



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERIA**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE  
SISTEMAS**

“Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima”

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE  
INGENIERO DE SISTEMAS**

**AUTOR:**

Lopez Roca, Kevin Alex

**ASESOR:**

Mg. Rivera Crisostomo, Rene

**LÍNEA DE INVESTIGACIÓN:**

Sistemas de Información y Comunicaciones

**LIMA – PERÚ**

**2018**

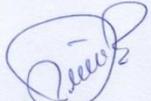
El Jurado encargado de evaluar la tesis presentada por don  
 (a).....Kevin Alex Lopez Roca  
 cuyo título es: "Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima"

Reunido en la fecha, escuchó la sustentación y la resolución de preguntas por el estudiante, otorgándole el calificativo de: 13.....(número)  
trece.....(letras).

Trujillo (o Filial).....01 de diciembre del 2018

  
 .....  
 PRESIDENTE

  
 .....  
 SECRETARIO

  
 .....  
 VOCAL

			
Revisó	Vicerrectorado de Investigación/ <b>DEVAC</b>	Responsable del <b>SEC</b>	Aprobó <b>PERUVALLEJO</b>

NOTA: Cualquier documento impreso diferente del original, y cualquier archivo electrónico que se encuentren fuera del Campus Virtual Trilce serán considerados como COPIA NO CONTROLADA.

## **Dedicatoria**

Al Creador por brindarme la luz para alcanzar mis metas.

A mi madre y a mis tíos, por su apoyo y guía en este camino.

## **Agradecimiento**

Al Creador por guiarme en este camino y fortalecerme día a día para enfrentar las adversidades del hoy y del mañana.

A la Asociación de Sordos Región Lima, por su comprensión, consejos y apoyo en el presente trabajo de investigación.

## DECLARACIÓN DE AUTENTICIDAD

Yo Kevin Alex Lopez Roca con DNI N.º 48270133, a efecto de cumplir con las disposiciones vigentes consideradas en el Reglamento de Grados y Títulos de la Universidad César Vallejo, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, declaro bajo juramento que toda la documentación que acompaño es veraz y auténtica. Asimismo, declaro también bajo juramento que todos los datos e información que se presenta en la presente tesis son auténticos y veraces.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada por lo cual me someto a lo dispuesto en las normas académicas de la Universidad César Vallejo.

Lima, 1 de diciembre de 2018



---

**Kevin Alex Lopez Roca**  
**DNI: 48270133**

## **Presentación**

Señores miembros del Jurado:

En cumplimiento del Reglamento de Grados y Títulos de la Universidad César Vallejo presento ante ustedes la tesis titulada “Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima”, con el objetivo de “Desarrollar la aplicación móvil de interpretación del lenguaje de señas para los discapacitados auditivos de la asociación de sordos región Lima”. En el primer capítulo se explica la realidad problemática y las tecnologías empleadas para la implementación del proyecto tales como el uso de las librerías OpenCV en Android Studio permitió la creación de una aplicación móvil capaz de interpretar los gestos humanos que representan el lenguaje de señas como también posibilidad de comunicarnos con la comunidad sorda mediante la representación del alfabeto en señas; en el segundo capítulo se muestra la metodología de investigación aplicada durante la investigación, en el tercer capítulo se detalla el análisis de los datos, con las pruebas aplicadas para obtener los resultados de la implementación de la solución propuesta. Finalmente se presenta la discusión de los resultados obtenidos, las conclusiones del estudio y las recomendaciones para futuras investigaciones.



**Kevin Alex Lopez Roca**

## Resumen

Este trabajo de investigación tiene como objetivo el desarrollo de una “Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima”, y determinar el efecto de la aplicación del mismo para dar una explicación sobre las deficiencias que existen en el tiempo de comunicación con las personas sordas y el costo de contratación de intérpretes que conlleva el proceso de comunicación.

Para realizar el presente trabajo de investigación se tomó como población de las misma, las 8 personas sordas que son socios activos. Se usó como técnica de recopilación de datos la observación que hizo uso como instrumento una ficha de registros.

De los resultados obtenidos con esta investigación se llegó a la conclusión que el aplicativo móvil permite detectar mediante reconocimiento óptico de patrones el alfabeto dactilológico del lenguaje de señas y que el sordo pueda formar oraciones deletreando las palabras para que el aplicativo móvil pueda interpretarlo. De esa forma lograr entender y lograr comunicarse con las personas sordas. Su eficacia para interpretar oraciones ha sido demostrada con experimentación en situ en un ambiente controlado. En cuanto al rendimiento del aplicativo móvil se concluye que solo la cámara del dispositivo permite reconocer símbolos a partir de imágenes previamente aprendidas en un ambiente controlado, por lo que puede ser la base para futuras investigaciones. También se ha determinado una disminución en promedio del tiempo de comunicación con una persona sorda en una reducción del 12% (4 a 6 minutos) con respecto de 8 a 10 minutos que se demora entender a una persona sorda. Además, se redujo el costo promedio de contratación de intérpretes para el proceso de comunicación de la persona sorda que le costaba contratar semanalmente entre 40 a 50 soles, con el aplicativo se logra reducir a un rango representativo de 0 a 10 soles.

Palabras Clave: aplicativo móvil, reconocimiento óptico de patrones, alfabeto dactilológico.

## **Abstract**

This research work aims to develop a "Mobile application of interpretation of the Peruvian sign language for hearing impaired in the Association of the Deaf in the Lima Region", and determine the effect of the application of the same to give an explanation on the deficiencies that exist in the time of communication with deaf people and the cost of hiring interpreters involved in the communication process.

In order to carry out the present research work, the 8 deaf people who are active partners were taken as their population. The observation that used a record card as an instrument was used as a data collection technique.

From the results obtained with this research it was concluded that the mobile application allows detecting by means of optical pattern recognition the sign language alphabet and that the deaf man can form sentences by spelling the words so that the mobile application can interpret it. In this way, we can understand and achieve communication with deaf people. Its effectiveness in interpreting sentences has been demonstrated through on-site experimentation in a controlled environment. Regarding the performance of the mobile application, it is concluded that only the camera of the device allows to recognize symbols from images previously learned in a controlled environment, so it can be the basis for future research. An average decrease in communication time with a deaf person has also been determined in a reduction of 12% (4 to 6 minutes) with respect to 8 to 10 minutes that a deaf person is delayed in understanding. In addition, the average cost of hiring interpreters was reduced for the deaf person's communication process, which cost him to contract weekly between 40 to 50 soles, with the application being reduced to a representative range of 0 to 10 soles.

**Keywords:** mobile application, optical pattern recognition, fingerprint alphabet.

# INDICE GENERAL

<b>Dedicatoria</b>	III
<b>Agradecimiento</b>	IV
<b>Resumen</b>	VII
<b>Abstract</b>	VIII
<b>INDICE DE TABLAS</b>	XI
<b>INDICE DE FIGURAS</b>	XII
<b>I. INTRODUCCIÓN</b>	13
<b>1.1 Realidad Problemática</b>	14
<b>1.2 Trabajos previos</b>	17
<b>1.2.1 Antecedentes Internacionales</b>	17
<b>1.2.2 Antecedentes Nacionales</b>	19
<b>1.3 Teorías Relacionadas al Tema</b>	20
<b>1.3.1 Variable Independiente: Aplicación Móvil</b>	20
<b>1.3.2 Variable Dependiente: Interpretación de Lenguaje de Señas</b>	30
<b>1.4 Formulación del Problema</b>	40
<b>1.4.1 Problemas General</b>	40
<b>1.4.2 Problemas Específicos</b>	40
<b>1.5 Justificación del estudio</b>	41
<b>1.6 Hipótesis</b>	42
<b>1.6.1 Hipótesis General</b>	42
<b>1.6.2 Hipótesis Especificas</b>	42
<b>1.7 Objetivos</b>	42
<b>1.7.1 Objetivo General</b>	42
<b>1.7.2 Objetivo Especifico</b>	43
<b>II. METODO</b>	44
<b>2.1 Diseño de la investigación</b>	45
<b>2.2 Variables, operacionalización</b>	46
<b>2.3 Población y Muestra</b>	50
<b>2.3.1 Población</b>	50
<b>III. RESULTADOS</b>	53
<b>3.1. Análisis Descriptivo</b>	54
<b>IV. DISCUSIÓN</b>	70
<b>V. CONCLUSIÓN</b>	72

<b>VI. RECOMENDACIONES</b>	74
<b>VII. REFERENCIAS</b>	76
<b>ANEXOS</b>	80
Anexo 1: Matriz De Consistencia	
Anexo 2: Cronograma	
Anexo 3: Instrumento De Recolección De Datos	
Anexo 4: Instrumento De Medición De La Variable Dependiente	
Anexo 5: Aspectos Administrativos	
Anexo 6: Aspectos Tecnicos	
Anexo 7: Curso De Lenguaje De Señas Peruanas	
Anexo 8: Analisis Y Diseño	
Anexo 9: Codigo Fuente	
Anexo 10: Acta De Aprobación De Originalidad De Los Trabajos Académicos De La UCV	
Anexo 11: Pantallazo Turnitin	
Anexo 12: Autorización De Publicación De Tesis	
Anexo 13: Autorización De La Versión Final Del Trabajo De Investigación	

## INDICE DE TABLAS

Tabla 1- Diferencias de Traducción e Interpretación.....	37
Tabla 2- Operacionalización de la variable dependiente .....	49
Tabla 3- Matriz de operacionalización de las variables.....	51
Tabla 4- Total de la Población Inscrita.....	52
Tabla 5- Técnicas e Instrumentos de Recolección de Datos .....	53
Tabla 6- Indicadores – prueba de hipótesis .....	54
Tabla 7- Matriz de Confusión para gesto letra “E” del clasificador de configuración simple.....	56
Tabla 8- Resultados de las pruebas de rendimiento del gesto letra “E” .....	56
Tabla 9- Matriz de Confusión para gesto letra “F” del clasificador de configuración simple .....	57
Tabla 10- Resultados de las pruebas de rendimiento del gesto letra “F” .....	57
Tabla 11- Matriz de Confusión para gesto letra “G” del clasificador de configuración simple .....	58
Tabla 12- Resultados de las pruebas de rendimiento del gesto letra “G”.....	58
Tabla 13- Matriz de Confusión para gesto letra “H” del clasificador de configuración simple .....	59
Tabla 14- Resultados de las pruebas de rendimiento del gesto letra “H”.....	59
Tabla 15- Matriz de Confusión para gesto letra “I” del clasificador de configuración simple .....	60
Tabla 16- Resultados de las pruebas de rendimiento del gesto letra “I” .....	60
Tabla 17- Matriz de Confusión para gesto letra “K” del clasificador de configuración simple .....	61
Tabla 18- Resultados de las pruebas de rendimiento del gesto letra “K”.....	61
Tabla 19- Estudio N°1 Pruebas en Ambiente Controlado Resultados de las pruebas de rendimiento .....	62
Tabla 20 - Tiempo en entender a una persona sorda.....	63
Tabla 21 - Frecuencia Pre y Post tiempo promedio .....	64
Tabla 22 - Costo de contratación de intérpretes de señas .....	67
Tabla 23 - Frecuencia Pre y Post costo promedio.....	67
Tabla 24 - Cuadro Comparativo, después de aplicar la solución .....	71
Tabla 25 - Matriz de consistencia.....	81
Tabla 26 - Matriz de Confusión para gesto letra “A” del clasificador de configuración simple .....	84
Tabla 27 - Resultados de las pruebas de rendimiento del gesto letra “A” (clasificador de configuración simple).....	85
Tabla 28 - Recursos y presupuesto .....	86
Tabla 29 - Caso de Uso Detectar Mano.....	92
Tabla 30 - Caso de Uso Guardar Gesto .....	93
Tabla 31 - Caso de Uso Entrenar Gesto.....	94
Tabla 32 - Caso de Uso Reconocer Gesto .....	95

## INDICE DE FIGURAS

Figura 1- Abecedario usado por la dactilología.....	33
Figura 2- Cascada de Clasificadores .....	38
Figura 3- Árbol de problemas .....	43
Figura 4- Datos de Entrenamiento del aplicativo movil .....	55
Figura 5- Pruebas de entrenamiento del gesto letra “E” .....	56
Figura 6- Pruebas de entrenamiento del gesto letra “F” .....	57
Figura 7- Pruebas de entrenamiento del gesto letra “G” .....	58
Figura 8- Pruebas de entrenamiento del gesto letra “H” .....	59
Figura 9- Pruebas de entrenamiento del gesto letra “I” .....	60
Figura 10- Pruebas de entrenamiento del gesto letra “K” .....	61
Figura 11- Gráfica Frecuencia Pre-Test -Tiempo promedio en comunicación .....	64
Figura 12- Gráfica Frecuencia Post-Test -Tiempo promedio en comunicación.....	65
Figura 13- Gráfica Frecuencia Pre-Test -Costo promedio de contratación .....	68
Figura 14- Gráfica Frecuencia Post-Test – Costo promedio de contratación de intérprete .....	68
Figura 15- Cronograma de ejecución de desarrollo del proyecto de investigación .....	82
Figura 16- Pruebas de entrenamiento del gesto letra “A” (clasificador de configuración simple) ...	85
Figura 17- Boleta de Matrícula y Pensión del curso de Lenguaje de Señas Peruanas.....	88
Figura 18- Lugar de Pruebas: Aula de Cases de Lenguaje de Señas ASSORELI.....	90
Figura 19- Caso de Uso General .....	92
Figura 20- Caso de Uso Detectar Mano .....	93
Figura 21- Caso de Uso Guardar Gesto .....	94
Figura 22- Caso de Uso Entrenar Gesto .....	95
Figura 23- Caso de Uso Reconocer Gesto .....	96
Figura 24- Pantallas del Sistema Inicio de Sistema .....	96
Figura 25- Menú de Configuraciones .....	97
Figura 26- Configuración de Resolución de Cámara.....	97
Figura 27- Resolución 320*240 seleccionada .....	98
Figura 28- Detección de Fondo y Detección de la Mano.....	98
Figura 29- Imagen Binarizada.....	98
Figura 30- Mano detectada con extracción de características, enumerada de 0 a 4 .....	99
Figura 31- Entrenamiento de la Letra B .....	99
Figura 32- Entrenamiento de la Letra C .....	100
Figura 33- Entrenamiento de la Letra D.....	100

# **I. INTRODUCCIÓN**

## 1.1 Realidad Problemática

En los últimos años la Comunidad de Personas Sordas están en constante cambio desde su concepción de personas con derechos a ser miembros de una minoría lingüística y cultural. La lengua de señas al igual que las lenguas auditivo fonéticas es una Lengua plena. Este cambio se evidencia en que ahora tienen un mayor acceso a tecnologías, medios de comunicación y servicios que ofrece el Estado a través de personal capacitado para interpretar el lenguaje de señas como así también para brindar sus servicios.

En el Perú la población con discapacidad asciende a 1 619 88 que representa el 5.2% (en virtud de una población total de 31 151 643), a partir de ello se estima que la Población con Discapacidad Auditiva serían 560 mil 730 personas, estas representan el 1,8% de la población total en el Perú según CENSO de INEI (Instituto Nacional de Estadística e Informática) del Año 2015.

La población con discapacidad inscrita son 157 mil 355 personas de las cuales los inscritos por discapacidad auditiva son 22 674 que representan el 14,4% de inscritos de ahí solo 10 867 personas tienen discapacidad auditiva severa. Además, se refleja que la mayor concentración está en Lima Metropolitana con 6 mil 962 personas hasta la fecha del 31 de agosto del 2016. La situación de las personas con discapacidad auditiva en el Perú que se refleja en el sector de Educación Básica son el 66.28% de la Población se encuentra en Escuelas Especializadas y el 33.72% en Escuelas Inclusivas, recalando que por primera vez se están incluyendo intérpretes de Lenguaje de Señas Peruanas en las Escuelas Especializadas. De tal forma se puede afirmar que solo la Población Sorda en etapa escolar básica se beneficia de los intérpretes que amoldan el lenguaje oral al señado según CONADIS (Consejo Nacional para la Integración de la Persona con Discapacidad)

En el Sector Laboral de Empresas que sí contratan a personas con discapacidad en el Perú se estiman solo el 9,3% (13 mil 847 personas aprox.) representan a los discapacitados auditivos según el Estudio del MTPE (Ministerio de Trabajo y Promoción del Empleo) en el año 2012.

Se estima que en el Perú hay solo 23 intérpretes calificados y reconocidos por ASISEP en el año 2015 (Asociación de intérpretes y guías de intérpretes de lengua de señas del Perú), Existen intérpretes empíricos que aprendieron por algún amigo o familiar sordo, los cuales deben de especializarse ya que no aplican técnicas de interpretación (Isabel Rey Clemente, presidenta de ASISEP).

Flavio Figallo ex-viceministro de Gestión Pedagógica, reconoció deficiencias (falta de intérpretes de lenguaje de señas, limitaciones para acceder a diferentes servicios públicos, etc.) refiriéndose a que ha sido consensuado el reglamento de ley N° 29535 y esta para aprobación de la misma para el año presente (2015). Además, identificó que existen deficiencias acerca de el aproximado de sordos en el Perú que se estima 100 mil jóvenes en etapa escolar con el cual se deberían tener 5000 intérpretes para cubrir la demanda (Perú 21, 2015).

El 15 de agosto del 2017 fue aprobada mediante Decreto Supremo la Ley que Otorga el Reconocimiento Oficial a la Lengua de Señas Peruanas con la finalidad de establecer condiciones de igualdad, protección y realización para las personas discapacitadas, a su vez promueve la inclusión plena y efectiva en la sociedad, economía, tecnología, cultura y política; disponiendo que se reconoce la lengua de señas peruanas como lengua predominante para las personas sordas en el Perú, aunque no es obligatorio. Y cabe la posibilidad a la elección de tres métodos de comunicación en la vida cotidiana de las personas sordas. (Ley N° 29535, que Otorga el Reconocimiento Oficial a la Lengua de Señas Peruanas)

De acuerdo a la Entrevista con la presidenta de la Asociación de Sordos Región Lima menciona que las personas inscritas en la asociación están compuestas de personas discapacitadas auditivas de dos tipos: parcial y total. Las cuales necesitan de atención personalizada al iniciar la comunicación con las personas hablantes de su entorno, debido a que estos en general solo se comunican con familiares que entienden su lenguaje. De tal modo es necesidad un canal de comunicación efectiva para que puedan hacerse entenderse con la comunidad oyente, el lenguaje de señas como tal no es reconocido por la sociedad ya que el Ministerio de Cultura no lo ha declarado como una lengua adicional en nuestra nación. Por tal motivo solo algunas instituciones del estado brindan el servicio interno de intérpretes que puedan apoyar al discapacitado auditivo a llevar a cabo

una determinada solicitud ante una entidad pública de esta forma le es difícil acceder a servicios sociales que el Estado Peruano nos brinda como ciudadanos. Además, la presidenta de la Asociación de Sordos menciona que los asociados generalmente tienen trabajo de actividad manual como jardinería, orfebrería, procesos mecánicos, mecanografía, etc. De este modo les imposibilita acceder a contratar servicios de un intérprete ya que su condición económica no les favorece.

Esta situación es reflejo que el mercado laboral actual no se interesa en contratar personas discapacitadas a pesar de que existe un marco de ley para las personas con discapacidad que prima de una cuota de empleo del 5% de contratación en el sector público y un 3% en el sector privado. (Ley N° 29973, Ley General de las Personas con Discapacidad)

Esta es la problemática que enfrenta ASSORELI (Asociación de Sordos Región Lima) la cual es una organización sin fines de lucro que brinda apoyo social y cultural a discapacitados auditivos, a su vez enseña el Lenguaje de Señas Peruanas a discapacitados auditivos como también a personas hablantes. Además, fomenta capacitaciones tecnológicas con la finalidad de la inserción laboral de las personas con discapacidad auditiva.

En la presente investigación se estudia el desarrollo de un Sistema de Información móvil de interpretación del lenguaje de señas peruanas para mejorar la comunicación de los discapacitados y a su vez suplir la necesidad de intérpretes de lenguaje de señas. Se propone un aplicativo móvil cuyo objetivo es eliminar las barreras de la comunicación de las personas sordas ante las diversas situaciones que enfrentan con la sociedad hablante; es por ello que se integran herramientas de la computación visual, sintetizadores de voz para lograr realizar una herramienta de ayuda para la comunicación de personas sordas con las personas oyentes. Este tipo de investigación que se desarrolló fue aplicado y pre-experimental.

## 1.2 Trabajos previos

### 1.2.1 Antecedentes Internacionales

A nivel internacional, en el año 2017, Fang, B., Co, J. y Zhang, M., en la Tesis: “DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation”. Dicho Proyecto tiene el objetivo de romper la barrera de comunicación entre las personas sordas y la mayoría de la audiencia, y por lo tanto tiene el potencial significativo de cambiar fundamentalmente la vida de las personas sordas. En este artículo, presentamos el diseño, la implementación y la evaluación de DeepASL, una tecnología transformadora de lenguaje de señas basado en el aprendizaje profundo que permite la traducción de ASL ubicua y no intrusiva en niveles de palabras y oraciones. A nivel de palabra, DeepASL logra una precisión de traducción promedio del 94.5% sobre 56 palabras de ASL comúnmente usadas. En el nivel de las oraciones, DeepASL logra una tasa promedio de error de 8.2% al traducir oraciones ASL no vistas y una tasa promedio de error de palabras de 16.1% al traducir oraciones ASL realizadas por usuarios no vistos más de 100 oraciones ASL comúnmente usadas. Dada la solución de innovación que ofrece y su desempeño prometedor, creemos que DeepASL ha hecho una contribución significativa al avance de la tecnología de traducción ASL.

A nivel internacional, en el año 2015, Rodríguez, V., Grijalva, J. y Gallar, Y., en la Tesis: “Sistema de traducción simultánea de lenguaje de señas a voz mediante una interfaz natural de usuario para personas con discapacidad”. Dicho Proyecto tiene el objetivo de traducir de forma automática las señas a voz, lo cual facilita la comunicación con personas que no entienden el lenguaje gestual. El sistema fue desarrollado de un algoritmo en el lenguaje C# ya que se utilizó tecnologías de Microsoft como la cámara Kinect, a su vez se desarrolló una interfaz Natural de Usuario NUI la cual permite reconocer coordenadas corporales y detecta movimientos todo ello implementado en un módulo con partes de ordenador de escritorio. Como

conclusión se obtuvo los siguientes datos: al realizar el sistema en un sistema operativo base como es Microsoft Windows se aprovecha el narrador para leer el texto asociado a la seña identificada, en la traducción se determinó que el número de fotogramas de captura deberá ser mayor o igual a cero ya que habría una sobrecarga de datos cuando la persona discapacitada se mueve a mayor velocidad de la recomendada todo ello con la finalidad de compartir la esencia del traductor de señales y el rol de apoyo para las personas discapacitadas auditivos.

A nivel internacional, en el año 2014, Guerrero, J. y Pérez, W., en la Tesis: “Sistema traductor de Lenguaje de Señas Colombiana a texto basado en el dispositivo FPGA”. Dicho Proyecto tiene el objetivo de apoyar la comunicación de personas con discapacidad auditiva severa con las demás personas. El sistema empleó técnicas de visión artificial para el reconocimiento de las señas estadísticas de la Lengua de Señas Colombianas (LSC). Este sistema se implementó en el dispositivo FPGA Cyclone II EP2C70F896C6 el cual no requiere el uso de guantes o marcadores visuales para su correcto funcionamiento ya que se le acondicionó una cámara digital TRDB-D5M para la captura de imagen y para el reconocimiento se constituye de una red neuronal artificial perceptrón multicapa (MLP) potenciado en MatLab. Como conclusión se obtuvo la capacidad de que el sistema logró una tasa de efectividad del 98.15% al reconocer todas las 23 señas estáticas que pertenecen al abecedario de la Lengua de Señas Colombianas.

A nivel internacional, en el año 2014, Vintimilla Sarmiento, G., en la Tesis: “Desarrollo e Implementación de una Aplicación que traduzca el abecedario y los números del uno al diez del lenguaje de señas a texto para ayuda de discapacitados auditivos mediante dispositivos móviles Android”. Dicho Proyecto tiene el objetivo reconocer letras y números del lenguaje de señas, para ello se realizó los procedimientos de tratamiento de imagen y entrenamiento de la red neuronal. La arquitectura se basa en un servidor de inteligencia artificial ubicado en la nube la cual mediante un algoritmo de back-propagation toma los valores para realizar una comparación con el valor

de entrada hasta obtener una respuesta que será visualizada en pantalla del dispositivo móvil con la finalidad de no saturar recursos del dispositivo móvil. Como conclusión se logró identificar que el aplicativo solo puede reconocer imágenes previamente aprendidas, si no las reconoce el aplicativo no arrojará resultado por lo que la investigación se puede tomar como base para futuras aplicaciones que reconozcan palabras completas, la variación de iluminación y la variación de color de fondo pueden provocar imposibilidad para reconocer ciertas imágenes, las letras que requieren movimiento no podrán identificarse ya que dependerán de reconocimiento de movimiento, se aplicó una estructura de 3 capas con la finalidad de mejorar la velocidad de respuesta dos de los niveles se encuentran en un web service, se determinó que la precisión de la respuesta no depende de la resolución de la cámara ya que la imagen capturada pasa por un proceso para luego convertir en bits.

### **1.2.2 Antecedentes Nacionales**

En el Perú, en el año 2015, Vílchez, K. (2015), en la Tesis: “Sistema Intérprete de Lenguaje Alternativo para mejorar la comunicación de las personas sordas en la Asociación de Sordos de la Libertad”. Dicho Proyecto tiene el objetivo de mejorar la comunicación de las personas sordas mediante un sistema de escritorio que integra herramientas de desarrollo de redes neuronales artificiales con MatLab y desarrollo ágil con la metodología Iconix. Se tomo como base de medición el tiempo promedio de comunicación en las personas sordas y con ella obtuvo los siguientes resultados: con el sistema actual el 36.6% demora en entender a una persona normal entre 8 – 10 minutos y con el sistema propuesto el 39.0% demora en entender a una persona normal entre 4 – 6 minutos, reduciendo en un 2.4% el tiempo promedio la comunicación en las personas sordas; con respecto a los medios de comunicación entre la sociedad y las personas sordas con el sistema actual el 2.4% emplean 3 medios de comunicación y con el sistema propuesto el 0.8% emplean 5 medios de comunicación, aumentando en un 1.6% los medios de comunicación entre la sociedad y las personas sordas; por otro lado el número

de personas sordas en la inserción laboral con el sistema actual el 15.4% trabaja y con el sistema propuesto el 20.3% trabaja, aumentando en un 4.9% la inserción laboral en las personas sordas, los costos de contratación de traductores de señas (semanal) con el sistema actual el 21.1% paga entre 40 – 50 nuevos soles y con el sistema propuesto el 27.6% paga entre 20 – 30 nuevos soles, reduciendo el costo de contratación de traductores de señas. Concluyendo, el sistema mejoró significativamente la comunicación de las personas sordas en la Asociación de sordos de La Libertad.

### **1.3 Teorías Relacionadas al Tema**

#### **1.3.1 Variable Independiente: Aplicación Móvil**

##### **1.3.1.1 Definición de Dispositivos móviles**

Generalmente son aparatos computacionales de menor tamaño que está relacionado a la portabilidad de su uso y de las capacidades como almacenamiento de datos, aplicaciones y comunicación que puedan tener y poder accederlos en cualquier lugar como, por ejemplo: Smartphones, Tablets, PDAs, entre otros.

##### **1.3.1.2 Aplicación Móvil**

Software desarrollado para ejecutarse bajo un Terminal Móvil. Diseñada para enseñar, entretener o de apoyo cotidiano dirigido a sus usuarios. Tiene funcionalidades que solo existían en ordenadores personales y ahora pueden ser ejecutadas en los dispositivos móviles. (Avilés, 2011, “Estrategia y desarrollos de aplicaciones móviles”).

##### **1.3.1.2.1 Android**

Es un sistema operativo móvil de código abierto que como plataforma nos brinda un soporte para terminales móviles el cual está basado en un kernel Linux (versión 2.6).

## Características de Android

Android es una plataforma neutral por la cual el desarrollo de aplicaciones resulta ideal para el programador. A continuación, se lista las características funcionales más resaltantes:

- Variedad de diseños (VGA, bibliotecas de gráficos 2D y 3D, etc.)
- Repositorio de data en SQLite (Base de Datos no relacionales)
- Conectividad (LTE, UMTS, CDMA, EV-DO, GSM/EDGE, WiFi, Bluetooth y NFC)
- Mensajería (MMS y SMS)
- Navegador Web
- Máquina virtual de Java
- Están escritas en lenguaje Java las aplicaciones las cuales son ejecutadas en la máquina virtual Dalvik, la cual esta diseñada exclusivamente para los terminales móviles.
- Soporte de formatos de imagen, video y audio (H.264, MPEG-4, AMR, OGG, AAC, GIF, PNG, JPEG)
- Cuenta con soporte para adicionar hardware (pantallas táctiles, GPS, acelerómetro, cámaras de video, etc.)
- Entorno de desarrollo (emuladores, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para otras IDEs).
- Plataforma realmente abierta basada en Linux y de código libre. Se puede usar y modificar el sistema sin pagar licencias.
- Portabilidad, gracias a la máquina virtual las aplicaciones se podrán ejecutar en gran variedad de dispositivos tanto presentes como futuros.
- Arquitectura basada en componentes inspirados en internet como es el caso del XML permite el uso de una misma interfaz en dispositivos de pantallas dispares.
- Filosofía de dispositivo siempre conectado a internet.
- Variedad de servicios incorporados: Reconocimiento y síntesis de voz, localización basada en GPS y torres de comunicación, potentes bases de datos, NFC, etc.

- Nivel de seguridad alto ya que las aplicaciones se encuentran aisladas unas de otras porque disponen de una serie de permisos que limitan su rango de actuación.
- Optimización de recursos por baja potencia y/o baja memoria.
- Calidad de gráficos de alta definición en sonido también por la gran variedad de formatos soportados.

### **1.3.1.3 Visión Artificial**

También conocida como visión por computador, la cual consiste en extraer atributos relevantes visuales desde la captura de información de un entorno físico mediante tareas automáticas (Platero, 2011).

#### **1.3.1.3.1 OpenCV**

Es una librería escrita en lenguaje de programación C++ que se utiliza para el procesamiento de imágenes (OpenCV, 2014), puede usarse en otros lenguajes de programación, ejecutarse bajo distintos sistemas operativos y plataformas de hardware. Esta librería provee procedimientos para manipular imágenes y videos. También incorpora herramientas de inteligencia artificial. Permite realizar tareas de adquisición y escritura de imágenes, transformaciones, reconocimiento y seguimiento de objetos, etc. Los métodos de OpenCV están altamente optimizados y utilizan recursos de hardware para acelerar la ejecución de programas.

#### **1.3.1.3.2 Imágenes Digitales**

Es un conjunto de píxeles que forman la imagen, la cual también puede estar formada por puntos, en las fotografías estos puntos se representan de forma irregular y aleatoriamente, en cambio en la imagen digital está conformada por una matriz de filas y columnas.

#### **Tipos de Imágenes Digitales**

Existen dos modos para la manipulación de la data de imágenes digitales. Las cuales son:

- **Imagen de Mapa de Bits:**  
Conformada por una grilla de celdas. Estas celdas a su vez se representan como píxeles, asignándoles un valor de color e iluminación propias. Es por ello que cuando se observa la totalidad del conjunto de celdas, ellas nos dan la percepción de un archivo de imagen continua en tono.
- **Imágenes Vectoriales:**  
Son llamados gráficos orientados a objetos, de tamaño pequeño comparado con los de tipo mapas de bits. Ya que estos organizan la información de una imagen de forma simple ya que se basa en la generación de objetos en base a trazos geométricos determinados por cálculos matemáticos. En forma que estos gráficos puedan ser identificadas en base a coordenadas de una línea guardada como referencia.

#### **1.3.1.3.3 Procesamiento de Imágenes**

Son aquellos procedimientos (algoritmos) cuyo objetivo es lograr conseguir mejorar la imagen original. Además, consiste en resaltar características determinadas de la imagen o en eliminar aquello que las oculta con la finalidad de que la imagen sea analizada de forma más simple. Para ello se utilizan diversas técnicas:

- **Manipulación de Contraste:**  
Las variaciones locales del brillo, incrementan el cambio de luminosidad entre zonas más oscuras o más claras, de esta forma se logra simular un enfoque de imagen mas clara. La finalidad es mejorar unas zonas perjudicando otras para así distanciar más los píxeles con valores bajos o a la inversa.
- **Eliminación de Ruido:**  
Es el método de eliminar valores, dependiendo del sensor de la cámara o el medio de transmisión de señal, siempre el ruido digital se manifiesta generando píxeles que toman valores de color gris que difiere de los píxeles vecinos.

- **Realce de Bordes:**

Consiste en transformar una imagen de en forma de mostrar sólo el detalle de bordes o fronteras. Este efecto es lo opuesto a la eliminación de ruido, resaltando los píxeles de gris distinto a los contiguos.

#### **1.3.1.3.4 Extracción de Características**

Luego de procesar la imagen, se identifican los objetos existentes, en la cual se ubican características en la imagen como textura, bordes, líneas, esquinas e incluso movimiento que definen al objeto.

- **Detección de Bordes:**

Es una característica útil que se encuentra en una imagen, la constituyen los bordes, ya que se utilizan para definir los límites entre sí y el fondo de la imagen. La técnica ideal para la detección de bordes es binarizar la imagen de manera que se representen los píxeles en un bit (0 o 1), ya que sea completamente negro o completamente blanco, de esta manera los bordes quedarán completamente marcados.

- **Texturas:**

Es una de las características importantes en la segmentación, identificación de objetos de interés en una imagen. Es una característica de difícil definición ya que está definida mediante la intensidad, regularidad, rugosidad, grosor, densidad, uniformidad y direccionalidad de medidas discretas en el tono de los píxeles y de sus relaciones espaciales.

- **Detección de movimiento:**

Es un tipo de características que apoyan a la segmentación ya que los puntos del mismo objeto presentan velocidades semejantes. Como pueden ser: el movimiento de la cámara, de objetos, cambios de iluminación, cambios de estructura, forma o tamaño del objeto.

#### **1.3.1.3.5 Segmentación de Imágenes**

En este proceso la imagen se fragmenta en partes iguales con la finalidad de aislar objetos de interés basado en un algoritmo de segmentación que tiene

como propiedad básica los valores de gris que diferencia discontinuidad o similitud entre los píxeles vecinos.

- Similitud: Los píxeles de un elemento poseen valores similares para alguna propiedad.
- Discontinuidad: Los objetos destacan del entorno y tienen, por tanto, unos bordes definidos.
- Conectividad: Los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados.

#### **1.3.1.3.6. Aprendizaje de Maquina (Machine Learning)**

Se le conoce al conjunto de técnicas de Inteligencia Artificial (I.A) por el cual la inteligencia se logra obtener a partir de ejemplos la cual se basados en 2 principios:

- Algoritmo de entrenamiento, el cual se encarga de generar un modelo que se empleará en el algoritmo de predicción.
- Algoritmo de predicción, el cual toma un modelo generado por el algoritmo de entrenamiento para luego clasificar las futuras entradas.

#### **1.3.1.5 Metodología de Programación Extrema (XP)**

Esta metodología ágil se enfoca en mejorar el equipo de desarrollo de software en base a las relaciones interpersonales promover el trabajo y aprendizaje en equipo, además mejorando el clima laboral.

La metodología XP tiene las siguientes fases:

- Exploración. -  
Es la fase por el cual, se plantean las historias de usuario por los clientes como primera versión del entregable de producto. También el equipo de trabajo se va familiarizando con herramientas, tecnologías y buenas prácticas que se emplearán en el proyecto. Toma pocas semanas la fase de exploración ya que depende del tamaño y familiaridad con la que se puedan adaptar los programadores con la tecnología.

- **Planificación de la Entrega. -**  
Es la fase donde el cliente formula prioridades en cada historia de usuario, y correspondientemente, los programadores hacen una estimación de esfuerzo necesario para lograr presentar los entregables. Se debe tomar acuerdos sobre el contenido del primer documento entregable. La planificación se puede basar mediante el tiempo o alcance. Si se basa en el tiempo la velocidad del proyecto toma el papel de establecer unas horas adicionales para que se pueda implementar previo a una fecha determinada o cuánto tiempo tomara implementar un conjunto de historias. Si se basa en el alcance se divide la suma de puntos de historias de usuario escogidas entre la velocidad del proyecto obteniendo así el número de iteraciones necesarias para su implementación.
- **Iteraciones. -**  
Es la fase que incluye muchas iteraciones del sistema antes de ser entregado. La primera iteración se intenta establecer la arquitectura del sistema, para que al final de la última iteración el sistema quede listo para entrar en producción. Para ello deberá tomarse en cuenta un Plan de la Iteración que son las historias de usuarios no abordadas, agilidad del proyecto, pruebas de aceptación no superadas, tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación de modo que sea asignada cada programador responsable, pero llevadas a cabo por parejas de programadores.
- **Producción. -**  
Es la fase que requiere de pruebas y testeos previo a que el sistema sea trasladado al ambiente del cliente.
- **Mantenimiento. -**  
En esta fase se versionan las fases, mientras se encuentra el sistema en producción. Esto requiere actividades de soporte al cliente.

- Muerte del Proyecto. -

En esta fase el cliente se queda sin historias que sean incluidas al sistema de ese modo se satisfacen las necesidades del cliente como la confiabilidad y rendimiento del sistema. Generando así la documentación definitiva del sistema.

En la Metodología XP, los principios y prácticas pueden ser llevadas al extremo, de ahí proviene su nombre. De ahí se puede presentar las siguientes características esenciales de historias, roles, procesos y prácticas.

Esta metodología asume que la planificación no se podrá llevar a cabo como uno desea, además variará en función de cómo sea las necesidades del negocio. Por consiguiente, el valor real reside en obtener agilmente un plan inicial, y contar con mecanismos de feedback que ayude a conocer dónde estamos. La planificación es iterativa ya que un representante del negocio puede decidir al comienzo de cada iteración las características finales que se van a implementar.

- Historias de usuario: Son tarjetas en la cual el cliente debe describir las características que debe tener el software, la entrada de datos y lo que espera que salga ya sean de requerimientos funcionales o no funcionales. El orden no se toma en cuenta, siempre se podrá agregar más historias luego.

En el inicio de cada iteración se registrarán los cambios de las historias de usuario ya que a partir de ello se planifica la próxima iteración. Se descomponen en tareas programadas las historias de y a su vez estas se asignan a los programadores que luego implementaran en cada iteración.

- Roles, En el desarrollo se identifican a los involucrados donde se definen los cargos, los cuales son: el cliente, programador, encargado de pruebas, entrenador, encargado de seguimiento, gestor de proyectos y consultor.
- Proceso, En caso de un proyecto XP se logra tener éxito cuando el cliente identifica el valor de negocio a implementar basándose en las habilidades del equipo el cual ayuda a medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo se da cuando

el cliente define el valor del negocio, Para lograr la implementación el programador debe estimar esfuerzos, además el cliente acorde a sus prioridades y tiempo debe decidir lo que desea construir para que finalmente el programador pueda construir el valor de negocio que luego este proceso será iterativo.

- Prácticas, la reducción de costos supone que la metodología ágil XP está comprendida para disminuir el costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP se basa en el crecimiento paulatino del costo del cambio y con un comportamiento asintótico.

Esta metodología propone 12 prácticas técnicas de fácil entendimiento que juntas garantizan el buen resultado del proyecto. A continuación, se detallan estas 12 prácticas. (Bahit, 2012)

#### PRÁCTICA 1: Cliente in-situ

Disposición del cliente para interactuar activamente con el equipo en todas las fases del proyecto.

#### PRÁCTICA 2: Semana de 40 Horas

Se requiere para garantizar la calidad del equipo de desarrollo por la cual no se tiene que sobrecargar responsabilidades de mayor esfuerzo del que se disponga en el.

#### PRÁCTICA 3: Metáfora

Proponer el uso de metáforas haciendo una comparativa entre una funcionalidad del sistema y la vida real.

#### PRÁCTICA 4: Diseño Simple

Se trata de un principio de preservar un diseño simple, estándar, de fácil comprensión y refactorización. “Hacer lo mínimo indispensable, tan legible como sea posible”.

#### PRÁCTICA 5: Refactización

Se entiende como una técnica de modificar el código fuente de un software sin afectar a su comportamiento externo, de tal modo que al desarrollar nuevas funcionalidades impedir las redundancias.

#### PRÁCTICA 6: Programación de a pares

Consiste en programar de dos desarrolladores los cuales pueden intercambiar roles de tal modo que tendrán la libertad de “ser originalmente creativos”.

#### PRÁCTICA 7: Entregas cortas

Se pretende generar entregas cortas en el tiempo, agregando funcionalidades cortas, de tal modo que el cliente tenga buena experiencia con el software, pudiendo continuar mejorando con facilidad en la fase de implementación.

#### PRÁCTICA 8: Testing

Según lo propuesto por esta metodología se establecen tres tipos de test: Test de Integración, Aceptación, Unitarios.

#### PRÁCTICA 9: Código Estándar

Es esencial a la hora de programar el uso de un estándar de escritura de código fuente ya que permite generar un código mas limpio y entendible para proveer a otros programadores un mayor entendimiento de este.

#### PRÁCTICA 10: Propiedad colectiva

En esta metodología no existe dueño de un determinado código o funcionalidad. La propiedad del código fuente es colectiva cuyo objetivo es que que todo el equipo de desarrollo conozcan “qué” y “cómo” se está desarrollando el sistema.

### PRÁCTICA 11: Integración continua

Propone que todo el código encuentre un repositorio común en el cual deban administrarse las versiones previo testeo con la finalidad de verificar lo nuevo y que no “rompa” con lo anterior.

### PRÁCTICA 12: Juego de planificación

La dinámica es la siguiente:

1. La lista de funcionalidades es presentada por el cliente.
2. El grupo de desarrollo estima el esfuerzo que demanda desarrollar las funcionalidades propuestas por el cliente.
3. Las Historias de Usuario a desarrollar y en qué orden lña decide el Cliente XP es una metodología ágil que tiene un enfoque:

- Para proyectos de alto riesgo en la variación de requerimientos que puede tener.
- Incentiva que el cliente participe de manera activa en el proceso del desarrollo, además se logra la programación en parejas como también de equipo para lograr una integración comunicativa.
- En esta metodología, la retroalimentación continua se da entre el cliente y el equipo de desarrollo.
- Entre los participantes existe la comunicación fluida y practicidad y coraje para los cambios como soluciones a enfrentar.

## **1.3.2 Variable Dependiente: Interpretación de Lenguaje de Señas**

### **1.3.2.1 Lenguaje de Señas**

Es un sistema de comunicación producido por el cuerpo y que se percibe a través de la vista (Ministerio de Educación, 2015, p.12).

El lenguaje de señas se divide en pequeñas unidades llamadas queremas las cuales se estudian desde 7 aspectos o parámetros (Stokoe, 1960).

Las cuales son:

#### **1. Punto o lugar de articulación**

Es el espacio donde la mano o las manos realiza la seña.

Se utiliza generalmente 4 áreas del cuerpo:

- Cabeza
- Tronco
- Cuello
- Brazos

## **2. Configuración de las manos**

Es la forma en que se coloca las manos al realizar una seña, o las posiciona para expresar alguna palabra. Se pueden dividir según configuración (actos), postura y manos:

### **Según configuración. -**

- Configuración Simple: se ejecuta en un acto.
- Configuración Compuesta: se ejecuta en dos actos.
- Configuración Doble: tiene un acto de inicio y otro final.

### **Según cantidad de manos. -**

- Seña Monomanual: Se realiza con 1 sola mano.
- Seña Bimanual: Se realiza a 2 manos.

### **Según postura de las manos. -**

- Señas Simétricas: Son aquellas que tienen la misma configuración en las dos manos.
- Señas Asimétricas: Son aquellas señas que difieren la configuración de una mano y otra.

## **3. Movimiento de la palma de la mano**

Se realiza dentro del espacio donde se realiza la seña, la mano puede moverse de múltiples maneras: hacia arriba, abajo, izquierda o derecha.

Además, se toma en cuenta los diferentes movimientos como frontal, diagonal y la postura sin movimiento. Cualquier cambio en dicha orientación implica una modificación y significado de la seña.

#### **4. Orientación de las manos**

Es la relación que se da entre la posición de la mano respecto al cuerpo de la persona que habla en señas: las cuales pueden ser:

Manos paralelas al respecto del cuerpo, manos horizontales con respecto al cuerpo, manos oblicuas con respecto al cuerpo, los dedos hacia abajo con respecto al cuerpo, etc.

#### **5. Punto de contacto**

Se refiere a la parte de la mano principal cuando hace contacto con una parte del cuerpo, se realizan como apoyo en los gestos dinámicos que se realizan a nivel de la mano, codo, muñeca.

#### **6. Plano**

Se refiere al punto situado delante de la persona que se comunica, además se percibe como la distancia de la mano con respecto al cuerpo al realizar la seña. Se dividen en 4 planos:

- 1er plano: Se realiza en el propio cuerpo
- 2do plano: Se realiza a corta distancia del cuerpo. En este plano se realizan la gran mayoría de señas.
- 3er plano: Es usado por la dactilología. (Abecedario de señas)
- 4to plano: Es la zona más lejana del cuerpo del que hace la seña.

#### **7. Componentes no manuales**

Esta comprendido por expresiones del rostro, como movimiento de labios, cabeza, que se realizan de forma paralela a la configuración de las manos.

##### **1.3.2.1 Dactilología**

Es la representación de las letras del alfabeto, pero con gestos de la mano. Se utiliza para los nombres propios, apellidos, nombres de instituciones y palabras que no tienen una seña específica. El alfabeto manual está compuesto por 27 letras, mediante las cuales, es posible conformar cualquier palabra (Ministerio de Educación, 2015, p.41).

Figura 1. Abecedario usado por la dactilología



Fuente: Ministerio de Educación, 2015, p.41

### 1.3.2.2 Lengua de Señas Peruana

Creada por la comunidad sorda del Perú, originaria lengua peruana que posee como todas las lenguas, identidad, valores y cultura de usuarios (I Coloquio Internacional sobre la Lengua de Señas Peruanas, 2016, p.1).

### **1.3.2.3 Interpretación del Lenguaje de Señas**

Acción de verter información original de una lengua determinada y transformarla al lenguaje de señas y en reversa (Grupo de Accesibilidad Audiovisual del CERMI, 2006, p.55).

Es el proceso donde el intérprete transfiere en tiempo real el mensaje del emisor ya sea hablado o en señas, a la velocidad que lo diga, sorteando vocabulario, terminología y logrando toma de decisiones en microsegundos que generan que la formación del intérprete sea completa para producir la interpretación a gran velocidad (Burad, 2009, p.69).

#### **1.3.2.3.1 Interpretar**

Consiste en comprender información de un ser humano de forma hablada o en señas de un contexto o situación determinada, a su vez debe de transferir lo que este enunciado en otra lengua (Burad, 2009, p.4).

Traducir: “Es la conversión de un texto escrito en otro texto escrito.” (González, R, 2015, p.14)

##### **1.3.2.3.1.1 Técnicas de Interpretación**

Basado en el tiempo entre la emisión del mensaje hasta la interpretación.

- **Interpretación Simultánea:** Se realiza cuando la intervención es amplia, este procedimiento es de transmisión casi instantánea del discurso del emisor. Que puede ejecutarse al mismo momento que el emisor emita el discurso.
- **Interpretación Consecutiva:** Se realiza cuando el intérprete pronuncia partes del discurso en periodos cortos de tiempos, de tal forma que el expositor pueda emitir un mensaje en periodos de tiempo realizando pausas con la finalidad de la intervención del intérprete para que transmita la información del discurso acorde la lengua destino, se puede realizar luego de cada frase, conjunto de frases o al final del breve discurso.

### 1.3.2.3.1.2 Tipos de Interpretación

- **Interpretación de contacto o Bilateral:** Se le conoce como traducción frente a frente, se refiere al procedimiento por el cual el intérprete transmite un mensaje hablado al lenguaje de señas o viceversa, al mismo tiempo que se produce todo tipo de comunicación. Este tipo se vincula al área de negocios, comercial, social, institucional, etc., el cual solo es realizado por un intérprete de forma bidireccional.
- **Interpretación Unilateral:** Consiste en el proceso de interpretación unidireccional, el cual se realiza en un solo canal de comunicación como TV o publicidad.

### 1.3.2.4 Diferencia entre intérprete y traductor de lengua de signos

Tabla 1. Diferencias de traducción e interpretación

TRADUCCIÓN	INTERPRETACIÓN
Comprensión del significado de un registro en una lengua, llamada registro origen o registro de salida, para producir otro registro con significado equivalente en otra lengua, llamada registro meta.	Traspaso de un mensaje oral desde una lengua a otra sin posibilidad de revisión ya que no tiene registro.
Dispone de un plazo de tiempo para la entrega de su traducción.	El intérprete trabaja a tiempo real y de forma inmediata.
El traductor debe entregar una traducción lo más precisa y detallada posible. Para ello dispone del tiempo necesario para documentarse y consultar diferentes materiales de referencia.	La inmediatez de la interpretación le brinda al intérprete cierta libertad para expresar en otra lengua lo que acaba de recibir.
Deben denominar la lengua de partida para comprender el registro, pero su especialidad es la lengua de llegada.	El intérprete debe tener una gran fluidez y dominio en ambas lenguas, ya que debe comprender la lengua origen y comunicar inmediatamente en la lengua de llegada sin la posibilidad de consultar algún material.
Trabaja solitario y personal.	Permanente contacto con otras personas e interacción constante.

Fuente: Siré, S. (2016) Intérprete profesional en lengua de Señas.

### **1.3.3 Comunicación de discapacitados auditivos**

Se define como sistema visual de símbolos escritos que sirven como canal de comunicación para las personas con sordas. Siendo sólo un canal de los muchos que existen con el fin de eliminar la brecha de interacción y comunicación entre sordos y oyentes no sólo a nivel local, sino también universal (Muñoz Ramírez, Ana, 2014).

En la comunidad sorda participan personas sordas, sordo-ciegas y oyentes, estos últimos actúan como intérpretes que participan en objetivos en común para la inclusión social y cultural de las personas sordas (Geldres Antezana, Diana, 2015).

El acto comunicativo se establece entre las dos comunidades (sorda y oyente) no se puede enmarcar en unas características generales puesto que siempre esa situación de encuentro depende de las particularidades de las estructuras cognitivas y sociales de los individuos. “Las formas de comunicación, en todas situaciones comunicativas es necesaria la mediación de la lengua ya sea la lengua de señas o uso del castellano, sobre lo cual se denota que pocas veces se media con la lengua de señas porque el oyente no maneja dicho lenguaje, mientras que generalmente los sordos tengan un manejo de castellano, así sea básico” (Abello Gómez, Viviana, 2017)

El investigador observó que el uso de la tecnología es evidente y entra a ser una forma de comunicación importante entre los dos grupos sociales. El uso de aplicaciones como WhatsApp y Facebook, permiten al sordo tener una mayor conexión y comunicación con familiares, amigos y conocidos oyentes, ya sea por medio del uso del castellano escrito o de mensajes visuales que permiten mediar nuevas formas de comunicación.

Es importante diferenciar que la comunicación no puede estandarizarse como lenguaje ya que existe variabilidad de dialecto y sistema de signos con distintas configuraciones de símbolos, códigos sociales de cada cultura (Hymes, 1962).

#### **1.3.3.1. Discapacitado Auditivo**

La discapacidad representa una deficiencia o resticción de ejecutar una actividad de forma normal en lo considerado para un ser humano (OMS, 1980).

El término “sordo” es correcto, porque la persona con discapacidad auditiva tiene las mismas capacidades, facultades de una persona oyente, con la única limitación de que no escucha.

El término “sordomudo”, es una denominación errónea, ya que antiguamente se pensaba que el sordo era un ser que no tenía la capacidad de comunicación, cuando en realidad si son capaces de comunicar mediante su lengua materna (LSP) e incluso de aprender a leer y escribir en lengua oral, recibiendo la educación adecuada (Geldres Antezana, Diana, 2015).

Existen dos tipos de discapacitados auditivos:

- **Los Profundos.** - Son aquellos que no escuchan
- **Los Hipoacúsicos.** - Son aquellos que tienen restos auditivos, algunos utilizan audífonos y poseen implante coclear

Para esta investigación la discapacidad trasciende del contexto de lo clínico e implicaciones físicas de la limitación auditiva, encontrando esta condición una problemática de carácter social, particularmente relacionado con prácticas comunicativas. El tipo de sordera profunda representa un fenómeno cultural más que solo un simple diagnóstico (Ladd, 2003).

#### **1.3.4 Reconocimiento de Lenguaje de Señas en la Computación Visual**

En esta fase se establecerá el proceso por el cual se medirá el rendimiento del aplicativo móvil en el proceso de reconocimiento de lenguaje de señas.

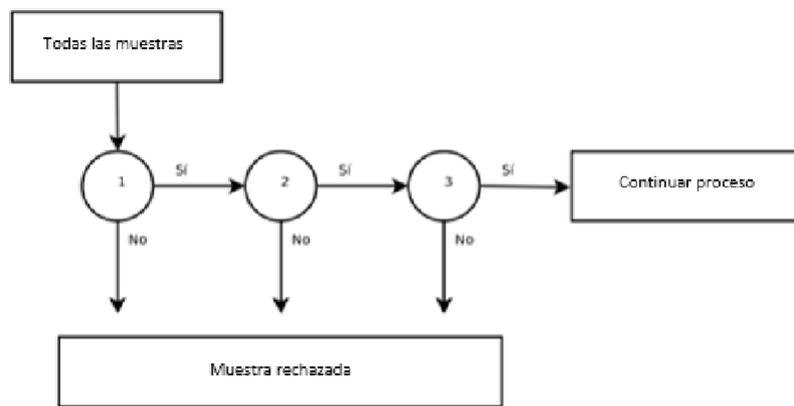
Este proceso se dividirá en clasificadores (etapas) las cuales serán medibles a partir de la dimensión de estudio Rendimiento, estas son:

- Reconocimiento de señas de configuración simple
- Reconocimiento de señas de configuración compuesta
- Reconocimiento de señas de configuración doble
- Reconocimiento de señas monomanuales
- Reconocimiento de señas bimanuales
- Reconocimiento de señas simétricas
- Reconocimiento de señas asimétricas

Las etapas se establecieron acorde la metodología de Stokoe, de ahí el investigador utilizará el aspecto principal de configuración de las manos para el reconocimiento de lenguaje de señas las cuales nos brindan un constructo para ser medible a partir de la dimensión rendimiento que nos brinda Vanrell en su metodología de Evaluación del Rendimiento en la Computación Visual.

“Se entrena un primer clasificador que es capaz de eliminar las muestras negativas fáciles, luego que otro que elimina las muestras negativas más difíciles y así hasta el número de etapas que se establezca”. (Planells, 2009)

Figura 2. Cascada de Clasificadores



Fuente: Planells Lerma, 2009

Para conocer el rendimiento se hace uso de la Matriz de Confusión, la cual funciona como herramienta básica, que permite visualizar el nivel de confusión de un clasificador. De ahí podemos encontrar los siguientes términos en la investigación:

- Positivos Reales: Imágenes etiquetadas como positivas y que el clasificador detecta como positivas.
- Falsos Positivos: Imágenes etiquetadas como negativas y que el clasificador las reconoce como positivas.
- Falsos Negativos: Imágenes etiquetadas como positivas y que el clasificador no detecta.
- Negativos Reales: Imágenes etiquetadas como negativas y que el clasificador correctamente no reconoce.

El Rendimiento de los Clasificadores será evaluado mediante cuatro análisis

- **Exactitud:** Analiza la proximidad entre el resultado del clasificador y la clasificación exacta; cuya fórmula es:

$$Exactitud = \frac{\text{Positivos Reales} + \text{Negativos Reales}}{\text{Predicciones totales}}$$

- **Precisión:** Analiza la calidad de respuesta del clasificador. Su fórmula es la siguiente:

$$Precisión = \frac{\text{Positivos Reales}}{\text{Positivos Reales} + \text{Falsos Positivos}}$$

- **Sensibilidad:** Analiza la eficiencia en la clasificación de todos los elementos que pertenecen a la clase que se desea detectar; mediante el uso de la siguiente fórmula:

$$Sensibilidad = \frac{\text{Negativos Reales}}{\text{Negativos Reales} + \text{Falsos Positivos}}$$

- **Especificidad:** Analiza la eficiencia en la clasificación de todos los elementos que no son de la clase que se desea detectar, haciendo uso de su fórmula:

$$Especificidad = \frac{\text{Negativos Reales}}{\text{Negativos Reales} + \text{Predicciones totales}}$$

- **Factor Rendimiento o F1 Score:** Para poder realizar las pruebas de un clasificador debemos tomar en cuenta realizar una clasificación binaria: es decir, de las muestras para el test de rendimiento, se tendrá como resultado si existe o no lo que se desea detectar. Debido a lo cual, se hará uso de la métrica de evaluación disponible para los modelos de clasificación binaria de Machine Learning (Microsoft, 2017).

$$Factor\ Rendimiento = \frac{2(\text{Precisión} * \text{Sensibilidad})}{\text{Precisión} + \text{Sensibilidad}}$$

## 1.4 Formulación del Problema

### 1.4.1 Problemas General

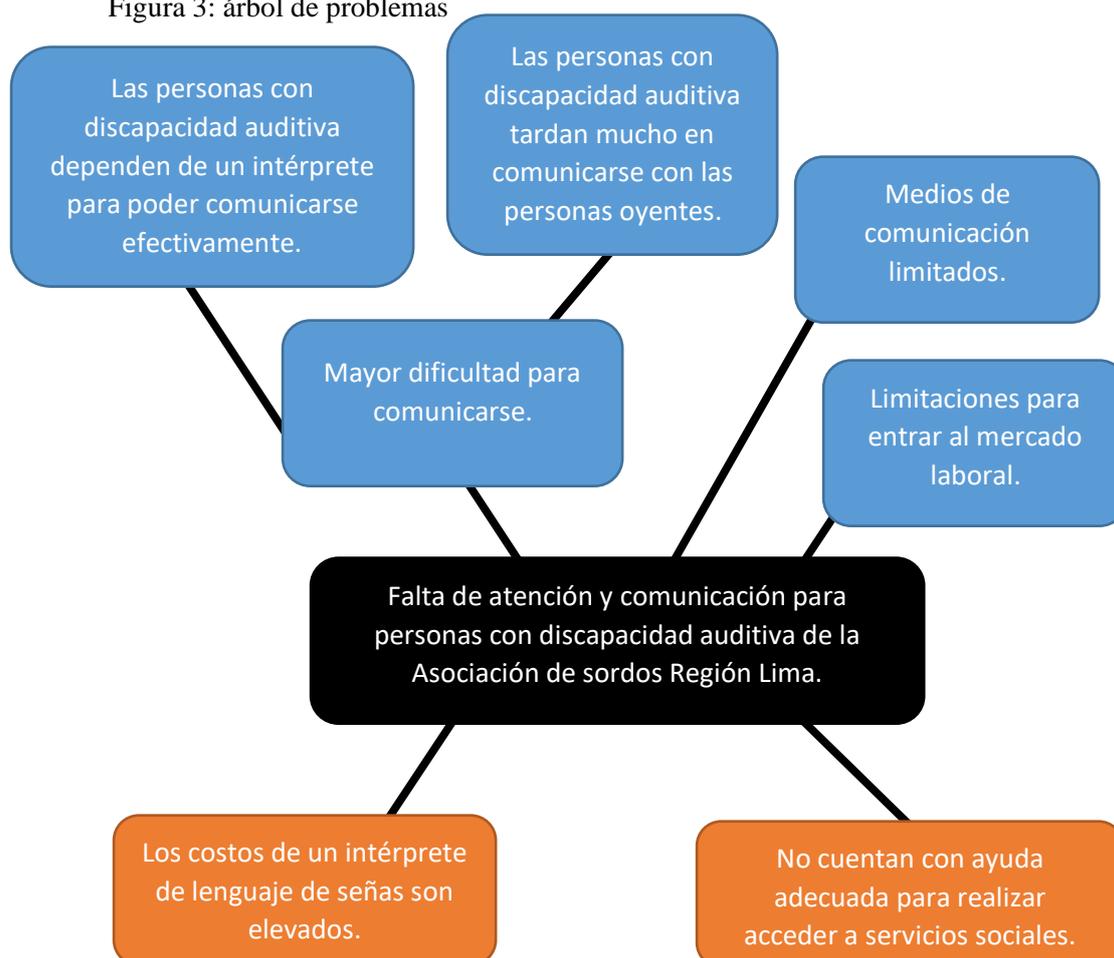
¿De qué manera la implementación de una aplicación móvil influirá en la interpretación del lenguaje de señas peruanas de los discapacitados auditivos de la asociación de sordos región Lima?

### 1.4.2 Problemas Específicos

- ¿De qué manera la implementación de una aplicación móvil reducirá el tiempo de interpretación del lenguaje de señas de las personas con discapacidad auditiva?
- ¿De qué manera el aplicativo móvil mejorara el rendimiento en el proceso de interpretación del lenguaje de señas?
- ¿De qué manera la implementación de una aplicación móvil reducirá el costo de contratación de intérpretes de lenguaje de señas para las personas con discapacidad auditiva?

### 1.4.3 Árbol de Problemas

Figura 3: árbol de problemas



Fuente: Elaboración Propia

## **1.5 Justificación del estudio**

El presente trabajo de investigación propone una solución móvil orientada a personas con deficiencias auditivas para lo cual se ha tenido en cuenta los diferentes ámbitos teóricos, metodológicos, tecnológicos y económicos que se justifican a continuación.

### **1.5.1 Justificación teórica**

Se justifica teóricamente que el presente proyecto de investigación se realiza con el fin de resaltar la importancia de los aplicativos móviles como herramienta de interpretación de lenguaje de señas para mejorar la comunicación de los discapacitados auditivos, para lo cual se toma como referencia aplicaciones realizadas como antecedentes. Según Rosales Romero (2017, p.81) menciona que: “Promover la práctica de proyectos orientados a la asistencia de personas con discapacidad, involucra buenos valores como el de servir y velar por el bienestar de los demás”. El presente trabajo servirá para posteriores investigaciones en este campo de estudios.

### **1.5.2 Justificación metodológica**

Existe un enfoque para las personas sordas y la comunicación con otras personas que no entienden el lenguaje de señas. Las cuales son dos posibles soluciones: una de tipo de reconocimiento de lenguaje de señas y otra es el uso de avatares como mediadores en la comunicación de lenguaje de señas (Ruiz y Villegas, 2014, p.45).

### **1.5.3 Justificación tecnológica**

“Mediante este proyecto se pretende aportar al desarrollo tecnológico respecto al reconocimiento gestual, tecnologías de reconocimiento de gestos como Kinect de Microsoft han dado un importante avance al reconocimiento gestual, además se han construido importantes librerías de software libre como privativo que aportan en el reconocimiento gestual.” (Gonzales y Yimes, 2016, p.2).

#### **1.5.4 Justificación económica**

Lo que nos lleva al desarrollo de este proyecto es involucrar el alcance de recursos que no dependan de inversión adicional, ya que en su desarrollo se utilizan librerías de software libre, además; el dispositivo y la plataforma de implementación son recursos propios de la muestra de estudio (Ruiz y Villegas, 2014, p.87).

### **1.6 Hipótesis**

#### **1.6.1 Hipótesis General**

La implementación de una aplicación móvil mejorará significativamente la interpretación del lenguaje de señas peruanas de discapacitados auditivos de la asociación de sordos región Lima.

#### **1.6.2 Hipótesis Específicas**

- La implementación de una aplicación móvil mejorará significativamente el tiempo promedio de interpretación de lenguaje de señas de las personas sordas.
- La implementación de una aplicación móvil mejorará el rendimiento en el proceso de interpretación del lenguaje de señas.
- La implementación de una aplicación móvil reducirá el costo promedio de contratación de intérpretes de lenguaje de señas para las personas sordas.

### **1.7 Objetivos**

#### **1.7.1 Objetivo General**

Desarrollar la aplicación móvil de interpretación del lenguaje de señas para los discapacitados auditivos de la asociación de sordos región Lima.

### **1.7.2 Objetivo Especifico**

- Reducir el tiempo promedio de interpretación del lenguaje de señas de las personas con discapacidad auditiva.
- Mejorar el rendimiento en el proceso de la interpretación de lenguaje de señas.
- Reducir el costo promedio de contratación de intérpretes de lenguaje de señas para las personas sordas.

## **II. METODO**

## **2.1 Diseño de la investigación**

En esta investigación se reducirá la problemática que existe en la Asociación de Sordos Región Lima a través de un aplicativo móvil de interpretación de lenguaje de señas peruanas. Es necesario tener técnicas y herramientas con métodos ordenados y confiables donde se tome en cuenta la tecnología para lograr establecer conclusiones y recomendaciones que estén relacionadas a la investigación. De tal forma se han considerado dos tipos de estudio para este proyecto de investigación:

### **Tipo de estudio**

Investigación Aplicada: “Es importante en un planteamiento cuantitativo ya que son útiles para evaluar, comparar, interpretar, establecer precedentes y determinar causalidad y sus implicaciones. Teniendo como justificación adelantos y productos tecnológicos”. (Lester y Lester, 2012)

Es de tipo Explicativa ya que se exponen eventos sociales y físicos en los cuales se da a conocer cómo ocurre el fenómeno, además de argumentar de forma deductiva en base a un conocimiento profundo de este. Con la finalidad de intentar encontrar las causas del mismo en relación causal acercándose al problema y describiéndolo (Baptista, 2006).

El tipo de estudio a utilizar es la Investigación aplicada.

### **Diseño**

El diseño de investigación es pre experimental en la medida que el investigador introduce de manera intencional la variable aplicativo móvil que va a diseñar y aplicar para la interpretación del lenguaje de señas.

El diseño de investigación es pre experimental de una sola medición. “Consiste en administrar un estímulo o tratamiento a un grupo y después aplicar la medición de una o más variables para observar cuál es el nivel del grupo en éstas”. (Sampieri, 2010, p.174).

Este diseño podría diagramarse de la siguiente manera:



Dónde:

**G:** Grupo de sujetos o casos.

**X:** Tratamiento, estímulo o condición experimental (presencia de algún nivel o modalidad de la variable independiente).

**O:** Medición de los sujetos de un grupo o de la variable dependiente

El diseño de la investigación a utilizar es Experimental de tipo Pre Experimental debido a que luego de la aplicación de estímulo o tratamiento experimental (implementación del Sistema de Información) se observa su efecto sobre una o más variables dependientes

## **2.2 Variables, operacionalización**

### **2.2.1 Variable Independiente**

#### **Aplicación Móvil:**

Programa informático diseñado para ejecutar una tarea específica en teléfonos inteligentes, tablets y otros dispositivos móviles, el aplicativo móvil puede ser de cualquier tipo; a nivel profesional, de entretenimiento, educativo, entre otros., ayudando en la gestión de procesos y actividades (Santiago et al. 2015).

### **2.2.2 Variable Dependiente**

#### **Interpretación de lenguaje de señas:**

Es la acción que ejerce el intermediario en lenguaje de señas y lenguaje oral, el cual tiene la capacidad de interpretar de forma eficaz los mensajes hacia otra lengua equivalente (Méndez, 2010).

## Operacionalización de Variables

a) **Variable Independiente:** Aplicación Móvil

Indicador	Índice
Presencia - Ausencia	No, Sí

b) **Variable Dependiente:** Interpretación de lenguaje de señas

Tabla 2: Operacionalización de la variable dependiente

DIMENSIÓN	INDICADOR	UNIDAD DE MEDIDA	FÓRMULA	UNIDAD DE OBSERVACIÓN
<b>Tiempo</b>	Tiempo promedio de comunicación con las personas sordas (TPCPS)	Minutos	$TPCPS = \frac{\sum_{i=1}^n (TPC)}{n}$ <p>TPC = Tiempo de Comunicación n = Número de personas sordas inscritas en la asociación</p>	Registro Manual  Observación Directa
<b>Rendimiento</b>	Exactitud (X)	Porcentual (%)	$X = \frac{PR + NR}{PT}$ <p>PR = Positivos Reales NR = Negativos Reales PT = Predicciones Totales</p>	Registro Manual
	Precisión (P)	Porcentual (%)	$P = \frac{PR}{PR + FP}$ <p>PR = Positivos Reales FP = Falso Positivo</p>	Registro Manual
	Sensibilidad (S)	Porcentual (%)	$S = \frac{PR}{PR + FN}$ <p>PR = Positivos Reales</p>	Registro Manual

			FN = Falso Negativo	
	Especificidad(E)	Porcentual (%)	$E = \frac{NR}{NR + FP}$ NR = Negativos Reales FP = Falso Positivo	Registro Manual
	Factor Rendimiento (R)	Porcentual (%)	$R = \frac{2(P * S)}{P + S}$ P = Precisión S = Sensibilidad	Registro Manual
<b>Costo</b>	Costo de contratación de intérpretes de lenguaje de señas (CCILS)	Minutos	CCILS $= \frac{\sum_{i=1}^n (CILS)}{n}$ CILS = Costo de Intérpretes de Lenguaje de Señas n = Número de personas sordas inscritas en la asociación	Registro Manual

Fuente: Elaboración Propia

## Matriz de Operacionalización de las variables

### Matriz de Operacionalización de las variables

Tabla 3: Matriz de operacionalización de las variables de la investigación

Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicador	Instrumento	Escala de Medición
INTERPRETACIÓN DE LENGUAJE DE SEÑAS	Se define como la acción que ejerce el intermediario en la o las lenguas de señas y orales del contexto con la capacidad de interpretar el mensaje del emisor hacia una lengua igual al del receptor de forma eficaz (Méndez Mendoza, Rocío).	La variable interpretación de lenguaje de señas se medirá en base a la métrica F1 Score donde se evaluará los factores que permitirán mejorar los problemas específicos de la asociación de sordos región lima.	Tiempo	Tiempo Promedio de comunicación en las personas sordas $TPCPS = \frac{\sum_{i=1}^n (TPC)}{n}$	Medición del Tiempo/ Cronómetro	De Razón
			Rendimiento	Exactitud $X = \frac{PR + NR}{PT}$	Métrica F1 Score	De Razón
				Precisión $P = \frac{PR}{PR + FP}$	Métrica F1 Score	De Razón
				Sensibilidad $S = \frac{PR}{PR + FN}$	Métrica F1 Score	De Razón
				Especificidad $E = \frac{NR}{NR + FP}$	Métrica F1 Score	De Razón
				Factor Rendimiento $R = \frac{2(P * S)}{P + S}$	Métrica F1 Score	De Razón
			Costo	Costo de contratación de intérpretes de lenguaje de señas para las personas sordas $CCILS = \frac{\sum_{i=1}^n (CILS)}{n}$	Medición del Tiempo/ Cronómetro	De Razón

## 2.3 Población y Muestra

### 2.3.1 Población

Personas con discapacidad auditiva que estén inscritas y asistan actualmente en la “Asociación de Sordos de la Región Lima”.

Tabla 4: Total de población inscrita

POBLACIÓN	N°
<b>Personas sordas inscritas en la Asociación y asisten</b>	<b>8</b>
<b>Total</b>	<b>8</b>

Fuente: Elaboración Propia

### 2.3.2 Muestra

Se aplicará el muestreo no probabilístico por conveniencia para este proyecto ya que: La muestra puede ser seleccionada de casos accesibles que sean incluidos por accesibilidad, conveniencia y proximidad de los sujetos de prueba para el investigador (Otzen & Manterola, 2017).

En la presente investigación serán 8 personas con discapacidad auditiva severa inscritas en la Asociación de Sordos de la Región Lima.

#### 2.3.2.1 Criterios de inclusión

Personas con discapacidad auditiva que pertenezcan a la “Asociación de Sordos Región Lima”.

#### 2.3.2.2 Criterios de exclusión

Personas que no pertenezcan a la “Asociación de Sordos Región Lima”

Personas inscritas en la “Asociación de Sordos Región Lima” y no asisten actualmente.

Personas sordas que no dispongan de un smartphone.

## Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Tabla 5: Técnicas e Instrumentos de Recolección de Datos

TECNICA	INSTRUMENTO	FUENTE	INFORMANTE
<b>Entrevista</b>	Guía de entrevista	Presidente de la Asociación de Sordos de la Región Lima	Presidenta
<b>Cuestionario</b>	Cuestionario	Personas con discapacidad auditiva inscritas en la Asociación de Sordos de la Región Lima	Personas con Discapacidad auditiva
<b>Observación</b>	Ficha de observación	Personas con discapacidad auditiva inscritas en la Asociación de Sordos de la Región Lima y personas oyentes	Personas sordas y oyentes
<b>Resumen</b>	Fichas Bibliográficas	Libros, Tesis, Internet	Conceptos, Antecedentes, Información Web

Fuente: Elaboración Propia

- **Guía de Entrevista:**

Se utilizó con la finalidad de recolectar información para la situación problemática.

- **Cuestionario:**

Se utilizó para recolectar información de problemas específicos de las personas sordas.

- **Ficha de Observación:**

Se utilizó para obtener datos de los involucrados en el proceso de interpretación de lenguaje de señas referentes recolección de información previa a la implementación de la solución del caso de estudio.

## Métodos de análisis de datos

Se extrae la data desde el aplicativo móvil que luego será procesada en Excel con apoyo de una herramienta llamada Matriz de Confusión o cálculo del F1 Score. Además, se diseñaron tablas de datos obtenidos de los instrumentos de medición con las cuales se generan gráficos circulares, histogramas de frecuencia con la finalidad de exponer los resultados de la investigación propuesta.

## **Aspectos éticos**

Este proyecto de investigación se realiza bajo la necesidad de brindar una solución informática a una asociación de sordos sin fines de lucro, la cual da soporte para los sordos y su interpretación de lenguaje de señas mediante la dactilología. La información procesada es en base a datos obtenidos por el autor, los cuales no fueron modificados de ninguna manera, solo utilizados para fines de esta investigación.

# **III. RESULTADOS**

En este apartado se exponen resultados obtenidos de la investigación haciendo uso de los indicadores “porcentaje de tiempo empleado en la comunicación con las personas sordas” y “porcentaje de costo por tiempo de interpretación” además se medirá los indicadores de la dimensión “rendimiento”. Mostrando el impacto de la implementación de un aplicativo móvil que interpreta el lenguaje de señas en el proceso de comunicación con los discapacitados auditivos, y también se realiza el procesamiento de los datos obtenidos de las muestras de cada indicador (tanto para el pre-test y el post-test) con el software IBM SPSS Statistics 25. Debido a que la investigación fue pre-experimental, se utilizan datos antes de la implementación del sistema (pre-test) y después que el sistema fue implementado y puesto en marcha (post-test).

### 3.1. Análisis Descriptivo

En el estudio se empleó un aplicativo móvil para evaluar el rendimiento en la interpretación de lenguaje de señas en un ambiente controlado. De tal forma que en el Primer indicador solo se evaluará el sistema mediante la prueba F1 SCORE

Tabla N° 6: Indicadores – prueba de hipótesis

INDICADOR	TIPO
Rendimiento en el proceso de interpretación de lenguaje de señas	Cuantitativo
Tiempo promedio de comunicación en las personas sordas.	Cuantitativo
Costos de contratación de intérpretes de lenguaje de señas.	Cuantitativo

## Prueba de hipótesis para indicador 1: Cuantitativo

Rendimiento en el proceso de interpretación de lenguaje de señas

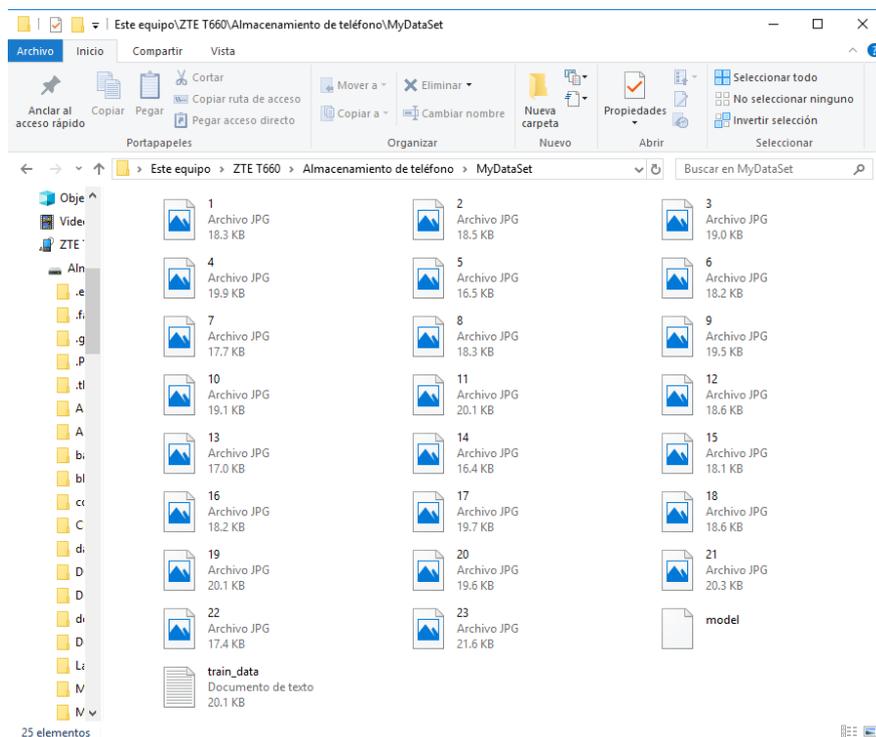
### Reconocimiento de señas de configuración simple

Mediante la librería OpenCV que cuenta con varios clasificadores entrenados en cascada, los cuales sirven para la detección de rostro, placas de auto, objetos. Pero si se desea detectar gestos como es en este caso de estudio se necesita crear nuestros propios clasificadores. Para crear los clasificadores es necesario imágenes positivas (donde se encuentra el objeto a reconocer), imágenes negativas (donde no se encuentra el objeto a reconocer) de tal modo se ha decidido crear un clasificador por cada tipo de gesto para el cual el aplicativo móvil generará archivos de registro txt (train\_data.txt) y jpg (imágenes base), a partir de ello se evaluarán: vectores, positivos, negativos, en base a clasificadores. Se realizó en entrenamiento con resolución de imagen 320x240

En general aquí se entrenan gestos de un solo acto como un Total de datos.

En esta etapa esta comprendido el Abecedario de Lenguaje de señas que se estudia en la Dactilología (no se cuenta con las letras Ñ, J, S, Z) Total 23

Figura 4 – Datos de Entrenamiento del Aplicativo Movil



## GESTO: LETRA “E”

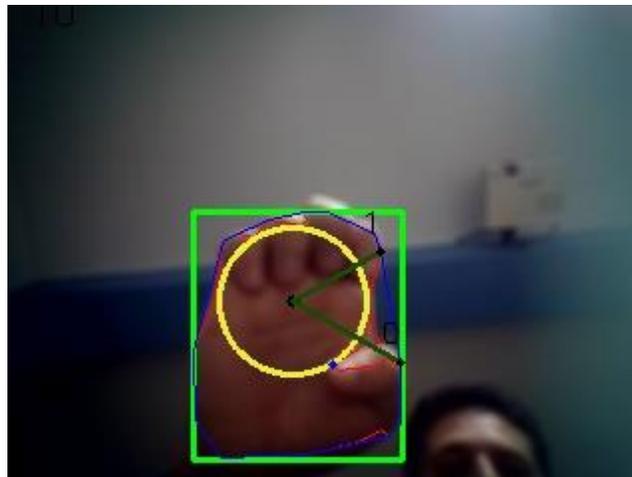
- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 21	Falsos Negativos: 6
	<b>NO – GESTO</b>	Falsos Positivos: 7	Negativos Reales: 16

Tabla 7: Matriz de Confusión para gesto letra “E” del clasificador de configuración simple



Fuente: Propia

Figura 5: Pruebas de entrenamiento del gesto letra “E” (clasificador de configuración simple)

Tabla 8: Resultados de las pruebas de rendimiento del gesto letra “E” (clasificador de configuración simple)

<b>Análisis</b>	<b>Porcentaje</b>
<b>Exactitud</b>	74 %
<b>Precisión</b>	75 %
<b>Sensibilidad</b>	77 %
<b>Especificidad</b>	69 %
<b>F1 Score</b>	75.99 %

## GESTO: LETRA “F”

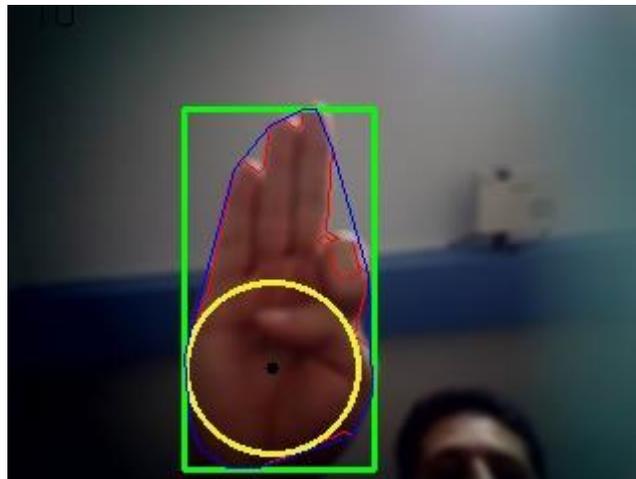
- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 25	Falsos Negativos: 9
	<b>NO – GESTO</b>	Falsos Positivos: 0	Negativos Reales: 19

Tabla 9: Matriz de Confusión para gesto letra “F” del clasificador de configuración simple



Fuente: Propia

Figura 6: Pruebas de entrenamiento del gesto letra “F” (clasificador de configuración simple)

Tabla 10: Resultados de las pruebas de rendimiento del gesto letra “F” (clasificador de configuración simple)

Análisis	Porcentaje
Exactitud	82 %
Precisión	100 %
Sensibilidad	73 %
Especificidad	100 %
F1 Score	84.39 %

## GESTO: LETRA “G”

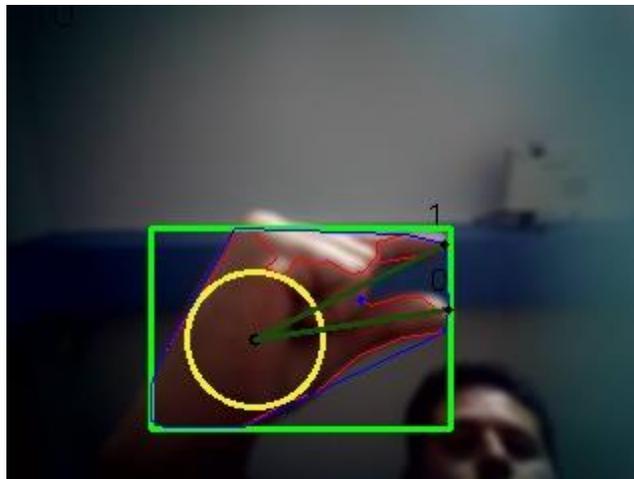
- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 24	Falsos Negativos: 5
	<b>NO – GESTO</b>	Falsos Positivos: 7	Negativos Reales: 14

Tabla 11: Matriz de Confusión para gesto letra “G” del clasificador de configuración simple



Fuente: Propia

Figura 7: Pruebas de entrenamiento del gesto letra “G” (clasificador de configuración simple)

Tabla 12: Resultados de las pruebas de rendimiento del gesto letra “G” (clasificador de configuración simple)

<b>Análisis</b>	<b>Porcentaje</b>
<b>Exactitud</b>	76 %
<b>Precisión</b>	77 %
<b>Sensibilidad</b>	82 %
<b>Especificidad</b>	66 %
<b>F1 Score</b>	79.42 %

## GESTO: LETRA “H”

- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 25	Falsos Negativos: 14
	<b>NO – GESTO</b>	Falsos Positivos: 2	Negativos Reales: 9

Tabla 13: Matriz de Confusión para gesto letra “H” del clasificador de configuración simple



Fuente: Propia

Figura 8: Pruebas de entrenamiento del gesto letra “H” (clasificador de configuración simple)

Tabla 14: Resultados de las pruebas de rendimiento del gesto letra “H” (clasificador de configuración simple)

<b>Análisis</b>	<b>Porcentaje</b>
<b>Exactitud</b>	68 %
<b>Precisión</b>	92 %
<b>Sensibilidad</b>	64 %
<b>Especificidad</b>	81 %
<b>F1 Score</b>	75.49 %

## GESTO: LETRA “I”

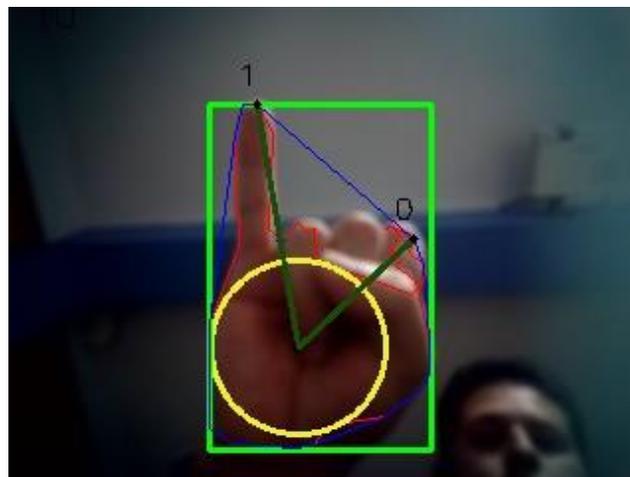
- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 21	Falsos Negativos: 5
	<b>NO – GESTO</b>	Falsos Positivos: 7	Negativos Reales: 17

Tabla 15: Matriz de Confusión para gesto letra “I” del clasificador de configuración simple



Fuente: Propia

Figura 9: Pruebas de entrenamiento del gesto letra “I” (clasificador de configuración simple)

Tabla 16: Resultados de las pruebas de rendimiento del gesto letra “I” (clasificador de configuración simple)

<b>Análisis</b>	<b>Porcentaje</b>
<b>Exactitud</b>	76 %
<b>Precisión</b>	78 %
<b>Sensibilidad</b>	83 %
<b>Especificidad</b>	70 %
<b>F1 Score</b>	80.42 %

## GESTO: LETRA “K”

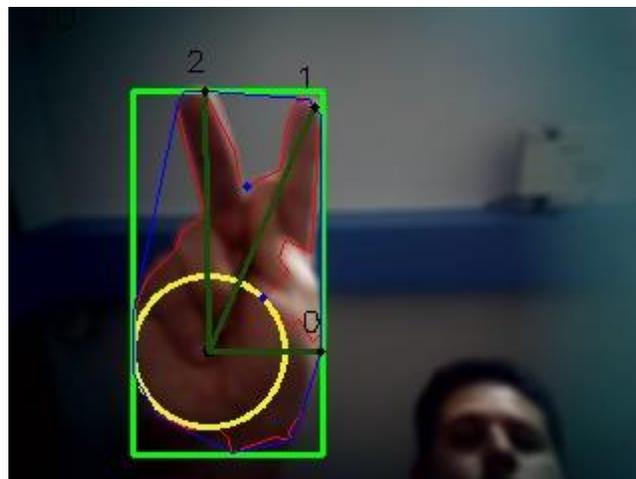
- Imágenes Negativas: 600
- Imágenes Positivas: 150
- Imágenes para Test: 50

Donde se obtienen los siguientes resultados de rendimiento:

### RESULTADOS DEL CLASIFICADOR

		<b>GESTO</b>	<b>NO – GESTO</b>
<b>INSTANCIAS REALES</b>	<b>GESTO</b>	Positivos Reales: 24	Falsos Negativos: 9
	<b>NO – GESTO</b>	Falsos Positivos: 2	Negativos Reales: 15

Tabla 17: Matriz de Confusión para gesto letra “K” del clasificador de configuración simple



Fuente: Propia

Figura 10: Pruebas de entrenamiento del gesto letra “K” (clasificador de configuración simple)

Tabla 18: Resultados de las pruebas de rendimiento del gesto letra “K” (clasificador de configuración simple)

<b>Análisis</b>	<b>Porcentaje</b>
<b>Exactitud</b>	78 %
<b>Precisión</b>	92 %
<b>Sensibilidad</b>	72 %
<b>Especificidad</b>	88 %
<b>F1 Score</b>	80.78 %

## Resultados Caso de Estudio Ambiente Controlado

Tabla 19: Estudio N° 1, Pruebas en Ambiente Controlado

LETRA	N° PRUEBAS	ACIERTO	RENDIMIENTO F1 SCORE	FALSOS POSITIVOS	FALSOS NEGATIVOS
E	50	74%	75,99%	14%	12%
F	50	82%	84,39%	0%	18%
G	50	76%	79,42%	14%	10%
H	50	68%	75,49%	4%	28%
I	50	76%	80,42%	14%	10%
K	50	78%	80,78%	4%	18%

### Prueba de hipótesis para indicador 2: Cuantitativo

Tiempo promedio de comunicación de las personas sordas

#### Definición de Variables

**TPCPSa** = Tiempo promedio de comunicación de las personas sordas con el sistema actual de comunicación.

**TPCPSs** = Tiempo promedio de comunicación de las personas sordas con el sistema propuesto de comunicación.

#### a) Hipótesis estadística

**Hipótesis Ho** = El tiempo promedio de las personas sordas con el sistema actual de comunicación es menor o igual que el tiempo promedio de comunicación de las personas sordas con el sistema propuesto (minutos).

$$H_0 = TPCPSa - TPCPSs \leq 0$$

**Hipótesis Ha** = El tiempo promedio de las personas sordas con el sistema actual de comunicación es mayor que el tiempo promedio de comunicación de las personas sordas con el sistema propuesto (minutos).

$$H_a = TPCPSa - TPCPSs > 0$$

**b) Nivel de Significancia**

Se define el margen de error, confiabilidad **95%**

Usando un nivel de significancia ( $\alpha = 0.05$ ) del **5%**. Por lo tanto, el nivel de confianza ( $1 - \alpha = 0.95$ ) será del **95%**

**c) Estadígrafo de contraste**

Tomando como población a las personas sordas inscritas en la Asociación de sordos de la región Lima (ASSORELI), se calcula una población de 8 personas que es igual a la muestra. De tal modo para realizar la prueba de normalidad cualitativa ordinal por lo tanto se utilizará la prueba no paramétrica de Wilconxon.

**Resultados:** Para calcular el Tiempo promedio de comunicación de las personas sordas en la Asociación de sordos de la Región Lima, se ha estimado un universo de 8 personas sordas.

Tabla 20: Tiempo en entender a una persona sorda

N°	¿Cuánto es el tiempo que se demora en entender a una persona sorda? TPCPSa - Antes	¿Cuánto es el tiempo que se demora en entender a una persona sorda? TPCPSs – Después
1	3	1
2	5	3
3	3	3
4	3	1
5	5	2
6	4	1
7	5	2
8	5	3
<b>SUMA</b>	<b>33</b>	<b>16</b>

Tabla 21 – Frecuencia Pre y Post tiempo promedio

¿Cuánto es el tiempo que se demora en entender a una persona sorda (minutos)?		Frecuencia	%	Porcentaje acumulado
Pre-test	1 - 2	0	0	0
	2 - 4	0	0	0
	4 - 6	3	37,5	37,5
	6 - 8	1	12,5	50,0
	8 - 10	4	50,0	100,0
	Total	8	100	
Post-test	1 - 2	3	37,5	37,5
	2 - 4	2	25,0	62,5
	4 - 6	3	37,5	100
	6 - 8	0	0	100
	8 - 10	0	0	100
	Total	8	100	
Prueba de Wilcoxon				
Z	-2,588			
P	0,015			

¿Cuánto es el tiempo que se demora en entender a una persona sorda (minutos)?

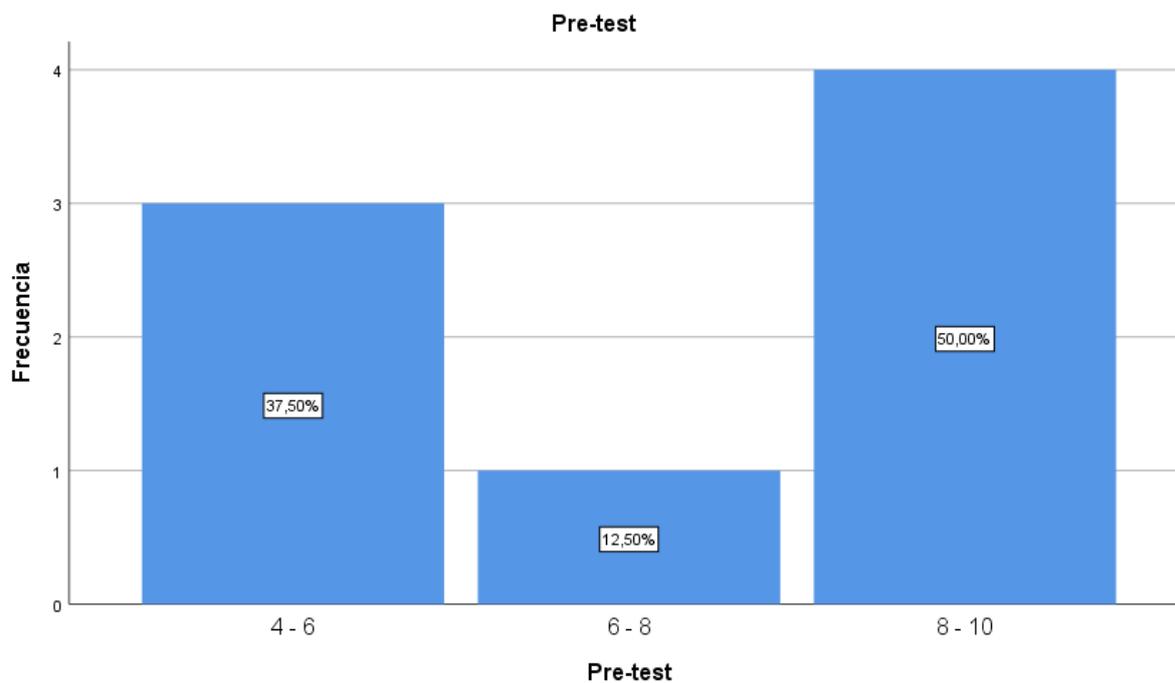
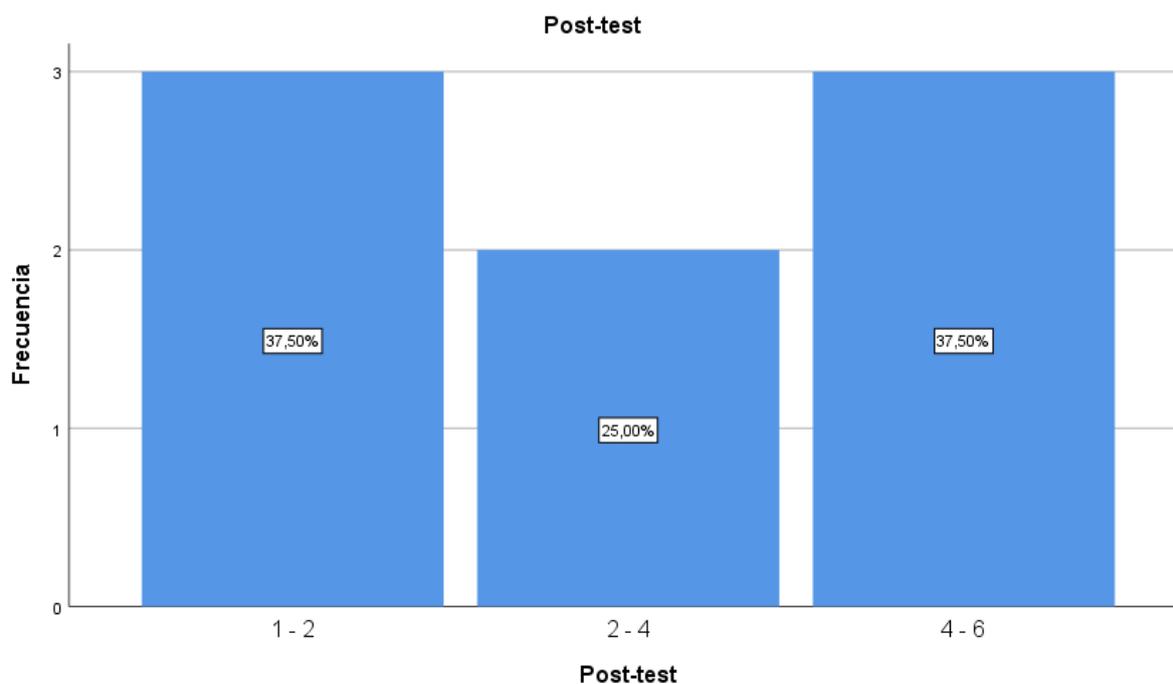


Figura 11 – Gráfica Frecuencia Pre-Test – tiempo promedio en comunicación



**Figura 12 – Gráfica Frecuencia Post-Test – tiempo promedio en comunicación**

#### d) Conclusión

En el análisis Wilconxon se muestra que  $P < 0.05$  y siendo el valor menor que 0.05 entonces se rechaza  $H_0$  y por consiguiente se acepta  $H_a$ .

Se concluye entonces que el tiempo promedio que se demora entender a una persona sorda con el sistema actual el 50,0% demora en entender a una persona sorda entre 8 – 10 minutos y con el sistema propuesto el 37,5 % demora en entender a una persona sorda entre 4 – 6 minutos, reduciendo en un 12,0% el tiempo promedio de comunicación de las personas sordas con la sociedad hablante con un nivel de error del 5% y un nivel de confianza del 95%.

#### **Prueba de hipótesis para indicador 3: Cuantitativo**

Costo de contratación de intérpretes de lenguaje de señas peruanas

##### a) Definición de Variables

**CCISa** = Costo de contratación de intérpretes de lenguaje de señas con el sistema actual de comunicación.

**CCISs** = Costo de contratación de intérpretes de lenguaje de señas con el sistema propuesto de comunicación.

**b) Hipótesis estadística**

**Hipótesis Ho** = El costo de contratación de intérpretes de lenguaje de señas con el sistema actual de comunicación es menor o igual que el Costo de contratación de intérpretes de señas con el sistema propuesto.

$$H_o = CCILS_a - CCILS_s \leq 0$$

**Hipótesis Ha** = El costo de contratación de traductores de señas con el sistema actual de comunicación es mayor que el Costo de contratación de traductores de señas con el sistema propuesto.

$$H_a = CCILS_a - CCILS_s > 0$$

**c) Nivel de Significancia**

Se define el margen de error, confiabilidad **95%**

Usando un nivel de significancia ( $\alpha = 0.05$ ) del **5%**. Por lo tanto, el nivel de confianza ( $1 - \alpha = 0.95$ ) será del **95%**

**d) Estadígrafo de contraste**

Tomando como población a las personas sordas inscritas en la Asociación de sordos de la región Lima (ASSORELI), se calcula una población de 8 personas que es igual a la muestra. De tal modo para realizar la prueba de normalidad cualitativa ordinal por lo tanto se utilizará la prueba no paramétrica de Wilcoxon.

**Resultados:** Para calcular el Costo de contratación de intérpretes de lenguaje de señas en la asociación de sordos región lima, se ha estimado un universo de 8 personas sordas.

Tabla N 22: Costo de contratación de intérpretes de señas

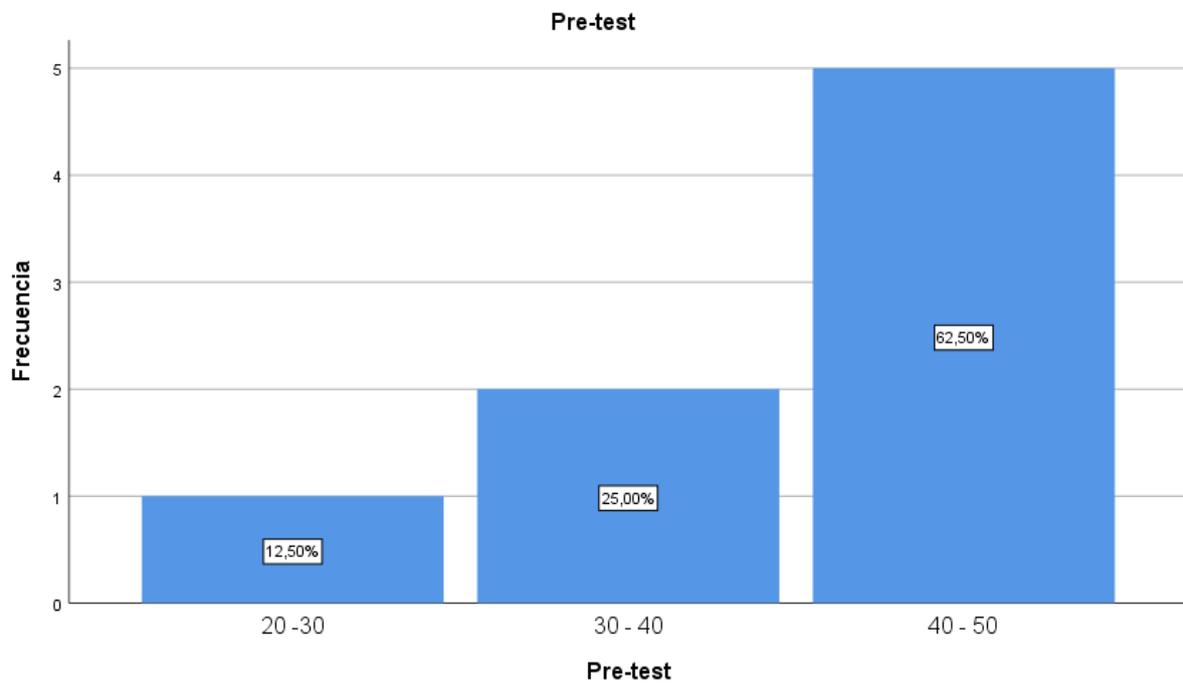
N°	¿Cuánto es el costo para contratar a un intérprete (semanal)? CCISa - Antes	¿Cuánto es el costo para contratar a un intérprete (semanal)? CCISs - Después
1	2	2
2	3	3
3	5	1
4	3	1
5	4	1
6	3	1
7	1	1
8	5	2
<b>SUMA</b>	<b>26</b>	<b>12</b>

Tabla 23: Frecuencia Pre y Post costo promedio

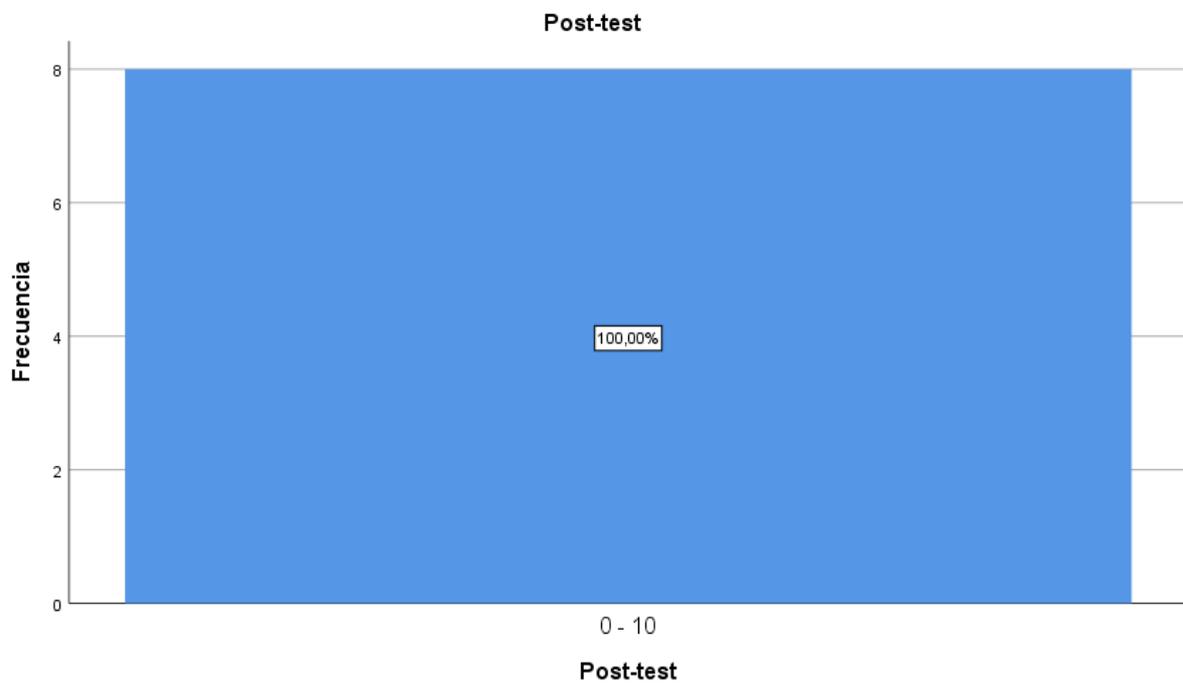
¿Cuánto es el costo para contratar a un intérprete (semanal) Soles?	Frecuencia	%	Porcentaje acumulado	
Pre-test	0 - 10	0	0	
	10 - 20	0	0	
	20 - 30	1	12,5	12,5
	30 - 40	2	25,0	37,5
	40 - 50	5	62,5	100,0
	Total	8	100	
Post-test	0 - 10	8	100	100
	10 - 20	0	0	100
	20 - 30	0	0	100
	30 - 40	0	0	100
	40 - 50	0	0	100
	Total	8	100	
Prueba de Wilcoxon				
Z	-2,588			
P	0,010			

¿Cuál es el costo para contratar a un intérprete (semanal)?

**Figura 13 – Gráfica Frecuencia Pre-Test – Costo promedio de contratación de intérprete**



**Figura 14 – Gráfica Frecuencia Post-Test – Costo promedio de contratación de intérprete**



**e) Conclusión**

En el análisis Wilcoxon se muestra que  $P < 0.05$  y siendo el valor menor que 0.05 entonces se rechaza  $H_0$  y por consiguiente se acepta  $H_a$ .

Se concluye que el costo de contratación de intérpretes de lenguaje de señas con el sistema actual 40 a 50 soles representa 62,5% de la muestra al costo promedio semanal de un intérprete y con el sistema propuesto el 100% de la muestra representa de 0 a 10 soles el costo promedio de un intérprete, reduciendo un 37,5% el costo semanal, ya que el aplicativo móvil es participe parcial o total en el proceso de interpretación del lenguaje de señas con un nivel de error del 5% y un nivel de confianza del 95%.

# **IV. DISCUSIÓN**

## Discusión de resultados

La Asociación de sordos de la Región Lima es una asociación que busca que los discapacitados auditivos tengan un sistema de comunicación apropiado para el mejor entendimiento con la sociedad, además capacita a los sordos a uso de tecnología como lo es el servicio de mensajería de los smartphones y otras herramientas de ofimática para su integración con la sociedad moderna, por consiguiente para poder tener una mejor comunicación se necesita tener un aplicativo móvil que cumpla con parte del proceso de interpretación del lenguaje de señas, como es en este caso el detectar el alfabeto dactilológico ya que es requerimiento básico para poder comunicarse en el lenguaje de señas en la nación y en el mundo.

**Tabla N° 24** Cuadro Comparativo, después de aplicar la solución.

INDICADOR	ANTES	CON LA SOLUCION	DIFERENCIA
Tiempo para entender a una persona sorda	50%	37,5%	12%
Costo de contratación de intérpretes	100%	62,5%	37,5%

**Fuente:** Elaboración Propia

Como se puede contrastar en el Grafico – Cuadro Comparativo, se nota la diferencia después de aplicar la solución, de donde se obtuvo los siguientes resultados:

- El tiempo para entender a una persona sorda, disminuye a un 12% (4 a 6 minutos) con respecto a la forma empírica de entender a la persona sorda (8 a 10 minutos).
- Se redujo costos de contratación de intérpretes ya que un 100% de la muestra ahora paga entre 0 a 10 soles (valor representativo) esto equivale a una diferencia de 37,5% (11 a 39 soles) en la aplicación total o parcial del aplicativo móvil.

Esta investigación obtuvo resultados parecidos al trabajo realizado por Vílchez, K. (2015), Donde se concluyó de que se redujo en un 2.4% el tiempo promedio la comunicación en las personas sordas y los costos de contratación de traductores de señas(semanal) con el sistema actual el 27.6% de una población de 123 personas paga entre 20 – 30 nuevos soles, reduciendo el costo de contratación de traductores de señas.

# **V.CONCLUSIÓN**

**Primera:** Se ha determinado que el rendimiento de la interpretación del lenguaje de señas del aplicativo móvil en un ambiente controlado fue: Detección de la letra “E” 74% de acierto, letra “F” 82% de acierto, letra “G” 76% de acierto, letra “H” 68% de acierto, letra “I” 76% de acierto, letra “K” 78% de acierto. Y el rendimiento promedio obtenido es de 79.41%, con estos resultados se demuestra que el aplicativo móvil puede detectar el alfabeto dactilológico de señas estáticas realizadas en un ambiente controlado.

**Segunda:** Se ha determinado que el tiempo promedio que se demora comprender a una persona sorda es entre 8 a 10 minutos que representa 50,0% en promedio de la muestra y con el aplicativo móvil el tiempo se reduce de 4 a 6 minutos que representa 37,5%, esto es equivalente a un 12% de reducción del tiempo en entender a una persona sorda. Y con esto se está demostrando que con un nivel de confianza del 95% se puede acelerar el proceso de entendimiento del lenguaje dactilológico de una persona sorda.

**Tercera:** Se ha determinado que el costo promedio semanal de un intérprete que le cuesta prestar sus servicios a las personas sordas ha pasado de un intervalo de 40 - 50 soles el 62,5% en promedio de la muestra a 0 - 10 soles el 100% de la muestra mediante la intervención total o parcial del aplicativo móvil en el proceso de interpretar el lenguaje dactilológico de las personas sordas. Y con estos resultados se demuestra que con un nivel de confianza del 95% hay una mejora en la reducción de costos de contratación de intérpretes de lenguaje de señas.

# **VI. RECOMENDACIONES**

### Recomendaciones para futuras investigaciones:

- Mejorar las tecnologías móviles utilizadas combinándolas con inteligencia artificial para completar el objetivo de interpretar el lenguaje natural del lenguaje de señas.
- Extender la población y la muestra en la investigación para la recolección de información acerca del desempeño de este tipo de tecnologías.
- Extender la investigación mediante la compatibilidad del sistema móvil para que sea ejecutado en otros sistemas operativos móviles, web y de escritorio.
- Aumentar las variables de estudio para medir el impacto de la aplicación móvil frente a la comunicación real del sordo con la sociedad.
- Implementar estas tecnologías en las instituciones del estado para brindar un servicio incluyente a las personas sordas.

## **VII. REFERENCIAS**

- Avilés, E. (14 de julio de 2011). *Estrategia de Desarrollo de Aplicaciones Móviles*. Recuperado el 7 de abril 2018 de: <https://es.slideshare.net/slashmobility/estrategia-y-desarrollos-de-aplicaciones-moviles-8596799>
- Burad, V. (2009). La interpretación del par de lenguas de señas – cultura sorda, lengua hablada – cultura oyente. *Brevísima aproximación a algunas conceptualizaciones generales*. Recuperado el 1 de junio 2018 de: [http://www.cultura-sorda.org/wp-content/uploads/2015/03/Burad\\_Viviana\\_Interpretacion\\_par\\_LSCS\\_LHCO\\_Brevissima\\_a\\_aproximacion\\_conceptualizaciones\\_generales\\_2009.pdf](http://www.cultura-sorda.org/wp-content/uploads/2015/03/Burad_Viviana_Interpretacion_par_LSCS_LHCO_Brevissima_a_aproximacion_conceptualizaciones_generales_2009.pdf)
- Fang, B., Co, J. y Zang, M. (2017). *Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation*. (Tesis de grado, Universidad del Estado de Michigan EEUU). Recuperada de: [https://www.egr.msu.edu/~fangbiyi/papers/2017\\_SenSys\\_DeepASL.pdf](https://www.egr.msu.edu/~fangbiyi/papers/2017_SenSys_DeepASL.pdf)
- Gonzáles, C. y Yimes, J. (2016). *Sistema de Reconocimiento Gestual de Lengua de Señas Chilena mediante cámara digital*. (Tesis de grado, Universidad Católica de Valparaíso, Chile). Recuperada de: [http://opac.pucv.cl/pucv\\_txt/txt-0500/UCC0990\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-0500/UCC0990_01.pdf)
- Gonzáles, R. (2015). *¿Existe la traducción de la lengua de signos?* Recuperado el 1 de junio 2018 de: <http://funcasor.org/wp-content/uploads/2015/07/R-3-traduccion-lengua-de-signos-.pdf>
- Grupo de Accesibilidad Audiovisual del CERMI Estatal. (2006). *Accesibilidad de la Televisión Digital para las Personas con Discapacidad*. (1ra ed.). España.
- Guerrero, J. y Pérez, W. (2014). *Sistema traductor de Lenguaje de Señas Colombiana a texto basado en dispositivo FPGA*. (Tesis de grado, Universidad Nacional de Colombia). Recuperada de: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0012-73532015000100022&lng=es&nrm=iso&tlng=es](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532015000100022&lng=es&nrm=iso&tlng=es)
- Ladd, P. (2003). *Understanding Deaf Culture: In Search of Deafhood*. Clevedon: The Comwell Press.
- Méndez, R. (2010). Figura del intérprete de lengua de signos en la educación. *Temas para la Educación*, N° 8. Recuperado el 1 de junio 2018 de: <https://www.feandalucia.ccoo.es/docu/p5sd7181.pdf>
- Microsoft. (2017). Evaluación del rendimiento de un modelo en Aprendizaje automático de Azure, *Azure Machine Learning Studio*. Recuperado el 11 de junio 2018 de: <https://docs.microsoft.com/es-es/azure/machine-learning/machine-learning-evaluate-model-performance>
- Ministerio de Educación. (2015). *Lengua de Señas Peruana*. (2da ed.). Perú.
- OpenCV. (2014). *The OpenCV Reference Manual*. Recuperado el 1 de junio 2018 de: <https://docs.opencv.org/3.0-beta/opencv2refman.pdf>

- Otzen, T. & Manterola, C. (2017). *Técnicas de Muestreo sobre una Población a Estudio*. Universidad de Tarapacá, Chile. Recuperado de: <https://scielo.conicyt.cl/pdf/ijmorphol/v35n1/art37.pdf>
- Planells Lerma, J. (2009). *Implementación del algoritmo de detección facial de Viola-Jones*. Recuperada de: [https://www.academia.edu/9503665/Implementacion\\_del\\_algoritmo\\_de\\_deteccion\\_facial\\_de\\_Viola-Jones\\_Autor\\_Joquin\\_Planells\\_Lerma\\_Director](https://www.academia.edu/9503665/Implementacion_del_algoritmo_de_deteccion_facial_de_Viola-Jones_Autor_Joquin_Planells_Lerma_Director)
- Platero, C. (2011). *Apuntes de Visión Artificial*. Recuperado el 1 de junio 2018 de: <http://www.elai.upm.es/moodle/course/view.php?id=23>
- Rodríguez, V., Grijalva, J. y Gallar, Y. (2015). *Sistema de traducción simultánea de lenguaje de señas a voz mediante una interfaz natural de usuario para personas con discapacidad*. (Tesis de grado, Universidad Internacional SEK Ecuador). Recuperada de: <http://runachayecuador.com/refcale/index.php/didascalia/article/view/493/324>
- Rosales Romero, C. (2017). *Prototipo de detección de expresiones corporales mediante visión artificial para mejorar la comunicación con los niños que tienen parálisis cerebral infantil*. (Tesis de grado, Universidad Nacional de Loja, Ecuador). Recuperada de: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/18591/1/Rosales%20Romero%20%20Cristhian%20Eduardo.pdf>
- Ruiz, W. y Villegas, S. (2014). *Soluciones adaptativas de tecnologías de información y comunicación en computadoras y dispositivos móviles para personas con discapacidad*. (Tesis de grado, Universidad Peruana de Ciencias Aplicadas). (Acceso el 14 de mayo de 2018)
- Sampieri, R. (2010). *Metodología de la Investigación*. (5ta ed.). Perú: Interamericana editores.
- Sausa, M. (diciembre, 2015). Grupo El Comercio. Perú 21. Recuperado de: <https://peru21.pe/lima/peru-hay-532-000-personas-sordas-23-interpretes-video-199711>
- Siré, S. (2016). *Interprete profesional en lengua de señas*. [Diapositivas]. Recuperado el 1 de junio 2018 de: [http://virtual.cudi.edu.mx:8080/access/content/group/da2986d3-c493-4b77-9db0-19b15b4de2f5/seminarios/sevida\\_noviembre.pdf](http://virtual.cudi.edu.mx:8080/access/content/group/da2986d3-c493-4b77-9db0-19b15b4de2f5/seminarios/sevida_noviembre.pdf)
- Stokoe, W. (1960). *Sign language structure: An outline of the visual communication system of the American deaf*, NY: Univ. of Buffalo.
- Vanrell, M. (s. f.). *Evaluación del Rendimiento*, Universidad Autónoma de Barcelona, España.
- Vílchez Sandoval, K. (2015). *Sistema Intérprete de Lenguaje Alternativo para mejorar la comunicación de las personas sordas en la Asociación de Sordos de la Libertad*. (Tesis de grado, Universidad César Vallejo). (Acceso el 7 de abril de 2018)

Vintimilla Sarmiento, G. (2014). *Desarrollo e implementación de una Aplicación que traduzca el abecedario y los números de uno al diez del lenguaje de señas a texto para ayuda de discapacitados auditivos mediante dispositivos móviles Android*. (Tesis de grado, Universidad de las Fuerzas Armadas Ecuador). Recuperada de: <https://repositorio.espe.edu.ec/bitstream/21000/9673/1/AC-RED-ESPE-048054.pdf>

[en línea] (s. f.). *Segmentación de imágenes*. [Diapositivas]. Recuperado el 1 de junio 2018 de: <http://alojamientos.us.es/gtocom/pid/tema4.pdf>

# **ANEXOS**

## ANEXO 1: MATRIZ DE CONSISTENCIA

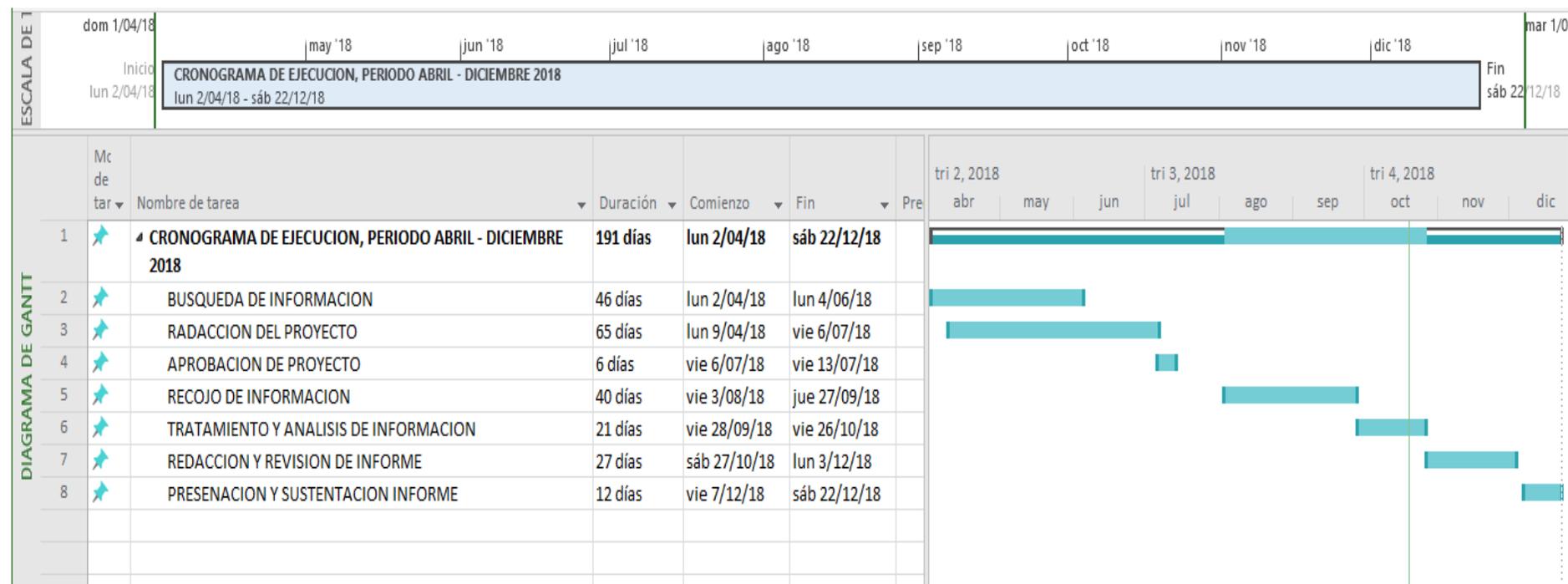
Tabla 25

*Matriz de consistencia*

PROBLEMAS	OBJETIVOS	HIPOTESIS	VARIABLE	DIMENSIONES	INDICADORES
General	General	General	INTERPRETACIÓN DE LENGUAJE DE SEÑAS	TIEMPO	Tiempo promedio de comunicación de las personas sordas
¿De qué manera la implementación de una aplicación móvil influirá en la interpretación del lenguaje de señas peruanas de los discapacitados auditivos de la asociación de sordos región Lima?	Desarrollar la aplicación móvil de interpretación del lenguaje de señas para los discapacitados auditivos de la asociación de sordos región Lima.	La implementación de una aplicación móvil mejorará significativamente la interpretación del lenguaje de señas peruanas de discapacitados auditivos de la asociación de sordos región Lima.		RENDIMIENTO	Exactitud
Específicos	Específicos	Específicos		Precisión	
PE1: ¿De qué manera la implementación de una aplicación móvil reducirá el tiempo de interpretación del lenguaje de señas de las personas con discapacidad auditiva?	OE1: Reducir el tiempo promedio de interpretación del lenguaje de señas de las personas con discapacidad auditiva.	HE1: La implementación de una aplicación móvil mejorará significativamente el tiempo promedio de interpretación de lenguaje de señas de las personas sordas.		Sensibilidad	
PE2: ¿De qué manera el aplicativo móvil mejorara el rendimiento en el proceso de interpretación del lenguaje de señas?	OE2: Mejorar el rendimiento en el proceso de interpretación de lenguaje de señas.	HE2: La implementación de una aplicación móvil mejorará el rendimiento en el proceso de interpretación del lenguaje de señas.		Especificidad	
				Factor Rendimiento	
PE3: ¿De qué manera la implementación de una aplicación móvil reducirá el costo de contratación de intérpretes de lenguaje de señas para las personas con discapacidad auditiva?	OE3: Reducir el costo promedio de contratación de intérpretes de lenguaje de señas para las personas sordas.	HE3: La implementación de una aplicación móvil reducirá el costo promedio de contratación de intérpretes de lenguaje de señas para las personas sordas.		COSTO	Costo de contratación de intérpretes de lenguaje de señas para las personas sordas

## ANEXO 2: CRONOGRAMA

Figura 15: Cronograma de ejecución de desarrollo del proyecto de investigación



### **ANEXO 3: INSTRUMENTO DE RECOLECCIÓN DE DATOS**

**Cuestionario dirigido a la presidenta de la Asociación de Sordos de la Región Lima.**

**COORDINADORA DE LA “ASOCIACIÓN DE SORDOS REGIÓN LIMA”**

1. ¿Cómo se fundó la “ASOCIACIÓN DE SORDOS REGIÓN LIMA”?
2. ¿Cuándo fue fundada la “ASOCIACIÓN DE SORDOS REGIÓN LIMA”?
3. ¿Qué actividades realizan dentro de la “ASOCIACIÓN DE SORDOS REGIÓN LIMA”?
4. ¿La “ASOCIACIÓN DE SORDOS REGIÓN LIMA” cuenta con ayuda del estado o privada?
5. ¿La “ASOCIACIÓN DE SORDOS REGIÓN LIMA” cuenta con algún sistema informático de ayuda para el asociado?
6. ¿Qué grado de educación tienen las personas inscritas en la “ASOCIACIÓN DE SORDOS REGIÓN LIMA”?
7. ¿Es fácil su relación de las personas con discapacidad auditiva en la “ASOCIACIÓN DE SORDOS REGIÓN LIMA”?
8. ¿Cuál es su opinión sobre la Ley de inserción laboral para personas con discapacidades?

## ANEXO 4: INSTRUMENTO DE MEDICIÓN DE LA VARIABLE DEPENDIENTE

### Entrenamiento de los Clasificadores en base a la Matriz de Confusión (F1 Score):

Mediante la librería OpenCV que cuenta con varios clasificadores entrenados en cascada, los cuales sirven para la detección de rostro, placas de auto, objetos. Pero si se desea detectar gestos como es en este caso de estudio se necesita crear nuestros propios clasificadores. Para crear los clasificadores es necesario imágenes positivas (donde se encuentra el objeto a reconocer), imágenes negativas (donde no se encuentra el objeto a reconocer) de tal modo se ha decidido crear un clasificador por cada tipo de gesto para el cual el aplicativo móvil generará archivos de registro txt(train\_data.txt) y jpg (imágenes base), a partir de ello se evaluarán: vectores, positivos, negativos, en base a clasificadores.

### Reconocimiento de señas de configuración simple

En general aquí se entrenan gestos de un solo acto como un Total de datos.

Ejm: el Abecedario de Lenguaje de señas que se estudia en la Dactilología (no se cuenta con las letras LL, Ñ, J y Z) Total 24

#### GESTO: LETRA “A”

- Imágenes Positivas: n°
- Imágenes Negativas: n°
- Imágenes para Test: n°

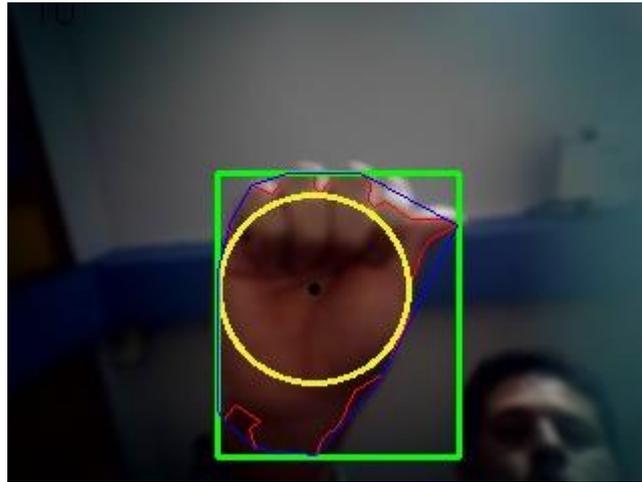
Donde se obtienen los siguientes resultados de rendimiento:

#### RESULTADOS DEL CLASIFICADOR

		GESTO	NO – GESTO
INSTANCIAS REALES	GESTO	Positivos Reales: n°	Falsos Negativos: n°
	NO – GESTO	Falsos Positivos: n°	Negativos Reales: n°

Tabla 26: Matriz de Confusión para gesto letra “A” del clasificador de configuración simple

Figura 16: Pruebas de entrenamiento del gesto letra “A” (clasificador de configuración simple)



Fuente: Propia

Tabla 27: Resultados de las pruebas de rendimiento del gesto letra “A” (clasificador de configuración simple)

Análisis	Porcentaje
Exactitud	n %
Precisión	n %
Sensibilidad	n %
Especificidad	n %
F1 Score	n %

## ANEXO 5: ASPECTOS ADMINISTRATIVOS

### Recurso y Presupuesto

Tabla 28 *Recursos y presupuesto*

Descripción	Cantidad	Precio Unitario	Subtotal
PC	1 ud.	S/ 1,500.00	S/ 1,500.00
Memoria flash	1 ud.	S/ 25.00	S/ 25.00
Lapicero	4 ud.	S/ 2.00	S/ 8.00
Bloc	1 ud.	S/ 5.00	S/ 5.00
CD-ROM	2 ud.	S/ 1.00	S/ 2.00
Internet	4 meses	S/ 66.00	S/ 264.00
Electricidad	4 meses	S/ 80.00	S/ 320.00
Matrícula Lenguaje de Señas	1 ud.	S/ 50.00	S/ 50.00
Curso de Lenguaje de Señas Nivel Básico	2 meses	S/ 120.00	S/ 240.00
Curso de Lenguaje de Señas Nivel Intermedio	2 meses	S/ 150.00	S/ 320.00
Transporte	--	--	S/ 100.00
Impresiones	8 ud.	S/ 7.00	S/ 56.00
Anillados	8 ud.	S/ 3.00	S/ 24.00
Empastado	2 ud.	S/ 10.00	S/ 20.00
Investigador	--	--	--
			S/ 2,934.00

Fuente: Elaboración propia

### Financiamiento

La Asociación de Sordos de la Región Lima, no cuenta con presupuesto para la inversión de proyectos de investigación, por lo tanto, el financiamiento del proyecto de tesis se realizará con recursos propios, estará a cargo del Tesista.

## **ANEXO 6: ASPECTOS TECNICOS**

Para el uso del programa es necesario que el terminal móvil posea las siguientes especificaciones:

- Sistema Operativo Android 4.0 en adelante
- Procesador Doble núcleo o superior
- 1GB de RAM o superior
- Cámara de 5mpx trasera y 2mpx delantera como mínimo
- Resolución de pantalla QVGA o superior
- Conexión a datos 3g o superior

Software adicional para el uso de la aplicación móvil:

OpenCV Manager 2.19 o superior se proporcionará por separado el archivo .apk ya que no se encuentra en la Play Store.

Software de desarrollo del aplicativo móvil:

- Andoid Studio 3.2
- MySQL Server
- SublimeText
- Navegador Chrome
- Office 2016





**ASOCIACIÓN DE SORDOS  
REGIÓN LIMA - ASSORELI**

Av. Venezuela Nº 28 Lima - Lima - Lima  
(espalda de la Parroquia Jesús Nazareno)  
Telf.: 564 0102 • Cel.: 965 395 701  
E-mail: info@assoreli.com

R.U.C. 20600312082

**BOLETA DE VENTA**

0001- **Nº 000321**

Señor(es): Kevin Alex López Roca  
Dirección: Jr. Los Quinuales 231 - SJL Doc. Ident.: 48270133

DÍA	MES	AÑO
27	06	18

CANT.	DESCRIPCION	P. UNIT.	IMPORTE
			120.00
	2do Mes Basico		
	Prof. Sunil		
	Miercoles 5:00 - 8:00 PM.		

PABLO R. ALBINO ARROSPIDE  
R.U.C.: 10094420666 Telf.: 331-1978  
Serie: 0001 del 0001 al 1000  
Nº Aut.: 12622656023 FA: 15/11/2016

CANCELADO  
Lima, 27 de 06 del 2018

TOTAL S/. 120.00  
**USUARIO**



**ASOCIACIÓN DE SORDOS  
REGIÓN LIMA - ASSORELI**

Av. Venezuela Nº 28 Lima - Lima - Lima  
(espalda de la Parroquia Jesús Nazareno)  
Telf.: 564 0102 • Cel.: 965 395 701  
E-mail: info@assoreli.com

R.U.C. 20600312082

**BOLETA DE VENTA**

0001- **Nº 000352**

Señor(es): Lopez Roca Kevin Alex  
Dirección: ..... Doc. Ident.: 48270133

DÍA	MES	AÑO
25	08	18

CANT.	DESCRIPCION	P. UNIT.	IMPORTE
	1er Mes Intermedio LSP		150.00
	Prof: Sunil Ponce		
	Horario: Sáb 3 - 6pm		

PABLO R. ALBINO ARROSPIDE  
R.U.C.: 10094420666 Telf.: 331-1978  
Serie: 0001 del 0001 al 1000  
Nº Aut.: 12622656023 FA: 15/11/2016

CANCELADO  
Lima, 25 de 08 del 2018

TOTAL S/. 150.00  
**USUARIO**

**Figura 18: Lugar de Pruebas: Aula de Cases de Lenguaje de Señas ASSORELI**



## **ANEXO 8: ANALISIS Y DISEÑO**

### **Analisis Requerimientos del Proyecto**

El requerimiento es necesario documentar sobre el contenido, forma o funcionalidad de un producto o servicio.

Los requerimientos son declaratorias que representan atributos, características y/o cualidades que necesita cumplir un sistema en este caso un software para dispositivo móvil ya que eso le dará valor y utilidad para el usuario. En conclusión los requerimientos muestran qué elementos y funciones son necesarias para un proyecto.

#### **Requerimientos Funcionales**

- Implementación de algoritmo para detección basado en imágenes.
- Implementación de algoritmo para el reconocimiento de patrones.
- Implementación de sistema de aprendizaje de máquina.
- Implementación de algoritmo de entrenamiento basado en coordenadas.
- Mostrar contorno, puntos y palma marcados en la cámara.
- Mostrar el número de dedos de 0 a 4.
- Guardar información de entrenamiento en la memoria interna del Smartphone

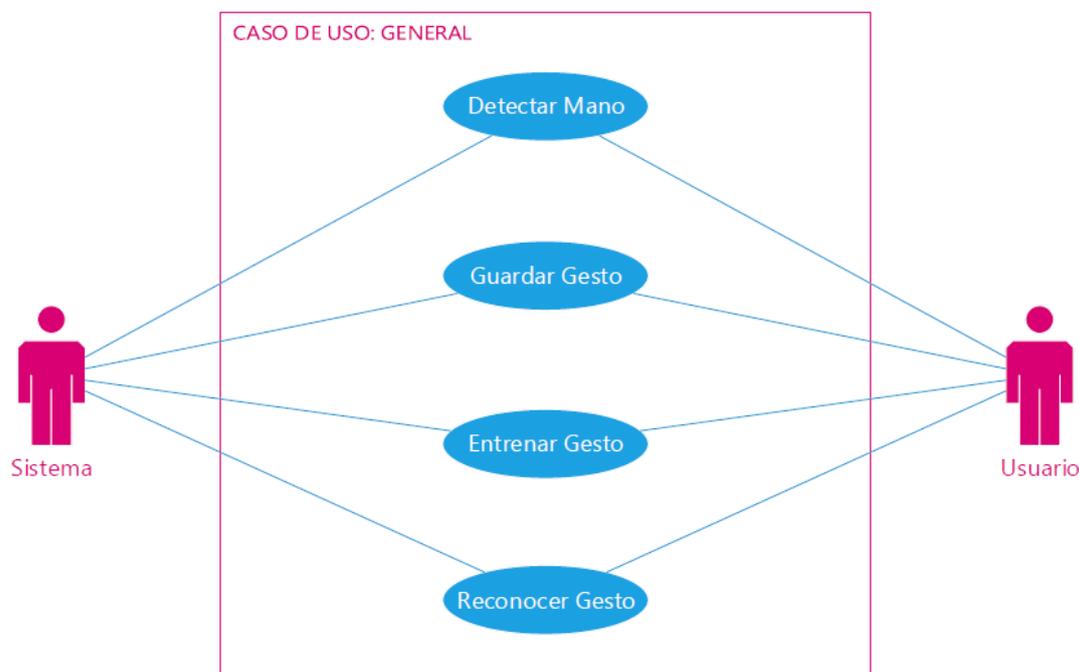
#### **Requerimientos No Funcionales**

- Diseño de una interfaz sencilla y con alta usabilidad.
- Software optimizado, alto rendimiento, procesos con bajos requerimientos de hardware.
- Documentación detallada del código y sus modificaciones

### **Diagrama de Casos de Uso**

Los casos de uso son una técnica de especificación del comportamiento de un sistema, En resumen: “el caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.”

**Figura 19 Caso de Uso General**

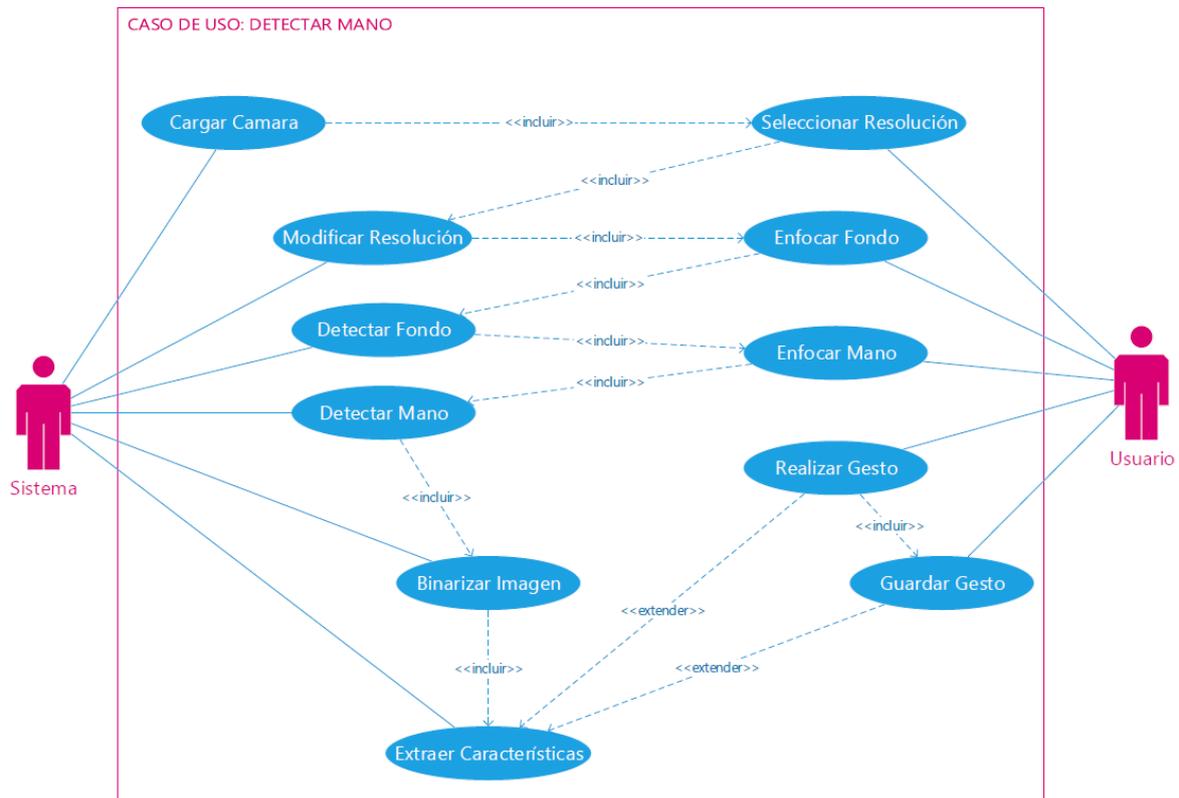


**Caso de Uso Detectar Mano**

**Tabla 29**

<b>ESPECIFICACION DE CASO DE USO: DETECTAR MANO</b>	
<b>Caso de Uso</b>	Detectar mano
<b>Actores</b>	Sistema, Usuario
<b>Objetivo</b>	Detectar la mano de manera precisa para posteriormente reconocer gestos.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar la mano del emisor de seña.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación hasta que el usuario finalice el proceso de detección.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
<ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación.</li> <li>2. El usuario inicia la cámara.</li> <li>3. El usuario selecciona resolución.</li> <li>4. El sistema configura la resolución seleccionada por el usuario.</li> <li>5. El usuario enfoca el fondo a contrastar con la mano.</li> <li>6. El sistema detecta el fondo.</li> <li>7. El usuario enfoca la mano.</li> <li>8. El sistema detecta la mano.</li> <li>9. El sistema debe binarizar la imagen.</li> <li>10. El sistema extrae características de la mano.</li> <li>11. El usuario realiza o no un gesto.</li> <li>12. El usuario guarda o no un gesto.</li> </ol>	

**Figura 20 Caso de Uso Detectar Mano**

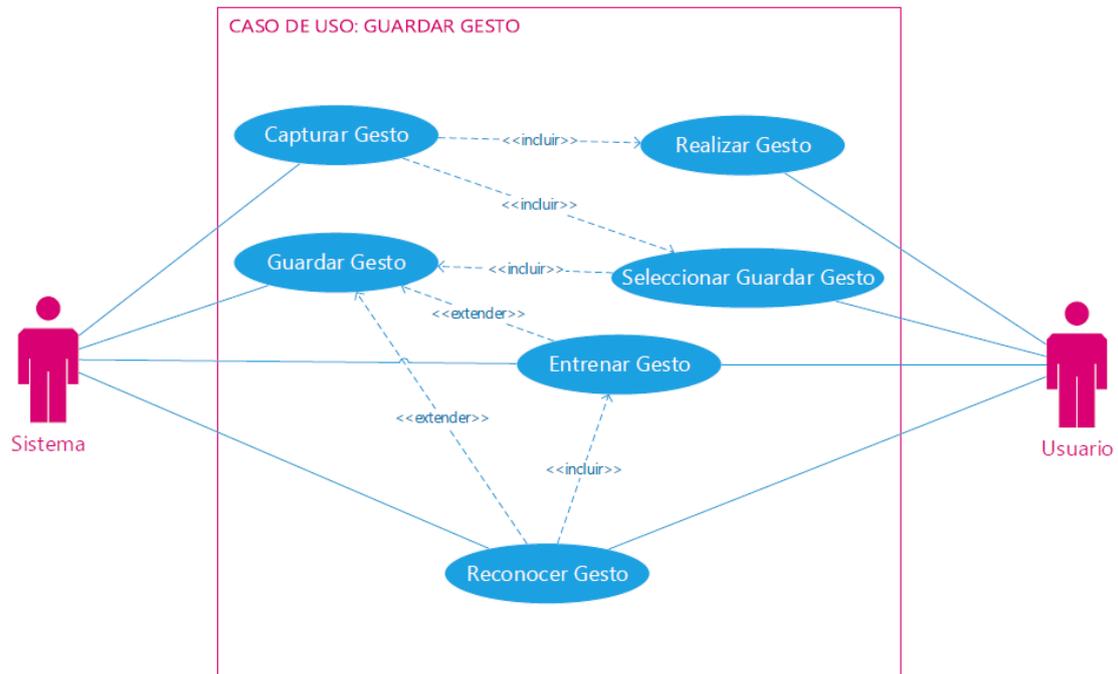


**Caso de Uso Guardar Gesto**

**Tabla 30**

<b>ESPECIFICACION DE CASO DE USO: GUARDAR GESTO</b>	
<b>Caso de Uso</b>	Guardar Gesto
<b>Actores</b>	Sistema, Usuario
<b>Objetivo</b>	Guardar gesto indicado por el usuario para su posterior reconocimiento.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación realizando un seleccionar guardar gesto, entrenando o no, para su reconocimiento.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
13. El usuario ingresa a la aplicación. 14. El sistema enfoca la mano. 15. El sistema guarda la imagen detectada de la mano.	

**Figura 21 Caso de Uso Guardar Gesto**

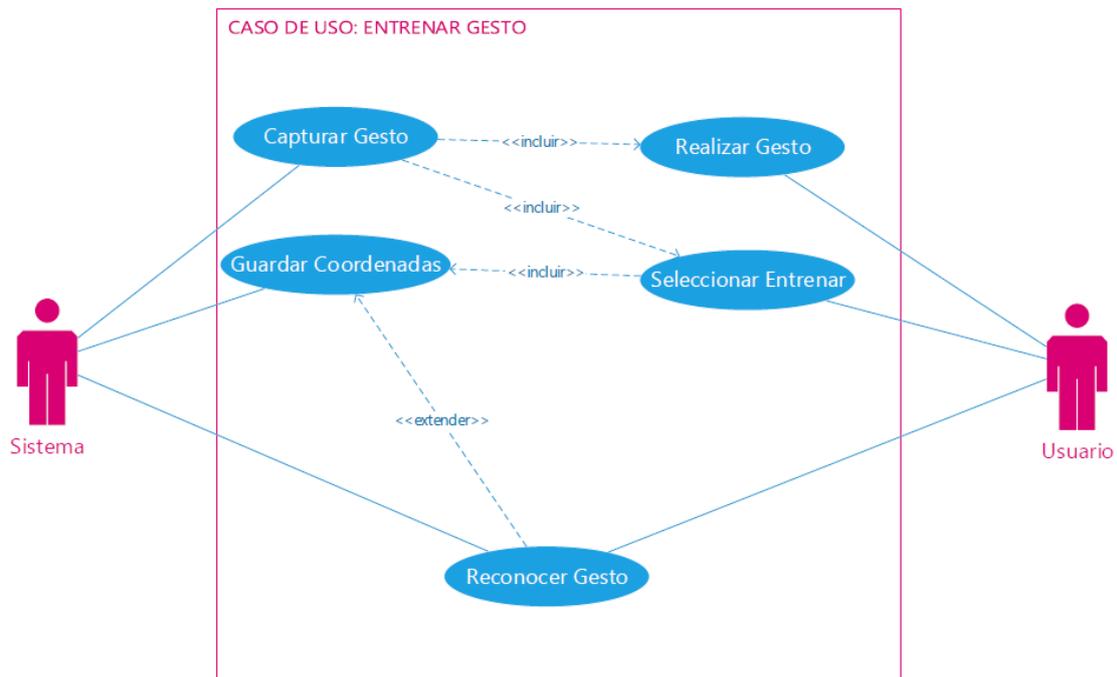


**Caso de Uso Entrenar Gesto**

**Tabla 31**

<b>ESPECIFICACION DE CASO DE USO: ENTRENAR GESTO</b>	
<b>Caso de Uso</b>	Entrenar Gesto
<b>Actores</b>	Sistema, Usuario
<b>Objetivo</b>	Entrenar cada tipo de seña para un mejor reconocimiento.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano.</li> <li>- El usuario realiza gesto.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación configurando la resolución de la cámara de OpenCV.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
16. El usuario ingresa a la aplicación. 17. El sistema enfoca la mano. 18. El sistema detecta los puntos. 19. El sistema guarda coordenadas de entrenamiento.	

**Figura 22 Caso de Uso Entrenar Gesto**

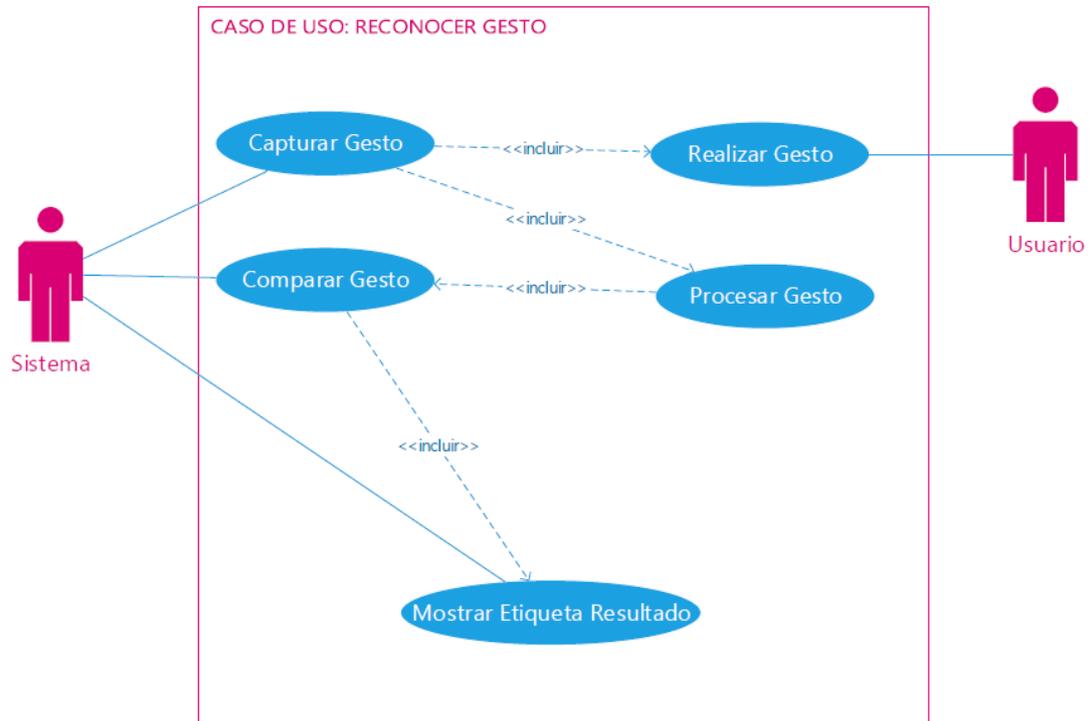


**Caso de Uso Reconocer Gesto**

**Tabla 32**

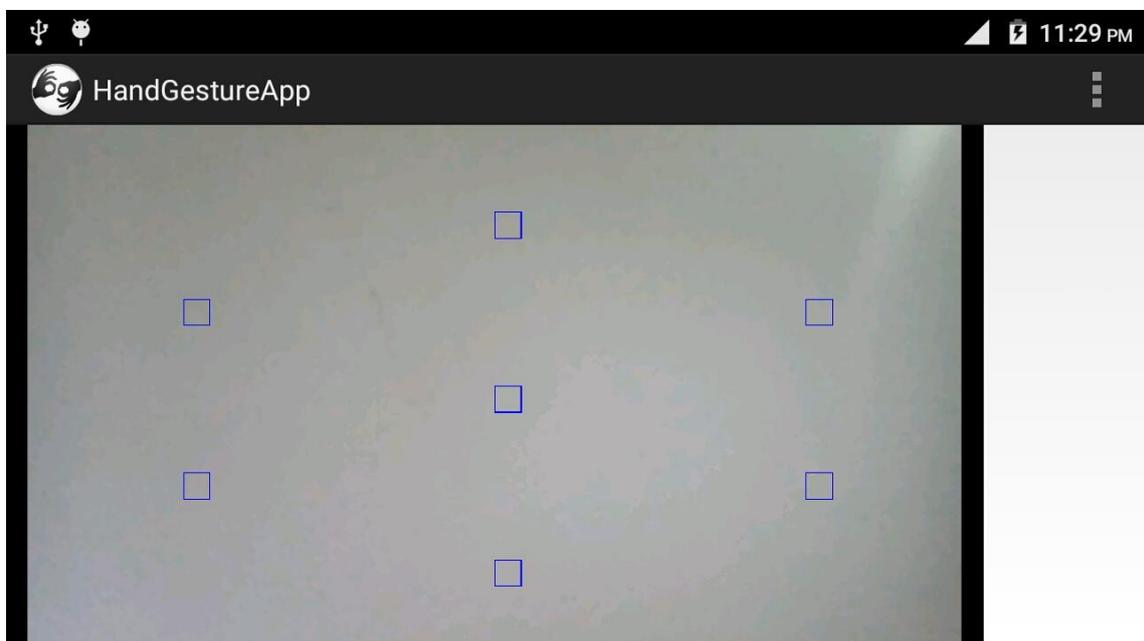
<b>ESPECIFICACION DE CASO DE USO: RECONOCER GESTO</b>	
<b>Caso de Uso</b>	Reconocer Gesto
<b>Actores</b>	Sistema, Usuario
<b>Objetivo</b>	Que el sistema sea capaz de reconocer los gestos indicados por el usuario, luego de haber realizado todo el procesamiento de imagen.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano.</li> <li>- El usuario debe mostrar la mano.</li> <li>- El usuario realiza gesto.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación configurando o no el preprocesamiento de la imagen hasta que el sistema enfoque la mano detectando separaciones y calcule la cantidad de dedos enfocados.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
20. El usuario ingresa a la aplicación. 21. El sistema enfoca la mano. 22. El sistema detecta los puntos. 23. Reconoce el gesto indicado por el usuario.	

**Figura 23 Caso de Uso Reconocer Gesto**

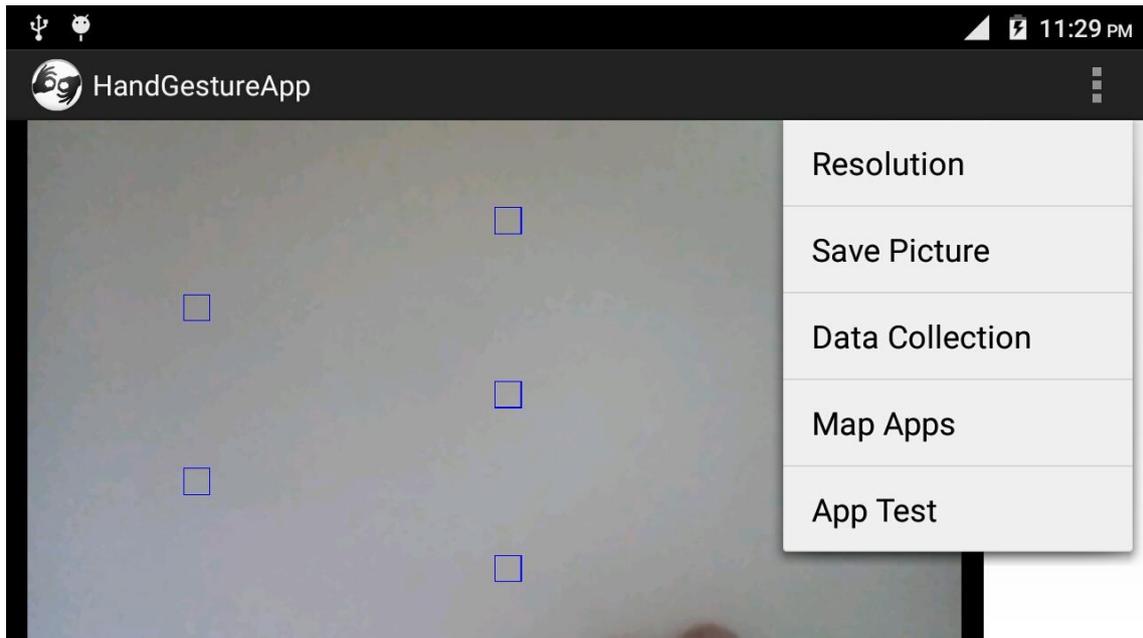


**Figura 24 Pantallas del Sistema**

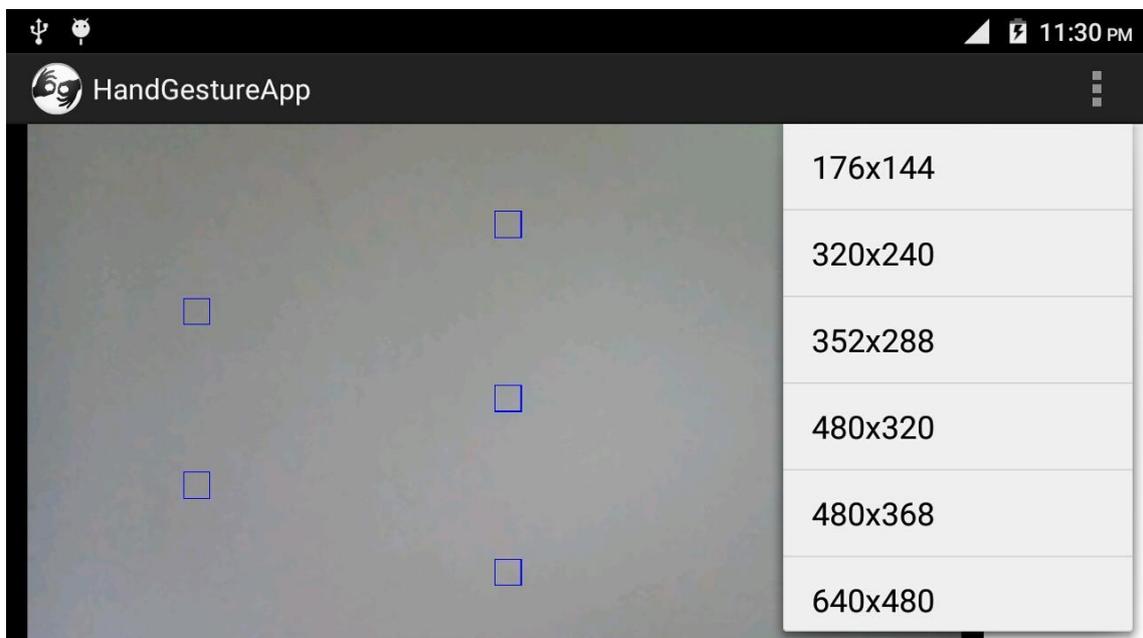
**Inicio de Sistema**



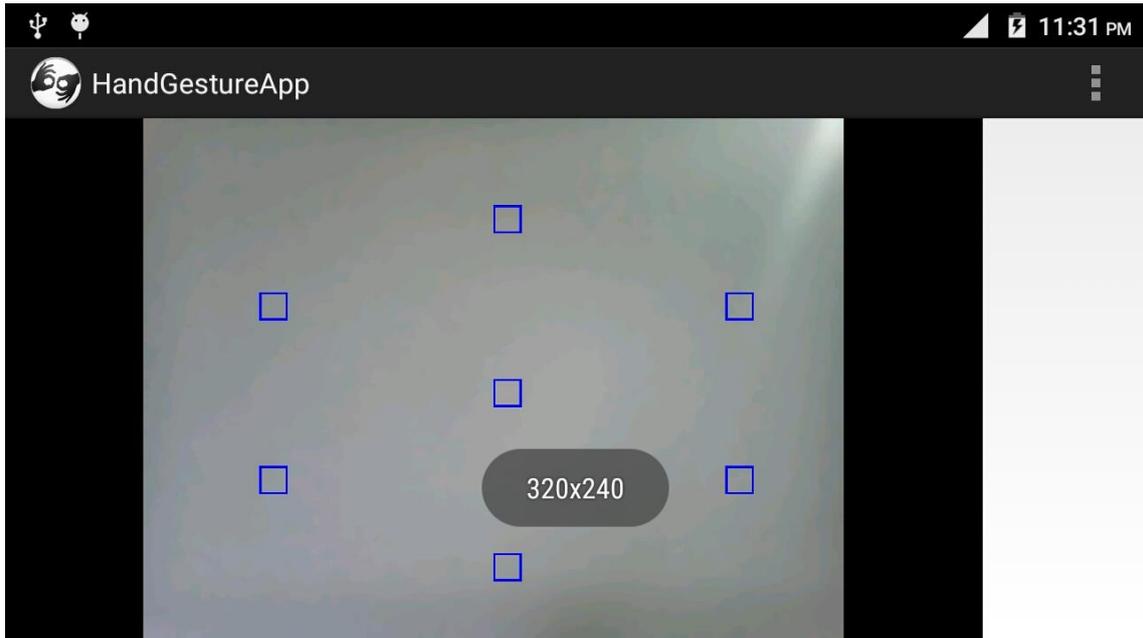
**Figura 25 Menú de Configuraciones**



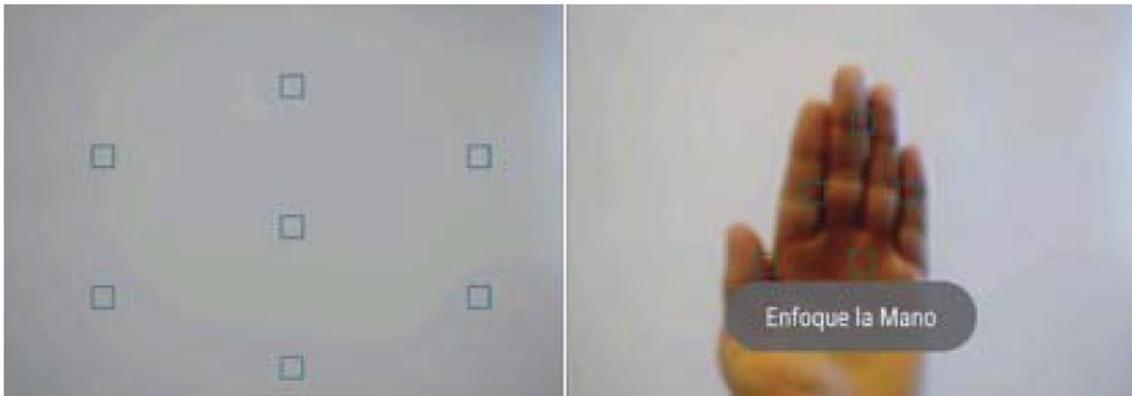
**Figura 26 Configuración de Resolución de Cámara**



**Figura 27 Resolucion 320\*240 seleccionada**



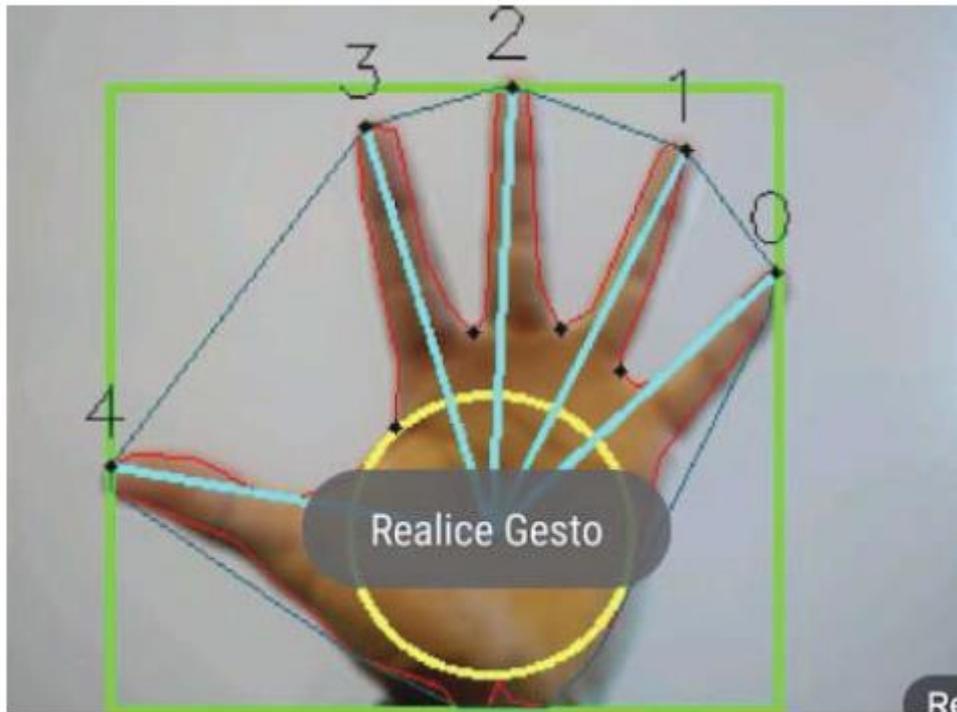
**Figura 28 Detección de Fondo y Detección de la Mano**



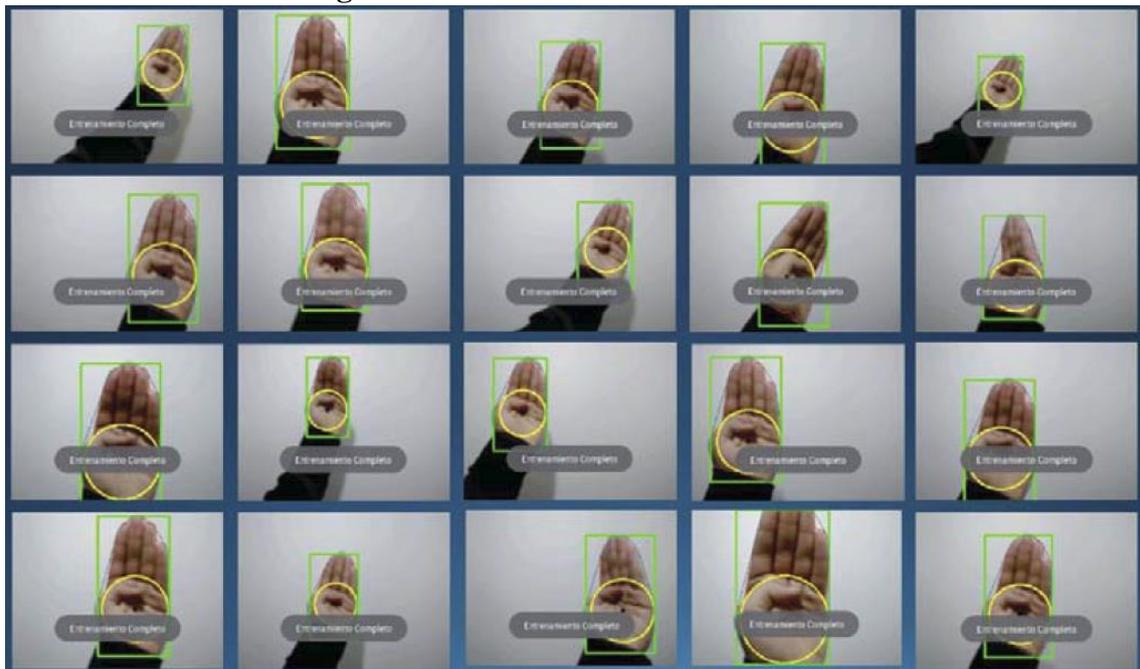
**Figura 29 Imagen Binarizada**



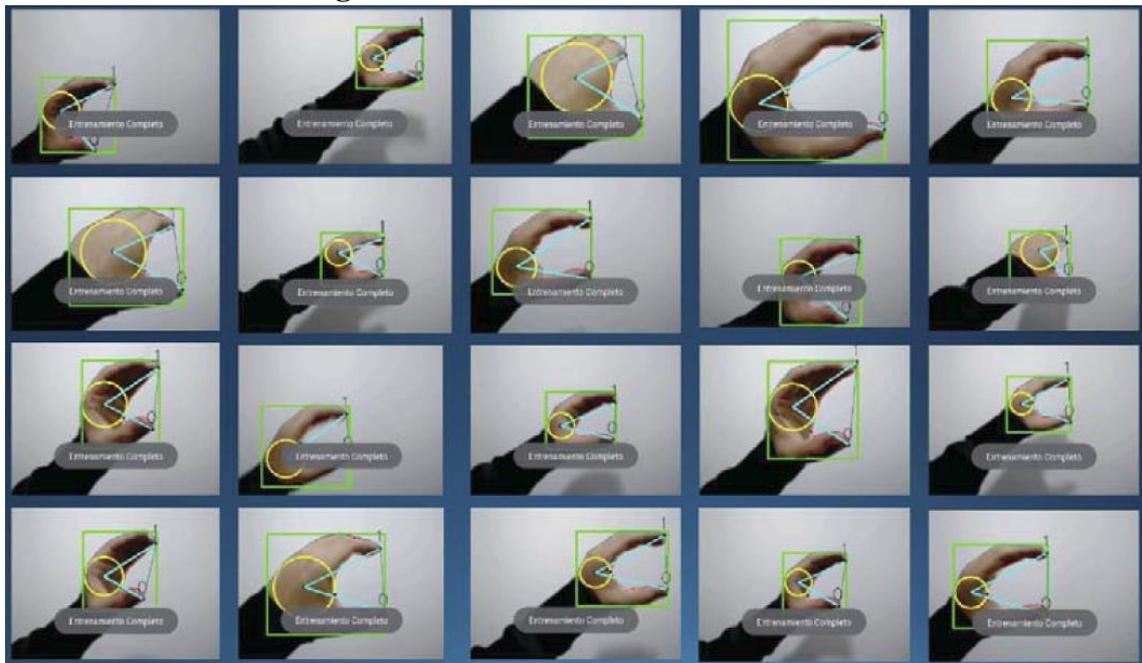
**Figura 30 Mano detectada con extracción de características, enumerada de 0 a 4**



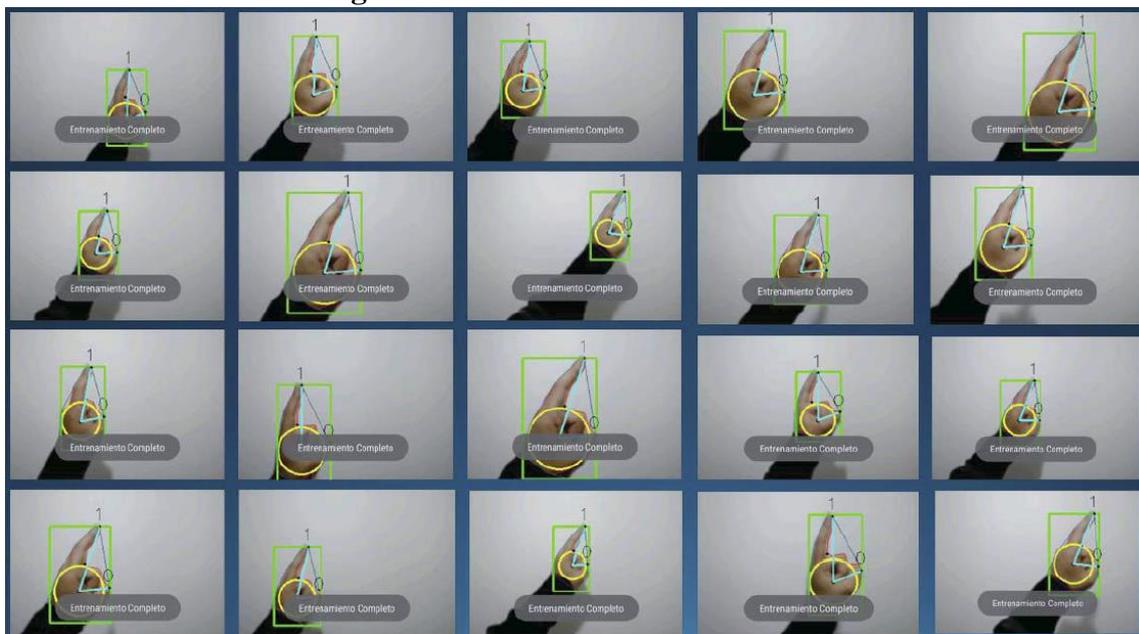
**Figura 31 Entrenamiento de la Letra B**



**Figura 32 Entrenamiento de la Letra C**



**Figura 33 Entrenamiento de la letra D**



## ANEXO 13: CODIGO FUENTE

### Codigo Fuente Java - Android:

#### MainActivity.java

```
package com.example.newdemo;

+ import java.io.File;

public class MainActivity extends Activity implements
CvCameraViewListener2 {

    //Just for debugging
    private static final String TAG = "HandGestureApp";

    //Color Space used for hand segmentation
    private static final int COLOR_SPACE = Imgproc.COLOR_RGB2Lab;

    //Number of frames collected for each gesture in the training set
    private static final int GES_FRAME_MAX= 10;

    public final Object sync = new Object();

    //Mode that presamples hand colors
    public static final int SAMPLE_MODE = 0;

    //Mode that generates binary image
    public static final int DETECTION_MODE = 1;

    //Mode that displays color image together with contours, fingertips,
    //defect points and so on.
    public static final int TRAIN_REC_MODE = 2;

    //Mode that presamples background colors
    public static final int BACKGROUND_MODE = 3;

    //Mode that is started when user clicks the 'Add Gesture' button.
    public static final int ADD_MODE = 4;

    //Mode that is started when user clicks the 'Test' button.
    public static final int TEST_MODE = 5;

    //Mode that is started when user clicks 'App Test' in the menu.
    public static final int APP_TEST_MODE = 6;

    //Mode that is started when user clicks 'Data Collection' in the
menu.
    public static final int DATA_COLLECTION_MODE = 0;

    //Mode that is started when user clicks 'Map Apps' in the menu.
    public static final int MAP_APPS_MODE = 1;

    //Number of frames used for prediction
    private static final int FRAME_BUFFER_NUM = 1;

    //Frame interval between two launching events
    private static final int APP_TEST_DELAY_NUM = 10;

    private boolean isPictureSaved = false;
    private int appTestFrameCount = 0;
```

```

private int testFrameCount = 0;
private float[][] values = new float[FRAME_BUFFER_NUM] [];
private int[][] indices = new int[FRAME_BUFFER_NUM] [];

// onActivityResult request
private static final int REQUEST_CODE = 6384;

private static final int REQUEST_SELECTED_APP = 1111;

private String diagResult = null;
private Handler mHandler = new Handler();
private static final String DATASET_NAME = "/train_data.txt";

private String storeFolderName = null;
private File storeFolder = null;
private FileWriter fw = null;

//Stores the mapping results from gesture labels to app intents
private HashMap<Integer, Intent> table = new HashMap<Integer,
Intent>();

private List<AppInfo> mlistAppInfo = null;
private MyCameraView mOpenCvCameraView;
private MenuItem[] mResolutionMenuItems;
private SubMenu mResolutionMenu;

private List<android.hardware.Camera.Size> mResolutionList;

//Initial mode is BACKGROUND_MODE to presample the colors of the hand
private int mode = BACKGROUND_MODE;

private int chooserMode = DATA_COLLECTION_MODE;

private static final int SAMPLE_NUM = 7;

private Point[][] samplePoints = null;
private double[][] avgColor = null;
private double[][] avgBackColor = null;

private double[] channelsPixel = new double[4];
private ArrayList<ArrayList<Double>> averChans = new
ArrayList<ArrayList<Double>>();

private double[][] cLower = new double[SAMPLE_NUM][3];
private double[][] cUpper = new double[SAMPLE_NUM][3];
private double[][] cBackLower = new double[SAMPLE_NUM][3];
private double[][] cBackUpper = new double[SAMPLE_NUM][3];

private Scalar lowerBound = new Scalar(0, 0, 0);
private Scalar upperBound = new Scalar(0, 0, 0);
private int squareLen;

private Mat sampleColorMat = null;
private List<Mat> sampleColorMats = null;

private Mat[] sampleMats = null ;

```

```

private Mat rgbaMat = null;

private Mat rgbMat = null;
private Mat bgrMat = null;

private Mat interMat = null;

private Mat binMat = null;
private Mat binTmpMat = null;
private Mat binTmpMat2 = null;
private Mat binTmpMat0 = null;
private Mat binTmpMat3 = null;

private Mat tmpMat = null;
private Mat backMat = null;
private Mat difMat = null;
private Mat binDifMat = null;

private Scalar          mColorsRGB[] = null;

//Stores all the information about the hand
private HandGesture hg = null;

private int imgNum;
private int gesFrameCount;
private int curLabel = 0;
private int selectedLabel = -2;
private int curMaxLabel = 0;
private int selectedMappedLabel = -2;

//Stores string representation of features to be written to
train_data.txt
private ArrayList<String> feaStrs = new ArrayList<String>();

File sdCardDir = Environment.getExternalStorageDirectory();
File sdFile = new File(sdCardDir, "AppMap.txt");

private BaseLoaderCallback mLoaderCallback = new
BaseLoaderCallback(this) {
    @Override
    public void onManagerConnected(int status) {
        switch(status) {
            case LoaderCallbackInterface.SUCCESS: {
                Log.i("Android Tutorial", "OpenCV loaded successfully");

                System.loadLibrary("HandGestureApp");

                try {
                    System.loadLibrary("signal");
                } catch (UnsatisfiedLinkError ule) {
                    Log.e(TAG, "Hey, could not load native library
signal");
                }

                mOpenCvCameraView.enableView();

                mOpenCvCameraView.setOnTouchListener(new OnTouchListener()

```

```

{
    //Called when user touch the view screen
    //Mode flow: BACKGROUND_MODE --> SAMPLE_MODE -->
    DETECTION_MODE <--> TRAIN_REC_MODE
    public boolean onTouch(View v, MotionEvent event) {
        // ... Respond to touch events
        int action =
MotionEventCompat.getActionMasked(event);

        switch(action) {
            case (MotionEvent.ACTION_DOWN) :
                Log.d(TAG, "Action was DOWN");
                String toastStr = null;
                if (mode == SAMPLE_MODE) {
                    mode = DETECTION_MODE;
                    toastStr = "Sampling Finished!";
                } else if (mode == DETECTION_MODE) {
                    mode = TRAIN_REC_MODE;

((Button) findViewById(R.id.AddBtn)).setVisibility(View.VISIBLE);

((Button) findViewById(R.id.TrainBtn)).setVisibility(View.VISIBLE);

((Button) findViewById(R.id.TestBtn)).setVisibility(View.VISIBLE);
                    toastStr = "Binary Display
Finished!";

                    preTrain();

                } else if (mode == TRAIN_REC_MODE) {
                    mode = DETECTION_MODE;

((Button) findViewById(R.id.AddBtn)).setVisibility(View.INVISIBLE);

((Button) findViewById(R.id.TrainBtn)).setVisibility(View.INVISIBLE);

((Button) findViewById(R.id.TestBtn)).setVisibility(View.INVISIBLE);

                    toastStr = "train finished!";
                } else if (mode == BACKGROUND_MODE) {
                    toastStr = "First background
sampled!";

                    rgbaMat.copyTo(backMat);
                    mode = SAMPLE_MODE;
                }

Toast.makeText(getApplicationContext(), toastStr,
Toast.LENGTH_LONG).show();

                return false;
            case (MotionEvent.ACTION_MOVE) :
                Log.d(TAG, "Action was MOVE");
                return true;
            case (MotionEvent.ACTION_UP) :
                Log.d(TAG, "Action was UP");
                return true;
            case (MotionEvent.ACTION_CANCEL) :

```

```

        Log.d(TAG, "Action was CANCEL");
        return true;
    case (MotionEvent.ACTION_OUTSIDE) :
        Log.d(TAG, "Movement occurred outside
bounds " +
element");
        return true;
    default :
        return true;
    }
    });
    } break;
default: {
    super.onManagerConnected(status);
}break;
}
};

// svm native
private native int trainClassifierNative(String trainingFile, int
kernelType,
int cost, float gamma, int isProb, String modelFile);
private native int doClassificationNative(float values[][], int
indices[][],
int isProb, String modelFile, int labels[], double probs[]);

//SVM training which outputs a file named as "model" in MyDataSet
private void train() {
    // Svm training
    int kernelType = 2; // Radial basis function
    int cost = 4; // Cost
    int isProb = 0;
    float gamma = 0.001f; // Gamma
    String trainingFileLoc = storeFolderName+DATASET_NAME;
    String modelFileLoc = storeFolderName+"/model";
    Log.i("Store Path", modelFileLoc);

    if (trainClassifierNative(trainingFileLoc, kernelType, cost,
gamma, isProb,
modelFileLoc) == -1) {
        Log.d(TAG, "training err");
        finish();
    }
    Toast.makeText(this, "Training is done", 2000).show();
}

public void initLabel() {

    File file[] = storeFolder.listFiles();

    int maxLabel = 0;
    for (int i=0; i < file.length; i++)
    {

```

```

String fullName = file[i].getName();

final int dotId = fullName.lastIndexOf('.');
if (dotId > 0) {
    String name = fullName.substring(0, dotId);
    String extName = fullName.substring(dotId+1);
    if (extName.equals("jpg")) {
        int curName = Integer.valueOf(name);
        if (curName > maxLabel)
            maxLabel = curName;
    }
}

curLabel = maxLabel;
curMaxLabel = curLabel;
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
0);
    setContentView(R.layout.activity_main);

    try{
        FileInputStream fis = new FileInputStream(sdFile);
        ObjectInputStream ois = new ObjectInputStream(fis);
        while(true)
        {
            try{

                int key = ois.readInt();
                String value =(String) ois.readObject();

                Intent intent = Intent.parseUri(value, 0);
                table.put(key, intent);
            }
            catch(IOException e)
            {
                break;
            }
        }
        ois.close();
        Log.e("ReadFile", "read succeeded.....");
    }catch(Exception ex)
    {
        Log.e("ReadFile", "read ended.....");
    }

    mOpenCvCameraView = (MyCameraView)
findViewById(R.id.HandGestureApp);
mOpenCvCameraView.setVisibility(SurfaceView.VISIBLE);
mOpenCvCameraView.setCvCameraViewListener(this);
}

```

```

samplePoints = new Point[SAMPLE_NUM][2];
for (int i = 0; i < SAMPLE_NUM; i++)
{
    for (int j = 0; j < 2; j++)
    {
        samplePoints[i][j] = new Point();
    }
}

avgColor = new double[SAMPLE_NUM][3];
avgBackColor = new double[SAMPLE_NUM][3];

for (int i = 0; i < 3; i++)
    averChans.add(new ArrayList<Double>());

//HLS
//initCLowerUpper(7, 7, 80, 80, 80, 80);

//RGB
//initCLowerUpper(30, 30, 30, 30, 30, 30);

//HSV
//initCLowerUpper(15, 15, 50, 50, 50, 50);
//initCBackLowerUpper(5, 5, 80, 80, 100, 100);

//YcrCb
// initCLowerUpper(40, 40, 10, 10, 10, 10);

//Lab
initCLowerUpper(50, 50, 10, 10, 10, 10);
initCBackLowerUpper(50, 50, 3, 3, 3, 3);

SharedPreferences numbers = getSharedPreferences("Numbers", 0);
imgNum = numbers.getInt("imgNum", 0);

    initOpenCV();

    Log.i(TAG, "Created!");
}

public void initOpenCV() {

}

//Things triggered by clicking any items in the menu start here
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection

    switch (item.getItemId()) {
        case R.id.action_save:
            isPictureSaved = true;
            // savePicture();
            return true;
        case R.id.data_collection:
            callDataCollection();
            return true;
        case R.id.map_apps:

```

```

        callMapApps();
        return true;
    case R.id.app_test:
        if (mode == APP_TEST_MODE) {
            mode = TRAIN_REC_MODE;
            Toast.makeText(getApplicationContext(), "App testing
ends!", Toast.LENGTH_LONG).show();
        } else {
            mode = APP_TEST_MODE;
            Toast.makeText(getApplicationContext(), "App testing
begins!", Toast.LENGTH_LONG).show();
            appTestFrameCount = 0;
        }
        return true;
    // default:
    //     return super.onOptionsItemSelected(item);
}

int groupId = item.getGroupId();

if (item.getGroupId() == 2) {
    int id = item.getItemId();
    Camera.Size resolution = mResolutionList.get(id);
    mOpenCvCameraView.setResolution(resolution);
    resolution = mOpenCvCameraView.getResolution();
    String caption =
Integer.valueOf(resolution.width).toString() + "x" +
Integer.valueOf(resolution.height).toString();
    Toast.makeText(this, caption, Toast.LENGTH_SHORT).show();

    return true;
}

return super.onOptionsItemSelected(item);
}

public void restoreUI() {

    setContentView(R.layout.activity_main);

    mOpenCvCameraView = (MyCameraView)
findViewById(R.id.HandGestureApp);
    mOpenCvCameraView.enableView();

    mOpenCvCameraView.setOnTouchListener(new OnTouchListener() {
        public boolean onTouch(View v, MotionEvent event) {
            // ... Respond to touch events
            int action = MotionEventCompat.getActionMasked(event);

            switch(action) {
                case (MotionEvent.ACTION_DOWN) :
                    Log.d(TAG, "Action was DOWN");
                    String toastStr = null;
                    if (mode == SAMPLE_MODE) {
                        mode = DETECTION_MODE;
                        toastStr = "Sampling Finished!";
                    } else if (mode == DETECTION_MODE) {
                        mode = TRAIN_REC_MODE;
                    }
            }
        }
    });

    ((Button) findViewById(R.id.AddBtn)).setVisibility(View.VISIBLE);
}

```

```

((Button) findViewById(R.id.TrainBtn)).setVisibility(View.VISIBLE);

((Button) findViewById(R.id.TestBtn)).setVisibility(View.VISIBLE);
    toastStr = "Binary Display Finished!";

    preTrain();

    } else if (mode == TRAIN_REC_MODE) {
        mode = DETECTION_MODE;

((Button) findViewById(R.id.AddBtn)).setVisibility(View.INVISIBLE);

((Button) findViewById(R.id.TrainBtn)).setVisibility(View.INVISIBLE);

((Button) findViewById(R.id.TestBtn)).setVisibility(View.INVISIBLE);

        toastStr = "train finished!";
    } else if (mode == BACKGROUND_MODE) {
        toastStr = "First background sampled!";
        rgbaMat.copyTo(backMat);
        mode = SAMPLE_MODE;
    }

    Toast.makeText(getApplicationContext(),
toastStr, Toast.LENGTH_LONG).show();
    return false;
    case (MotionEvent.ACTION_MOVE) :
        Log.d(TAG, "Action was MOVE");
        return true;
    case (MotionEvent.ACTION_UP) :
        Log.d(TAG, "Action was UP");
        return true;
    case (MotionEvent.ACTION_CANCEL) :
        Log.d(TAG, "Action was CANCEL");
        return true;
    case (MotionEvent.ACTION_OUTSIDE) :
        Log.d(TAG, "Movement occurred outside bounds
" +
                "of current screen element");
        return true;
    default :
        return true;
    }

    }

});

mOpenCvCameraView.setVisibility(SurfaceView.VISIBLE);
mOpenCvCameraView.setCvCameraViewListener(this);

((Button) findViewById(R.id.AddBtn)).setVisibility(View.VISIBLE);

((Button) findViewById(R.id.TrainBtn)).setVisibility(View.VISIBLE);
    ((Button) findViewById(R.id.TestBtn)).setVisibility(View.VISIBLE);

}

```

```

    public void showDialogBeforeAdd(String title,String message){
        Log.i("Show Dialog", "Entered");
        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
            this);
        // set title
        alertDialogBuilder.setTitle(title);
        // set dialog message
        alertDialogBuilder
            .setMessage(message)
            .setCancelable(false)
            .setPositiveButton("Yes",new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,int
id) {

                    doAddNewGesture();

                    synchronized(sync) {
                        sync.notify();
                    }

                    dialog.cancel();

                }
            })
            .setNegativeButton("No",new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,int
id) {

                    // if this button is clicked, just close
                    // the dialog box and do nothing

                    synchronized(sync) {
                        sync.notify();
                    }

                    dialog.cancel();

                }
            });
        // create alert dialog
        AlertDialog alertDialog = alertDialogBuilder.create();
        // show it
        alertDialog.show();
    }

    public void showDialog(final Context v, String title,String message,
        String posStr, String negStr, String neuStr){

        diagResult = null;

        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
            v);
        // set title
        alertDialogBuilder.setTitle(title);

```

```

        // set dialog message
        alertDialogBuilder
            .setMessage(message)
            .setCancelable(false)
            .setPositiveButton(posStr, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
id) {

                    diagResult = "Positive";

                    Toast.makeText(getApplicationContext(), "Add
more to Gesture "
                    + selectedLabel,
Toast.LENGTH_SHORT).show();

                    curLabel = selectedLabel - 1;

                    dialog.cancel();

                }
            })
            .setNegativeButton(negStr, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
id) {

                    // if this button is clicked, just close
                    // the dialog box and do nothing

                    diagResult = "Negative";

                    doDeleteGesture(selectedLabel);

                    Toast.makeText(getApplicationContext(),
"Gesture "
                    + selectedLabel + " is deleted",
Toast.LENGTH_SHORT).show();

                    curLabel = selectedLabel - 1;
                    dialog.cancel();

                }
            });

        if (neuStr != null) {
            alertDialogBuilder.setNeutralButton(neuStr, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {

                    diagResult = "Neutral";

                    Toast.makeText(getApplicationContext(), "Canceled"
                    , Toast.LENGTH_SHORT).show();

```

```

        selectedLabel = -2;
        dialog.cancel();

    }
    }) ;
}

// create alert dialog
AlertDialog alertDialog = alertDialogBuilder.create();
// show it
alertDialog.show();
}

```

## HandGesture.java

```

package com.example.newdemo;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.TreeMap;

import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfInt;
import org.opencv.core.MatOfInt4;
import org.opencv.core.MatOfPoint;
import org.opencv.core.MatOfPoint2f;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.RotatedRect;
import org.opencv.core.Scalar;
import org.opencv.imgproc.Imgproc;
import org.opencv.imgproc.Moments;

public class HandGesture {
    public List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
    public int cMaxId = -1;
    public Mat hie = new Mat();
    public List<MatOfPoint> hullP = new ArrayList<MatOfPoint>();
    public MatOfInt hullI = new MatOfInt();
    public Rect boundingRect;
    public MatOfInt4 defects = new MatOfInt4();

    public ArrayList<Integer> defectIdAfter = new ArrayList<Integer>();

    public List<Point> fingerTips = new ArrayList<Point>();
    public List<Point> fingerTipsOrder = new ArrayList<Point>();
    public Map<Double, Point> fingerTipsOrdered = new TreeMap<Double,
Point>();

    public MatOfPoint2f defectMat = new MatOfPoint2f();
    public List<Point> defectPoints = new ArrayList<Point>();
    public Map<Double, Integer> defectPointsOrdered = new TreeMap<Double,
Integer>();
}

```

```

public Point palmCenter = new Point();
public MatOfPoint2f hullCurP = new MatOfPoint2f();
public MatOfPoint2f approxHull = new MatOfPoint2f();

public MatOfPoint2f approxContour = new MatOfPoint2f();

public MatOfPoint palmDefects = new MatOfPoint();

public Point momentCenter = new Point();
public double momentTiltAngle;

public Point inCircle = new Point();

public double inCircleRadius;

public List<Double> features = new ArrayList<Double>();

private boolean isHand = false;

private float[] palmCircleRadius = {0};

void findBiggestContour()
{
    int idx = -1;
    int cNum = 0;

    for (int i = 0; i < contours.size(); i++)
    {
        int curNum = contours.get(i).toList().size();
        if (curNum > cNum) {
            idx = i;
            cNum = curNum;
        }
    }

    cMaxId = idx;
}

boolean detectIsHand(Mat img)
{
    int centerX = 0;
    int centerY = 0;
    if (boundingRect != null) {
        centerX = boundingRect.x + boundingRect.width/2;
        centerY = boundingRect.y + boundingRect.height/2;
    }
    if (cMaxId == -1)
        isHand = false;
    else if (boundingRect == null) {
        isHand = false;
    } else if ((boundingRect.height == 0) || (boundingRect.width ==
0))
        isHand = false;
    else if ((centerX < img.cols()/4) || (centerX > img.cols()*3/4))
        isHand = false;
    else
        isHand = true;
    return isHand;
}

```

```

}

//Convert the feature indicated by label to the string used in SVM
input file
String feature2SVMString(int label)
{
    String ret = Integer.toString(label) + " ";
    int i;
    for (i = 0; i < features.size(); i++)
    {
        int id = i + 1;
        ret = ret + id + ":" + features.get(i) + " ";
    }
    ret = ret + "\n";
    return ret;
}

//Extract hand features from img
String featureExtraction(Mat img, int label)
{
    String ret = null;
    if ((detectIsHand(img))) {

        defectMat.fromList(defectPoints);

        List<Integer> dList = defects.toList();
        Point[] contourPts = contours.get(cMaxId).toArray();
        Point prevDefectVec = null;
        int i;
        for (i = 0; i < defectIdAfter.size(); i++)
        {
            int curDlistId = defectIdAfter.get(i);
            int curId = dList.get(curDlistId);

            Point curDefectPoint = contourPts[curId];
            Point curDefectVec = new Point();
            curDefectVec.x = curDefectPoint.x - inCircle.x;
            curDefectVec.y = curDefectPoint.y - inCircle.y;

            if (prevDefectVec != null) {
                double dotProduct = curDefectVec.x*prevDefectVec.x +
                    curDefectVec.y*prevDefectVec.y;
                double crossProduct = curDefectVec.x*prevDefectVec.y -
                    prevDefectVec.x*curDefectVec.y;

                if (crossProduct <= 0)
                    break;
            }

            prevDefectVec = curDefectVec;
        }

        int startId = i;
        int countId = 0;

        ArrayList<Point> finTipsTemp = new ArrayList<Point>();
    }
}

```

```

if (defectIdAfter.size() > 0) {
    boolean end = false;

    for (int j = startId; ; j++)
    {
        if (j == defectIdAfter.size())
        {

            if (end == false) {
                j = 0;
                end = true;
            }
            else
                break;
        }

        if ((j == startId) && (end == true))
            break;

        int curDlistId = defectIdAfter.get(j);
        int curId = dList.get(curDlistId);

        Point curDefectPoint = contourPts[curId];
        Point fin0 = contourPts[dList.get(curDlistId-2)];
        Point fin1 = contourPts[dList.get(curDlistId-1)];
        finTipsTemp.add(fin0);
        finTipsTemp.add(fin1);

        //Valid defect point is stored in curDefectPoint
        Core.circle(img, curDefectPoint, 2, new Scalar(0, 0,
255), -5);

        countId++;
    }
}

int count = 0;
features.clear();
for (int fid = 0; fid < finTipsTemp.size(); )
{
    if (count > 5)
        break;

    Point curFinPoint = finTipsTemp.get(fid);

    if ((fid%2 == 0)) {

        if (fid != 0) {
            Point prevFinPoint = finTipsTemp.get(fid-1);
            curFinPoint.x = (curFinPoint.x + prevFinPoint.x)/2;
            curFinPoint.y = (curFinPoint.y + prevFinPoint.y)/2;
        }

        if (fid == (finTipsTemp.size() - 2) )
            fid++;
        else

```

```

        fid += 2;
    } else
        fid++;

    Point disFinger = new Point(curFinPoint.x-inCircle.x,
curFinPoint.y-inCircle.y);
    double dis =
Math.sqrt(disFinger.x*disFinger.x+disFinger.y*disFinger.y);
    Double f1 = (disFinger.x)/inCircleRadius;
    Double f2 = (disFinger.y)/inCircleRadius;
    features.add(f1);
    features.add(f2);

    //curFinPoint stores the location of the finger tip
Core.line(img, inCircle, curFinPoint, new Scalar(24, 77, 9),
2);

Core.circle(img, curFinPoint, 2, Scalar.all(0), -5);

Core.putText(img, Integer.toString(count), new
Point(curFinPoint.x - 10,
curFinPoint.y - 10),
Core.FONT_HERSHEY_SIMPLEX, 0.5, Scalar.all(0));

    count++;

}

ret = feature2SVMString(label);

}

return ret;
}

```

```

public native double findInscribedCircleJNI(long imgAddr, double
rectTLX, double rectTLY,
double rectBRX, double rectBRY, double[] incircleX, double[]
incircleY, long contourAddr);

```

```

// Find the location of inscribed circle and return the radius and
the center location

```

```

void findInscribedCircle(Mat img)
{

    Point tl = boundingRect.tl();
    Point br = boundingRect.br();

    double[] cirx = new double[]{0};
    double[] ciry = new double[]{0};

    inCircleRadius = findInscribedCircleJNI(img.getNativeObjAddr(),
tl.x, tl.y, br.x, br.y, cirx, ciry,
approxContour.getNativeObjAddr());
    inCircle.x = cirx[0];
    inCircle.y = ciry[0];
}

```

```

        Core.circle(img, inCircle, (int)inCircleRadius, new
Scalar(240,240,45,0), 2);
        Core.circle(img, inCircle, 3, Scalar.all(0), -2);
    }

}

```

## MyCameraView.java

```
package com.example.newdemo;
```

```
import java.util.List;
```

```
import org.opencv.android.JavaCameraView;
```

```
import android.content.Context;
```

```
import android.hardware.Camera;
```

```
import android.hardware.Camera.Size;
```

```
import android.util.AttributeSet;
```

```

public class MyCameraView extends JavaCameraView{
    private static final String TAG = "Sample::MyCameraView";
    private String mPictureFileName;
    // private Camera.Parameters mParameters = mCamera.getParameters();
    public MyCameraView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public List<String> getEffectList() {
        return mCamera.getParameters().getSupportedColorEffects();
    }

    public boolean isEffectSupported() {
        return (mCamera.getParameters().getColorEffect() != null);
    }

    public String getEffect() {
        return mCamera.getParameters().getColorEffect();
    }

    public void setEffect(String effect) {
        Camera.Parameters params = mCamera.getParameters();
        params.setColorEffect(effect);
        mCamera.setParameters(params);
    }

    public List<Size> getResolutionList() {
        return mCamera.getParameters().getSupportedPreviewSizes();
    }
}

```

```

}

public void setResolution(Size resolution) {
    disconnectCamera();
    mMaxHeight = resolution.height;
    mMaxWidth = resolution.width;
    connectCamera(getWidth(), getHeight());
}

public Size getResolution() {
    return mCamera.getParameters().getPreviewSize();
}

public boolean isAutoWhiteBalanceLockSupported() {
    //return mParameters.isAutoWhiteBalanceLockSupported();
    return mCamera.getParameters().isAutoWhiteBalanceLockSupported();
}

public boolean getAutoWhiteBalanceLock () {
    //return mParameters.getAutoWhiteBalanceLock();
    return mCamera.getParameters().getAutoWhiteBalanceLock();
}

public void setAutoWhiteBalanceLock (boolean toggle) {
    Camera.Parameters params = mCamera.getParameters();
    params.setAutoWhiteBalanceLock(toggle);
    mCamera.setParameters(params);
}

public String getFocusMode () {
    return mCamera.getParameters().getFocusMode();
}

public void setFocusModeFixed () {
    Camera.Parameters params = mCamera.getParameters();
    params.setFocusMode(params.FOCUS_MODE_FIXED);
    mCamera.setParameters(params);
}

public void setFocusModeCon() {
    Camera.Parameters params = mCamera.getParameters();
    params.setFocusMode(params.FOCUS_MODE_CONTINUOUS_VIDEO);
    mCamera.setParameters(params);
}

public void startAutoFocus(Camera.AutoFocusCallback cb) {
    mCamera.autoFocus(cb);
}

```

```
}  
  
public int getMaxNumFocusAreas () {  
    return mCamera.getParameters().getMaxNumFocusAreas();  
}  
public boolean isAutoExposureLockSupported () {  
    return mCamera.getParameters().isAutoExposureLockSupported();  
}  
  
public boolean getAutoExposureLock () {  
    return mCamera.getParameters().getAutoExposureLock();  
}  
  
public void setAutoExposureLock (boolean toggle) {  
    Camera.Parameters params = mCamera.getParameters();  
    params.setAutoExposureLock(toggle);  
    mCamera.setParameters(params);  
}  
}
```

**ANEXO 10 : ACTA DE APROBACIÓN DE ORIGINALIDAD DE LOS  
TRABAJOS ACADÉMICOS DE LA UCV**



**ACTA DE APROBACIÓN DE ORIGINALIDAD  
DE TESIS**

Código : F06-PP-PR-02.02  
Versión : 02  
Fecha : 23-03-2018  
Página : 1 de 1

Yo, MG. RENEÉ RIVERA CRISÓSTOMO, docente de la Facultad Ingeniería y Escuela Profesional Ingeniería de Sistemas de la Universidad César Vallejo Lima Este, revisor (a) de la tesis titulada "APLICACIÓN MÓVIL DE INTERPRETACIÓN DEL LENGUAJE DE SEÑAS PERUANAS PARA DISCAPACITADOS AUDITIVOS EN LA ASOCIACIÓN DE SORDOS DE LA REGIÓN LIMA", del (de la) estudiante **LOPEZ ROCA KEVIN ALEX**, constato que la investigación tiene un índice de similitud de 29% verificable en el reporte de originalidad del programa Turnitin.

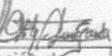
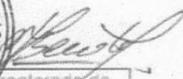
El/la suscrito (a) analizó dicho reporte y concluyó que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

Lima, San Juan de Lurigancho 01 de Diciembre del

2018

.....  
MG. RENEÉ RIVERA CRISÓSTOMO

DNI: ...08557341...

 Elaboró	 Dirección de Investigación	Revisó	 Responsable del SOC	 Vicerectorado de Investigación
--	---	--------	--	---

**ANEXO 11 : PANTALLAZO TURNITIN**

feedback studio Lopez Roca, Kevin Alex -- /0

**UNIVERSIDAD CÉSAR VALLEJO**  
**FACULTAD DE INGENIERIA**  
**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS**

"Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima"

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS**

**AUTOR:**  
Lopez Roca, Kevin Alex

**ASESOR:**  
Mg. Rivera Crisostomo, Rene

**LÍNEA DE INVESTIGACIÓN:**  
Sistemas de Información y Comunicaciones

**LIMA - PERÚ**  
2018



**Resumen de coincidencias** ✕

**29 %**

1	Entregado a Universida... Trabajo del estudiante	8 %
2	repositorio.ucv.edu.pe Fuente de Internet	8 %
3	opac.pucv.cl Fuente de Internet	2 %
4	cybertesis.urp.edu.pe Fuente de Internet	1 %
5	repositorio.espe.edu.ec Fuente de Internet	1 %
6	documents.mx Fuente de Internet	1 %
7	Entregado a Universida... Trabajo del estudiante	1 %

Página: 1 de 66    Número de palabras: 12540    Text-only ...    Turnitin C...    High Reso...    Activ...    🔍

**ANEXO 12: AUTORIZACIÓN DE PUBLICACIÓN DE TESIS**



**AUTORIZACIÓN DE PUBLICACIÓN DE TESIS  
EN REPOSITORIO INSTITUCIONAL UCV**

Código : F08-PP-PR-02.02  
Versión : 10  
Fecha : 10-06-2019  
Página : 1 de 1

Yo Kevin Alex Lopez Roca, identificado con DNI N° 48270133, egresado de la Escuela Profesional de Ingeniería de Sistemas de la Universidad César Vallejo, autorizo () , No autorizo (  ) la divulgación y comunicación pública de mi trabajo de investigación titulado "Aplicación móvil de interpretación del lenguaje de señas peruanas para discapacitados auditivos en la Asociación de Sordos de la Región Lima"; en el Repositorio Institucional de la UCV (<http://repositorio.ucv.edu.pe/>), según lo estipulado en el Decreto Legislativo 822, Ley sobre Derecho de Autor, Art. 23 y Art. 33

Fundamentación en caso de no autorización:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

  
FIRMA

DNI: 48270133

FECHA: 1 de diciembre del 2018

			
Revisó	Vicerrectoría de Investigación / DEVAC	Responsable de la SCC	Aprobó Rectorado

NOTA: Cualquier documento impreso diferente del original, y cualquier archivo electrónico que se encuentren fuera del Campus Virtual Trilce serán considerados como COPIA NO CONTRIBUADA.

**ANEXO 13: AUTORIZACIÓN DE LA VERSIÓN FINAL DEL TRABAJO DE  
INVESTIGACIÓN**



# UNIVERSIDAD CÉSAR VALLEJO

## AUTORIZACIÓN DE LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN

CONSTE POR EL PRESENTE EL VISTO BUENO QUE OTORGA EL ENCARGADO DE INVESTIGACIÓN DE

Mg. Renée Rivera Crisostomo

A LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN QUE PRESENTA:

Kevin Alex Lopez Roca

INFORME TÍTULADO:

"Aplicación móvil de interpretación del lenguaje de señas peruanas para  
discapitados auditivos en la Asociación de sordos de la Región Lima"

PARA OBTENER EL TÍTULO O GRADO DE:

Ingeniero de Sistemas

SUSTENTADO EN FECHA: 01 de diciembre del 2018

NOTA O MENCIÓN: 13



FIRMA DEL ENCARGADO DE INVESTIGACIÓN