificación para poder determinar la emoción que se encuentra realizando el usuario.
Flujo normal de eventos:	**1.** Se recibe un cuadro procesado. **2.** El cuadro procesado se usa como entrada del algoritmo de clasificación. **3.** El algoritmo regresa la emoción más probable relacionada a la expresión actual del usuario.
Flujo alternativo de eventos:	**3.** En caso de que exista un error al retornar la emoción más probable, se le avisará al usuario.

Tabla 3.8 Caso de uso: Ejercicios

Caso de uso:	Ejercicios
Actores:	Usuario
Descripción:	Al usuario se le presentarán diferentes ejercicios para practicar sus expresiones faciales.
Flujo normal de eventos:	**1.** Se le presenta una lista de ejercicios al usuario. **2.** El usuario selecciona un ejercicio de la lista. **3.** El ejercicio inicia.

Caso de uso:	Ejercicios
Flujo alternativo de eventos:	**2.** El usuario no selecciona ningún ejercicio.

Tabla 3.9 Caso de uso: Ejercicios: Imitar emociones

Caso de uso:	Ejercicios: Imitar emociones
Actores:	Usuario
Descripción:	Cuando el usuario seleccione este ejercicio, se le presentará una lista de expresiones faciales, el usuario tendrá que imitar una por una.
Flujo normal de eventos:	**1.** El usuario entra al ejercicio. **2**. Se le presenta una lista *n* expresiones faciales aleatorias. **3.** El usuario debe imitar cada una de ellas con la ayuda de la cámara y su algoritmo de clasificación. **4.** Al acabarse las expresiones a imitar, se le presenta un diálogo con los resultados del ejercicio.
Flujo alternativo de eventos:	**4.** En caso de que el usuario cancele el ejercicio antes de terminarlo, no se presentarán ni se guardarán los resultados.

Tabla 3.10 Caso de uso: Ejercicios: Memoria

Caso de uso:	Ejercicios: Memoria
Actores:	Usuario
Descripción:	Cuando el usuario seleccione este ejercicio, se le presentará un tablero que simula el juego del "memorama", donde el contenido de las tarjetas son expresiones faciales. El objetivo de usuario es emparejar cada una de estas tarjetas e imitar con la cámara la expresión que muestran.

Caso de uso:	Ejercicios: Memoria
Flujo normal de eventos:	**1.** El usuario entra al ejercicio. **2**. Se le presenta un tablero con tarjetas que contienen expresiones faciales. **3.** El usuario seleccionará dos tarjetas diferentes del tablero. **4.** Si ambas tienen el mismo contenido, se presentará un diálogo con el módulo de la cámara para que el usuario imite la expresión. **5.** En caso de que el contenido sea diferente, el usuario tendrá que volver a elegir otras dos cartas. **6.** El ejercicio termina una vez que no haya más cartas por elegir y se le presenta un diálogo con sus resultados.
Flujo alternativo de eventos:	**4.** Si el usuario realiza una expresión diferente, tendrá que olver a intentarlo.

Tabla 3.11 Caso de uso: Ejercicios: Situaciones

Caso de uso:	Ejercicios: Situaciones
Actores:	Usuario
Descripción:	Cuando el usuario seleccione este ejercicio, comenzará a responder cómo reaccionaría a diferentes situaciones sociales usando sus expresiones faciales. Cada situación social estará compuesta de una imagen y una breve descripción de la situación social.
Flujo normal de eventos:	**1.** El usuario entra al ejercicio. **2**. Se le presenta una situación social aleatoria. **3.** El sistema espera la respuesta del usuario. **4.** Si es la expresión facial adecuada, el usuario avanza a la siguiente situación. **5.** Se repite *n* veces hasta terminar y se le presenta un diálogo mostrándole los resultados.
Flujo alternativo de eventos:	**4.** Si el usuario realiza una expresión facial adecuada, tendrá que volver a intentarlo.

Tabla 3.12 Caso de uso: Reportes

Caso de uso:	Reportes
Actores:	Usuario, psicólogo
Descripción:	Los usuarios y psicólogos tendrán la posibilidad de ver cuáles ejercicios se han realizado en la aplicación. En la sección de reportes se podrá ver cuáles ejercicios se han realizado, cuántas veces se han hecho por día, y cuánto tiempo se ha invertido en ellos.
Flujo normal de eventos:	**1.** El usuario entra a la sección de reportes. **2**. Se selecciona un rango de fechas donde se desea ver la actividad del usuario. **3.** El sistema presenta los resultados a través de distintas gráficas.
Flujo alternativo de eventos:	**3.** En caso de que no haya actividad registrada, se mostrará una advertencia y las gráficas estarán vacías.

3.2. Diseño del sistema

3.2.1. Diagrama de actividades

Aquí se presentarán los diagramas de actividades para cinco secciones diferentes que se encuentran disponibles en el sistema, estos son: la selección de actividades, imitar emociones, memorama, reacción a siuaciones, y muestra de reportes. Los diagramas de actividades fueron diseñados para mostrar una visión simplificada de lo que ocurre durante una operación o un proceso.

Selección de actividad

Esta actividad se presenta al iniciar el programa. El usuario tiene la opción de elegir uno de los tres ejercicios disponibles, o bien, mostrar el reporte de su actividad. El diagrama se muestra en la figura 3.4.

Imitar emociones

En este ejercicio, el usuario tiene como objetivo el de imitar una serie de emociones usando la cámara de su dispositivo. Cuando el sistema detecta que el usuario se encuentra realizando la misma expresión que se le está indicando, se le suma un punto y pasa a la siguiente emoción. El ejercicio termina una vez que el usuario haya acabado con todas las emociones. El diagrama se muestra en la figura 3.5.

Memorama

En este ejercicio, se le permite al usuario jugar un juego de memoria. En el cual el objetivo es emparejar cartas con la misma expresión facial para luego imitarla con la cámara. El ejercicio termina una vez que el usuario haya emparejado todas las cartas. El diagrama se muestra en la figura 3.6.

Reacción a situaciones

En este ejercicio, al usuario se le mostrará una serie de situaciones sociales a través de texto e imágenes. Se tiene como objetivo realizar la expresión facial adecuada a la situación utilizando la cámara. El diagrama se muestra en la figura 3.7.

Muestra de reportes

Por último, en la actividad de reportes, el usuario podrá ver su progreso a través de gráficas. También, el usuario podrá seleccionar un rango de fechas para ver su progreso a mayor o menor detalle. El diagrama se muestra en la figura 3.8.

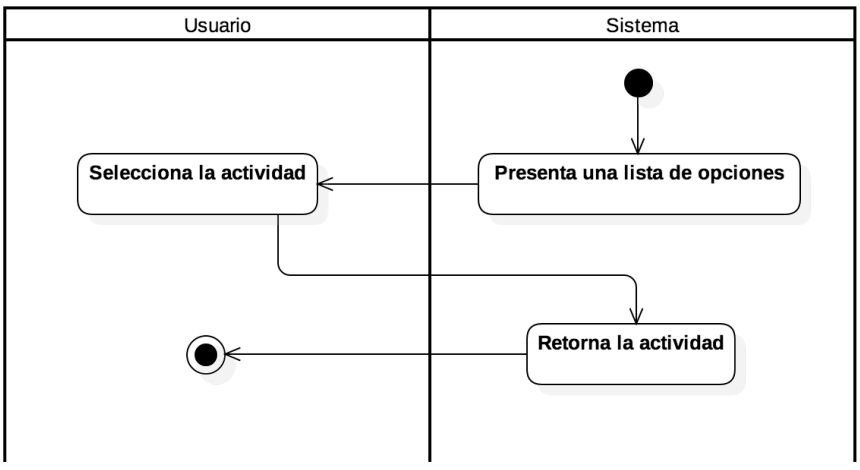

Figura 3.4: Diagrama de selección

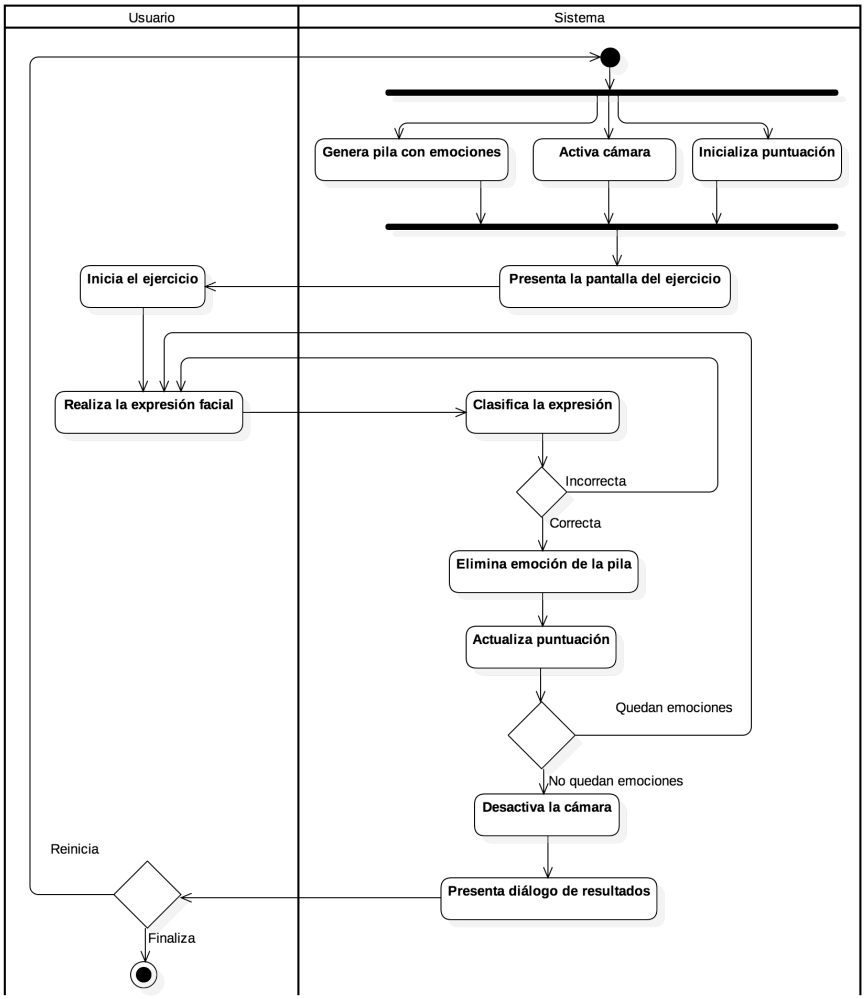

Figura 3.5: Diagrama para imitar emociones

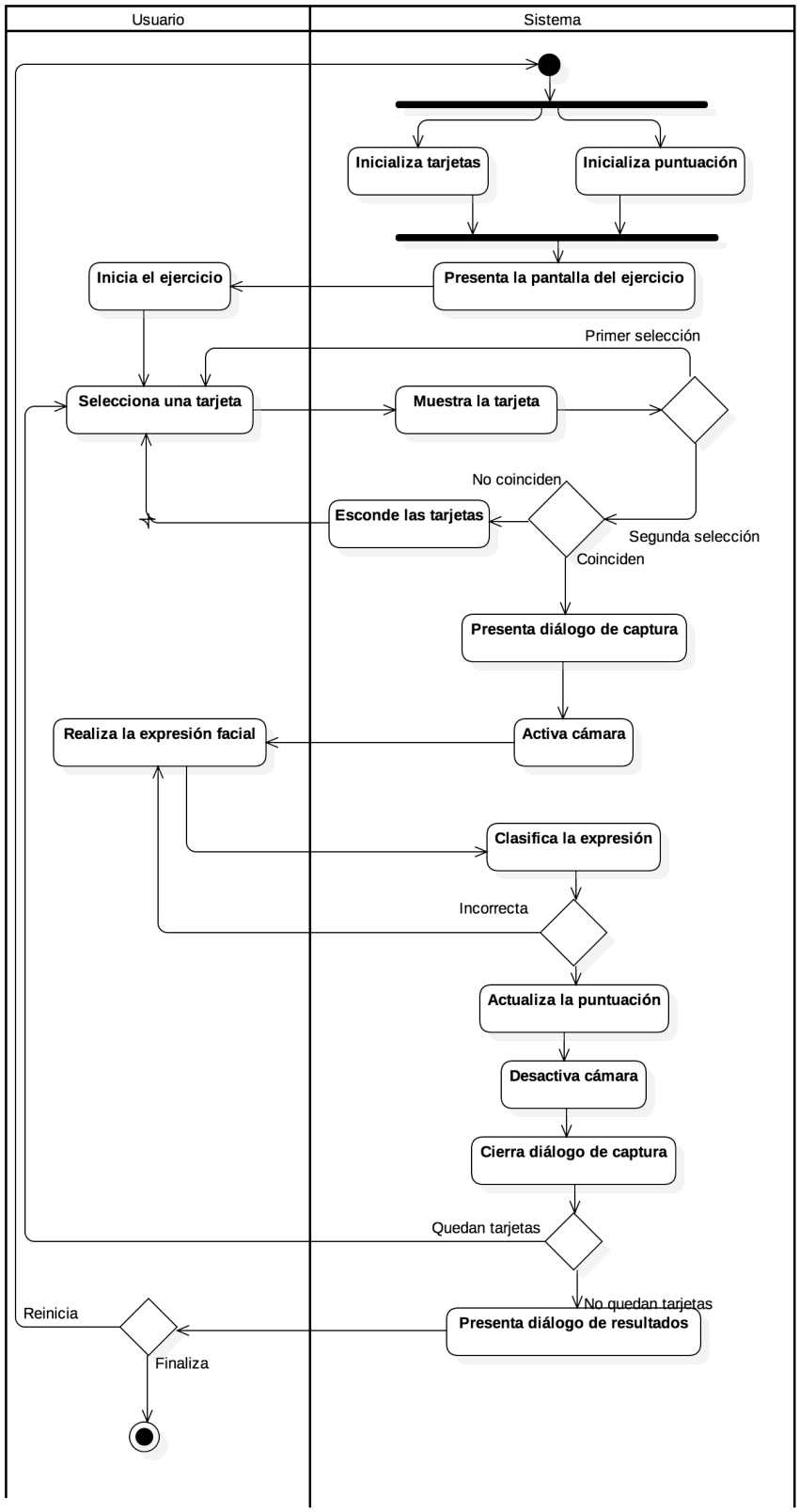

Figura 3.6: Diagrama para el memorama

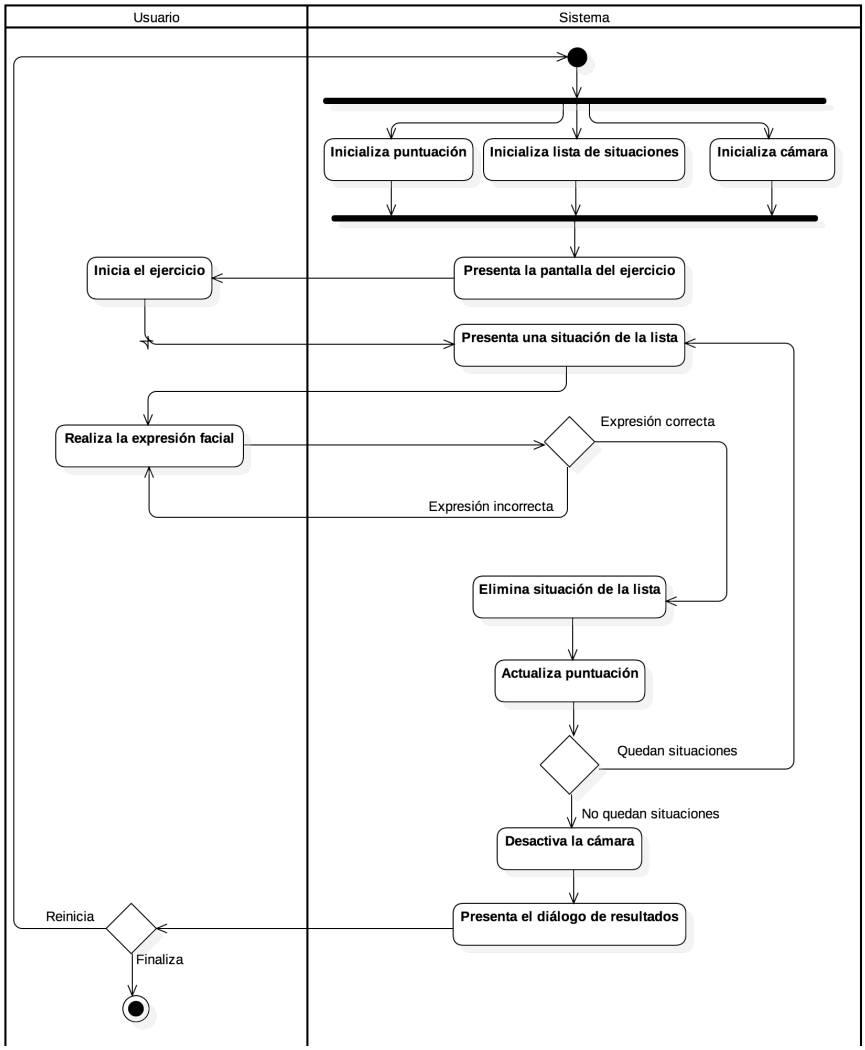

Figura 3.7: Diagrama para reacción a situaciones

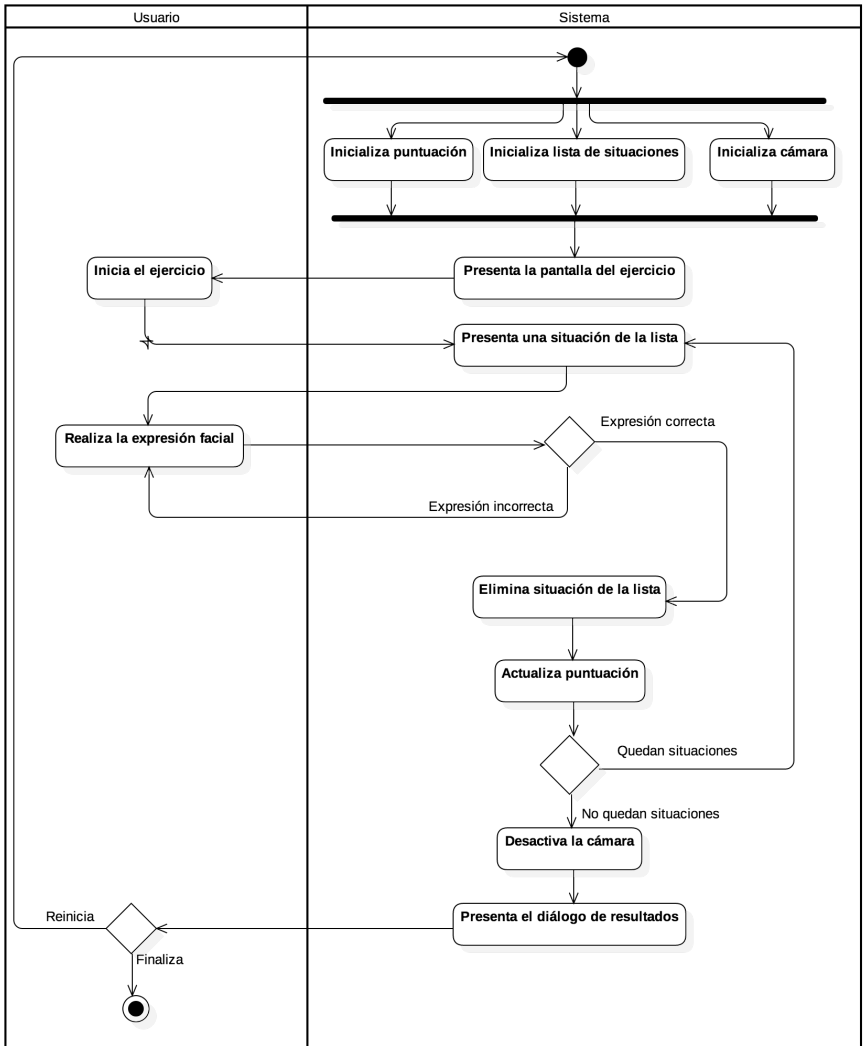

Figura 3.8: Diagrama para la muestra de reportes

3.2.2. Diagrama de clases

A continuación se describirán a través de diagramas de clases tres de los componentes más importantes del proyecto: el modelado de datos, la arquitectura MVP y la cámara con su clasificador de expresiones faciales.

Modelos de datos

Para poder manipular los datos que se manejará en el caso de uso “Catálogo”, se tienen que modelar de la forma que se indica en la figura 3.10. En este caso se conside-

ró el uso de un ORM (*Object Relational Mapping*) para hacer la implementación más sencilla.

Arquitectura MVP

La arquitectura MVP (Model View Presenter) es una forma de separar la lógica de la aplicación, para hacerla más sencilla de programar y de probar. Esta arquitectura se explica en la figura 3.9. En la figura 3.11 puede verse como un diagrama de clases.

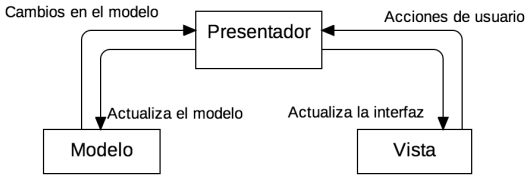

Figura 3.9: Arquitectura MVP

Cámara

El módulo para la cámara tiene la tarea de capturar y mostrar en tiempo real una vista previa de lo que se está capturando con la cámara del dispositivo. Al mismo tiempo, cada *frame* capturado debe de ser procesado y clasificado para que el sistema pueda reconocer la expresión facial que el usuario se encuentra haciendo. En la figura 3.12 se muestra cómo se implementaría el módulo.

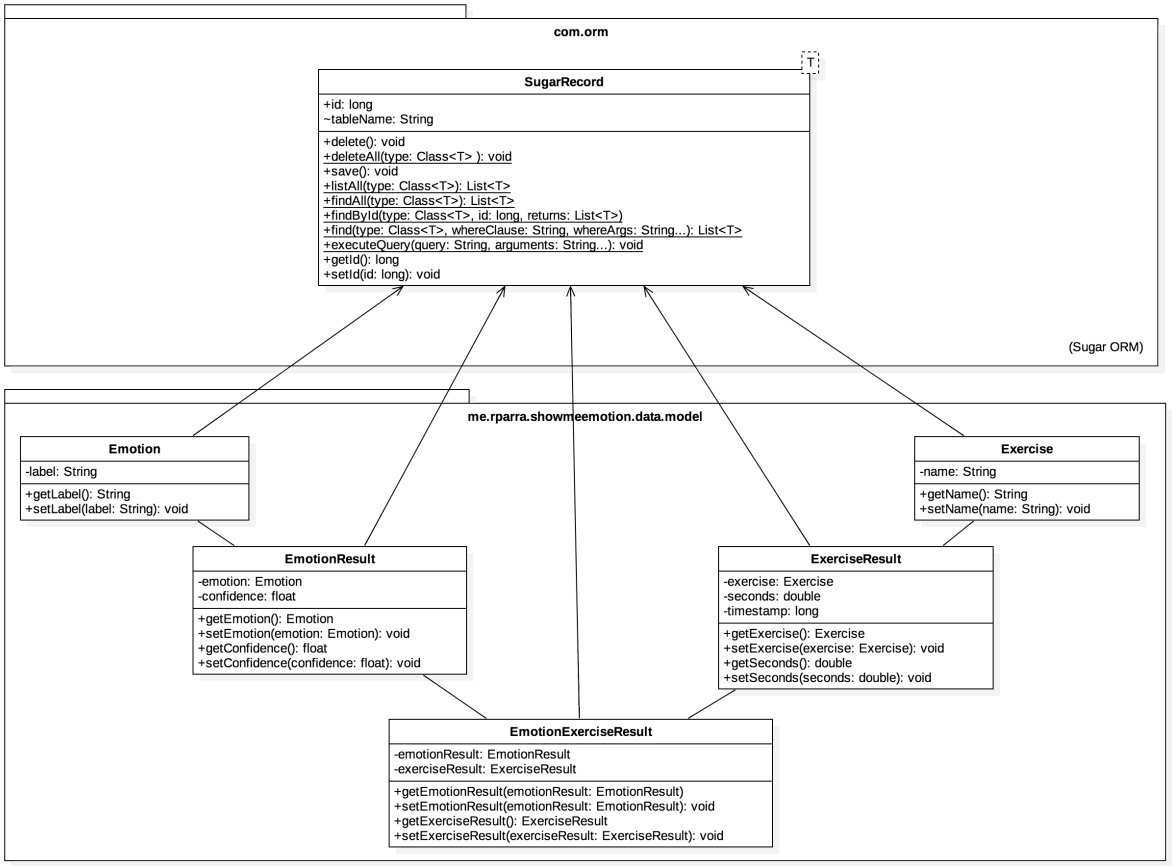

Figura 3.10: Diagrama de clases para los datos

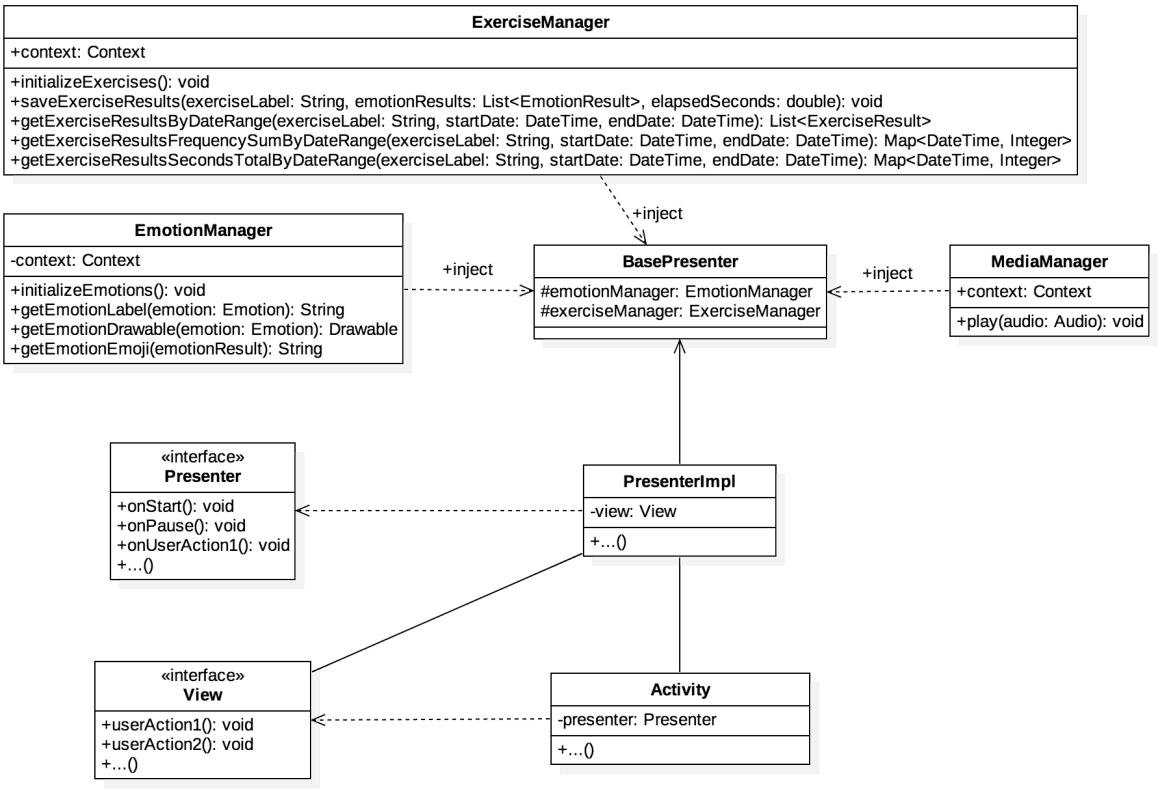

Figura 3.11: Diagrama de clases para los datos

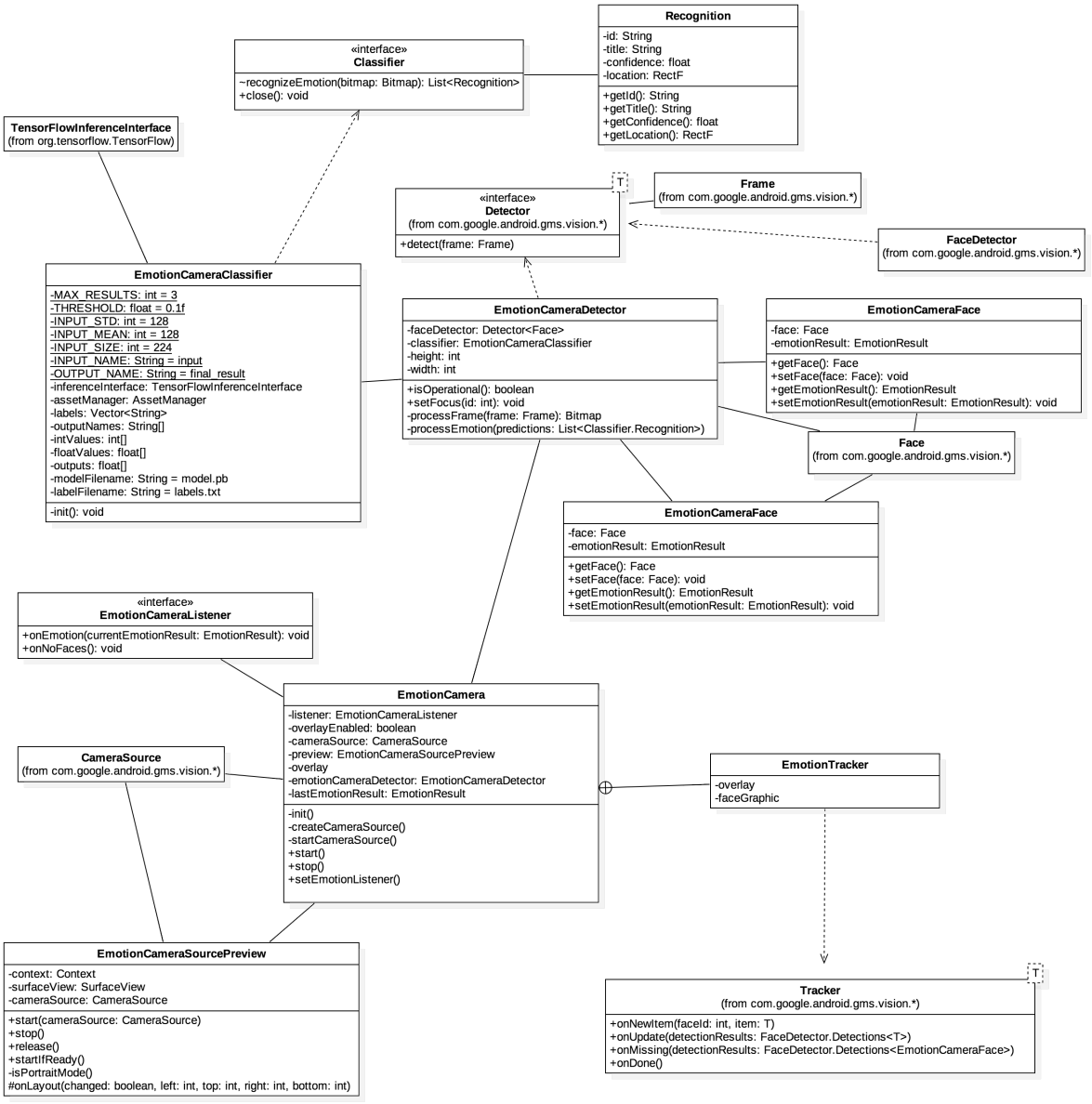

Figura 3.12: Diagrama de clases la cámara

3.2.3. Diagrama de componentes y despliegue

En la figura 3.13 se muestra cómo quedaría la aplicación finalizada. Debido a que se planea utilizar Android para implementar el sistema, se siguió el mismo modelo de despliegue que cualquier otra aplicación Android. En el diagrama se muestra la aplicación de Android compilada, y el dispositivo móvil con su cámara y almacenamiento.

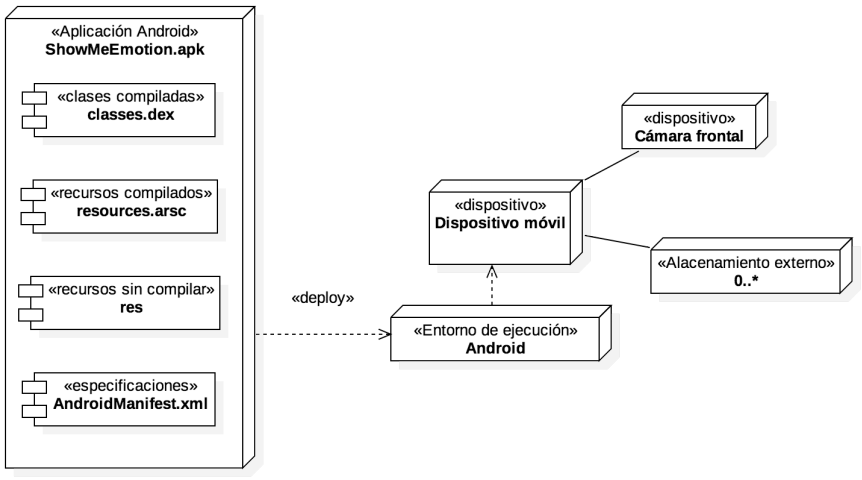

Figura 3.13: Diagrama de componentes y despliegue

3.3. Diseño de la base de datos

Para saber si los usuarios tienen un progreso en los ejercicios que realizan, se necesita tener de un registro con lo que se ha hecho. Para eso se implementará una base de datos con el esquema que se muestra en la figura 3.14.

Las tablas `Emotion` y `Exercise` definen las emociones y tipos de ejercicios disponibles en el programa. Mientras que las tablas `EmotionResult` y `ExerciseResult` son aquellas que guardan los resultados relacionados a cuando se termina un determinado ejercicio. Por último, la tabla `EmotionExerciseResult` sirve para relacionar cuáles emociones se usaron en un ejercicio.

Con este diseño, podemos preguntarle al sistema datos de interés como:

- Cuántos ejercicios ha hecho el usuario en cierto tiempo.
- Cuánto tiempo ha invertido en cierto ejercicio.
- Qué emociones realizó en un determinado ejercicio.
- Cuál es el ejercicio que más ha realizado.
- Entre otros.

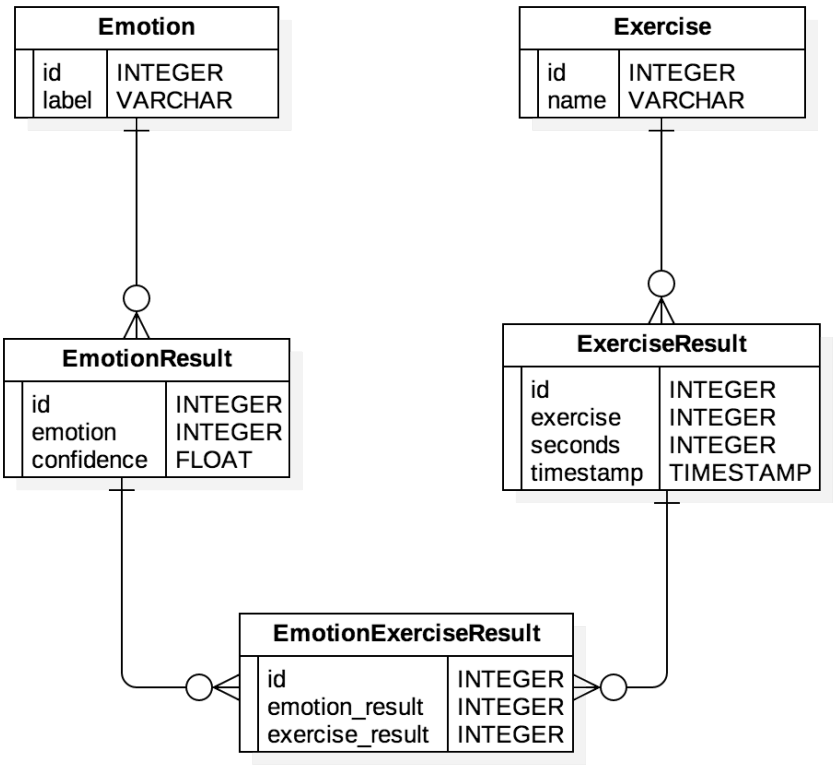

Figura 3.14: Diagrama entidad-relación de la base de datos

3.4. Diseño de la red neuronal

Para la tarea de clasificación en un sistema de visión por computadora, las redes convolucionales son aquellas que desempeñan mejor su trabajo. Sin embargo, muchas de las arquitecturas disponibles son muy grandes y no es conveniente usarlos para sistemas móviles.

Debido a lo anterior, se optó por utilizar una red neuronal ya existente: MobileNet [29], la cual se trata de una red convolucional especialmente optimizada para sistemas móviles y embebidos.

En la figura 3.15 se puede apreciar la arquitectura de esta red. La primer capa sirve como entrada, la cual recibe una imagen de 224 por 224 pixeles con tres canales de colores. Le sigue una serie de capas de convolución para procesar la imagen hasta

llegar a la última, la salida, la cual es una capa clasificadora de tipo *softmax*. Esta última capa se volvería a entrenar para cuatro diferentes etiquetas, las cuales serían: felicidad, tristeza, sorpresa y enojo.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 2. Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

Figura 3.15: Arquitectura MobileNet. Fuente: adaptado de [29]

3.5. Arquitectura propuesta

En la figura 3.16 se muestra la arquitectura del sistema, donde se describen dos entornos y cuatro módulos. A continuación se explicará cada uno de ellos.

3.5.1. Exterior

El entorno exterior es donde se encuentra el usuario. El usuario tiene la tarea de resolver los ejercicios que la aplicación le presente utilizando solamente su rostro.

3.5.2. Dispositivo móvil

Este es el entorno en el que el usuario interactúa con el sistema, en otras palabras es la aplicación que se utiliza en el celular. Este entorno cuenta con cuatro módulos:

- **Interfaz de gráfica:** Es la parte que el usuario ve. Se encarga de el manejo de la cámara y le presenta al usuario lo que se está capturando. Además, este módulo se encarga de la retroalimentación: le dirá al usuario la clasificación final del rostro que se encuentra haciendo.
- **Procesamiento de imágenes:** Tiene el propósito de procesar el cuadro capturado por la cámara para que pueda ser usado por el algoritmo de clasificación.
- **Clasificación:** Se usa una red neuronal previamente entrenada para clasificar los rostros en una emoción. La evaluación se la presentará al usuario (a través de la interfaz gráfica) y también la guardará en la base de datos (utilizando el módulo de reportes)
- **Reportes:** Se encarga de almacenar los resultados de la clasificación y de los ejercicios. Además de realizar las diferentes peticiones a la base de datos para que se puedan generar reportes.

3.5.3. Entorno de desarrollo

En este entorno no interactúa el usuario, es un paso previo a la publicación de la aplicación. Aquí se carga el algoritmo de aprendizaje máquina con los resultados de un dataset de expresiones faciales.

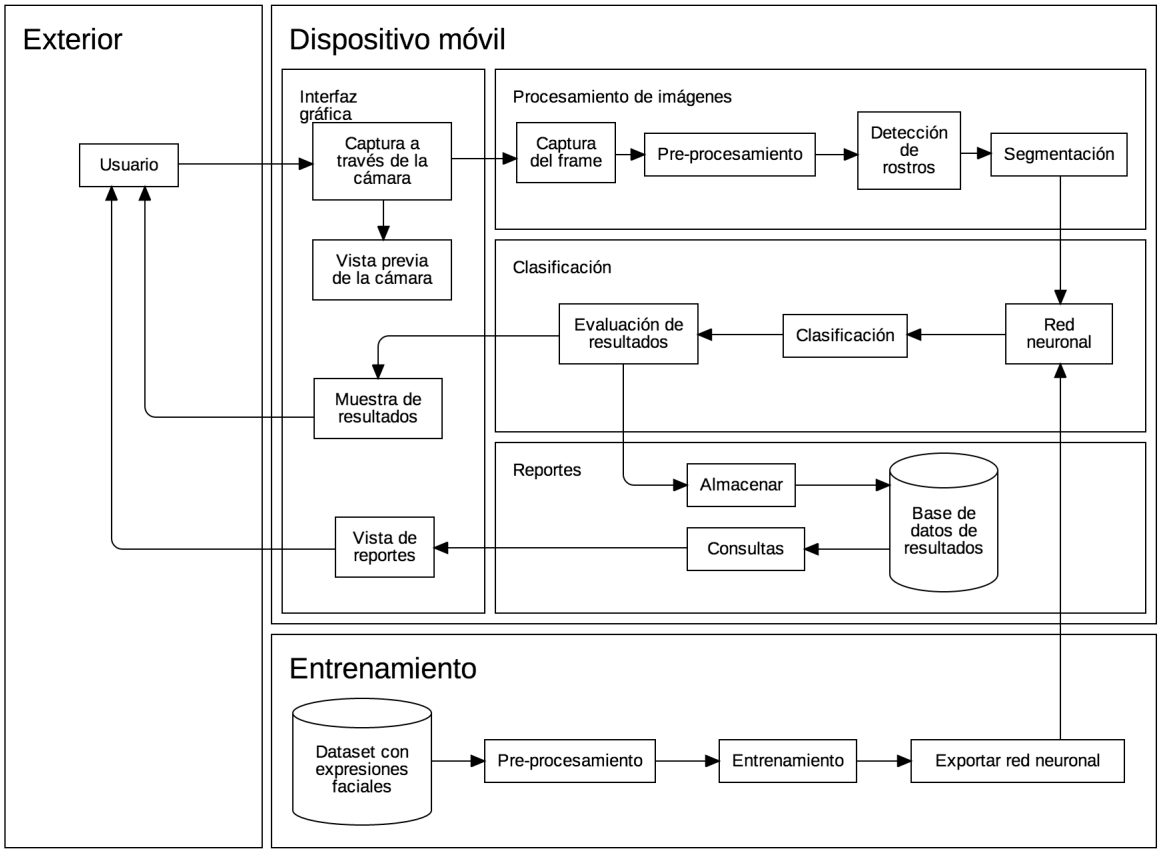

Figura 3.16: Arquitectura propuesta

Capítulo 4

Implementación del sistema

En el capítulo anterior pudimos observar cuáles eran los requerimientos para desarrollar el sistema. En este capítulo veremos cómo es que se llevó a cabo la implementación para hacerlo realidad. La implementación del sistema puede dividirse en dos partes: la primera, es el entrenamiento de la red neuronal la cual se usa para clasificar las expresiones faciales de los usuarios, y la aplicación móvil, la interface con la que interactúa el usuario final. Pero antes, se discutirá un poco de cómo se creó un entorno de desarrollo para poder trabajar con estas tecnologías.

La red neuronal fue hecha utilizando el *framework* para aprendizaje máquina de código libre desarrollado por Google: TensorFlow. Originalmente este framework fue hecho por los investigadores e ingenieros del proyecto *Google Brain* de la organización *Google AI*. Pero fue liberado al público recientemente.

La aplicación móvil se programó para el sistema operativo Android utilizando Java. Android es una plataforma para dispositivos móviles desarrollada igualmente por Google.

4.1. Entorno de desarrollo

Uno de los problemas más grandes, y a la vez comunes, que muchos desarrolladores enfrentan al crear sistemas que utilizan redes neuronales profundas es que necesitan contar con el equipo de cómputo adecuado. Actualmente, ya no es factible entrenar re-

des neuronales solamente con procesadores convencionales, es mucho más eficiente y rápido entrenarlas con equipos que cuenten con tarjetas gráficas u otro hardware especializado.

El equipo con el que se entrenó la red fue proporcionado por el laboratorio de sistemas inteligentes del Instituto Tecnológico de Hermosillo. Sus características se describen en la tabla 4.1, y puede verse cómo se instaló la tarjeta de video para auxiliar el entrenamiento en la figura 4.1.

Tabla 4.1 Características de la computadora de desarrollo

Sistema operativo:	Ubuntu 18.04 LTS
Procesador:	Intel Core i7-4790 @ 3.60GHz × 8
Memoria:	16 GiB
Tarjeta gráfica:	GeForce GTX 1060 6GB / PCIe / SS2
Disco:	1.0 TB

Figura 4.1: Instalando la tarjeta de video

Ahora, no basta con tener solamente el hardware, hay que crear un entorno de software que pueda aprovecharlo al máximo. La forma más sencilla y limpia que vimos para realizarlo fue a través de un contenedor Docker.

Docker es una plataforma para manejar contenedores de software. Los contenedores son paquetes ligeros, autónomos y ejecutables que incluyen todo lo que un sistema necesita para funcionar, ya sea código, sistemas de tiempo de ejecución (*runtimes*), configuración, herramientas y librerías del sistema [30]. Usando esta herramienta, creamos una imagen personalizada para tener un entorno de desarrollo basado en Jupyter Notebook que incluyera los paquetes necesarios para usar librerías como: OpenCV, TensorFlow, Numpy, y Pandas, además de tener instalado la librería CUDA para poder aprovechar la tarjeta de video. A continuación se muestra el archivo `Dockerfile` con el cual se construyó la imagen personalizada:

```
FROM nvidia/cuda:9.0-cudnn7-devel-ubuntu16.04
MAINTAINER Ramón Parra <ramon@rparra.me>

# update ubuntu
RUN apt-get update
RUN apt-get upgrade -y

# install opencv dependencies
RUN apt-get install -y build-essential cmake pkg-config
RUN apt-get install -y libjpeg8-dev libtiff5-dev libjasper-dev
 ↪ libpng12-dev
RUN apt-get install -y libavcodec-dev libavformat-dev libswscale-dev
 ↪ libv4l-dev
RUN apt-get install -y libxvidcore-dev libx264-dev
RUN apt-get install -y libgtk-3-dev
RUN apt-get install -y libhdf5-serial-dev graphviz
RUN apt-get install -y libopenblas-dev libatlas-base-dev gfortran
RUN apt-get install -y python2.7-dev python3.5-dev
RUN apt-get install -y wget unzip

WORKDIR /tmp

# opencv source
RUN wget -O opencv.zip
 ↪ https://github.com/Itseez/opencv/archive/3.4.0.zip
RUN unzip opencv.zip

RUN wget -O opencv_contrib.zip
 ↪ https://github.com/Itseez/opencv_contrib/archive/3.4.0.zip
RUN unzip opencv_contrib.zip

# setup python
RUN apt-get install -y python3-pip
RUN pip3 install --upgrade pip
```

```
RUN pip3 install numpy

# compile opencv
RUN mkdir -p /tmp/opencv-3.4.0/build
WORKDIR /tmp/opencv-3.4.0/build
RUN cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D WITH_CUDA=OFF \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D INSTALL_C_EXAMPLES=OFF \
    -D OPENCV_EXTRA_MODULES_PATH=/tmp/opencv_contrib-3.4.0/modules \
    -D PYTHON_EXECUTABLE=/usr/bin/python3 \
    -D BUILD_EXAMPLES=ON ..
RUN make -j4
RUN make install

RUN mv /usr/local/lib/python3.5/dist-packages/cv2*.so
 ↪  /usr/local/lib/python3.5/dist-packages/cv2.so

# install and configure jupyter notebook
RUN apt-get install -y npm nodejs-legacy
RUN npm install -g configurable-http-proxy
RUN pip3 install jupyter
RUN pip3 install jupyterhub

RUN useradd -ms /bin/bash jupyter
RUN echo 'jupyter:jupyter' | chpasswd

VOLUME /home/jupyter
WORKDIR /home/jupyter/.jupyterhub

EXPOSE 8000
EXPOSE 6006
```

```
# install deep learning libraries
RUN pip3 install scipy matplotlib pillow
RUN pip3 install imutils h5py requests progressbar2
RUN pip3 install scikit-learn scikit-image
RUN pip3 install pandas
RUN pip3 install tensorflow-gpu
RUN pip3 install tensorflow_hub
RUN pip3 install keras

CMD [”jupyterhub”]
```

Una vez construida, se ejecuta la imagen usando `nvidia-docker`, el cual permite la comunicación de los contenedores con la tarjeta gráfica. A continuación se muestran la forma de ejecutarlo:

```
nvidia-docker run -d \
    --restart always \
    -p 8000:8000 \
    -p 6006:6006 \
    -v /home/admn/notebooks:/home/jupyter \
    --name dl-server \
    dl-server
```

Una vez que el contenedor se ejecuta, podemos entrar al entorno de desarrollo que creamos entrando a la dirección http://localhost:8000 desde nuestro navegador, como se muestra en la figura 4.2.

4.2. Red neuronal

A continuación se explicará cómo fue que se entrenó la red neuronal para resolver el problema de la clasificación de expresiones faciales.

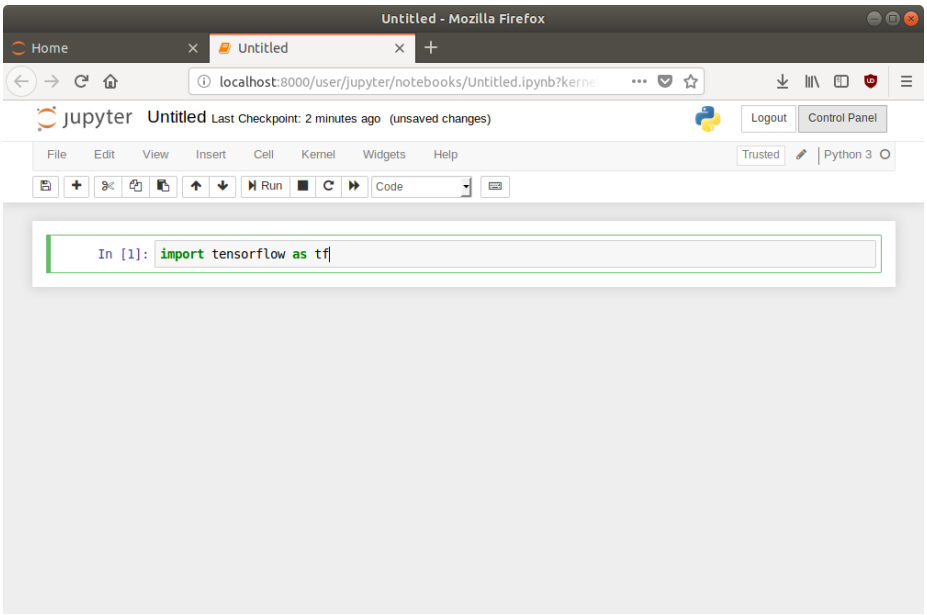

Figura 4.2: Entorno de desarrollo con Jupyter

4.2.1. Procesamiento del conjunto de datos

Las redes neuronales ocupan aprender de una enorme cantidad de datos, y si queremos que puedan reconocer cuál expresión existe en una cara, ocuparemos un conjunto de datos (o *dataset*) con miles de rostros etiquetados con su respectiva expresión facial.

Afortunadamente, ya existe un conjunto de datos para entrenar clasificadores de expresiones faciales. Este conjunto de datos se trata de uno llamado FER2013, publicado por primera vez en "*Challenges in representation learning: A report on three machine learning contests*" [31]. En su momento, su objetivo fue el de ser parte de una competencia probar la efectividad de distintos modelos de clasificación, y actualmente se encuentra disponible en la página de Kaggle[1] para que otros desarrolladores puedan utilizarlo.

Este conjunto de datos incluye alrededor de 35,000 imágenes en blanco y negro de personas realizando una de las siguientes expresiones faciales: felicidad, tristeza, sorpresa, disgusto, enojo, y neutral. Se puede apreciar una muestra de este conjunto de datos en la figura 4.3. Para nuestro objetivo, sólo tomaremos cuatro categorías para hacer más rápido el entrenamiento y eliminar posibles incógnitas a las cuales nos podríamos en-

[1] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

frentar y se encuentran fuera del objetivo del trabajo de tesis, por ejemplo: ¿a qué nos referimos con una expresión facial neutra?

Figura 4.3: Muestra de FER2013

El conjunto de datos se distribuye a partir de un archivo tipo csv (*comma separated values*), el cual es básicamente un archivo de texto donde, en cada línea, viene la información relacionada a una expresión distinta. Cada línea contiene un número que representa la categoría de la expresión facial, una serie de números que representan los pixeles de la imagen, y por último, una etiqueta para saber si el registro se encuentra catalogado para entrenamiento o para prueba. La estructura del archivo puede verse en la figura 4.4.

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Figura 4.4: Estructura del archivo .csv

Como se mencionó anteriormente, ocupamos eliminar las categorías que no nos interesan además de la información que no ocupamos. Adicionalmente, para lograr una mejor distribución, ocupamos mezclar los rostros etiquetados con “enojo” y “disgusto”. Una forma muy práctica de manejar grandes conjuntos de datos es con la librería pandas

de Python. A continuación se muestra cómo fue que se realizó el proceso de carga y limpieza:

```
import pandas as pd

# load the dataset
data = pd.read_csv('fer2013.csv')

# cleaning
data = data.drop('Usage', axis=1)
data = data[data.emotion != 2]
data = data[data.emotion != 6]
data.index = pd.RangeIndex(len(data.index))
data.index = range(len(data.index))

# merge disgust with angry
data['emotion'] = data['emotion'].replace([1], 0)
```

La nueva distribución del conjunto de datos puede verse en la figura 4.5.

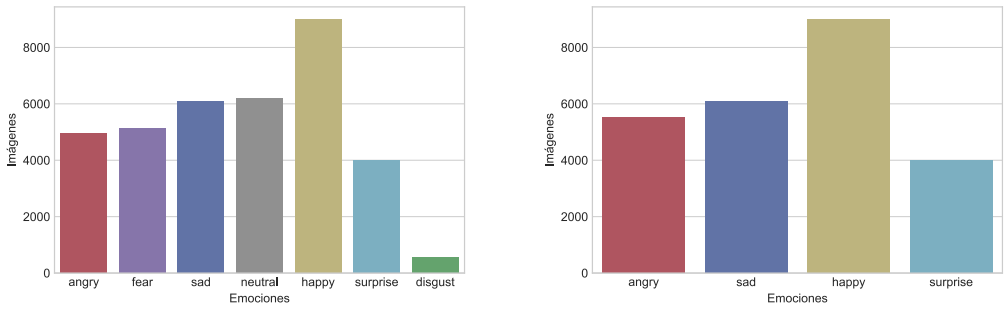

Figura 4.5: Nueva distribución de FER2013, resultados pregunta 6

Ahora bien, para poder utilizar todos estos datos para la clasificación, optamos por exportar cada renglón del archivo de texto a imagen. Este proceso fue hecho gracias a la librería OpenCV y puede verse a continuación:

```
import numpy as np
import cv2
import os
```

```
import shutil

# dictionary with the available emotions
emotions = {
    0: 'angry',
    3: 'happy',
    4: 'sad',
    5: 'surprise'
}

# pixel width and height
w = h = 48

# separate labels and features
n_samples = len(data)
labels = data['emotion']

faces = np.zeros((n_samples, w, h))
for i in range(n_samples):
    faces[i] = np.fromstring(data['pixels'][i], dtype=np.uint8, sep='
 ↪  ').reshape(w, h)

# create the directory to export
outloc = './datasets/fer2013/'
if os.path.exists(outloc):
    shutil.rmtree(outloc)

os.makedirs(outloc)

for emotion in emotions:
    os.makedirs(outloc + emotions[emotion])

# export the set
for idx, face in enumerate(faces):
```

```
    file_name = outloc + emotions[labels.data[idx]] + '/' + str(idx) +
↪   '.jpg'
    cv2.imwrite(file_name, face.reshape(w, h))
    print('Exported: ' + file_name)

print('Done!')
```

El resultado puede verse en la figura 4.6.

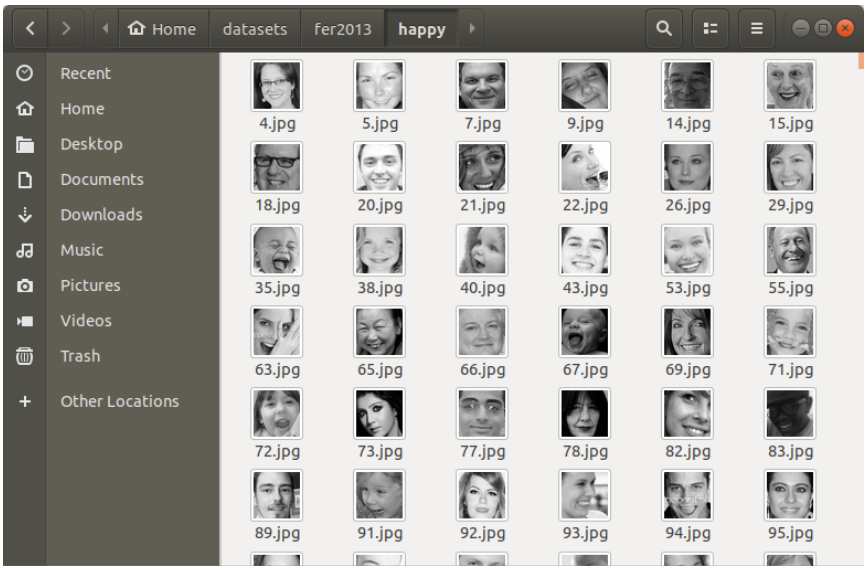

Figura 4.6: Archivos de FER2013

4.2.2. Entrenamiento de la red neuronal

Una vez que tenemos las imágenes del conjunto de datos, podemos entrenar una red neuronal para que realice la clasificación. Como se mencionó anteriormente, estamos utilizando la librería TensorFlow para realizar el entrenamiento de una red de tipo MobileNet. Siendo ésto una tarea relativamente común en TensorFlow, existen herramientas que facilitan dicho entrenamiento.

El entrenamiento se realizó siguiendo los pasos publicados en [32]. Se ocupa descargar un archivo llamado `retrain.py`[2] y ejecutarlo desde el entorno de desarrollo de la siguiente manera:

[2]https://github.com/tensorflow/hub/raw/r0.1/examples/image_retraining/retrain.py

```
python3 retrain.py \
  --image_dir ./datasets/fer2013 \
  --learning_rate=0.001 \
  --testing_percentage=15 \
  --validation_percentage=15 \
  --train_batch_size=32 \
  --validation_batch_size=-1 \
  --flip_left_right True \
  --random_scale=30 \
  --random_brightness=30 \
  --random_crop=5 \
  --eval_step_interval=100 \
  --how_many_training_steps=30000 \
  --tfhub_module
 ↪  https://tfhub.dev/google/imagenet/mobilenet_v2_100_224/feature_vector/1
 ↪  \
  --output_graph=$(date '+%Y%m%d%H%M%S')_fer2013.pb \
  --output_labels=fer2013.txt
```

Adicionalmente a los parámetros que se observan en el ejemplo citado anteriormente, se añadió otros tales como: `--flip_left_right`, `--random_scale` o `--random_brightness`. Esto parámetros crean variaciones en los datos de entrenamiento para hacer un clasificador más robusto, a esta técnica se le llama aumento de datos (*data augmentation*).

Los resultados que se obtuvieron después del entrenamiento pueden verse en la figura 4.7, éstos indican que la red neuronal tuvo un 63 % de acierto en el conjunto de entrenamiento y un 60 % en el conjunto de prueba después de 30,000 pasos de entrenamiento. Si comparamos nuestro resultado con el estado del arte publicado en [31], podemos ver que éste llega hasta un 71 %. Tomando en cuenta la complejidad del problema, el reducido tamaño de la red que necesita un dispositivo móvil, el hardware y el tiempo disponible, no se pudo llegar a la misma puntuación. Sin embargo, creemos que nuestro resultado es suficiente para el objetivo de esta aplicación.

La salida del script se define con el parámetro `--output_graph`, el cual es un archivo que representa una red congelada, donde el modelo y los pesos están listos para

ser implementados en otra aplicación sin necesidad de volver a entrenar.

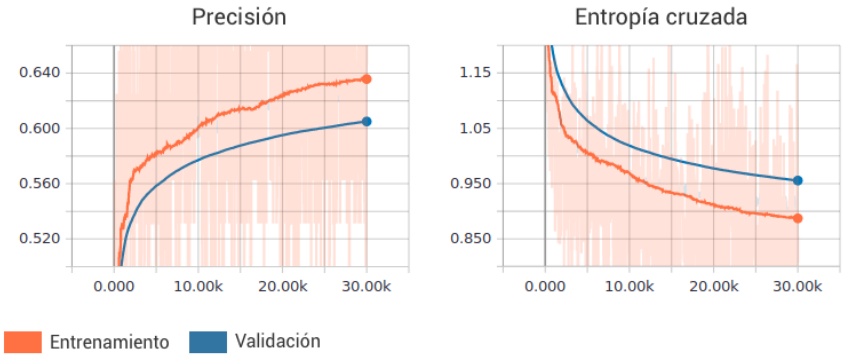

Figura 4.7: Resultados del entrenamiento

4.3. Aplicación móvil

En esta sección se explicará cuál fue el proceso de desarrollo de la aplicación móvil, y cómo fue que se implementó la red neuronal que se obtuvo anteriormente.

4.3.1. Estructura de la aplicación

La estructura del proyecto está basada en la arquitectura modelo-vista-presentador. La capa del modelo define los datos que maneja la base de datos y define algunas de las reglas para inicializar y hacer peticiones a la base de datos. La vista es la parte que ve el usuario, tiene el objetivo de mostrar la información y esperar las acciones del usuario. Por último, el presentador es el que tiene la lógica de la aplicación, se comunica con los modelos y actualiza la vista correspondiente.

Modelos

Los modelos de la aplicación se encuentran basados en la base de datos que se presentó en el capítulo 3. Todos están definidos como POJO (*Plain Old Java Object*) y extienden a la clase `SugarRecord` para poder manejarlos con un ORM (*Object Relational Mapping*).

A continuación, se muestra un ejemplo de cómo se implementa un modelo dentro de la aplicación:

```
public class Emotion extends SugarRecord<Emotion> {
    @Ignore private final String TAG = Emotion.class.getSimpleName();

    private String label;

    public Emotion() { /* */ }

    public Emotion(String label) {
        this.label = label;
    }

    public String getLabel() {
        return label;
    }

    public void setLabel(String label) {
        this.label = label;
    }

    public boolean equals(Emotion emotion) {
        return this.label.equals(emotion.getLabel());
    }

    @Override
    public String toString() {
        return String.format("%s: id: %d, label: %s",
                TAG.toUpperCase(), id, label);
    }
}
```

Para manejar la lógica común de la base de datos (como inicializarse, hacer peticiones comunes), para cada modelo de datos se incluye una clase tipo *Manager*, como se puede ver en el siguiente ejemplo:

```
public class EmotionManager extends BaseManager {
    // ...
```

```
    private Context context;

    public EmotionManager(Context context) {
        this.context = context;
    }

    private void initializeEmotionsDatabase() {
        // ...
    }

    public void initializeEmotions() {
        // ...
    }

    public Emotion getEmotionByLabel(String label) {
        // ...
    }

    public String getEmotionLocalizedLabel(Emotion emotion) {
        // ...
    }

    // ...

}
```

Vistas

En Android lo más similar a una vista es una actividad, para que éstas puedan usarse como tal sólo deben de tener dos objetivos: actualizarse con los datos que provengan del presentador y esperar una interacción del usuario para avisarle al presentador.

Las vistas se definen en una interface y se implementan en la actividad, y cada actividad crea su objeto presentador como puede verse en el siguiente ejemplo:

```
interface EmotionsView {
```

```
    void showCameraOverlay();
    void removeCameraOverlay();
    void startEmotionsCamera();
    void stopEmotionsCamera();
    void showEmotionOnStatus(String label, String emoji);
    void showNoFacesOnStatus();
    void showSuccessDialog(double elapsedSeconds);

    // ...

}

public class EmotionsActivity extends BaseActivity
    implements EmotionsView {

    // ...

    private EmotionsPresenter emotionsPresenter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // ...
        emotionsPresenter = new EmotionsPresenterImpl(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        emotionsPresenter.start();
    }

    @Override
    protected void onPause() {
        emotionsPresenter.pause();
```

```
        super.onPause();
    }

    // ...

}
```

Presentadores

Al igual que la vista, las acciones del presentador se definen en una interface y luego se implementan en una clase por separado. Los presentadores pueden actualizar su vista al ejecutar sus métodos gracias a que al momento de crearlos se pasa la instancia de la vista:

```
interface EmotionsPresenter {
    void start();
    void pause();
    void restart();
    // ...
}

class EmotionsPresenterImpl extends BasePresenter
    implements EmotionsPresenter {

    // ...

    private EmotionsView emotionsView;

    // ...

    EmotionsPresenterImpl(EmotionsView emotionsView) {
        this.emotionsView = emotionsView;
        // ...
    }

    // ...
```

```
    @Override
    public void start() {
        emotionsView.startEmotionsCamera();
        initialize();
    }

    @Override
    public void pause() {
        emotionsView.stopEmotionsCamera();
    }

    // ...

}
```

4.3.2. Vista previa de la cámara y detección de rostros

Para tener una cámara que pueda clasificar rostros en tiempo real, es necesario que podamos realizar dos tareas desde la aplicación: la primera es la de poder capturar y manipular los cuadros que retorna la cámara en modo vista previa, y la segunda es detectar si existe o no un rostro en ese cuadro.

La forma más práctica de realizar esta tarea en Android es usando la librería *Mobile Vision* de Google, la cual pone a nuestra disposición una forma de inicializar y manejar la cámara del dispositivo móvil y además nos puede decir si existen o no rostros en esa imagen.

Para crear la cámara se implementó el siguiente código en una vista personalizada de Android:

```
public class EmotionCamera extends BaseView {
    // ...

    private EmotionCameraDetector emotionCameraDetector;
```

```
    // ...

    private void createCameraSource() {
        emotionCameraDetector = new
↪   EmotionCameraDetector(getContext());
        emotionCameraDetector.setProcessor(
                new MultiProcessor.Builder<>(new
                 ↪  EmotionTrackerFactory())
                        .build());

        // ...

        MultiDetector multiDetector = new MultiDetector.Builder()
                .add(emotionCameraDetector)
                .build();

        cameraSource = new CameraSource.Builder(getContext(),
↪   multiDetector)
                .setRequestedPreviewSize(224, 224)
                .setFacing(CameraSource.CAMERA_FACING_FRONT)
                .setRequestedFps(30.0f)
                .build();
    }

    // ...

}
```

Una vez creada, puede llamarse el método `startCameraSource` desde otra parte del programa para iniciar la captura de cuadros:

```
public class EmotionCamera extends BaseView {
    // ...

    private void startCameraSource() {
        // check that the device has play services available
```

```
        int code = GoogleApiAvailabi-
         ↪  lity.getInstance().isGooglePlayServicesAvailable(
                getContext());
        if (code != ConnectionResult.SUCCESS) {
            Log.e(TAG, "startCameraSource: GooglePlayServices not
 ↪  available");
        }

        if (cameraSource != null) {
            try {
                preview.start(cameraSource, overlay);
            } catch (IOException e) {
                Log.e(TAG, "startCameraSource: unable to start camera
 ↪  source", e);
                cameraSource.release();
                cameraSource = null;
            }
        }
    }

    // ...

}
```

Estos cuadros pueden capturarse en una clase que implementa la interface `Detector`:

```
public class EmotionCameraDetector extends Detector<EmotionCameraFace>
 ↪  {
    // ...

    private Detector<Face> faceDetector;

    // ...

    @Override
```

```
    public SparseArray<EmotionCameraFace> detect(Frame frame) {
        SparseArray<Face> faces = faceDetector.detect(frame);
        SparseArray<EmotionCameraFace> faceEmotion = new
↪   SparseArray<>();

        if (faces.size() > 0) {
            // detectar expresiones faciales
        }

        return faceEmotion;
    }
}
```

Como podemos ver en el código anterior, además de tener acceso directo al objeto `Frame`, la librería también nos indica si existen o no caras en la imagen. En caso de que se detecte una debemos llamar al clasificador para que realice su trabajo.

4.3.3. Red neuronal en Android

Antes de utilizar la red neuronal, necesitamos inicializar la librería TensorFlow y después cargar el modelo correspondiente. Por conveniencia, se realiza esta carga dentro de la misma clase del componente anterior. Para inicializar Tensorflow desde Android se debe incluir el siguiente código:

```
public class EmotionCamera extends BaseView {
    // ...

    static {
        System.loadLibrary("tensorflow_inference");
       Log.d(TAG, "load: Tensorflow version: " + TensorFlow.version());
    }

    // ...
}
```

En la sección anterior vimos cómo fue que después del entrenamiento de la red

neuronal se congeló en un archivo llamado `fer2013.pb`. Este archivo debe de cargarse al momento de que se inicializa la cámara en la aplicación, se hace a través del siguiente código:

```
class EmotionCameraClassifier implements Classifier {

    // ...

    private static final String OUTPUT_NAME = "final_result";

    private TensorFlowInferenceInterface inferenceInterface;
    private AssetManager assetManager;

    private Vector<String> labels = new Vector<>();
    private String[] outputNames;
    private int[] intValues;
    private float[] floatValues;
    private float[] outputs;

    private String modelFilename = "fer2013.pb";
    private String labelFilename = "fer2013.txt";

    public EmotionCameraClassifier(Context context) {
        this.assetManager = context.getAssets();
        init();
    }

    private void init() {
        // load labels
        BufferedReader br = null;
        try {
            br = new BufferedReader(new
↪   InputStreamReader(assetManager.open(labelFilename)));
            String line;
            while ((line = br.readLine()) != null) {
```

```
                labels.add(line);
            }
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        Log.i(TAG, String.format("Labels: %s", labels));

        // create the inference interface
        inferenceInterface = new
↪   TensorFlowInferenceInterface(assetManager, modelFilename);
        final Operation operation =
         ↪  inferenceInterface.graphOperation(OUTPUT_NAME);
        final int numClasses = (int)
         ↪  operation.output(0).shape().size(1);
        Log.i(TAG, String.format("Read %d labels, output layer size is
↪   %d",
                labels.size(), numClasses));

        // pre-allocate buffers
        outputNames = new String[] {OUTPUT_NAME};
        intValues = new int[INPUT_SIZE * INPUT_SIZE];
        floatValues = new float[INPUT_SIZE * INPUT_SIZE * 3];
        outputs = new float[numClasses];
    }

    // ...

}
```

4.3.4. Clasificación de expresiones faciales en Android

Una vez que se detecta que existe un rostro en el cuadro capturado por la cámara, esa imagen se procesa y se pasa por el clasificador previamente entrenado. En la

aplicación móvil, el clasificador se representa por la siguiente interface:

```
interface Classifier {
    class Recognition {
        private final String id;
        private final String title;
        private final Float confidence;
        private final RectF location;

        public Recognition(
                final String id, final String title, final Float
                 ↪  confidence, final RectF location) {
            this.id = id;
            this.title = title;
            this.confidence = confidence;
            this.location = location;
        }

        // ...

    }

    List<Recognition> recognizeEmotion(Bitmap bitmap);
    void close();
}
```

El método `recognizeEmotion` recibe un objeto tipo mapa de bits que se encuentra en el objeto `Frame` de la librería *Mobile Vision*, para clasificar y nos retorna una lista con las predicciones. Esta función está implementada de acuerdo a como se muestra en el código fuente del ejemplo para un clasificador en Android en el repositorio de TensorFlow [3]:

```
class EmotionCameraClassifier implements Classifier {
    private static final int MAX_RESULTS    = 3;
```

[3] https://github.com/tensorflow/tensorflow/blob/r1.3/tensorflow/examples/android/src/org/tensorflow/demo/TensorFlowImageClassifier.java

```
    private static final float THRESHOLD    = 0.1f;
    private static final float INPUT_STD    = 128;

    private static final int INPUT_MEAN     = 128;
    public static final int INPUT_SIZE      = 224;

    // ...

    private Vector<String> labels = new Vector<>();
    private String[] outputNames;
    private int[] intValues;
    private float[] floatValues;
    private float[] outputs;

    // ...

    @Override
    public List<Recognition> recognizeEmotion(Bitmap bitmap) {
        // Preprocess the image data from 0-255 int to normalized float
        ↪ based
        // on the provided parameters.
        bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0,
↪ bitmap.getWidth(), bitmap.getHeight());
        for (int i = 0; i < intValues.length; ++i) {
            final int val = intValues[i];
           floatValues[i * 3 + 0] = (((val >> 16) & 0xFF) - INPUT_MEAN)
↪ / INPUT_STD;
            floatValues[i * 3 + 1] = (((val >> 8) & 0xFF) - INPUT_MEAN)
↪ / INPUT_STD;
            floatValues[i * 3 + 2] = ((val & 0xFF) - INPUT_MEAN) /
↪ INPUT_STD;
        }
```

```
        // Copy the input data into TensorFlow.
        inferenceInterface.feed(INPUT_NAME, floatValues, 1, INPUT_SIZE,
↪  INPUT_SIZE, 3);

        // Run the inference call
        inferenceInterface.run(outputNames, false);

        // Copy the output Tensor back into the output array
        inferenceInterface.fetch(OUTPUT_NAME, outputs);

        // Find the best classifications
        PriorityQueue<Recognition> pq =
                new PriorityQueue<>(
                        3,
                        new Comparator<Recognition>() {
                            @Override
                            public int compare(Recognition lhs,
                             ↪  Recognition rhs) {
                              return Float.compare(rhs.getConfidence(),
                                  ↪  lhs.getConfidence());
                            }
                        });

        for (int i = 0; i < outputs.length; ++i) {
            if (outputs[i] > THRESHOLD) {
                pq.add(new Recognition("" + i, labels.size() > i ?
↪  labels.get(i) : "unknown", outputs[i], null));
            }
        }

        final ArrayList<Recognition> recognitions = new
         ↪  ArrayList<Recognition>();
        int recognitionsSize = Math.min(pq.size(), MAX_RESULTS);
        for (int i = 0; i < recognitionsSize; ++i) {
```

```
            recognitions.add(pq.poll());
        }

        return recognitions;
    }

    // ...
}
```

4.3.5. Clasificación en tiempo real con la cámara

La cámara de *Mobile Vision* y el clasificador con TensorFlow se usan en conjunto dentro de la clase `EmotionCamera`. La cual tiene como atributo la siguiente interface que sirve como un *listener*:

```
public interface EmotionCameraListener {
    void onEmotion(EmotionResult currentEmotionResult, boolean
     ↪  finalEmotion);
    void onNoFaces();
}
```

De esta forma, cualquier clase que lo implemente, puede saber cuál es el estado actual del rostro que se encuentra en la cámara. Un ejemplo se puede ver a continuación:

```
public class EmotionsActivity extends BaseActivity
    implements EmotionCameraListener {
    // ...

    @BindView(R.id.emotion_camera) protected EmotionCamera
     ↪  emotionCamera;

    // ...

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // ...
```

```
        emotionCamera.setEmotionListener(this);
        // ...
    }

    @Override
    protected void onStart() {
        emotionCamera.start();
    }

    @Override
    protected void onPause() {
        emotionCamera.stop();
    }

    // ...

    @Override
    public void onEmotion(EmotionResult currentEmotionResult, boolean
     ↪  finalEmotion) {
        Log.d(TAG, "current emotion label: " +
↪   currentEmotionResult.getEmotion().getLabel());
        Log.d(TAG, "current emotion confidence: " +
↪   currentEmotionResult.getConfidence())
    }

    @Override
    public void onNoFaces() {
        // ...
    }

    // ...

}
```

En el XML que describe la interface de la actividad de Android, solamente hay

que incluir el siguiente código:

```
<me.rparra.showmeemotion.ui.common.emotioncamera.EmotionCamera
    android:id="@+id/emotion_camera"
    android:layout_width="220dp"
    android:layout_height="220dp" />
```

4.3.6. Interface de usuario

Al iniciar la aplicación, se le presenta al usuario una pantalla de bienvenida, y si es la primera vez que la abre, se le mostrará un tutorial. Aquí se aprovecha para pedirle al usuario su nombre para hacer más amables los mensajes. La pantalla de inicio y el tutorial pueden verse en la figura 4.8.

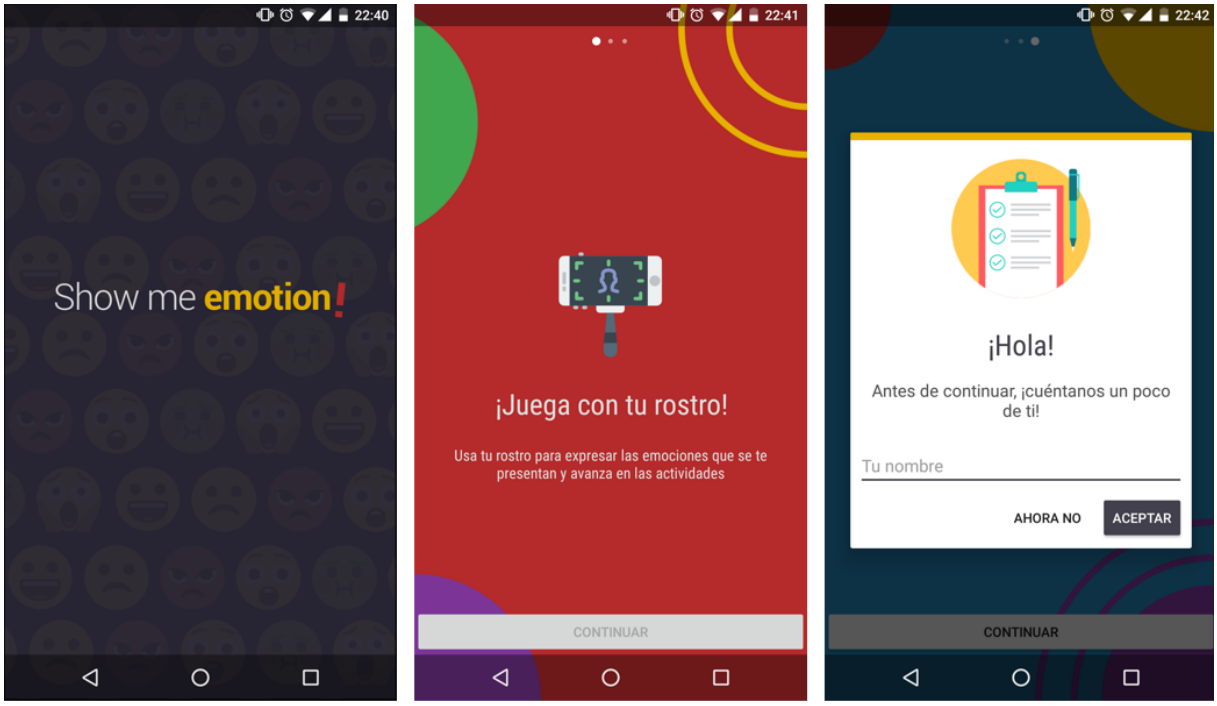

Figura 4.8: Pantalla de inicio y tutorial

Después del tutorial, se le presentará una pantalla con un menú para abrir una de las tres distintas actividades: reconocimiento de emociones, memorama, o reacción a situaciones. También está la opción de revisar los estadísticos. El menú se muestra en la figura 4.9. En esta misma pantalla, el usuario puede acceder a la configuración de

la aplicación para cambiar parámetros como la sensibilidad de la cámara, número de problemas a resolver, y otras opciones.

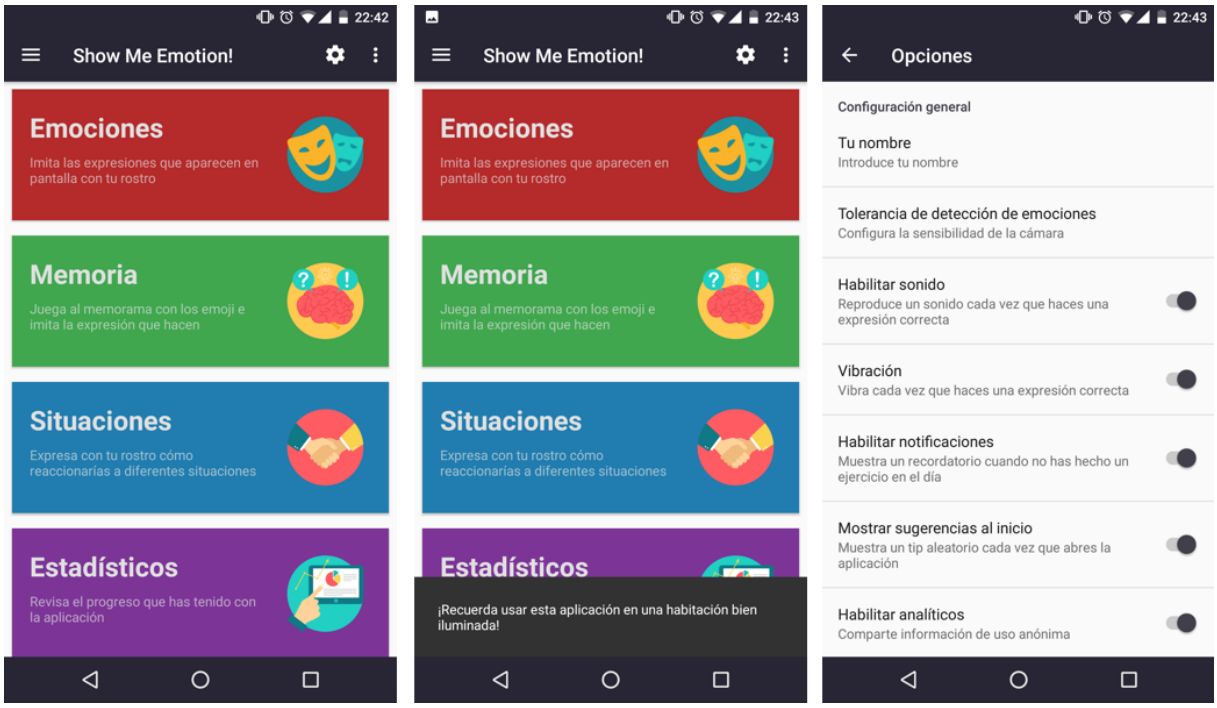

Figura 4.9: Pantalla de menú

En la actividad de reconocimiento de emociones, mostrada en la figura 4.10, consiste en las siguientes secciones:

- Una serie de imágenes que representan expresiones faciales
- Una vista previa de la cámara frontal del dispositivo móvil.
- Un mensaje de estado que evalúa el rostro actual del usuario.
- Un contador de puntuación.

El objetivo en esta actividad es la de imitar con el rostro la expresión facial seleccionada en la lista de la parte superior (la primera de izquierda a derecha). Cuando el mensaje de estado y la expresión facial seleccionada son iguales, la expresión se elimina de la parte superior y se le agrega un punto. La actividad termina una vez que ya no hay más expresiones por imitar.

Al finalizar la actividad se muestra un mensaje felicitando al usuario y muestra el tiempo que tardó en realizar el ejercicio. En este diálogo, el usuario puede elegir entre repetir el ejercicio o volver al menú principal.

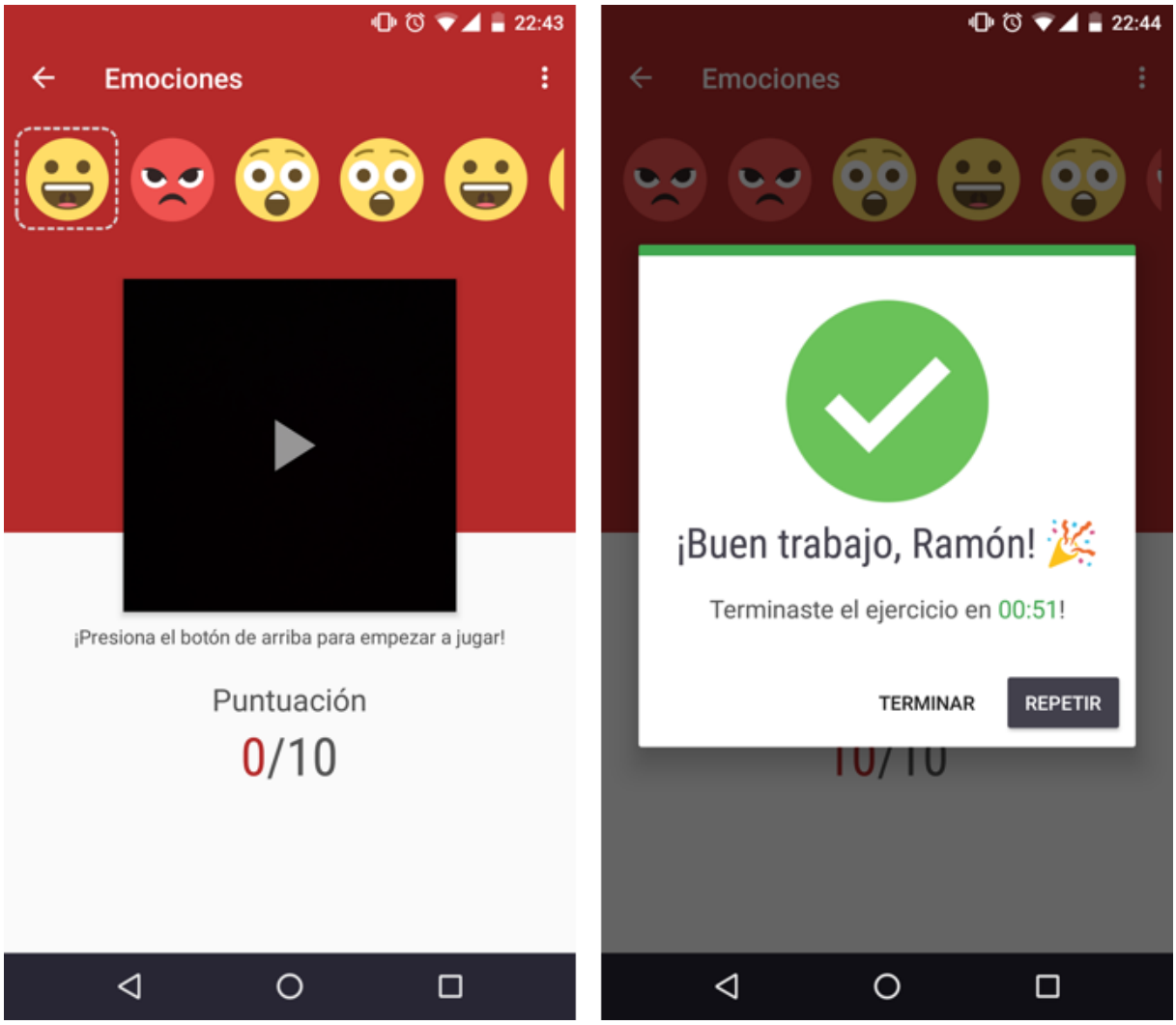

Figura 4.10: Pantalla de reconocimiento de emociones

En la actividad para el memorama, mostrada en la figura 4.11, consiste en un juego de memoria representado por los siguientes componentes:

- Un tablero con una serie de tarjetas.
- Un mensaje de estado.

La meta para el usuario en esta actividad es encontrar y juntar todos los pares de expresiones faciales. Al momento de encontrar un par, y si se trata de una expresión que podemos detectar con la cámara, el usuario tendrá que hacer la expresión facial que juntó para que pueda validarse.

La actividad termina una vez que ya no hay más pares que juntar. Igual que en el

ejercicio anterior, al finalizar se mostrará un mensaje felicitando al usuario, el tiempo que le llevó en terminarlo, y le dará la opción de repetir el ejercicio, o bien, de regresar al menú principal.

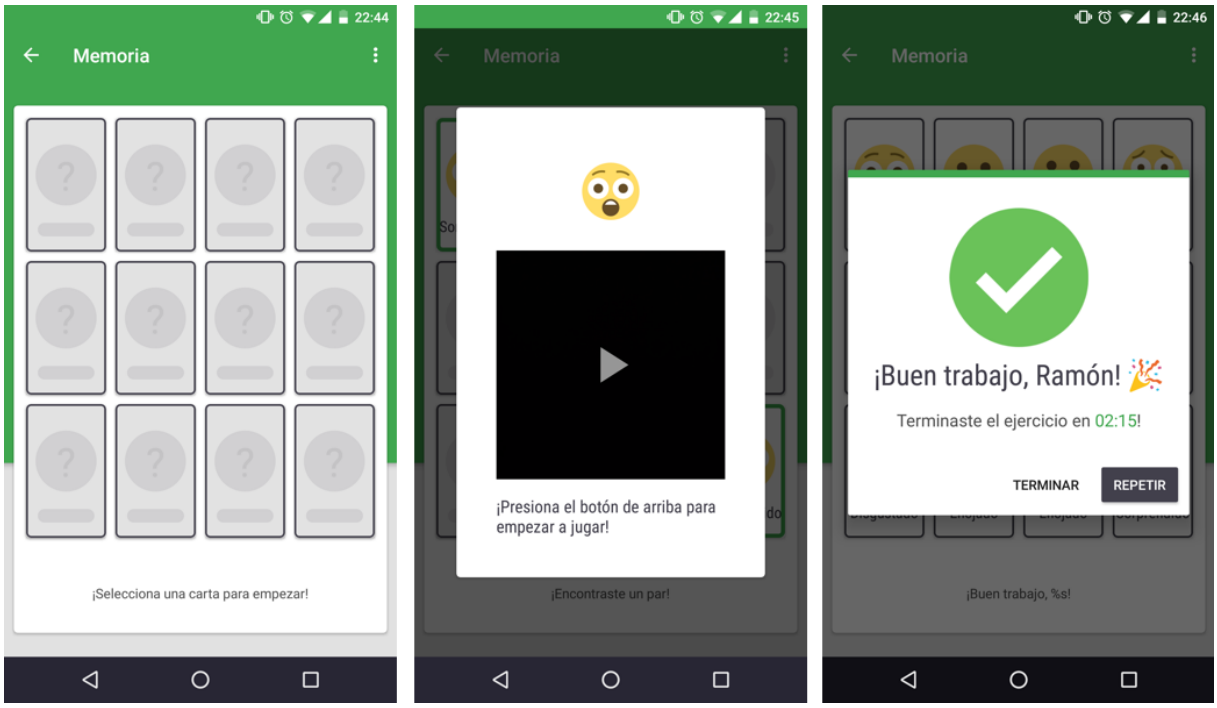

Figura 4.11: Pantalla de memorama

En la actividad de reacción a situaciones, mostrada en la figura 4.12, está conformada por los siguientes componentes:

- Una imagen que representa la situación.
- Una situación social escrita y su pregunta.
- La vista previa de la cámara frontal.
- Una sección que muestra la evaluación del rostro del usuario.

El objetivo de este ejercicio está en adivinar la expresión facial correcta que uno haría dentro de la situación social presentada. Una vez que el usuario hace la expresión facial correcta se le presentará la siguiente.

El ejercicio termina una vez que ya no hay más situaciones qué presentar. Igual que los demás ejercicios, al finalizar se le muestra al usuario un mensaje felicitándolo, el tiempo que le llevó en realizar el ejercicio y la opción de repetirlo o regresar al menú principal.

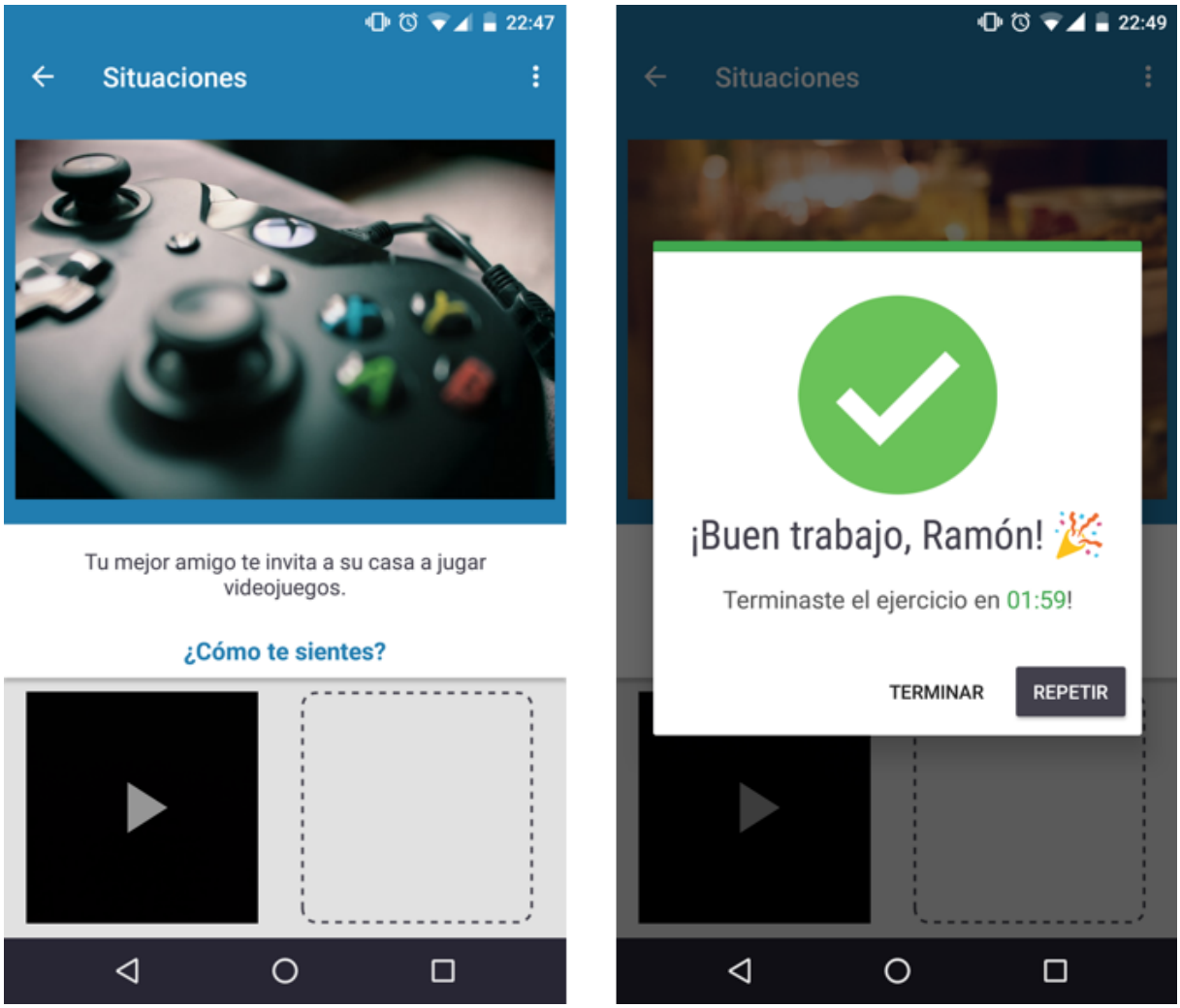

Figura 4.12: Pantalla de situaciones

En la sección de estadísticos, mostrada en la figura 4.13, le presenta al usuario una serie de gráficas que le indican el historial de las actividades que ha hecho. Cada ejercicio tiene dos diferentes gráficas, una que cuenta el tiempo que se ha invertido por día y otra qué tantas veces se ha realizado el ejercicio satisfactoriamente (la frecuencia).

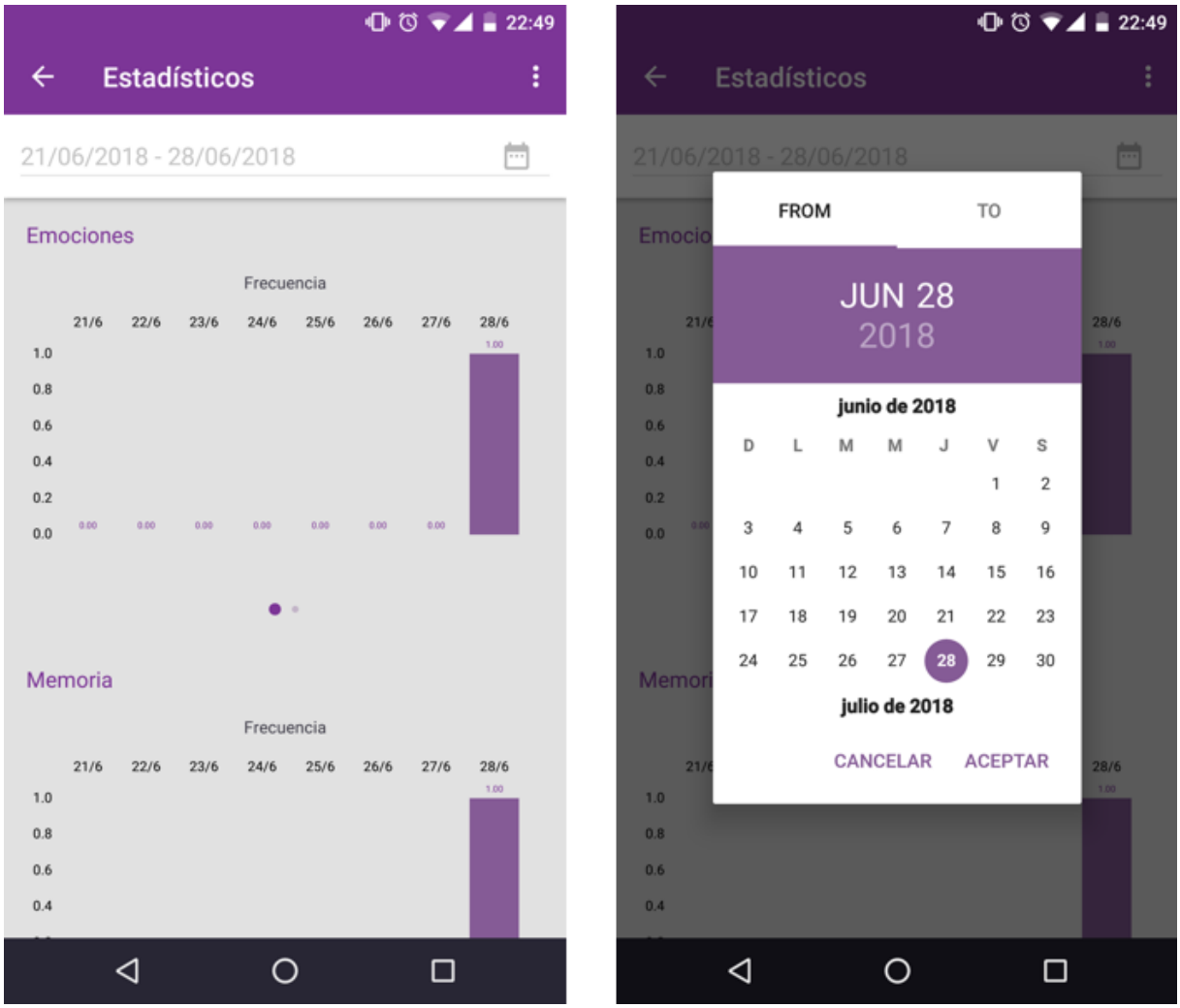

Figura 4.13: Pantalla de estadisticos

Capítulo 5

Análisis de resultados

Este capítulo se enfoca en el análisis de resultados que se obtuvieron al implementar el sistema y ponerlo en práctica con los maestros y psicólogos del colegio EDIA. A lo largo de este capítulo se verá cómo es que se distribuyó la aplicación a los maestros para que la pusieran a prueba con los alumnos, cómo fue que se pudieron detectar algunos problemas una vez que se distribuyó, y por último, se discutirá un modelo para evaluar la aceptación que tuvo la aplicación con los maestros y sus alumnos.

5.1. Distribución de la aplicación

Uno de los problemas más grandes que existen en la distribución de aplicaciones móviles, es que es casi imposible para el desarrollador probar todos los modelos de celulares en los que se puede instalar la aplicación. Estos dispositivos móviles pueden variar en cuanto a tamaño de la pantalla, resolución, densidad de pixeles, disponibilidad de teclas de hardware, número de cámaras, versión del sistema operativo y muchos otros factores. Así que, una vez que se haya distribuido la aplicación y encontramos algún error debido a alguno de los factores mencionados anteriormente, habrá que actualizar la aplicación para arreglarlo. Para actualizar una aplicación, se ocupa tener un sistema para manejar las versiones y un medio por el cual los usuarios puedan descargar la nueva versión que vaya saliendo del programa.

El sistema de versión que se utilizó se llama *Semantic Versioning*, y según

se describe en [33], consiste en nombrar cada versión usando el patrón "*MAJOR.MINOR.PATCH*", donde:

1. **MAJOR**: cambia cuando haces cambios incompatibles a la API.
2. **MINOR**: cambia cuando se le agrega funcionalidad nueva sin afectar a versiones anteriores.
3. **PATCH**: cambia al hacer arreglos al código sin afectar la funcionalidad ni otras versiones anteriores.

El segundo requerimiento para poder distribuir la aplicación, es tener un canal donde los usuarios puedan actualizarla cada vez que exista un cambio. Para cumplir este requerimiento, se usó *Google Play Store*, ya que usando una cuenta de desarrollador esta plataforma nos permite:

- Crear canales de distribución de manera privada, donde los usuarios tienen que darse de alta usando su correo electrónico y proporcionarles un enlace para entrar.
- Manejar distintas versiones de la aplicación.
- Actualizar la aplicación automáticamente, una vez instalada y si los usuarios no han cambiado la configuración por defecto de la *Play Store*, la aplicación se actualiza en el fondo.
- Recibir comentarios y calificaciones de los usuarios.
- Llevar un control de cuántas fallas y en dónde se han ocasionado, si se enlaza la aplicación con herramientas como *crashlytics* de *Firebase*.

En la figura 5.1 puede observarse el pánel de control, llamado *Google Play Console*, donde se muestra la aplicación publicada como una versión de pruebas interna y en la figura 5.2 la página en la *Play Store* donde los usuarios registrados pueden descargar y actualizar la aplicación. Adicionalmente, en 5.3 puede verse una captura de pantalla de un usuario que acaba de entrar a la versión de prueba de la aplicación.

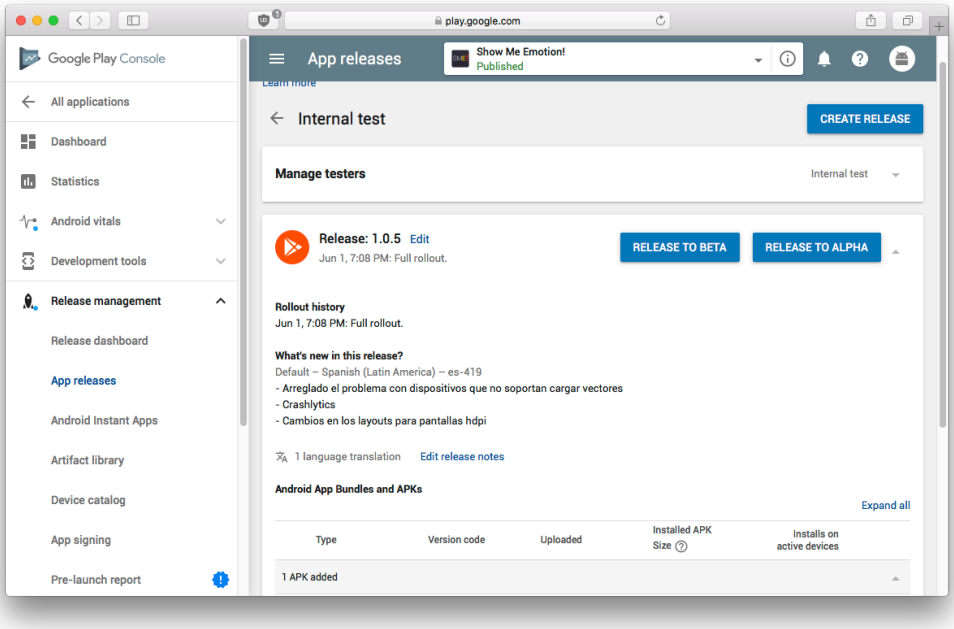

Figura 5.1: Google Play Console - Prueba interna

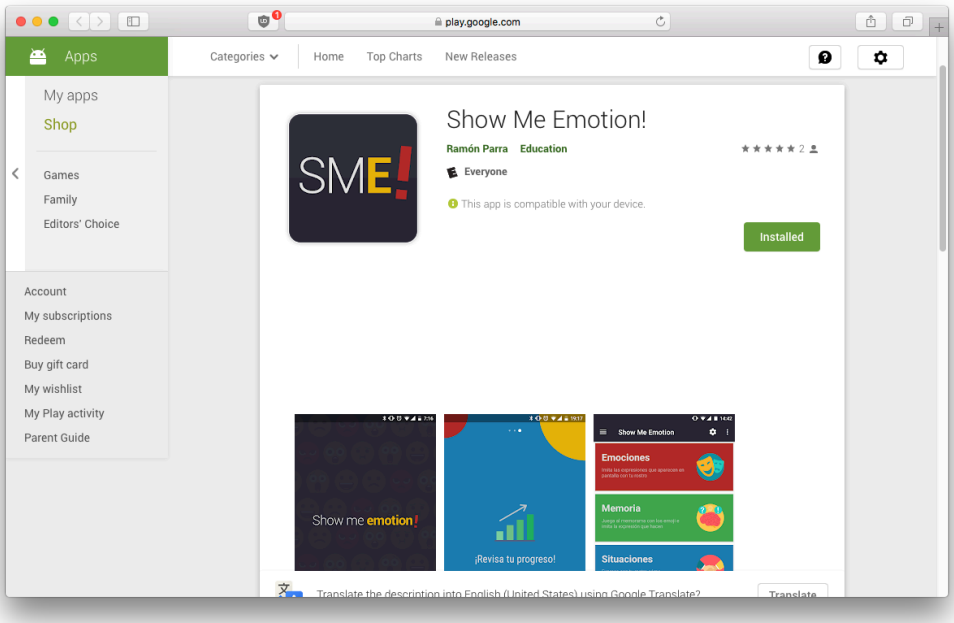

Figura 5.2: Aplicación publicada en Google Play

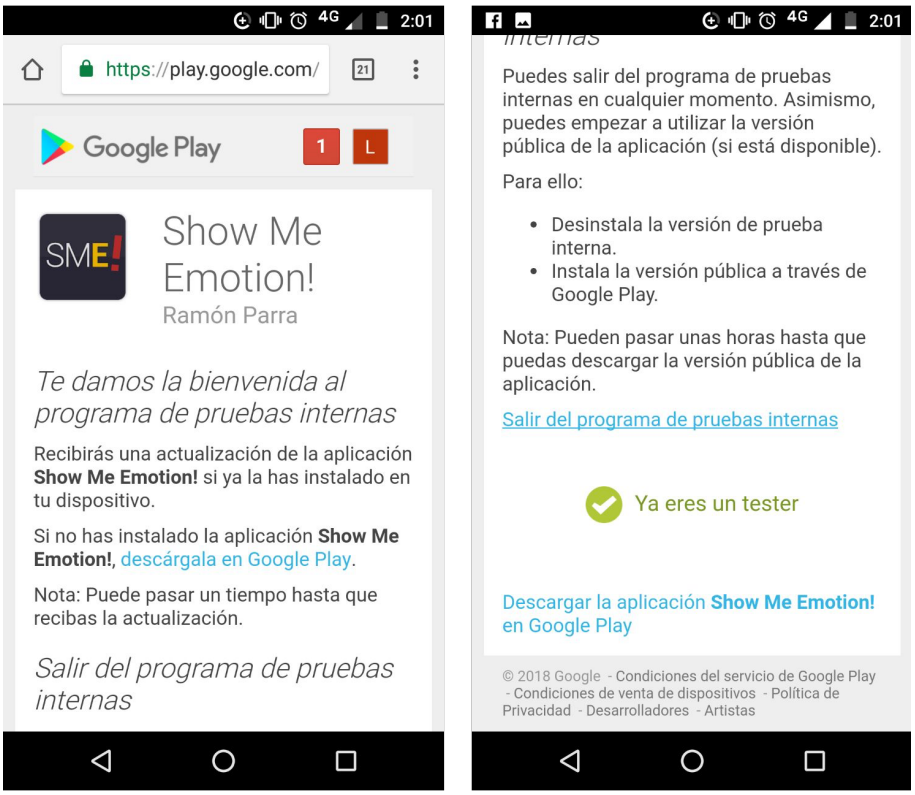

Figura 5.3: Capturas de pantalla de un tester

5.2. Pruebas en el Colegio EDIA

Para la prueba de implementación en el colegio EDIA, se eligió un grupo de estudiantes que presentan trastorno de comunicación social y se encuentran en el rango de 8 a 12 años de edad. El grupo consistió de 42 alumnos, los cuales fueron guiados por un grupo de 5 maestros para que utilicen la aplicación móvil en sus dispositivos móviles, dentro de las actividades de los maestros, se encuentran:

- Guiar a los alumnos en la interfaz de la aplicación.
- Auxiliarlos al hacer expresiones faciales.
- Registrar y avisar de posibles fallos en el uso de la aplicación.

Las pruebas se realizaron durante una semana, desde el 4 al 15 de junio del 2018 . Al finalizar, a los profesores se les envió por correo electrónico un cuestionario basado en el Modelo de Aceptación Tecnológica (*Technology Acceptance Model*, TAM). El TAM consiste en un modelo propuesto que evalúa la utilidad y facilidad de uso de una

nueva herramienta tecnológica a partir de una serie de preguntas que se hacen después de que un grupo de usuarios prueba dicha herramienta [34]. En la tabla 5.1 se presenta el cuestionario para la utilidad percibida y en la tabla 5.2 se presenta el cuestionario para la facilidad de uso percibida. La calificación es un valor numérico del 1 (no estoy de acuerdo) al 7 (muy de acuerdo).

Tabla 5.1 Cuestionario TAM: Utilidad percibida

Pregunta	**Calificación**
1. Usando Show Me Emotion, puedo reconocer con mayor facilidad las emociones básicas.	1-7
2. Usando Show Me Emotion, he mejorado mi forma de expresar las emociones básicas.	1-7
3. Usando Show Me Emotion, es posible mejorar mi interpretación de las emociones básicas.	1-7
4. Usando Show Me Emotion, siento que se disminuirán los conflictos con mis compañeros.	1-7
5. Usando Show Me Emotion, es más fácil interactuar con iguales.	1-7
6. Usando Show Me Emotion es posible mejorar mis habilidades sociales.	1-7

Tabla 5.2 Cuestionario TAM: Facilidad de uso percibida

Pregunta	**Calificación**
1. Aprender a usar Show Me Emotion es fácil.	1-7
2. Me resulta sencillo encontrar la actividad que quiero realizar en el programa.	1-7
3. La aplicación Show Me Emotion me resulta amigable.	1-7
4. Show Me Emotion es una aplicación sencilla para interactuar con ella.	1-7
5. El interactuar con una aplicación como Show Me Emotion me vuelve más hábil.	1-7
6. Encuentro que la aplicación es fácil de usar.	1-7

5.3. Resultados de la encuesta

Una vez finalizado el tiempo de pruebas, se le envió el cuestionario a cada uno de los profesores que participaron en la versión de pruebas para que pudieran contestarlo a partir de su experiencia. En esta sección se explicará y discutirá cada una de las preguntas.

5.3.1. Utilidad percibida

Pregunta 1. Usando Show Me Emotion, puedo reconocer con mayor facilidad las emociones básicas.

Los resultados se muestran en la figura 5.4 reflejan qué tan útil resultó la clasificación de expresiones faciales. A pesar de haber obtenido un porcentaje de acierto relativamente bajo a comparación del estado del arte, como se vio en la sección 4.2.2, la red neuronal pudo clasificar las emociones básicas del rostro de los usuarios. Lo anterior puede probarse con los resultados de la pregunta: donde la mayoría contestaron arriba de 6 puntos.

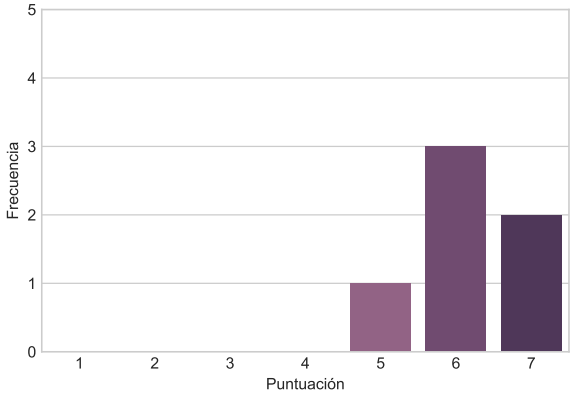

Figura 5.4: TAM: Utilidad percibida, resultados pregunta 1

Pregunta 2. Usando Show Me Emotion, he mejorado mi forma de expresar las emociones básicas.

Sin embargo, si observamos la figura 5.5, podemos observar que la mayoría dio

5 ó más puntos a la mejora de expresar las emociones básicas. Esto puede deberse al reducido tiempo en el que se puso a prueba.

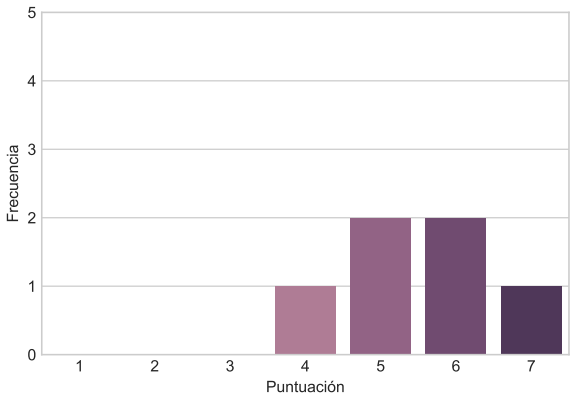

Figura 5.5: TAM: Utilidad percibida, resultados pregunta 2

Pregunta 3. Usando Show Me Emotion, es posible mejorar mi interpretación de las emociones básicas.

En la figura 5.6 puede verse de igual forma la falta de tiempo para poder decidir si el uso de la aplicación puede mejorar la interpretación de las emociones básicas. Sin embargo, también podemos observar que hubo maestros que consideran que la aplicación sí es útil.

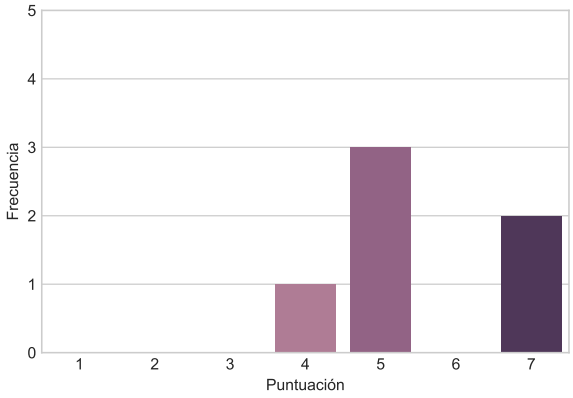

Figura 5.6: TAM: Utilidad percibida, resultados pregunta 3

Pregunta 4. Usando Show Me Emotion, siento que se disminuirán los conflictos con

mis compañeros.

Uno de los puntos más débiles del prototipo fue la falta de interacción con otros compañeros. Puede verse en la figura 5.7 que la mayoría de los maestros solamente evaluaron esta pregunta con cuatro puntos. Pudo deberse a la falta de un ejercicio en la aplicación prototipo que forzara a los niños a interactuar directamente con otras personas en lugar de realizar el ejercicio por sí solos.

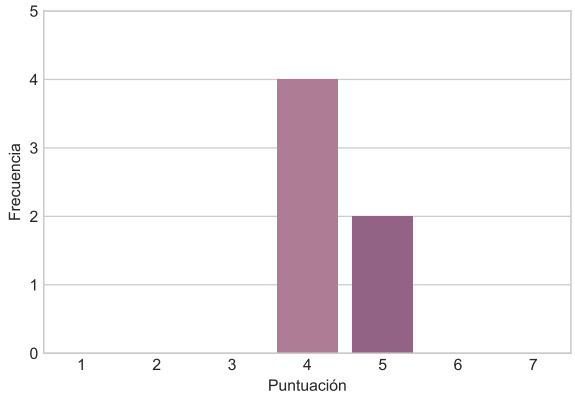

Figura 5.7: TAM: Utilidad percibida, resultados pregunta 4

Pregunta 5. Usando Show Me Emotion, es más fácil interactuar con iguales.

La figura 5.8 está relacionada a la anterior, sin embargo, aquí los resultados están más distribuidos. A partir de los datos podemos considerar lo mismo: a pesar que los ejercicios implementados son suficientes para una persona, ocupamos hacer nuevos ejercicios que hagan interactuar directamente a los niños con sus semejantes.

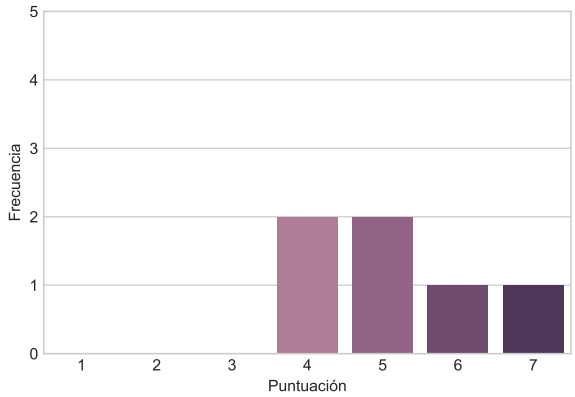

Figura 5.8: TAM: Utilidad percibida, resultados pregunta 5

Pregunta 6. Usando Show Me Emotion es posible mejorar mis habilidades sociales.

Para la pregunta final de la utilidad percibida, podemos observar la figura 5.9. Donde la mayoría de los maestros opinan que, aunque sea posible mejorar las habilidades sociales con la aplicación, aún faltan cosas por mejorar.

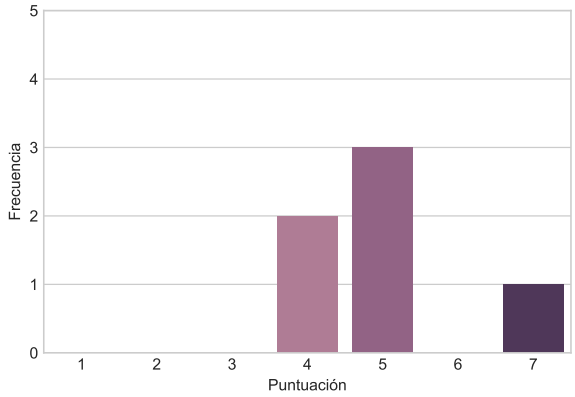

Figura 5.9: TAM: Utilidad percibida, resultados pregunta 6

5.3.2. Facilidad de uso percibida

Pregunta 1. Aprender a usar Show Me Emotion es fácil.

En general, como se puede observar en la figura 5.10, los maestros consideraron que la aplicación prototipo fue muy sencilla de usarse.

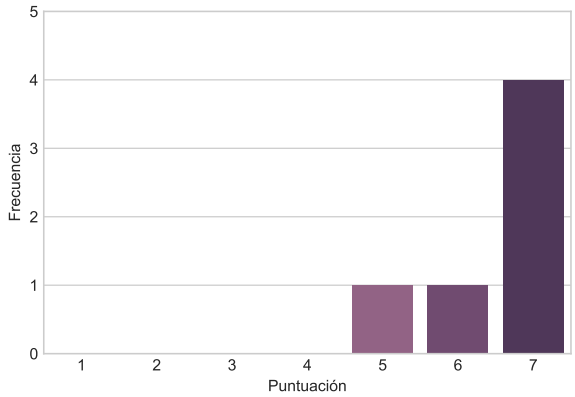

Figura 5.10: TAM: Facilidad de uso percibida, resultados pregunta 1

Pregunta 2. Me resulta sencillo encontrar la actividad que quiero realizar en el programa.

De igual manera, no existió problemas mayores con la navegación de actividades y ejercicios del programa, tal y como lo vemos en la figura 5.11.

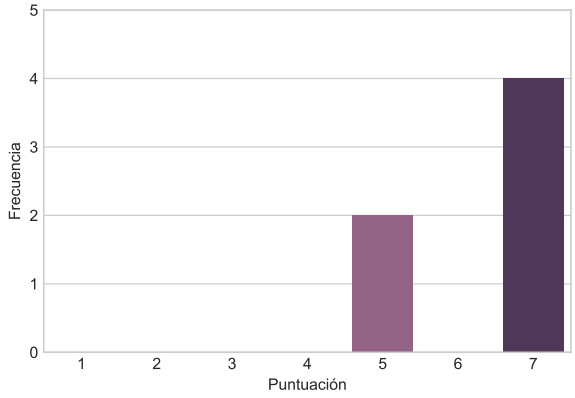

Figura 5.11: TAM: Facilidad de uso percibida, resultados pregunta 2

Pregunta 3. La aplicación Show Me Emotion me resulta amigable.

Los colores, iconografía y diseño de la aplicación tampoco causaron problemas para los maestros. Como se indica en la figura 5.12, la mayoría consideró que la aplicación es amigable al usuario.

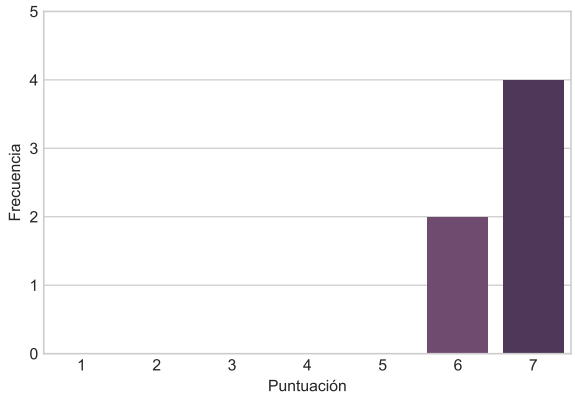

Figura 5.12: TAM: Facilidad de uso percibida, resultados pregunta 3

Pregunta 4. Show Me Emotion es una aplicación sencilla para interactuar con ella.

Salvo una incidencia, mostrada en la figura 5.13, los maestros consideran que la aplicación es sencilla para interactuar con ella. Esta incidencia se pudo deber a un problema que se tuvo al instalar la aplicación en el dispositivo de uno de los maestros.

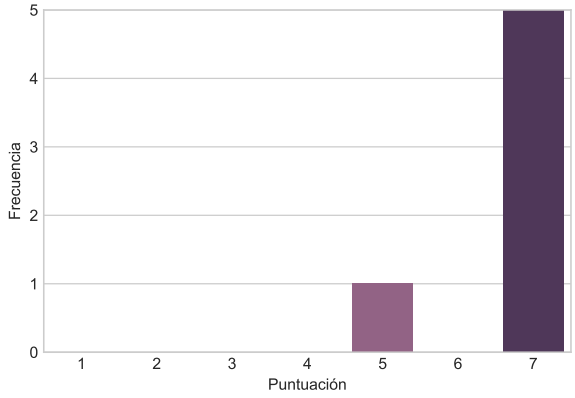

Figura 5.13: TAM: Facilidad de uso percibida, resultados pregunta 4

Pregunta 5. El interactuar con una aplicación como Show Me Emotion me vuelve más hábil.

Como se ve en la figura 5.14, la opinión fue más distribuida. Los maestros consideran que al usar la aplicación los vuelve más hábiles en el mismo manejo de ella.

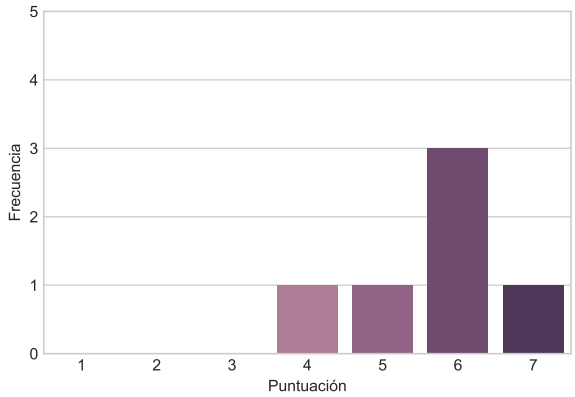

Figura 5.14: TAM: Facilidad de uso percibida, resultados pregunta 5

Pregunta 6. Encuentro que la aplicación es fácil de usar.

Para concluir, la figura 5.15 nos hace creer que todos los maestros consideraron que la aplicación es fácil de usar.

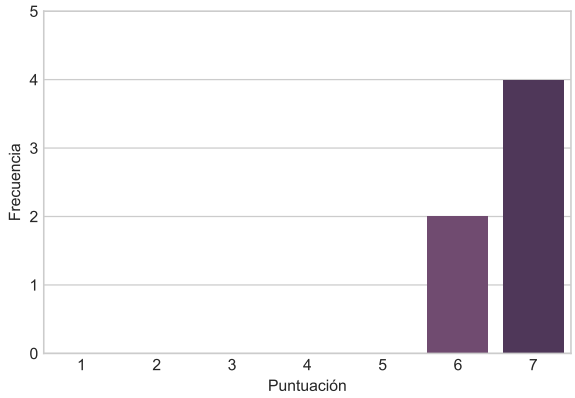

Figura 5.15: TAM: Facilidad de uso percibida, resultados pregunta 6

5.3.3. Comentarios

Adicionalmente al cuestionario, se pidió a los profesores que incluyeran algún comentario adicional que tuvieran. Algunos de los comentarios más relevantes son los siguientes:

- Fue un tiempo de prueba muy breve, por lo que no se pudo observar si el reconocimiento de emociones se vió reflejado en la relación que los alumnos tienen con sus compañeros.
- Igualmente, los maestros opinaron que faltó tiempo para probar si existía una relación directa con el esfuerzo para manifestar la emoción usando la cámara con lo que los niños experimentan día a día.
- Sin embargo, a muchos de ellos les pareció que la aplicación es muy sencilla de utilizar.
- Además, consideran que es un prototipo muy atractivo y los niños mostraron interés por realizar las actividades.

5.3.4. Puntuación total

Para concluir con el capítulo, se mostrará en las tablas 5.3 y 5.4 los promedios que se tuvieron en el cuestionario. Estos promedios reflejan que a pesar de haber tenido algunas debilidades, tales como en el punto cuatro del cuestionario de facilidad de uso percibida, la aceptación de la aplicación fue favorable en el periodo que se puso a prueba dentro del colegio EDIA.

Tabla 5.3 Cuestionario TAM: Puntuación total de utilidad percibida

Pregunta	**Calificación promedio**
1. Usando Show Me Emotion, puedo reconocer con mayor facilidad las emociones básicas.	6.16 / 7.00
2. Usando Show Me Emotion, he mejorado mi forma de expresar las emociones básicas.	5.50 / 7.00
3. Usando Show Me Emotion, es posible mejorar mi interpretación de las emociones básicas.	5.50 / 7.00
4. Usando Show Me Emotion, siento que se disminuirán los conflictos con mis compañeros.	4.43 / 7.00
5. Usando Show Me Emotion, es más fácil interactuar con iguales.	5.16 / 7.00
6. Usando Show Me Emotion es posible mejorar mis habilidades sociales.	5.00 / 7.00

Tabla 5.4 Cuestionario TAM: Puntuación total de facilidad de uso percibida

Pregunta	**Calificación promedio**
1. Aprender a usar Show Me Emotion es fácil.	6.50 / 7.00
2. Me resulta sencillo encontrar la actividad que quiero realizar en el programa.	6.33 / 7.00
3. La aplicación Show Me Emotion me resulta amigable.	6.66 / 7.00
4. Show Me Emotion es una aplicación sencilla para interactuar con ella.	6.66 / 7.00
5. El interactuar con una aplicación como Show Me Emotion me vuelve más hábil.	5.66 / 7.00
6. Encuentro que la aplicación es fácil de usar.	6.66 / 7.00

Capítulo 6

Conclusiones

A partir del trabajo desarrollado, esto es, la investigación como trabajo de tesis y el prototipo en forma de aplicación móvil puesto a prueba en el colegio EDIA, se pretende ayudar a niños con trastorno de comunicación social a desarrollar su habilidad de reconocer e imitar las emociones que se presentan a través de las expresiones faciales. En este capítulo se presentan las conclusiones a las que se llegaron y trabajos a futuro a considerar.

6.1. Conclusiones

En el segundo capítulo se vieron dos casos de éxito que tuvieron aplicaciones dedicadas a desarrollar la sensibilidad emocional en niños con distintos trastornos mentales, y el éxito que tuvieron implementando cada uno de ellos. Después, propusimos un prototipo similar usando tecnologías nuevas – tales como aplicaciones hechas para dispositivos móviles y visión por computadora usando redes neuronales convolucionales – y se probó que, efectivamente tuvo cierto éxito en el periodo de prueba que se realizó en el colegio EDIA con niños diagnosticados con el trastorno de comunicación social que se encuentran en el rango de 8 a 12 años de edad.

6.2. Trabajos a futuro

A pesar de que el prototipo tuvo éxito en las pruebas realizadas en el colegio EDIA, consideramos que podemos tener una mejor aplicación en trabajos futuros si se consideran los puntos que se expondrán a continuación.

El primer punto es considerar otras plataformas además de Android. La plataforma no debería de ser exclusiva para estas plataformas y debería de no sólo considerarse iOS, sino otras plataformas que no se clasifican como móviles, tales como la web. De esta forma, la aplicación sería mucho más accesible para otras personas e instituciones.

Siguiendo con el segundo punto, la aplicación debería de considerarse hacer para múltiples usuarios con diferentes roles. En su estado actual, sólo se pensó para que los alumnos la usaran de forma local, sin embargo, creemos que también debería de considerarse otros tipos de usuarios como administradores y maestros para que puedan observar las estadísticas de uso de la aplicación de distintos alumnos. Si se cumple este punto, y se anexa a una plataforma web, la sección de estadísticos sería mucho más robusta y útil para los maestros y psicólogos de la institución de interés.

Por último, como tercer punto, debería de replantearse usar otro tipo de conjunto de datos para el reconocimiento de expresiones faciales, para que permita publicar la aplicación de una forma más libre y que sea de mayor interés para empresas privadas.

Referencias

[1] Brain Centers, “Trastornos del Neurodesarrollo | CRECER EN SALUD.” [Online]. Available: http://crecerensalud.com/trastornos-del-neurodesarrollo. [Accessed: 21-Mar-2018]

[2] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders*. 2013, p. 991.

[3] Understood.org, “Entender el trastorno de la comunicación social.” [Online]. Available: https://www.understood.org/es-mx/learning-attention-issues/child-learning-disabilities/communication-disorders/understanding-social-communication-disorder. [Accessed: 21-Mar-2018]

[4] Asperger México A.C., “Preguntas Frecuentes.” [Online]. Available: https://www.asperger.org.mx/recursos/preguntas-frecuentes/

[5] T. S. Huang, “Computer Vision: Evolution and Promise,” *University of Illinois at Urbana-Champaign*, 1997.

[6] L. Seven Minds, “The Social Navigator App: a social skills app for kids with autism, ADD/ADHD, Asperger’s and special needs.” 2012 [Online]. Available: http://www.socialnavigatorapp.com/aspergers_apps.htm. [Accessed: 23-Mar-2018]

[7] CRFDIES, “CRFDIES - Quiénes somos.” [Online]. Available: http://www.crfdies.edu.mx/sitiov2/index.php?r=sitio/quienes_somos

[8] EDIA, “Colegio EDIA :: Colegio.” [Online]. Available: https://www.edia.edu.mx/colegio/. [Accessed: 12-May-2017]

[9] T. Attwood, *The Complete Guide to Asperger’s Syndrome*. 2008, p. 401.

[10] L. Wing, “Asperger’s syndrome: A clinical account,” *Psychological Medicine*, vol. 11, no. 1, pp. 115–129, Feb. 1981 [Online]. Available: http://www.journals.cambridge.org/abstract_S0033291700053332

[11] R. Echeverría, *Ontología del lenguaje*. JC Sáez editor, 2003, p. 245.

[12] R. W. Picard, “Affective Computing,” *MIT press*, no. 321, pp. 1–16, 1995 [Online]. Available: papers3://publication/uuid/9C02FCAE-FE2E-4D2C-9707-766804777DC9

[13] L. E. Sucar and G. Gomez, *Vision Computacional*. 2003, p. 185.

[14] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," *MIPRO, 2012 Proceedings of the 35th International Convention*, pp. 1725–1730, 2012 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6240859

[15] OpenCV, "OpenCV 3.3.0 Documentation." [Online]. Available: https://docs.opencv.org/3.3.0/. [Accessed: 30-Apr-2018]

[16] OpenCV, "Mat - the basic image container." 2017 [Online]. Available: https://docs.opencv.org/3.3.0/d6/d6d/tutorial_mat_the_basic_image_container.html. [Accessed: 12-Jun-2018]

[17] D. M. Brown, Christopher; H. Ballard, *Computer vision*, vol. 3. 2008, p. 8 pp. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19943627

[18] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, vol. 25. 2002, p. 693 [Online]. Available: http://www.worldscinet.com/ijprai/25/2503/S0218001411008853.html

[19] Y. Tian, T. Kanade, and J. F. Cohn, *Handbook of Face Recognition*. 2011, pp. 487–519.

[20] M. F. Henrik Brink, Joseph Richards, *Real World Machine Learning*. Manning Publications Co., 2017, p. 266.

[21] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, no. May 2014, pp. 253–256, 2010.

[22] S. Theodoridis, "Neural Networks and Deep Learning," in *Machine learning*, Determination Press, 2015, pp. 875–936 [Online]. Available: http://neuralnetworksanddeeplearning.com/

[23] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *CoRR*, vol. abs/1404.7828, 2014 [Online]. Available: http://arxiv.org/abs/1404.7828

[24] F. Vázquez, "Deep learning made easy with deep cognition." 2017 [Online]. Available: https://www.kdnuggets.com/2017/12/deep-learning-made-easy-deep-cognition.html. [Accessed: 12-Jun-2018]

[25] Aphex34, "Typical cnn [cc by-sa 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], from wikimedia commons." 2015 [Online]. Available: https://commons.wikimedia.org/wiki/File:Typical_cnn.png. [Accessed: 12-Jun-2018]

[26] M. Silver and P. Oakes, "Evaluation of a new computer intervention to teach people with autism or Asperger syndrome to recognize and predict emotions in others," *Autism*, vol. 5, no. 3, pp. 299–316, 2001 [Online]. Available: papers2://publication/uuid/BDD2A488-B964-4E1C-903C-02E668BA5D63

[27] B. S. Myles, "Using Assistive Technology to Teach Emotion Recognition to Students With Asperger Syndrome," *SPINAL CORD INJURIES: Management and Rehabilitation*, vol. 28, no. 3, pp. 174–181, 2007 [Online]. Available: http://dx.doi.org/10.1016/B978-0-323-00699-6.10013-9

[28] P. Kruntchen, “Architectural blueprints–the” 4+ 1” view model of software architecture,” *IEEE Software*, vol. 12, no. November, pp. 42–50, 1995.

[29] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.

[30] Docker Inc., “What is a Container | Docker.” 2017 [Online]. Available: https://www.docker.com/what-container. [Accessed: 12-Jun-2018]

[31] I. J. Goodfellow *et al.*, “Challenges in representation learning: A report on three machine learning contests,” *Neural Networks*, vol. 64, pp. 59–63, 2015.

[32] TensorFlow, “How to Retrain Inception’s Final Layer for New Categories.” 2017 [Online]. Available: https://www.tensorflow.org/tutorials/image_retraining

[33] Tom Preston Werner, “Semantic Versioning 2.0.0.” 2015 [Online]. Available: https://semver.org/. [Accessed: 04-Jun-2018]

[34] F. Davis, “Perceived ease of use, and user acceptance of information technology,” *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.