

**DISEÑO CENTRADO EN EL USUARIO DE UNA PLATAFORMA PARA LA COMUNICACIÓN POR LENGUAJE  
DE SEÑAS BASADO EN MACHINE LEARNING**



FUNDACIÓN  
**UNIVERSITARIA  
DE POPAYÁN**  
35 ANIVERSARIO

**NOMBRE:**

**CRISTIAN YERALDO URCUQUI ORTEGA  
JOSÉ CAMILO JIMÉNEZ TASCÓN**

**MONOGRAFÍA PARA OPTAR AL TÍTULO DE:  
INGENIERO DE SISTEMAS**

**DIRECTOR:**

**JOSE ARMANDO ORDOÑEZ**

**FUNDACIÓN UNIVERSITARIA DE POPAYÁN  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE SISTEMAS  
GRUPO DE INVESTIGACIÓN IMS**

**Popayán, agosto de 2020**

## Contenido

1. Introducción.....	7
1.1. Planteamiento del problema.....	7
1.2. Formulación del problema.....	8
2. Objetivos.....	8
2.1. Objetivo general.....	8
2.2. Objetivos específicos.....	8
3. Justificación.....	9
4. Marcos de referencia.....	9
4.1. Marco Teórico - Conceptual.....	9
4.1.1. Sordomudez y mudez.....	9
4.1.2. Diagnostico de la sordomudez.....	10
4.1.3. Inteligencia artificial.....	10
4.1.4. Procesamiento de imágenes.....	11
4.1.5. Machine Learning.....	11
4.1.6. Deep Learning.....	11
4.1.7. Neurona.....	12
4.1.8. Red neuronal artificial.....	13
4.1.9. Red neuronal convolucional.....	14
4.1.10. TensorFlow.....	16
4.1.11. Keras.....	16
4.1.12. Diseño centrado en el usuario.....	17
4.1.13. Revisión sistemática.....	17
4.2. Antecedentes y estado del arte.....	18
4.2.1. Definición para la búsqueda.....	18
4.2.2. Ejecución de la búsqueda.....	18
4.2.3. Análisis de los resultados.....	19
5. Implementación de la metodología.....	23

5.1. Modelo centrado en el usuario (Design Thinking).....	23
5.1.1. Entender el problema (empatizar) .....	24
5.1.2. Definir problemática (definir).....	26
5.1.3. Posibles soluciones (idear).....	26
5.1.4. Prototipado (prototipar).....	29
5.1.5. Evaluación(evaluar).....	30
5.1.6. Implementación.....	32
5.2. Modelado crisp-dm.....	32
5.2.1. Entendimiento del negocio.....	33
5.2.2. Entendimiento de los datos.....	34
5.2.3. Preparación de datos.....	35
5.2.4. Modelado.....	35
5.2.5. Selección del tipo de modelo.....	36
5.2.6. Entrada.....	37
5.2.7. Entrenamiento.....	38
5.2.8. Evaluación de precisión.....	41
5.2.9. Exportar el modelo.....	43
5.2.10. Evaluación.....	43
5.3. Implementación .....	44
5.4. Funcionamiento de la aplicación.....	53
6. Resultados.....	55
7. Conclusiones.....	58
8. Recomendaciones y trabajos futuros .....	59
9. Bibliografía.....	60

## Lista de tablas

Tabla 1 : Resultados de búsqueda sistemática.....	18
Tabla 2 : Artículos seleccionados de búsqueda sistemática.....	19

Tabla 3 : Tabla POV de usuario .....	26
Tabla 4 : Resultados de valoración y factibilidad de ideas.....	28
Tabla 5 : Encuesta de evaluación de prototipo.....	30
Tabla 6 : Cronograma.....	44
Tabla 7 : Pruebas de aplicación etapa 1 .....	48
Tabla 8 : Pruebas de aplicación etapa 2 .....	50
Tabla 9 : Pruebas de aplicación etapa 3.....	51
Tabla 10 : Tabla de resultados de prueba sobre clases.....	55

## Lista de ilustraciones

Figura 1 : Deep learning .....	12
Figura 2 : Red neuronal artificial .....	13
Figura 3 : Funcionamiento de red neuronal .....	14
Figura 4 : Funcionamiento de convolución .....	15
Figura 5 : Extracción de características .....	16
Figura 6 : Encuesta de usuario .....	25
Figura 7 : Lluvia de ideas para el planteamiento de la solución.....	27
Figura 8 : Tabla de valoración de factibilidad soluciones elegidas.....	27
Figura 9 : Resultados de valoración y factibilidad de ideas.....	28
Figura 10 : Diagrama de la solución.....	29
Figura 11 : Mockup-prototipo inicial de aplicación.....	30
Figura 12 : Evaluación de prototipo.....	31
Figura 13 : Prototipo-mockup final .....	32
Figura 14 : Segmentación de piel .....	35
Figura 15 : Dataset.....	36
Figura 16 : Operación de convolución.....	37
Figura 17 : Dataset en google colab.....	38
Figura 18 : Dataset de entrenamiento.....	38
Figura 19 : Código de alteración de imágenes.....	39

Figura 20 : Maxpooling.....	40
Figura 21 : Dropout.....	40
Figura 22 : Modelo entrenado.....	41
Figura 23 : Gráfica de precisión y pérdida.....	42
Figura 24 : Exportación del modelo.....	43
Figura 25 : Conversión de modelo a tensorflow lite.....	43
Figura 26 : Programación inicial de app móvil.....	46
Figura 27 : Creación de panel de las letras.....	47
Figura 28 : Implementación del módulo de la cámara.....	47
Figura 29 : Prueba etapa 1 (interfaz de escritorio).....	48
Figura 30 : Aplicativo móvil con modelo de pruebas.....	49
Figura 31 : Prueba etapa 2 (interfaz móvil).....	51
Figura 32 : Realización de encuestas de prueba.....	52
Figura 33 : Prueba etapa 3 (aplicación final).....	53
Figura 34 : Toma de la seña con cámara móvil.....	54
Figura 35 : Teclado indicador de letra. ....	54
Figura 36 : Mapa de calor, casos vs pruebas.....	56
Figura 37 : Aplicación final.....	57

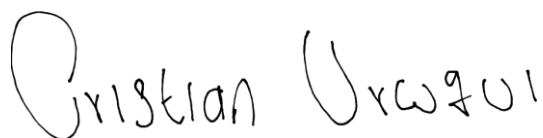
## Certificación de autoría

Certifico que conozco el concepto de plagiar según la Real Academia de la lengua ( “Copiar en lo sustancial obras ajenas, dándolas como propias.” )

Y certifico que el contenido de este documento son de mi autoría, no hay contenido que haya sido copiado directamente y al pie de la letra de ninguna fuente. En el caso de ideas, teorías, conceptos, resultados y otros contenidos tomados de otros autores se menciona explícitamente la fuente original, y sólo en unos pocos casos se han mantenido el mismo texto haciéndolo entre comillas.

Reconozco las consecuencias académicas, jurídicas y económicas que conlleva el plagio.

Firma



---

Cristian Yeraldo Urcuqui Ortega  
CC. 1061816387

Firma



---

José Camilo Jiménez Tascón  
CC. 1053852848

## Resumen

En Colombia y el mundo existe una gran población con mudez o incapacidad de comunicación verbal la cual debido a múltiples situaciones entre las cuales están la falta de ayudas y educación, se encuentra en una situación social complicada. Es por esto que en este trabajo traemos una propuesta para el diseño de una posible herramienta que le sirva a estas personas para poder comunicarse con otros que no tengan este tipo de dificultad verbal. El trabajo propuesto es un trabajo que incluye un aplicativo móvil que hace uso de un modelo de inteligencia artificial y deep learning para la traducción del lenguaje de señas al lenguaje escrito español mediante el uso de una red neuronal convolucional CNN, una aplicación desarrollada para Android y técnicas como las de transferencia de conocimiento y de entrenamiento en la nube.

**Palabras claves:** Prototipo, Machine learning, Red neuronal artificial, Deep learning, Tensor Flow, Aplicación móvil, detección de imágenes.

## 1. Introducción

### 1.1. Planteamiento del problema

Colombia es un país latinoamericano en donde existe una alta población con dificultades de comunicación en el habla ya que según cifras del Instituto Nacional para Sordos (INSOR), la población para el año 2015 ascendía a un número cercano a 450.000 [1]; esta problemática se agrava mucho más si se tiene en cuenta que según datos del INSOR, el Sistema de matrícula estudiantil (SIMAT) y el Departamento Nacional de Estadística (DANE) para el 2015 existían cerca de 167.246 de estas personas en estado de analfabetismo, es decir, cerca del 37.17% de la población sordo-muda del país [2].

Esta condición de dificultad en la comunicación no solamente afecta a estas personas dentro de su rendimiento académico, sino que además presenta un gran obstáculo para su desarrollo dentro de la sociedad, haciendo que en su mayoría este tipo de personas quede relegada a trabajos poco remunerados de difícil acceso o incluso a trabajos que pueden llegar a estar considerados dentro de la ilegalidad, esto se puede ver en [3], en donde basándose en las proyecciones dadas por el DANE para el año 2015 el 25% de la población sorda en Colombia se encontraba empleada o buscando trabajo y el restante 75% estaban en condiciones de no trabajo o trabajo informal.

Por esta razón desde la academia y la industria se han generado diversas aproximaciones que involucran herramientas tecnológicas para mejorar la comunicación de las personas con discapacidad auditiva. Algunas herramientas permiten la traducción entre computador-persona del lenguaje de señas a diversos idiomas. Estas aproximaciones utilizan o requieren cámaras de alta definición conectadas a el computador lo cual hace que sea un proceso costoso y poco práctico al no permite su uso en la vida cotidiana de las personas con discapacidad auditiva [4]. Otras aproximaciones realizan diseños

de reconocimiento de lenguaje de señas utilizando cámaras de baja resolución. En este caso el manejar este tipo de tecnología puede presentar dificultades a la hora de detectar las señas [5] . En la mayoría de las propuestas existentes no se evidencia la participación de las personas con discapacidad auditiva en el diseño de las soluciones si no como usuarios finales.

Nuestra propuesta busca aportar en este sentido por medio del diseño centrado del usuario de una aplicación que mejore la comunicación entre personas con discapacidad auditiva y personas sin esta discapacidad. Para conseguir este objetivo se utilizarán técnicas de machine learning para procesar imágenes del lenguaje para personas con discapacidad auditiva.

## **1.2. Formulación del problema**

¿Qué elementos debe tener una aplicación soportada en machine learning que permita la comunicación entre personas con discapacidad auditiva y las personas que no padecen de esta condición de forma práctica y económica?

## **2. Objetivos**

### **2.1. Objetivo general**

Realizar el diseño de un prototipo de aplicación centrado en el usuario de una plataforma que permita la traducción del abecedario de lenguaje de señas colombiano (CSL) a lenguaje natural (español) utilizando machine learning.

### **2.2. Objetivos específicos**

- Hacer una revisión del estado del arte sobre las aproximaciones tecnológicas existentes para la comunicación entre personas con discapacidad auditiva.
- Realizar el diseño centrado en el usuario de una plataforma soportada en machine learning que permita la interpretación del abecedario en lenguaje de señas traduciendo a lenguaje natural (español).
- Implementar y evaluar la plataforma diseñada.

Resultados obtenidos

- Monografía
- Prototipo desarrollado
- Solicitud de registro de software
- Borrador de artículo de investigación



### 3. Justificación

En [6], se exponen algunos criterios para evaluar la importancia potencial de una investigación, los cuales fueron adoptados para justificar este estudio:

- Valor metodológico de la investigación: Análisis del uso de inteligencia artificial en el diseño centrado en el usuario para la traducción del abecedario de lenguaje de señas a castellano.
- Valor práctico de la investigación: Las personas con discapacidad auditiva de la región podrán contar con una solución (herramienta software), que les ayude a comunicarse con las personas oyentes.
- Valor tecnológico: Se desarrollara una herramienta software integral de machine learning para la traducción de lenguaje para sordos mediante señales.
- Valor de Emprendimiento e Innovación: A corto plazo, el desarrollo generado podrá ser utilizado para la enseñanza a personas no sordas el lenguaje de señas y así solucionar la brecha de comunicación y a mediano plazo (fase siguiente de este proyecto) se buscará que esta solución pueda ser adoptada por instituciones que acogen a personas con discapacidad auditiva, con el objetivo de demostrar que la herramienta soluciona problemas de comunicación. Lo anterior, permitiría dar una mejor comunicación en instituciones educativas y laborales, finalmente esto contribuya a mejorar la calidad de vida de estas personas.

### 4. Marcos de referencia

#### 4.1. Marco Teórico - Conceptual

En este trabajo, presentamos un diseño para la creación de una aplicación prototipo necesaria para el apoyo al personal sordo-mudo en el país con el fin de poder dar el primer paso en búsqueda de un mejor bienestar y felicidad para dicha comunidad que actualmente se encuentra muy golpeada y olvidada en nuestro país haciendo uso de redes neuronales convolucionales (Convolutional Neural Network).

##### 4.1.1. Sordomudez y mudez

La sordomudez es una discapacidad en el desarrollo del lenguaje caracterizada como patología por la UNESCO [7] que se presenta en personas que nacen sin la capacidad física necesaria de poder escuchar o que luego del nacimiento pierden la capacidad total de escucha con lo cual tampoco desarrollan la capacidad de poderse comunicar verbalmente [8].

Por otra parte la mudez se define como la discapacidad para el hablar o comunicarse verbalmente sin ser necesario que exista algún problema auditivo.

En este trabajo nos referiremos a la población sordomuda y muda como una sola (sordomudos o sordo-mudos), por efectos prácticos debido a que el resultado final se vera reflejado en un prototipo que atiende a las dos poblaciones ya que ellas padecen los mismos problema de comunicación en el habla aunque sus causas sean diferentes.

#### **4.1.2. Diagnostico de la sordomudez**

Se detecta en la infancia cuando se trata de mudez o sordomudez desde antes de nacer debido a que el infante nunca muestra deseos de hablar o vocalizar como debería de suceder en los niños que no padecen de este tipo de discapacidad comunicacional. Ahora bien, cuando se trata de mudez o sordomudez adquirida a partir de un accidente o de alguna enfermedad, esta se ve reflejada en la perdida de expresión verbal por parte del paciente afectado [9].

Para realizar un diagnóstico preciso en los menores de edad se recomienda realizar algunas pruebas psicológicas y pediátricas ya que muchas veces este tipo de discapacidad puede llegar a ser confundido con trastornos de comunicación y otras anomalías psicológicas o psiquiátricas.

#### **4.1.3. Inteligencia artificial**

Desde la invasión informática la probabilidad de máquinas con genio de pensamiento y razonamiento ha resultado fascinante, visualizada por escritores y artistas se han ideado androides indistinguibles de individuos e inteligencias artificiales con capacidad sobrehumanas y entrada a infinitud de conocimiento. Para entender el porque se llegara a esta area del conocimiento es necesario entender que es la inteligencia artificial y como es su funcionamiento.

La inteligencia artificial (IA), es un área de las ciencias de la computación y se define como una serie de algoritmos con capacidades matemáticas de aprendizaje automáticas. Llegado a esto se tiene que los datos son patrones, puntos, bites o cualquier tipo de información medible y no procesada pero disponibles para su uso por algoritmos computacionales en donde estos últimos son los encargados de realizar un trabajo de aprendizaje artificial a través iteraciones, lecturas y correlaciones sobre los datos haciendo que el resultado final de sea un algoritmo capaz de predecir o detectar patrones en ejercicio predeterminado para el cual fue entrenado inicialmente [10][11]. En pocas palabras se puede definir a la inteligencia artificial como el conjunto de pasos necesarios para que a partir de datos un ordenador, computador o maquina sea capaz de detectar patrones de maneras similares a como lo hace el ser humano.

En el caso de este trabajo, se utilizaran algunos algoritmos de inteligencia artificial para poder realizar la detección de algunos patrones dados por la forma, posición y uso de las manos por parte de las personas sordo-mudas y así poder retornar una respuesta legible del significado de esas señales a través de la aplicación prototipo que mostrara la respuesta en forma de texto.

#### 4.1.4. Procesamiento de imágenes

El procesamiento de imágenes es una serie de técnicas, practicas y algoritmos que se utilizan para poder convertir una imagen en un conjunto de datos legible por un sistema de información, en nuestra investigación, este paso se realizo a través de herramientas como OpenCV, matplotlib, conv2s, maxploing, flatten y pandas.

#### 4.1.5. Machine Learning

Parte elemental de la inteligencia artificial es encontrar la manera mediante la cual una maquina pueda comportarse de una manera inteligente o que responda a patrones o estímulos de la manera mas humana posible. Es aquí en donde aparece el machine learning o también llamado aprendizaje automático, el cual no es mas que un conjunto de técnicas mediante las cuales se desarrollan algoritmos de reconocimiento de patrones matemáticos con la finalidad de que una maquina genere una respuesta al encontrar dichos patrones en objetos o datos que se le entregan para el análisis.

Todo lo anterior genera un entendimiento de patrones comunes que aceptan la elaboración de pronósticos cada vez más perfeccionados basándose en los datos y en los patrones suministrados a los algoritmos iniciales, es decir, las máquinas aprenden de cálculos anteriores para actuar, tomar decisiones y entregar resultados.

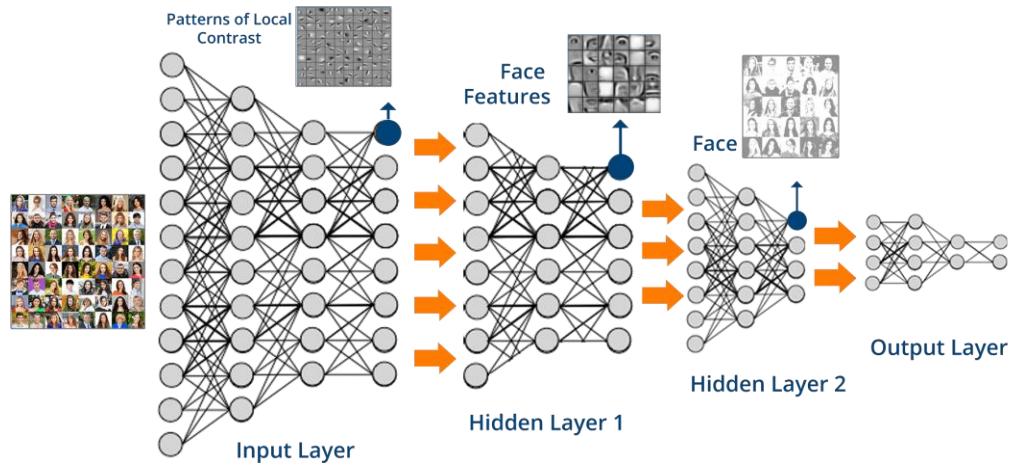
El problema del machine learning en su etapa mas básica, es que no es capaz de aprender a partir de ejemplos o de información, sino que necesita que se le suministren reglas manualmente y que se le dicten que tipo de patrones debe encontrar[12].

#### 4.1.6. Deep Learning

Dado el concepto anterior, dentro del mundo de las ciencias de la computación se llevo a un momento en el que el machine learning se hacía corto para su uso en sistemas mucho mas complejos, lo que llevo al desarrollo de una subdivisión de este campo de conocimiento a lo cual se le conoce como deep learning o aprendizaje profundo.

Este, a diferencia del aprendizaje automático, comprende el análisis de datos como una clasificación de conceptos y su división en patrones mas pequeños siguiendo una estructura jerárquica semejándose a como funciona el cerebro humano y sus redes de comunicación biológicas. Debido a esta manera de funcionar, la información y los datos dentro de un algoritmo de deep learning terminan divididos en capas según la importancia dentro del contexto o la aplicación en la que se quiere usar el algoritmo (ver fig 1), todo esto mediante el uso de patrones que fabrican representaciones de datos mas simples y las ordenan en pequeños grupos llamados neuronas las cuales pueden ser interpretadas mas fácilmente por la maquina de manera individual lo que lleva a que ya esta no solamente sea responsable de reconocer patrones iniciales administrados por un usuario y aprender

de ellos, sino que además va más allá de los patrones y en una de las etapas del procesamiento de datos comienza a aprender a partir de ejemplos y no de números como se hace en el machine learning tradicional [13].



*Figura 1: Deep learning*

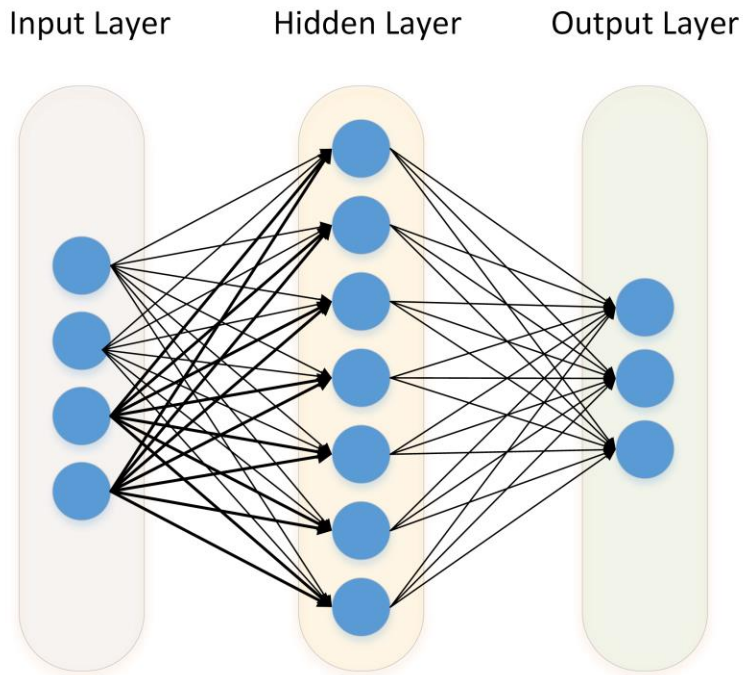
Todo lo dicho anteriormente hace que este tipo de técnicas clasifique la información relevante y elimine o de menos prioridad a la información no necesaria en donde dependiendo de la arquitectura o el modelo utilizado se generen conexiones entre las capas que hacen que su uso en el análisis de imágenes sea muy extendido y muy eficaz [14].

#### 4.1.7. Neurona

Una neurona dentro de un algoritmo de machine learning es básicamente una agrupación de una o más reglas que componen una red y que procesan única y exclusivamente un grupo de patrones que se extraen a partir de los datos iniciales; es decir, una neurona básicamente se encarga de reconocer una o unas pocas características que puedan ser extraídas de los datos [13].

#### 4.1.8. Red neuronal artificial

En el campo del deep learning una red neuronal artificial o red neuronal es un conjunto de neuronas conectadas entre sí que sirven como una cadena de toma de decisiones para realizar la entrega de un resultado final; las redes neuronales son básicamente la unión de muchas neuronas para que el algoritmo de aprendizaje sea capaz de retornar el mejor resultado, en otras palabras es una versión avanzada y mas compleja de lo que es una red Bayesiana (ver fig 2) [13].



*Figura 2: Red neuronal artificial*

#### 4.1.9. Red neuronal convolucional

Una red neuronal convolucional es un concepto en el que varias redes neuronales son divididas en capas, en donde la primer capa contiene neuronas encargadas de realizar convoluciones o en otras palabras, se encargan de discernir y extraer las características deseadas que el algoritmo necesita para poder reconocer los objetos o datos que recibirá en el futuro, luego de esta capa se encuentra otra en donde existen neuronas de reducción del muestreo que se encargan de tomar datos evitando repeticiones y por ultimo se encuentran las neuronas de clasificación que son las encargadas de determinar que patrones y características son mas o menos importantes [13][14] como puede verse en la figura 3 (ver fig 3).

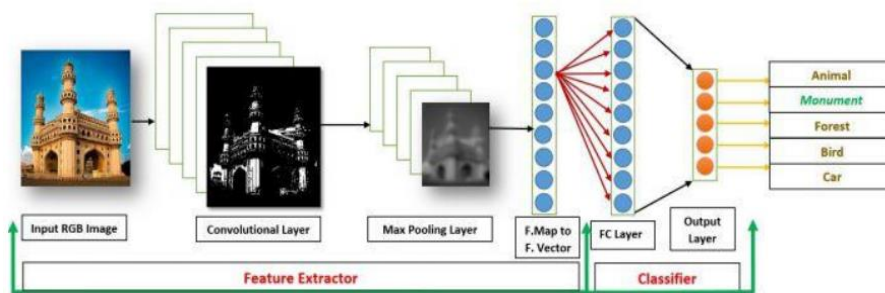
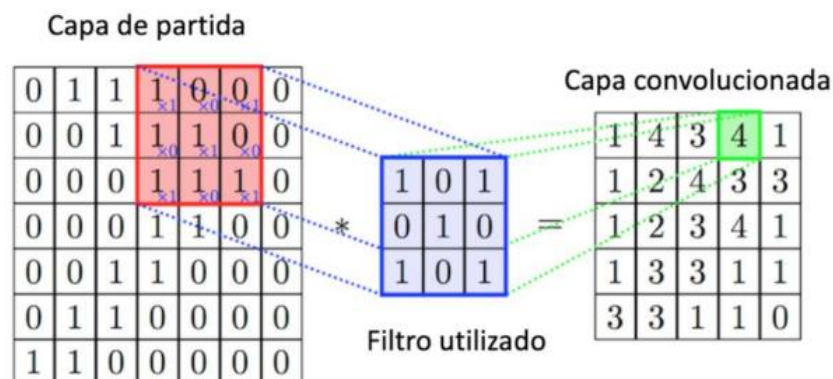


Figura 3: Funcionamiento de red neuronal

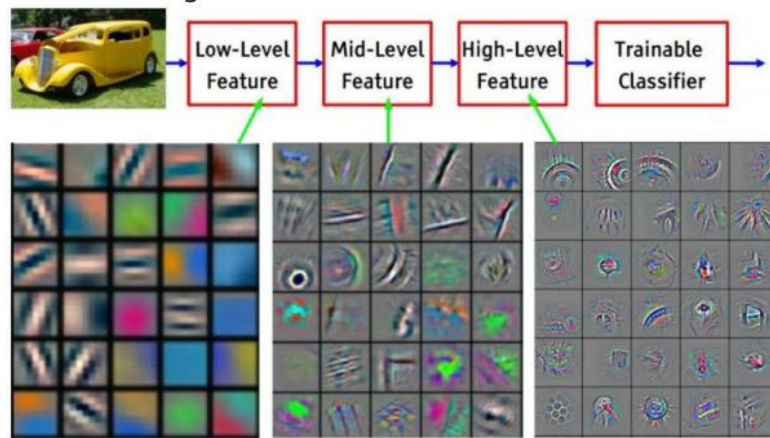
Por todo lo anterior es por lo cual este tipo de redes neuronales son utilizadas para el análisis de imágenes ya que permiten destacar características importantes, desechar las que no lo son y dar como resultado un modelo que permite identificar los patrones deseados con un alto grado de exactitud, todo esto haciendo

uso de capas que operan sobre los datos de entrada mediante el cálculo de convoluciones discretas con bancos de filtros finitos, tal y como se ve en la figura 4 (ver fig 4) [13].



*Figura 4: Funcionamiento de convolución*

En este tipo de capas, las operaciones de convolución permiten obtener características dominantes de la imagen de entrada relacionadas con los objetivos de entrenamiento. De forma experimental se observa que las primeras capas en redes de convolucionales se centran en la búsqueda de características simples, como podrían ser bordes, esquinas o regiones. A medida que se avanza hacia capas más profundas, se aumenta el nivel de abstracción del contenido de la imagen al que se muestran sensibles [13], tal y como se observa a continuación (ver fig 5).



*Figura 5: Extracción de características*

#### 4. 1. 10. TensorFlow

TensorFlow es una librería desarrollada por google para el control, manejo y la aplicación de algoritmos matemáticos para la diferenciación numérica de información mediante el uso de formulas y algoritmos conocidos para el tratamiento de estos datos, es por esto que es actualmente conocida como una de las mejores alternativas para el desarrollo de modelos de machine learning y la creación de redes neuronales capaces de llevar a cabo tareas de visión artificial y análisis de imágenes [15].

Es por todo lo anterior que en el trabajo desarrollado se utilizo esta librería, ya que facilita los procesos de creación, entrenamiento y prueba de algoritmos de machine learning, y además permite su uso en dispositivos móviles directamente a través de tensor flow lite.

#### 4. 1. 11. Keras

Keras es una librería desarrollada en python para el desarrollo rápido de redes neuronales convolucionales y deep learning con alta capacidad de procesamiento de datos capaz de funcionar con librerías de análisis de flujo de datos como



tensorflow, theano, microsoft cognitive toolkit y lenguajes de programación desarrollados para este tipo de trabajos como R, Julia o Myia [16].

En el trabajo a llevar a cabo se utilizó esta librería junto con tensor flow para el desarrollo de un modelo basado en deep learning que sea capaz de reconocer los patrones e imágenes del lenguaje de señas colombiano y entregar un resultado.

#### **4.1.12. Diseño centrado en el usuario**

El diseño centrado en el usuario (DCU) es, una aproximación al diseño de productos y aplicaciones que ubica al usuario en el centro de todo el proceso. Por tal razón podemos entender el DCU como una filosofía cuya premisa es que, para garantizar el éxito de un producto, hay que tener en cuenta al usuario en las fases del diseño dando importancia a las fases de diseño simultáneamente se plantean elementos como Diseño conceptual, diseño iterativo y visualización del prototipo.

Hay que mencionar que el diseño conceptual se fundamenta en datos y posibilidades tecnológicas, y además es impulsada por el conocimiento de los usuarios, la primera gama de ideas de conceptos, herramientas se produce en una serie de sesiones de idealización como lluvia de ideas. Se pueden considerar varios medios para el apoyo a la toma de decisiones que incluyen carteles y sesiones de capacitación, tutoriales en video y aplicaciones de construcción.

Por otro lado, el diseño es el de estudio de los elementos con los que un usuario puede interactuar cuando usa un computador o producto interactivo. El objetivo es definir el comportamiento o la interacción de un sistema en respuesta a las acciones de sus usuarios. Así mismo este tipo de diseño se focaliza en cómo el usuario comunica o interactúa con el sistema y, por ello, se centra en aspectos como el flujo de la interacción, el diálogo entre la persona y el computador, cómo el estímulo de entrada se relaciona con la respuesta y los mecanismos de retroalimentación [17].

#### **4.1.13. Revisión sistemática**

La revisión sistemática es una metodología para la elaboración de desarrollos basados en la investigación la cual consiste en una serie de tres pasos ajustados a la investigación requerida; bajo esta se hace necesario el hacer gran énfasis en los trabajos ya realizados por otras personas, todo ello en un periodo de tiempo menor a 10 años desde la publicación de dichas obras hasta el momento en el que se pretende realizar una nueva iteración investigativa en el tema en el cual se quiera incursionar, esta metodología está comprendida en los siguientes pasos:

- Definición para realizar la búsqueda de datos.
- Ejecución de la búsqueda.
- Análisis de resultados.

En el primer paso lo que se hace es definir que se quiere buscar, en el segundo se definen los términos de búsqueda y se ajustan a los requeridos por las bases de información en donde se quiere conseguir la información y la tercera parte es básicamente una reflexión del porque se eligieron los trabajos que se seleccionaron y el impacto que tendrán en la nueva investigación y en sus resultados [18].

## 4.2. Antecedentes y estado del arte

Para esta actividad se utilizó la metodología de revisión sistemática o “systematic review” (en inglés), descrita en [18] que se divide en tres etapas, las cuales son:

### 4.2.1. Definición para la búsqueda

En esta etapa lo que se hace es partir de la pregunta de investigación inicial en busca de soluciones ya desarrolladas o similares a la que se quiere realizar.

**Pregunta 1:** ¿Que elementos se deben tener en cuenta en una aplicación desarrollada para la detección del lenguajes de señas colombiano?

– Para esta búsqueda se tuvieron en cuenta las fuentes encontradas dentro de las bases de datos de IEEE Xplore, Scopus, Springer y EBSCO con un periodo de búsqueda de 5 años desde el 2015 hasta el 2020.

Para generar la cadena de búsqueda nos enfocamos en la parte central de nuestra pregunta de investigación, que corresponde a la detección de objetos a partir de imágenes, con lo que se extrajeron las palabras claves: image, computer vision, sign language, machine learning, neural networks y deep learning. De esta manera se generaron las siguientes cadenas principales para la búsqueda.

- a) “Sign language and machine learning, deep learning or neural networks” .
- b) “Sign language with computer vision and machine learning, deep learning or neural networks” .
- c) “Sign language detection using computer vision and deep learning, machine learning or neural networks” .
- d) “Sign language and deep learning, machine learning or neural networks” .

### 4.2.2. Ejecución de la búsqueda

Luego de tener las cadenas de búsqueda se procedió a convertirlas al lenguaje de búsqueda utilizado por las bases de datos seleccionadas, el cual se compone de las palabras clave y conectores AND y OR para relacionar dichas palabras claves y englobando los resultados en el área de ciencias de la computación (computer science), quedando así las cadenas de búsqueda (ver tabla 1):

*Tabla 1: Resultados de búsqueda sistemática*

Cadena de busqueda	IEEE Xplore	Scopus	Springer	Ebsco
--------------------	-------------	--------	----------	-------

			(manual)	
“Sign language” AND ( “machine learning” OR “deep learning” OR neural networks)	25	497	129	2
“Sign language” AND “computer vision” AND ( “machine learning” OR “deep learning” OR “neural networks” )	101	113	129	0
“Sign language detection” AND “computer vision” AND ( “deep learning” OR “machine learning” OR “neuronal networks” )	19	1	129	0
“Sign language” AND ( “deep learning” OR “machine learning” OR “neural networks” )	409	630	129	2

Para la investigación se seleccionaron artículos, revistas, libros o conferencias donde se consideraron modelos, metodologías y procedimientos para la predicción de lenguaje de señas basada en imágenes.

Se descartan los artículos que no expliquen los procesos o las metodologías.

Para identificar los estudios se realizaron los siguientes filtros:

- Revisión de los títulos arrojados por las búsquedas.
- Lectura del resumen y/o abstract de los artículos elegibles por el título.

#### 4.2.3. Análisis de los resultados

Finalmente para el análisis se establecen los criterios necesarios para cumplir con nuestra pregunta de investigación, los cuales son (ver tabla 2):

- Año de implementación.
- Objetivo.
- Modelo implementado.
- Resultados obtenidos.

*Tabla 2: Artículos seleccionados de búsqueda sistemática*

	Objetivo	Aporte	Brecha	Año
American Sign Language Recognition using DeepLearning and Com-	Realizar un sistema capaz de reconocer el lenguaje de señas americanos	Hace uso de algunas técnicas de deep learning que sirven para demos-	La investigación se centra en el lenguaje de señas americano y hace uso	2018

puter Vision [19]	a través de la captura de video y los procesos de análisis de imagen mediante deep learning.	trar la manera en que se descompone una imagen en las subsecuentes características que ayudan a detectar las señas realizadas con las manos.	de una cámara de video externa al aplicativo. Se muestra el concepto pero no se demuestra su aplicación como se busca en el caso de nuestro trabajo.	
Indian Sign Language Gesture Recognition using Image Processing and Deep Learning [20]	Realizar un sistema capaz de traducir el sistema de señas Indú a texto	Hace uso de las redes neuronales y presenta resultados y procesos similares a los que se van a utilizar en la investigación	La investigación se centra en el uso de dispositivos costosos y poco ergonómicos como lo es el kinect, se basa en el lenguaje de señas Indú y no presenta un prototipo de uso practico como el que se desea crear.	2019
Deep learning-based sign language recognition system for static signs [21]	Este articulo describe una manera mediante la cual se puede detectar el lenguaje de señas Indú a partir de imágenes estáticas y deep learning.	Muestra técnicas de deep learning aplicadas esta vez a imágenes estáticas del lenguaje de señas Indú.	El trabajo se centra en la transformación de las imágenes, en el procesamiento pero de nuevo no se muestra un resultado utilizable por la población general.	2020
Detection of Alphabets for Machine Translation of Sign Language Using Deep Neural Net [22]	En este articulo se desarrolla un modelo de deep learning como respuesta a clasificadores de tipo kNN o SVM	En el articulo se muestra una técnica de deep learning fácil de entender y al mismo tiempo ayuda a que desarrollemos nuestro modelo debido a las herramientas que usa.	De nuevo, el trabajo mostrado responde a una necesidad investigativa pero no a una necesidad real, por lo cual no se menciona en ningun momento la aplicación de dicha investigación en un dispositivo real.	2019
Indian sign language converter system using an android app [23]	En este articulo se lleva a cabo el desarrollo de una aplicación android basada en MATLAB y redes neuronales.	El articulo resulta muy util ya que es el mas aproximado a lo que quiere realizarse en este trabajo y resulta como gran guía para llevarlo	Tal vez el punto que mas negativo que tiene el trabajo es que si bien se enfoca en la creación de una aplicación Android, se hace uso de herramientas	2017

a cabo ya que se y técnicas no muy  
puede usar como unidóneas para el  
antecedente trabajo y no llega  
a dar resultados  
suficientemente  
exactos como los  
que se buscan ya  
que se requieren de  
condiciones muy  
exactas para que  
funcione.

---

La selección de artículos se realizó en base a los criterios anteriores y dieron como resultado información importante como la encontrada en [19], en donde los autores proponen un sistema de reconocimiento de patrones en el lenguaje de señas americano (ISL) basándose en CNN y LSTM. Este artículo nos pareció de suma importancia ya que se destaca el uso de CNN para la detección de características del lenguaje de señas, para esto los autores plantean el uso de vídeos y de la descomposición de estos vídeos en frames en donde de estos frames se eligen los de mejor calidad para que la red CNN los utilice como datos de entrenamiento, de prueba y al mismo tiempo luego de finalizado el modelo de esta misma manera el algoritmo de deep learning utilizaría esta misma técnica para la extracción de las imágenes con las cuales se determinaría a que letra pertenece. Si bien en este trabajo destacan el uso de LSTM o RNN (Recursive Neural Networks), para la consecución de esta detección en vídeos, no tomamos en cuenta esta parte del trabajo ya que el alcance de nuestro proyecto solamente va hasta la detección de señas por imágenes y no por video.

En el segundo artículo [20], nos encontramos con otro acercamiento, esta vez para la detección del lenguaje de señas Indú mediante el uso de una red neuronal basada en la arquitectura LSTM (Long short-term memory), el cual es un tipo de red neuronal convolucional recurrente usado no solamente para imágenes sino también para el procesamiento de videos, si bien nuestro proyecto no se llega a este nivel de abstracción, es un trabajo que nos sirvió mucho como base para poder comprender mejor el funcionamiento de las redes neuronales convolucionales ya que en el artículo se explica muy bien el concepto y como funciona además de que como se hace uso de análisis de imágenes mediante kinect también lo encontramos como una base muy importante para el desarrollo de nuestro prototipo debido a que si bien nosotros no hacemos uso de este tipo de hardware, al leer el proceso y como lo llevaron a cabo, pudimos replicar el proceso pero mediante software a través de modelos de visión por computadora, librería como opencv otro tipo de técnicas utilizadas para el procesamiento de imágenes mediante la descomposición por 3 canales RGB.

Ya llegado a la literatura entregada por [21], nos encontramos con un proyecto para detección del lenguaje de señas inglés (ISL), en donde se comparan 50 mode-

los diferentes de CNN. Este artículo es muy importante debido a que de él se partió para la generación del modelo utilizado en este trabajo ya según lo demostrado por el artículo, es un modelo preciso, fácil de llevar a cabo y que no hace uso de mucho poder de procesamiento, por lo cual se puede llevar a cabo mediante el uso de herramientas gratuitas como las que se propusieron en este trabajo. De igual manera, al ser un trabajo enfocado más a la detección de palabras y al proceso académico detrás de la creación del modelo y no de un aplicativo final, no se hace uso de ciertos pasajes de el paper ya que no son necesarios en el trabajo a realizar.

Además de lo encontrado en [21] se procedió al analizar [22] en donde se encontró una base para elegir entre 3 modelos de machine learning que eran los que podrían haber sido utilizados en el proyecto final; estos modelos son el KNN (algoritmo K-Nearest Neighbor), SVM (Support Vector Machine) y DNN (Deep Neural Network). El artículo básicamente nos dio las pautas necesarias por las cuales DNN es vista como la mejor alternativa y debido a esto y a que CNN es un algoritmo que se encuentra clasificado dentro DNN se le dio validez a la investigación y al desarrollo que se quiere llevar a cabo ya que se demostró una alta precisión en los resultados y una gran facilidad para la creación y operación de modelos basados en esta arquitectura.

Luego de haberse encontrado los artículos suficientes que describieran el proceso de la creación de la red convolucional deseada se decidió buscar información más enfocada a la creación de aplicaciones móviles que realizaran una tarea similar a la que se quiere con este proyecto. En dicha búsqueda nos encontramos con el artículo denominado “Indian sign language converter system using an android app” [23]; este artículo lo encontramos altamente importante debido a que en él se describe el desarrollo de una aplicación para la detección del lenguaje de señas Indú para el sistema operativo Android y aunque fue de gran utilidad para saber como realizar la aplicación, solo se eligió este mismo por su aporte al desarrollo de la aplicación y no por su aporte al modelo de IA ya que en este caso el paper propone uso de herramientas diferentes a las que se van a utilizar como lo es Matlab.

Por último, luego del análisis de los puntos a favor que nos dan los artículos encontrados al desarrollo de nuestro prototipo se procedió a buscar en ellos brechas o puntos a los cuales podemos aportar desde este proyecto, algunos de los puntos a aportar son los de aplicación, los de desarrollo enfocado al usuario y si bien en [23] se nos expone el desarrollo de una app móvil, justamente este artículo carece de las bases fundamentales por las cuales elegimos los 4 artículos anteriores, ya que en el desarrollo de [23] las herramientas usadas son totalmente diferentes y algo inferiores a las herramientas que van a ser usadas en este proyecto. Con todo lo anterior entonces podemos decir que nos encontramos con que [19], [20], [21] y [22] son artículos enfocados al desarrollo del modelo de detección de señas pero sin presentar un enfoque dirigido al usuario y mucho menos una solución final como la propuesta en este trabajo la cual describimos como el diseño de prototipo de aplicación móvil enfocada a ayudar a

las personas mudas, por otra parte en [23] aunque nos encontramos con la aplicación y el desarrollo enfocado al usuario, los métodos de detección de las señas no son los más adecuados, por lo que nuestro trabajo es el de intentar unir tanto los conceptos dados por los 4 artículos anteriores a [23] y este último haciendo que se cree una nueva propuesta que pueda considerarse innovadora en el campo de trabajo debido a que los trabajos existentes no han llegado a una propuesta similar.

## **5. Implementación de la metodología**

La implementación de la metodología hay que destacar que el proyecto se divide en dos partes; por un lado encontramos el desarrollo de la aplicación móvil y por otro encontramos el desarrollo del modelo de IA para la detección de las señas, esto nos llevo básicamente a hacer uso de las metodologías Design Thinking como metodología enfocada al diseño centrado al usuario y crisp-dm como metodología para el desarrollo del modelo de deep learning los cuales serán explicados a continuación.

### **5.1. Modelo centrado en el usuario (Design Thinking)**

Para la creación del modelo que rige todo el trabajo realizado, se utilizo la metodología design thinking la cual nos permite llevar solucionar cualquier problemática partiendo desde una idea y llevándola a la implementación en donde el enfoque final sera dirigido al usuario, dicha metodología se divide en 6 pasos fundamentales (empatizar, definir, idear, prototipar, evaluar, implementar) [24], y sera explicada a continuación.

### 5.1.1. Entender el problema (empatizar)

En esta fase lo que se hace es buscar empatía con las personas, es decir, se busca acercarse a su problemática con el fin de poder entender el problema. Para ello se realizó un acercamiento teórico y práctico a la población afectada el cual consistió en una investigación de datos, estadísticas e información relacionada al respecto, y un conversatorio realizado con una persona perteneciente a la población afectada; la información se consiguió a través de distintas fuentes gubernamentales como el INSOR, el DANE y el Ministerio de Educación Nacional tal y como se puede ver en el apartado introductorio y de definición de objetivos de este documento, y el conversatorio se realizó con un usuario al que en este documento llamaremos José el cual accedió a la realización de una encuesta como método para poder determinar el problema a través del conversatorio llevado a cabo, dicha encuesta consiste de las siguientes preguntas:

- ¿los sordos pueden hablar?
- ¿todas las personas sordas “leen los labios” ?
- ¿las personas sordas pueden bailar?
- ¿pueden las personas sordas manejar un dispositivo móvil?
- ¿las personas sordas pueden tener pareja?
- ¿Cómo fue su proceso educativo en la escuela y colegio?
- ¿hay alguna dificultad para ser aceptado en un trabajo?
- ¿Cómo hace para comunicarse con personas oyentes que no conocen la lengua de señas?

A este cuestionario, el usuario respondió con las siguientes afirmaciones:

- No siempre ya que en la mayoría de los casos las personas sordas nacen con este problema por lo que no son capaces de aprender a comunicarse verbalmente, sin embargo hay personas sordas que pierden la capacidad de escuchar con el tiempo pero no pierden la capacidad del habla.
- No, aprender a leer labios no es tan fácil como parece y en muchas es preferible aprender a escribir y a comunicarse mediante la escritura en lugar de aprender a leer los labios.
- Si, a no ser de que la persona en cuestión tenga algún otro tipo de discapacidad física, las personas sordas pueden guiarse mediante las vibraciones generadas por la música.
- Si, actualmente el uso de redes sociales para comunicarse con las personas es nuestra principal herramienta de comunicación, ya que no es necesario hablar para que la otra persona pueda entendernos como sucedía anteriormente con las llamadas telefónicas.



- Claro que sí, aunque la verdad nos cuesta un poco más de trabajo sobre todo porque la idea de la conquista esta muy influenciada por la comunicación verbal.
- En la escuela y el colegio fue bastante complicado debido a que todas las tareas, trabajos y la comunicación con mis profesores era de manera escrita ya que muy pocos si quiera se animaban a aprender a saludar con las señas.
- Si, en la mayoría de trabajos la comunicación verbal es pieza clave, y por lo general los únicos trabajos que nos ofrecen son trabajos poco remunerados en donde no se considere necesaria la comunicación.
- Para comunicarnos con estas personas utilizamos el lenguaje escrito o hacemos uso de aplicaciones como whatsapp que nos facilitan la comunicación pero de manera escrita.

Cabe resaltar que José es una persona sencilla, se caracteriza por ser competitiva, en su pensamiento la palabra no está rotundamente nula, expresa que las personas sordas tienen la misma capacidad y amor para realizar las acciones que cualquier otra persona pudiese hacer en su vida cotidiana, vive su día a día al máximo, y siempre está en constante aprendizaje, tiene amor por su trabajo y amor por su familia, también clara que no se socialmente no se debería mirar diferente a las personas con problemas auditivos ya que algunas son mejores personas que aquellas personas que por así decirlo lo tienen todo en la vida, por ultimo José nos regaló una frase y es “no existe limites cuando haces las cosas de corazón” , la ilustración del desarrollo de la encuesta puede verse a continuación (ver fig 6).



*Figura 6: Encuesta de usuario*

### 5.1.2. Definir problemática (definir)

En esta etapa lo que se hace es definir el problema de tal manera que luego sea fácil hallar soluciones; para ello nos basamos en los resultados obtenidos en la etapa de la empatización con lo que se genero una tabla en donde se puede ver el punto de vista del usuario (POV – Point Of View en inglés), dicha tabla se mostrará a continuación (ver tabla 3).

*Tabla 3: Tabla POV de usuario*

<b>Usuario</b>	<b>Necesidad</b>	<b>Necesidad</b>
Un adulto joven que vive en la ciudad.	Tener comunicación con personas que no entiende el lenguaje de señas.	El usuario quiere sentirse normal demostrar que son personas del común. Buscan tener buena comunicación con sus compañeros del trabajo para así poderle dar una mejor calidad de vida a su familia.

Ya teniendo definido el problema o necesidad del usuario, se pasa a la siguiente fase.

### 5.1.3. Posibles soluciones (idear)

Llegado a este punto, se realizo una pequeña lluvia de ideas que pudieran enfrentarse a la problemática inicial dada, dicha lluvia de ideas se desarrollo con un equipo de 6 personas y el proceso puede verse en la siguiente ilustración (ver fig 7).



*Figura 7: Lluvia de ideas para el planteamiento de la solución*

Luego de esto, se eligieron 3 posibles soluciones las cuales fueron llevadas a una tabla de valoración de factibilidad en donde se tomo en cuenta la facilidad de implementación y el costo, esta tabla se puede ver en la siguiente figura (ver fig 8).

	Economía	Técnica
Traductor Simultáneo		
detectar al Imagen		
Traducción por texto		

*Figura 8: Tabla de valoración de factibilidad soluciones elegidas*

Para el desarrollo de esta tabla se decidió que entre todos los 6 participantes de la lluvia de ideas se iba a dar una valoración de 0 a 10 a las dos variables,

aplicándolo a las 3 ideas elegidas, en donde para el caso de la variable económica 10 significaba un alto costo y 0 muy bajo costo, y para la variable de la facilidad de implementación 0 significaba muy fácil y 10 muy difícil. Todo esto llevo a que el resultado final de las 3 ideas quedara de la siguiente manera (ver tabla 4):

*Tabla 4: Resultados de valoración y factibilidad de ideas*

Idea	Económica	Facilidad
Traductor simultaneo basado en texto (texto a voz)	10	3
Detector de señas basado en imágenes	5	3
Traducción de señas a texto	10	8

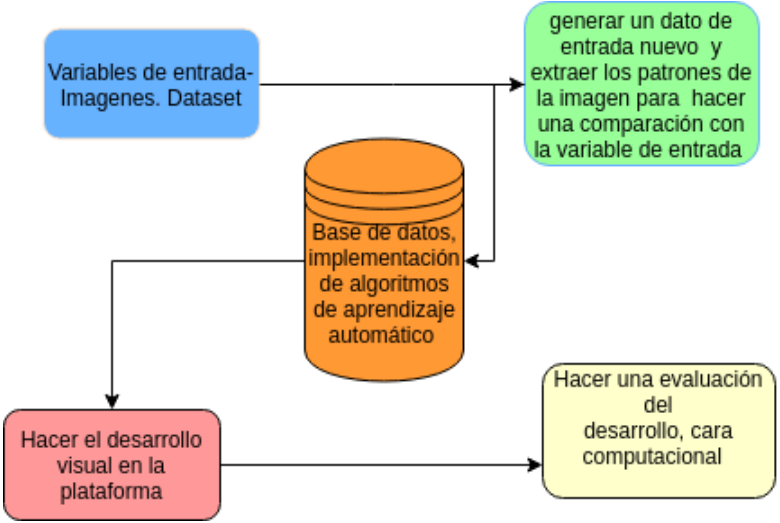
Dicha tabla también puede ser vista en la siguiente figura (ver fig 9):

Problyre

	Económica ✓	Facilidad (X) ↙
Traductor Simultaneo	10	3
detector de Imagenes	5	3
Traducción por texto	10	8

*Figura 9: Resultados de valoración y factibilidad de ideas*

Luego de esto se procedió con la realización de un diagrama que describiera la solución elegida, la cual fue la segunda (Detector de señas basado en imágenes), dicho diagrama puede ser visto a continuación (ver fig 10).



*Figura 10: Diagrama de la solución*

Por último y después de haberse encontrado la solución más adecuada, rápida y fácil de implementar paso a la etapa de prototipado.

#### 5.1.4. Prototipado (prototipar)

En esta etapa, se comenzó con el desarrollo de un prototipo de tipo mockup, el cual pudiera servirnos como guía para el desarrollo de la propuesta final, dicho prototipo luego sería evaluado por un usuario independiente, el prototipo puede verse a continuación .



*Figura 11: Mockup-prototipo inicial de aplicación*

### 5.1.5. Evaluación (evaluar)

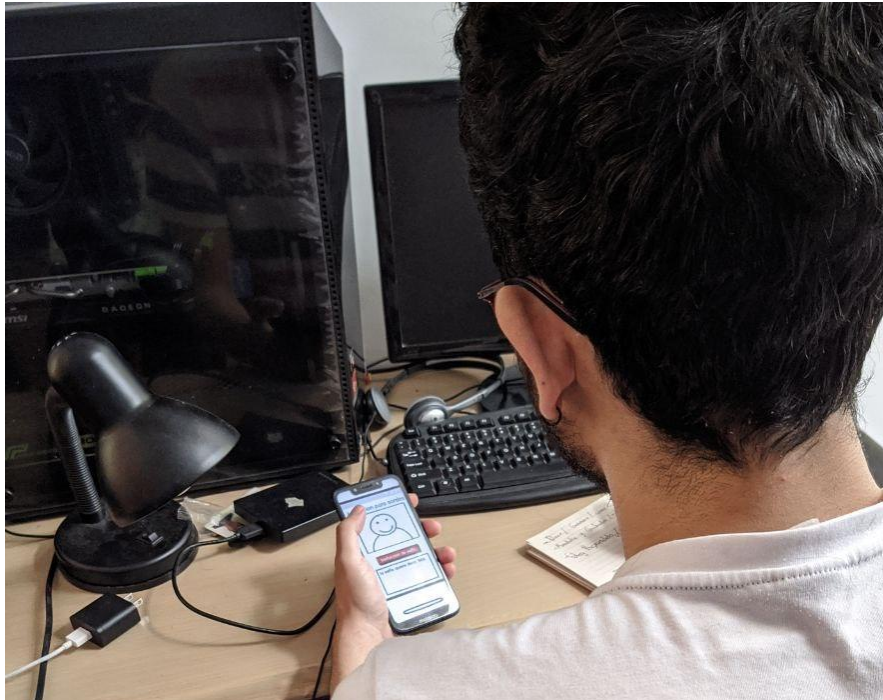
Para este punto, se tomo en cuenta lo encontrado en el estado del arte, y los resultados dados luego de realizarle a un usuario independiente las siguientes preguntas (ver tabla 5):

*Tabla 5: Encuesta de evaluación de prototipo*

Pregunta	Respuesta
¿ Cree usted que el prototipo propuesto puede funcionar?	Sí, ya que se ve sencillo, fácil de usar y al ser una aplicación móvil es muy portable.
¿ Cree que las personas mudas o sordomudas puedan hacer uso del prototipo propuesto?	Si, debido a que no requiere del uso de comandos de voz y todo se hace de manera muy intuitiva.

La realización de dicha evaluación puede verse a continuación (ver fig 12).





*Figura 12: Evaluación de prototipo*

Por último y debido a lo que encontramos en el estado del arte, decidimos que el producto final no lo enfocaríamos en la detección de palabras completas como se ve en el mockup (ver fig 11), ya que la detección de palabras es muy compleja, muy costosa y muy difícil de ser llevada a cabo debido a que el análisis de detección de señas debe realizarse con movimiento, lo cual se escapa de las capacidades de este proyecto; por todo esto, se propuso modificar el prototipo de tal manera que quedara de la siguiente manera (ver fig 13).



*Figura 13: Prototipo-mockup final*

#### **5.1.6. Implementación**

En esta etapa del desarrollo lo que se hace es comenzar con la implementación de la solución dada en la evaluación llevada anteriormente apoyándola en el estado del arte de este documento. Para ello se decidió comenzar con el desarrollo del modelado de la red neuronal sobre la cuál se va a basar esta aplicación, dicho modelado podría verse descrito en su totalidad en el capítulo 5.2 de este trabajo donde básicamente lo que se hizo fue seguir paso a paso la metodología crisp-dm para crear el modelo de deep learning que nos retorna cuando una seña corresponde a una letra y dicha letra y resultado deberá verse en una aplicación móvil.

#### **5.2. Modelado crisp-dm**

En esta sección del proyecto se mostrara el trabajo llevado a cabo bajo la metodología CRISP-DM como se propone en el punto 5.1.4 de este texto, la razón detrás de la elección de esta metodología desarrollada por IBM es que nos da una solución fácil y rápida para llevar a cabo el desarrollo de proyectos del área de Data Analytics debido a que nos proporciona una descripción de las actividades comunes que se deben efectuar para la realización de un proyecto de esta naturaleza [25]. La metodología consta de seis fases, aunque en nuestro caso se agregaron algunas fases extra para aclarar el trabajo realizado a la hora de seguir el modelo, estos pasos serán explicados a continuación.



### 5.2.1. Entendimiento del negocio

En este punto lo que se busca es comprender completamente el negocio a través de 3, las cuales serán mostradas a continuación.

#### ◆ Primera fase (salidas)

En esta fase se encuentran las salidas deseadas por el proyecto, estas son:

- **Objetivos:** El aplicativo se desarrollara como una herramienta de ayuda para que las personas mudas o sordomudas puedan comunicarse con personas sin este tipo de discapacidad tal y como puede verse en el capitulo 2 de este trabajo en donde están descritos de manera mas especifica.
- **Plan de trabajo:** En este punto, como plan de trabajo se siguieron los pasos dados por la metodología design thinking y siguiendo un cronograma (ver tabla 3).
- **Criterios de aprobación:** Por último, como criterios de aprobación se tuvieron en cuenta los resultados dados por la evaluación de presición del modelo, la cual podra ser vista en el capitulo 5.8, además de los criterios de evaluación del producto final que se encuentran en el capitulo 5.10 y 6 de este trabajo.

#### ◆ Segunde fase (situación actual)

En esta fase se describe la situación actual en la que se encuentra el proyecto y el estado del arte además de las soluciones que ya se encuentran planteadas en el mercado, esta fase puede ser descompuesta en las siguientes sub-fases:

- **Inventario:** En este punto se analizan los procesos actuales, el estado y/o los recursos con los que se cuentan, debido a que en este caso el foco de este trabajo es de I+D (Investigación y Desarrollo), el inventario se toma como el estado del arte actual, dicho estado del arte se encuentra descrito en su totalidad en el capitulo 4 de este trabajo.
- **Requerimientos y limitaciones:** En esta fase se analizan los requerimientos del trabajo y sus limitaciones, para el caso del trabajo a realizar, los requerimientos serían:
  - Un servidor con la capacidad suficiente para realizar análisis de imágenes.
  - Un celular con sistema operativo Android 5 o superior que cuente con camara.
  - La aplicación deberá ser fácil de usar.
  - La aplicación debe ser desarrollada para Android.

Luego de estos requerimientos nos encontramos con algunas limitaciones como:

- El servidor debe tener mucho poder de procesamiento (muy costoso).
- La aplicación final solamente se limitara a un nivel de prototipo.
- La arquitectura escogida para la red neuronal esta limitada a imágenes 2D.
- El prototipo final solo detectara las letras del alfabeto (exceptuado la J y Z).
- **Riesgos:** A nivel de riesgos se tiene que puedan pasar situaciones inesperadas, como bugs, problemas de reconocimiento, incompatibilidad con algunos dispositivos móviles y retrasos por situaciones inesperadas como en el caso de sucesos de carácter natural (incendios, terremotos, plagas, virus).
- **Terminología y glosario:** En este caso los términos mas importantes y relevantes fueron descritos en el capítulo 4.1 dentro del marco teórico.
- **Beneficios:** Por último se tiene que dentro de los beneficios es que el resultado final dará una aplicación capaz de identificar la mayoría de las letras encontradas dentro del CSL (lenguaje Colombiano de señas), tal y como se puede ver en las conclusiones de este trabajo.

#### ◆ Tercera y última fase (Metas y plan de proyecto)

Si bien, dentro de crisp-dm, las metas y el plan de proyecto se encuentran separadas, en este caso se presentaran juntas debido a que se encuentran muy relacionadas entre sí. Estas metas pueden verse dentro de los objetivos del proyecto (capítulo 2), y el plan de proyecto puede verse en la metodología sobre la cual se esta desarrollando el proyecto, la cual es design thinking, esta se puede consultar en el capítulo 5.1 .

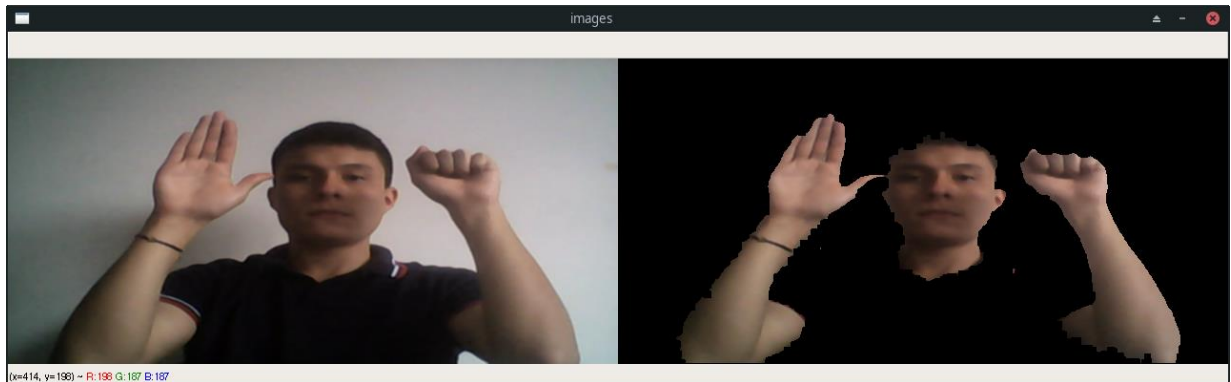
#### 5.2.2. Entendimiento de los datos

En este paso, lo que se realizo fue la descripción de datos la cual esta enfocada a entender que tipo de dato es el que necesitamos para poder detectar las señas de los sordomudos; para esto nos basamos en los capítulos 1 y 2, en donde a partir de la problemática y los objetivos identificamos se encontró que los datos necesarios para poder generar una solución, son imágenes, ya que a partir de estas se puede determinar si un signo corresponde a una letra o no.

Después de entender que los datos de entrada necesarios eran imágenes, pasamos a buscar fuentes desde las cuales pudiéramos extraerlas, para lo cual en primer lugar decidimos tomarlas por nosotros mismos con las características deseadas (las señas para letras del CLS) todo esto para formar un banco de datos con el que pudiéramos trabajar (dataset) y pasar a la siguiente etapa.

### 5.2.3. Preparación de datos

En el proceso para la preparación de los datos se implemento la segmentación basada en colores , utilizando el modelo de visión por computadora HVS ó YcbCr como lo indica [26]. Se adapta el umbral propuesto la imagen se segmenta, una vez que se segmenta la imagen se calcula la cantidad de piel ubicada en dicha imagen. Como se muestra en la ilustración 14 (ver fig 14).



*Figura 14: Segmentación de piel*

Las imágenes tiene un ancho y alto de 224 pixeles, en RGB(3 canales, Rojo, Verde, Azul), y es necesario conocer la forma de la entrada para poder dar el peso necesario a cada conexión entre neuronas, todo esto depende de la forma de las entradas.

### 5.2.4. Modelado

Luego de terminar el procesado de los datos se utilizo el método mostrado en la figura 14 (ver fig 14), de tal manera que luego de haberse segmentado lo necesario (en este caso la mano), se pudieran obtener datos limpios y descartar los datos en los cuales no se prendieron encontrar las manos.

Luego de esto, se realizo una composición del dataset quedando dividido en 24 clases que va de la A a la Y, omitiendo la J y la Z ya que la representación de dichas palabras contiene movimiento. En total se eligió el 70% para el entrenamiento y el 30% para la validación con una totalidad de 4400 imagines, en la siguiente ilustración se muestra parte del dataset (ver fig 15).



*Figura 15: Dataset*

Cabe destacar que para el modelado se opto por utilizar google colaboratory (google colab), ya que con esta herramienta se pueden llevar a cabo tareas de procesamiento de imagen con un alto nivel de poder de analisis grafico ya que en esta instancia de google se pudo llevar a cabo todo este trabajo haciendo uso de procesamiento por GPU.

para el entrenamiento del modelo, colab nos permite escribir y ejecutar código de python en un navegador, sin configuración requerida, acceso gratuito a una unidad de procesamiento grafico (GPU). Se decide usar colab por fácil acceso a una gpu a que nuestros computadores no cuentan con una unida grafica, esto causa que el modelo se demore en entrenar.

### 5.2.5. Selección del tipo de modelo

Para la selección del tipo de modelo a utilizar, nos basamos en lo encontrado en la literatura que se puede encontrar en el estado del arte y en el marco teórico, en donde a partir de su análisis, decidimos que el tipo de modelo mas conveniente para el desarrollo de este proyecto es el de una red neuronal convolucional (CNN), tal y como se demuestra en [19], [21] y [22], en donde se utiliza este tipo de modelo debido a su velocidad, facilidad de uso y a su precisión a la hora de llevar a cabo el análisis y detección de figuras a partir de imágenes 2D ya que hace uso de convoluciones, dicho proceso puede verse en la siguiente imagen (ver fig 16).

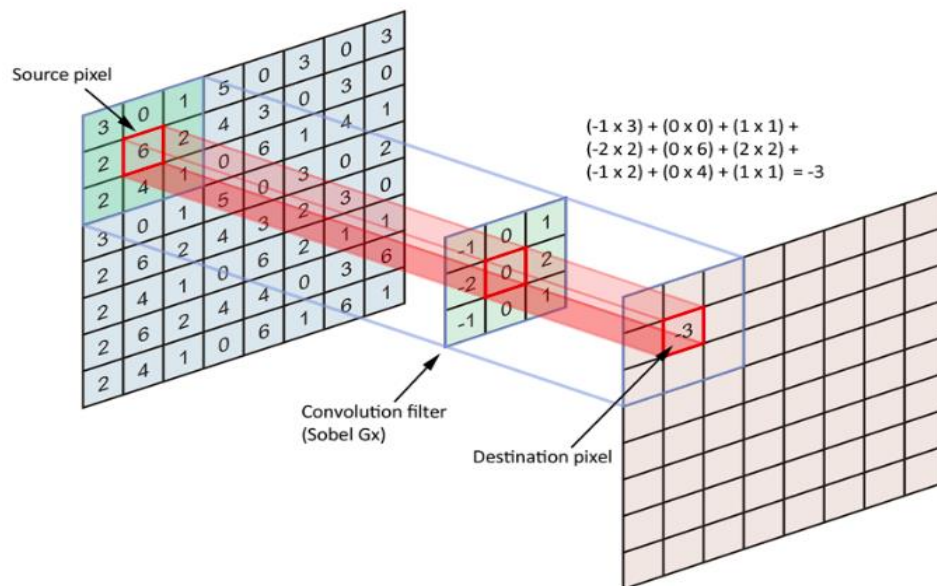


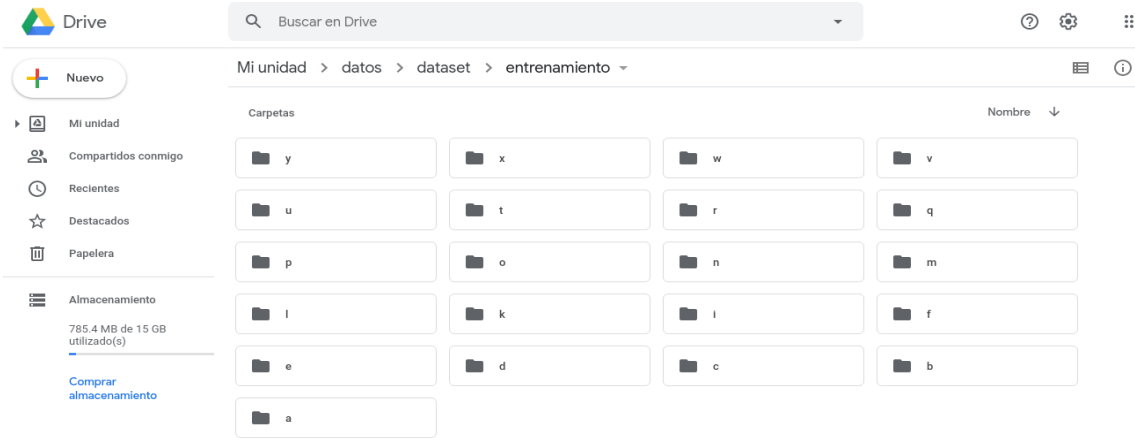
Figura 16: Operación de convolución

### 5.2.6. Entrada

Los datos de entrada para el entrenamiento del modelo vienen divididos en tres grupos.

- Entrenamiento: Datos clasificados que el modelo usara para aprender.
- Validación: Datos que el modelo usara para verificar el aprendizaje en cada iteración, la aplicación permite tomarlos aleatoriamente del conjunto de entrenamiento.
- Pruebas: Los datos que la aplicación usara para verificar la eficiencia del modelo, es importante que ninguna de estas imágenes estén en el conjunto de entrenamiento.

Ahora bien, para seleccionar los datos se debe tener en cuenta que el dataset debe estar cargado en google drive como se muestra en la ilustración 17 (ver fig 17), todo esto con el fin de simplificar la operación con Google colab.



*Figura 17: Dataset en google colab*

Adicionalmente se debe crear un bloc de notas de google colab en donde vamos a cargar los datos de entrenamiento y validación.

Por último y tal y como se puede ver en la imagen 18 (ver fig 18), se pueden observar 15 de las imágenes del dataset que fueron seleccionadas aleatoriamente de los datos de entrenamiento.



*Figura 18: Dataset de entrenamiento*

### 5.2.7. Entrenamiento

Una vez seleccionadas los diferentes conjuntos de datos debemos preparar las imágenes pero para evitar problemas de sobreajuste, necesitamos expandir artificialmente nuestro conjunto de datos haciendo que el conjunto de datos existentes sea aún mas grande; básicamente la idea seria hacer que algunos datos de entrenamiento incluyan pequeñas trasformaciones para reproducir las variaciones a

través de técnicas de aumento de datos. Algunos aumentos populares que las personas usan son escalas de grises, volteos horizontales, verticales, cultivos aleatorios, fluctuaciones de color, traducciones y rotaciones con lo cual y al aplicar solo un par de estas transformaciones a nuestros datos de entrenamiento, podemos fácilmente duplicar o triplicar el número de ejemplos de entrenamientos y crear un modelo más robusto.

Para el aumento de datos, se decide rotar al azar algunas imágenes de entrenamiento 10 grados de manera aleatoria, luego se aplicó un Zoom en un 10% a algunas imágenes de entrenamiento, después se alteraron aleatoriamente las imágenes horizontalmente en un 10% del ancho y por último se modificaron aleatoriamente las imágenes verticales en un 10% de la altura, a través del código que se muestra en la ilustración mostrada a continuación (ver fig 19).

```
[ ] # Con aumento de datos para evitar el sobreajuste

datagen = ImageDataGenerator(
    featurewise_center=False, # establecer la media de entrada a 0 sobre el conjunto de datos
    samplewise_center=False, # establecer cada muestra media a 0
    featurewise_std_normalization=False, # dividir entradas por std del conjunto de datos
    samplewise_std_normalization=False, # dividir cada entrada por su estándar
    zca_whitening=False, # aplicar blanqueamiento ZCA
    rotation_range=10, # rotar imágenes al azar en el rango (grados, 0 a 180)
    zoom_range = 0.1, # Ampliar aleatoriamente la imagen
    width_shift_range=0.1, # cambiar aleatoriamente las imágenes horizontalmente (fracción del ancho total)
    height_shift_range=0.1, # mover imágenes al azar verticalmente (fracción de la altura total)
    horizontal_flip=False, # voltear imágenes al azar
    vertical_flip=False) # voltear imágenes al azar

datagen.fit(x_train)
```

*Figura 19: Código de alteración de imágenes*

Una vez procesemos los datos podemos continuar con el entrenamiento del modelo para esto se ha creado 3 capas convolucionales 2D convirtiendo la matriz original a una (3x3) cada una con diferente número de neuronas haciendo un total de 150 neuronas, cada capa cuenta con un maxpooling se encarga de reducir el volumen de salida de las capas convoluciones de la CNN y que permite además incrementar el campo de percepción de la red, el resultado de las capas convoluciones se observa en la siguiente ilustración (ver fig 20).

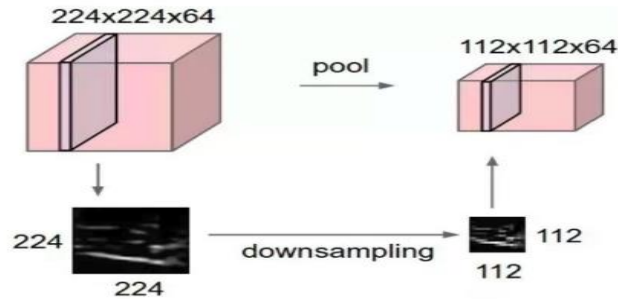
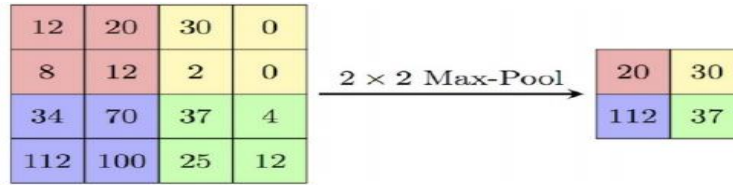


Figura 20: Maxpooling

Debemos darle una activación a nuestro modelo el cual se ha elegido la función relu esta función de activación es muy utilizada en las redes neuronales actuales debido a que experimentalmente se ha demostrado que dicho tipo de activación permite crear redes muchos mas profundas y facilitan el entrenamiento de las mismas debido a sus particularidades.

De igual forma la capa Dropout es una capa que se encarga de la regularización muy empleada para evitar el overfitting o sobreentrenamiento en las CNN; este termino se refiere a la eliminación de las contribuciones de ciertas neuronas justo a sus conexiones de entrada y salida. Dicha eliminación se realiza de forma aleatoria con una probabilidad de eliminación definida previamente, en la siguiente ilustración se muestra el funcionamiento (ver fig 21).

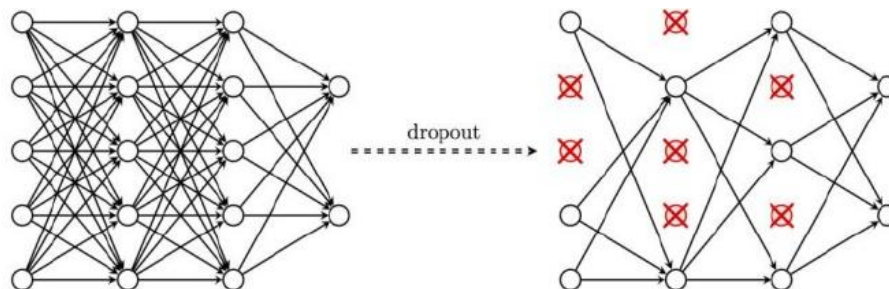


Figura 21: Dropout

Por ultimo la capa de normalización de batch tiene por objetivo aumentar la estabilidad del entrenamiento de la CNN, esta capa tiene un efecto de regularización en la red mediante la normalización de las salidas de las capas de activación anteriores actúa restando a estas la media del batch y dividiendo por su desviación típica. Esta capa requiere de dos parámetros entrenables de tal mane-



ra que los datos normalizados son multiplicados por un parámetro de desviación típica y un parámetro de media haciendo que el optimizador pueda deshacer la normalización previamente comentada para así asegurar la estabilidad del entrenamiento del sistema.

Dicho esto, el modelo código del modelo entrenado quedo de la siguiente manera (ver fig 22).

```
[ ] learning_rate_reduction = ReduceLRonPlateau(monitor='val_accuracy', patience = 2, verbose=1, factor=0.5, min_lr=0.00001)

model = Sequential()
model.add(Conv2D(75, (3,3), strides = 1, padding = 'same', activation = 'relu', input_shape = (224,224,3)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))
model.add(Conv2D(50, (3,3), strides = 1, padding = 'same', activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))
model.add(Conv2D(25, (3,3), strides = 1, padding = 'same', activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dropout(0.3))
model.add(Dense(units = 24, activation = 'softmax'))
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
model.summary()
```

*Figura 22: Modelo entrenado*

### 5.2.8. Evaluación de precisión

Llegados a la parte de la evaluación de la precisión del modelo, esta se realizo utilizando los datos que resultan de la predicción mediante el uso de las herramientas proporcionadas por el keras y tensorflow; luego partiendo de estos datos se ilustraron a través la librería de python matplotlib que se utiliza para generar gráficas a partir de datos matemáticos.

Llegado a las gráficas hay que decir que basándose en la figura 23 (ver fig 23), en la parte izquierda de la ilustración se visualiza un trazo verde que nos muestra que tan bien se entreno el modelo, en donde encontramos una aceptación del 99,88%; por otra parte la linea roja muestra los datos de validación o datos de prueba, los cuales son encargados de probar los datos de entrenamiento que en este caso arrojan un porcentaje de precisión del 99,64%.

Dejando de lado la gráfica izquierda, en la aparte derecha se mira la perdida que tiene nuestro modelo es un método es decir, que tanta precisión se perdió a la hora de evaluar el algoritmo especifico que modela los datos. En esta parte cabe destacar que si las predicciones se desvían demasiado de los resultados reales, la función de perdida arrojaría un porcentaje alto, pero en este caso nuestra perdida es de 0,56% con los datos de entrenamiento y 0,66% con los de la prueba.

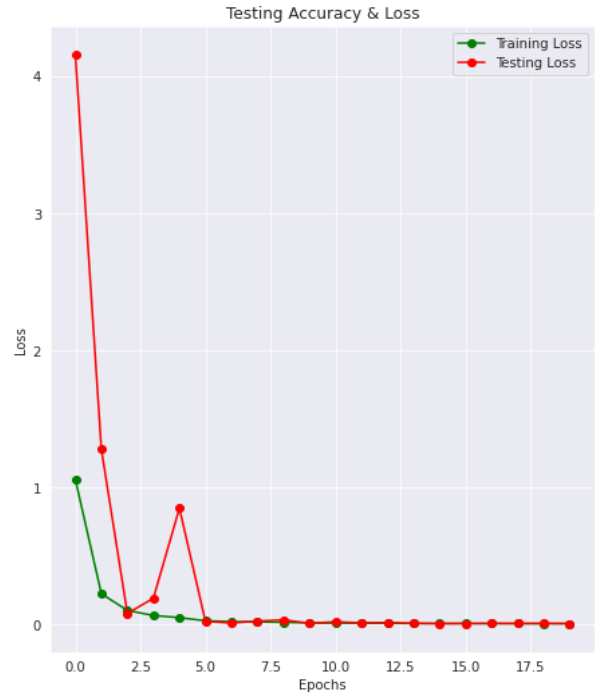
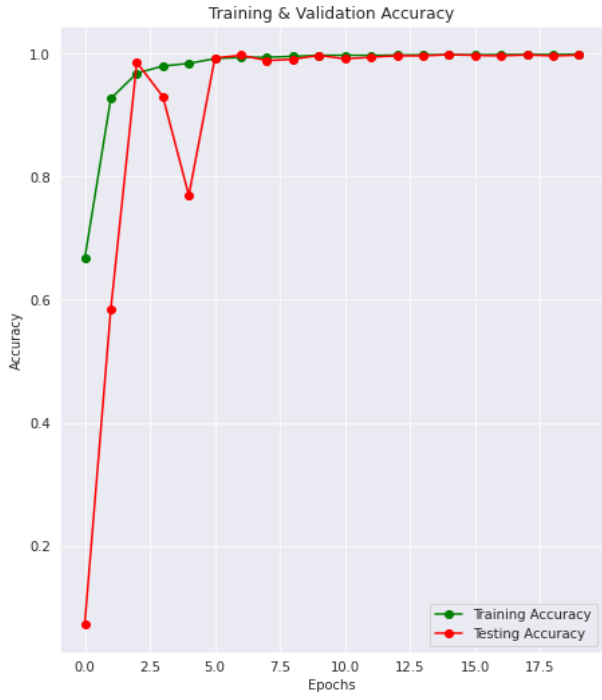


Figura 23: Gráfica de precisión y pérdida

### 5.2.9. Exportar el modelo

Cuando se acaba el entrenamiento podemos guardar nuestro modelo directamente en el drive y descargarlo para su respectiva implementación, tal y como se puede ver en el siguiente trozo de código (ver fig 24).

```
[ ] saved_model_path = "/tmp/model.h5"  
    model.save(saved_model_path)
```

*Figura 24: Exportación del modelo*

Cabe tener en cuenta que cuando tenemos el modelo se necesita hacer una transformación de la extensión .h5 ya que esta es la forma en la cual python exporta los modelos entrenados. Para hacer la conversión a un modelo que lo soporte una plataforma móvil el proceso se hace a través de una librería conocida como tensorflow lite, la cual es un conjunto de herramientas utilizadas para ayudar a los desarrolladores a implementar modelos de tensorflow en dispositivos móviles o de IoT, esta permite la inferencia de aprendizaje automático en dispositivos con una latencia baja y un tamaño de objeto binario pequeño.

En la siguiente ilustración se visualiza como se convierte un modelo de python con tensorflow lite (ver fig 25).

```
[ ] converter = tf.lite.TFLiteConverter.from_keras_model_file(model_name)  
  
[ ] tflite_model = converter.convert()  
  
[ ] with open("tf_model.tflite", "wb") as f:  
    f.write(tflite_model)
```

*Figura 25: Conversión de modelo a tensorflow lite*

### 5.2.10. Evaluación

La evaluación y consiguiente validación se realiza a juicio de expertos usando las gráficas dadas en la ilustración 17 y obtenidas del proceso de entrenamiento de la precisión del modelo respecto al grupo de imágenes de testeo y validación. Además de esto, la evaluación de usuario puede verse en el siguiente punto (5.3). Cabe resaltar que en este punto se pudo determinar que el modelo es útil según las necesidades del negocio y cumple con los objetivos para continuar con el prototipo de aplicación.

Por último, hay que decir que la ultima fase propuesta por crisp-md es la de implementación, y debido a que en este caso se utilizaron 2 metodologías, la implementación se hará a continuación pero no como parte de crisp-dm sino como parte del modelo design thinking.

### 5.3. Implementación

Para la implementación del modelo se desarrollo una aplicación para Android, la cual consume el modelo generado anteriormente y permite detectar las señales. Esta aplicación se realizo para Android, basándose en el cronograma propuesto en el anteproyecto, el cual constaba de 3 fases, las cuales son:

- Fase de investigación.
- Fase de creación de modelo.
- Fase de creación e implementación de la aplicación móvil.

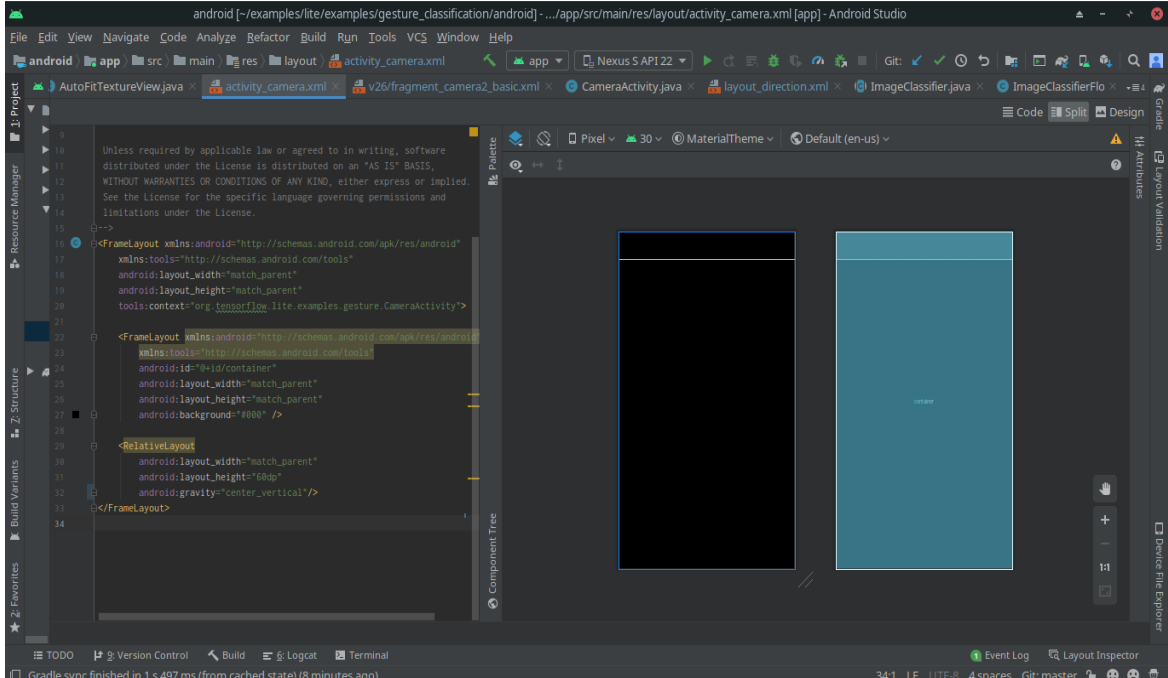
Todo esto puede verse a continuación en la tabla 6 (ver tabla 6).

Tabla 6: Cronograma

Tiempo	MES 1			MES 2				MES 3				MES 4				MES 5				MES 6				MES 7				MES 8					
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	
Fase I																																	
A1	X	X	X	X					X	X	X	X																					
Fase II																																	
B1					X	X							X							X													
B1							X	X					X						X						X								



Para esta tercer etapa, la aplicación se desarrollo utilizando Android Studio y como lenguaje de programación se utilizo Kotlin como se puede ver en las siguientes ilustraciones (ver fig 26, 27, 28).



*Figura 26: Programación inicial de app móvil*

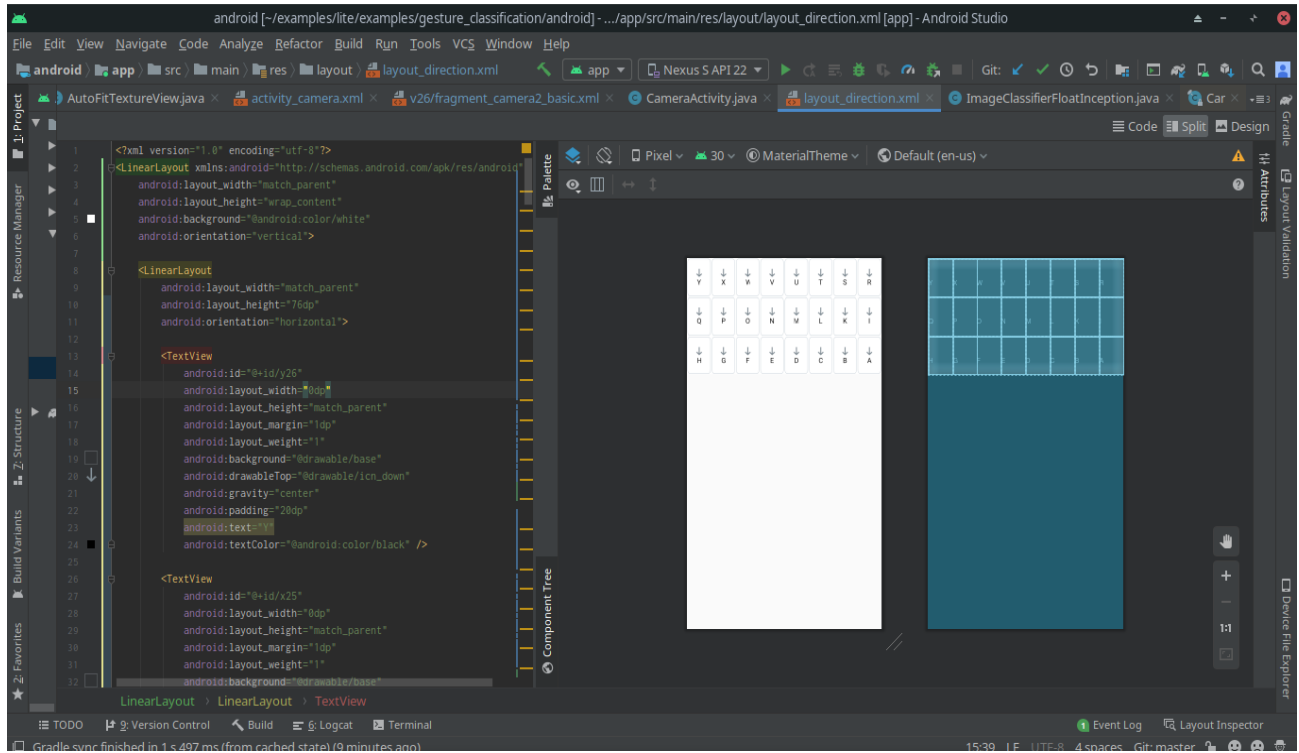


Figura 27: Creación de panel de las letras

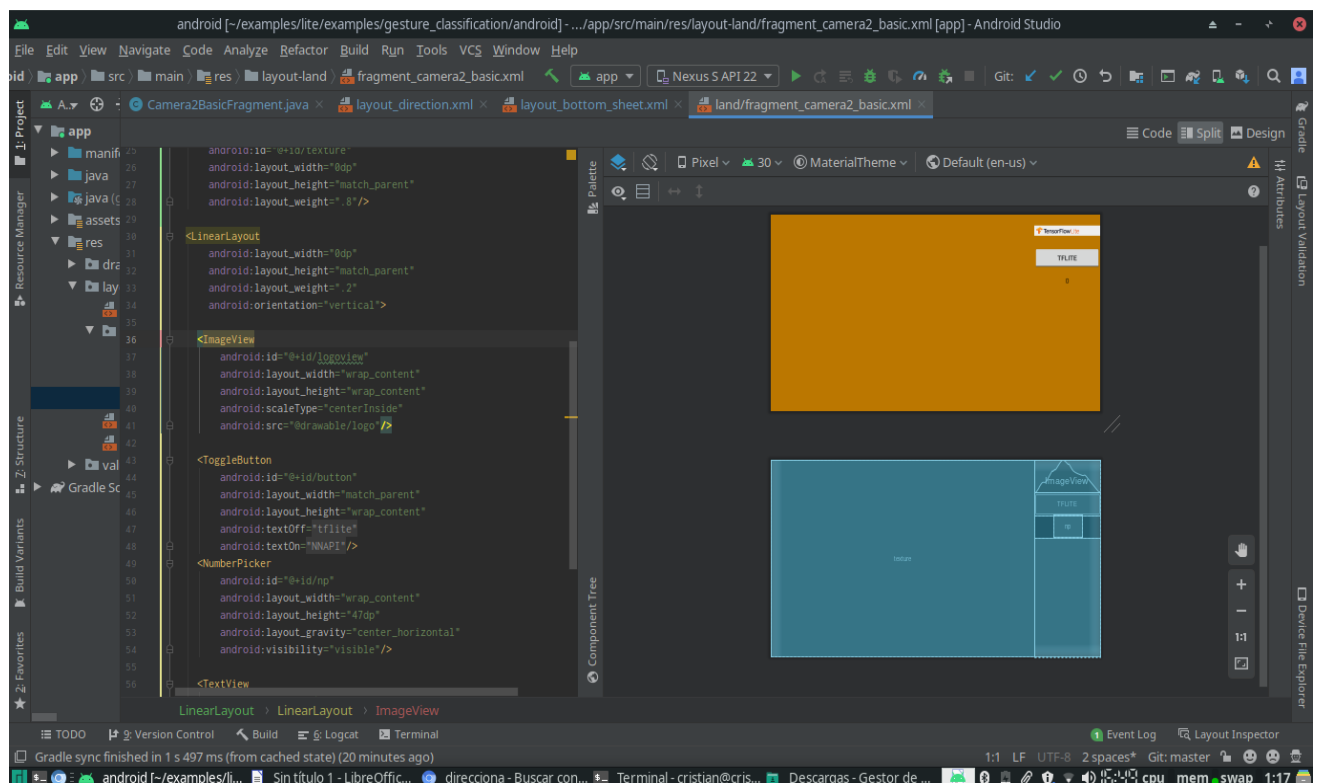


Figura 28: Implementación del módulo de la cámara

Cabe destacar que en la figura 26 fue en donde se comenzó la creación de la aplicación y donde se utilizó el modelo de deep learning, en la figura 27 es donde se creó un panel con las letras que se esperan detectar y en la 28 se agregó la interfaz de la cámara a la interfaz de usuario.

Luego de esto simplemente se realizaron unas pruebas de aceptación por parte de una tercera persona; esta persona tuvo la posibilidad de evaluar el aplicativo en 3 etapas a través de la resolución de un cuestionario correspondiente a 3 preguntas, las cuales son:

- ¿El aplicativo funciona?
- ¿Es fácil utilizar la aplicación?
- ¿Le parece cómodo el aplicativo?

Como primera etapa de aceptación, al usuario encuestado se le presentó inicialmente una interfaz de escritorio capaz de hacer la detección de las letras tal y como se puede ver en la ilustración 29 (ver fig 29).

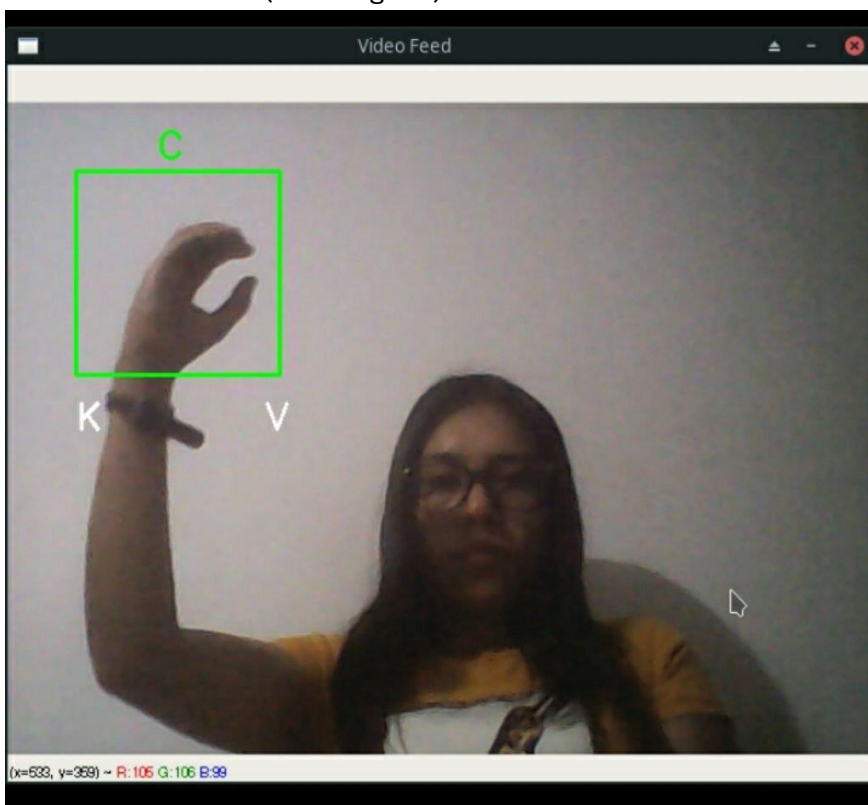


Figura 29: Prueba etapa 1 (interfaz de escritorio)

Llegada a esta etapa el usuario respondió el cuestionario con las siguientes respuestas (ver tabla 7).

Tabla 7: Pruebas de aplicación etapa 1

Pregunta	Respuesta	¿Por que?
¿El aplicativo funciona?	Si	El aplicativo detecta la seña de



na?

¿Es fácil utilizar la aplicación?

¿Le parece cómodo el aplicativo?

la mano.

Para ingresar al aplicativo, hay que utilizar una terminal para ejecutar el código en python.

No, debido a que solamente funciona en el escritorio y necesita tener python instalado.

Como se pudo ver, de cierta manera el objetivo de esta fase de implementación del modelo y de la aplicación se cumplió ya que el propósito de esta etapa era verificar si el modelo funcionaba en un entorno real y no solamente en un entorno simulado o computado.

Luego de realizada la fase 1 se prosiguió con la siguiente etapa implementación del modelo la cual consistió en la exportación de un modelo poco entrenado con tensorflow lite, esto se realizó única y exclusivamente para probar que efectivamente el modelo podría exportarse a una aplicación Android dando como resultado la aplicación que se puede ver en la figura 30 (ver fig 30).



Figura 30: Aplicativo móvil con modelo de pruebas

Después y tras la exportación de la aplicación a Android, se procedió con la evaluación por parte del usuario la cual dio como resultado lo que se puede ver en la tabla 8 (ver tabla 8).

*Tabla 8: Pruebas de aplicación etapa 2*

Pregunta	Respuesta	¿Por que?
¿El aplicativo funciona?	SI y NO	El aplicativo funciona en Android, sin embargo no detecta las letras que se le muestran a través de la cámara.
¿Es fácil utilizar la aplicación?	SI	Para utilizar la aplicación solamente hay que dar tap desde el celular.
¿Le parece cómodo el aplicativo?	SI	La interfaz es muy limpia y no requiere de la utilización de herramientas o comandos complicados para comprenderla.

La segunda fase de las pruebas puede verse ilustrada con la siguiente imagen (ver fig 31).

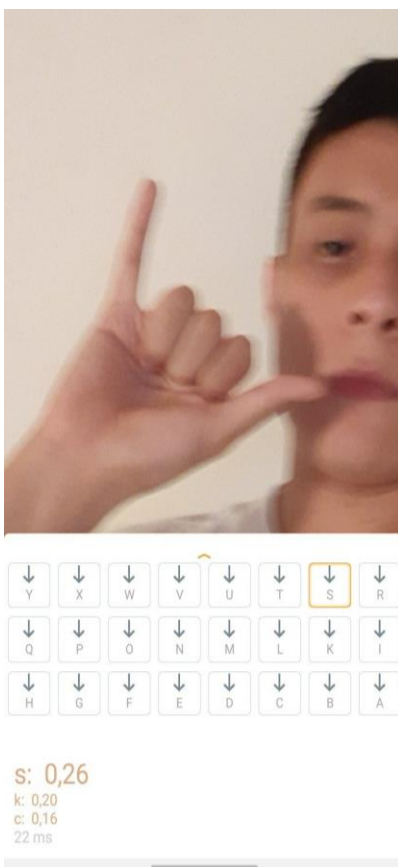


Figura 31: Prueba etapa 2 (interfaz móvil)

Cabe destacar que el signo realizado en la figura 31 corresponde a la letra Y y no a la letra S como la aplicación lo muestra; esto se debe a que el modelo transferido a la aplicación solamente era un modelo de prueba para evaluar si la transferencia con TensorFlow Lite si funcionaba de manera correcta.

Por último luego de verificar los objetivos propuestos en la fase 2, se procedió a pasar a la fase 3, en donde se evaluó el modelo completo, la exportación mediante tensorflow lite y la aplicación final en Android; para ello simplemente se registraron los datos de la encuesta al usuario la cual dio como resultado lo que se puede observar en la tabla 9 (ver tabla 9) y en la figura 32 (ver fig 32).

Tabla 9: Pruebas de aplicación etapa 3

Pregunta	Respuesta	¿Por que?
¿El aplicativo funciona?	SI	El aplicativo funciona en Android, y detecta las letras dadas por el lenguaje de señas.
¿Es fácil utilizar la aplicación?	SI	Para utilizar la aplicación solamente hay que dar tap

¿Le parece cómodo el SI aplicativo?

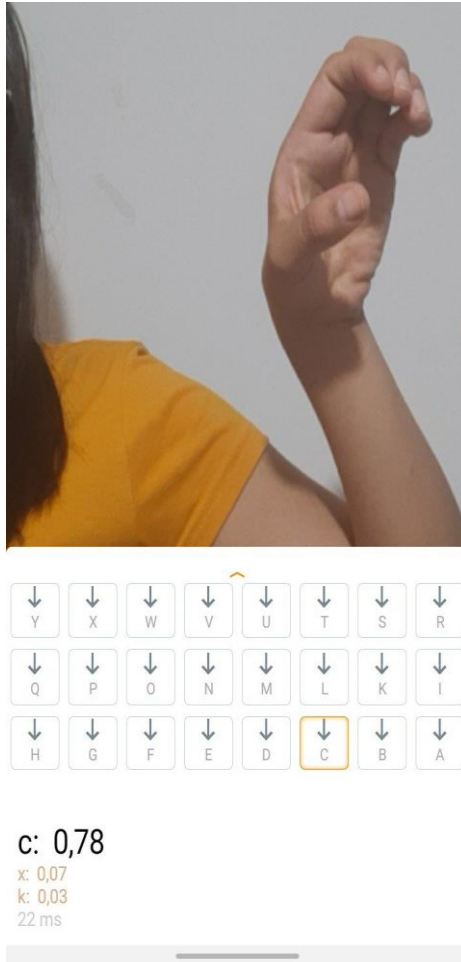
desde el celular.

La interfaz es muy limpia y no requiere de la utilización de herramientas o comandos complicados para comprenderla.



*Figura 32: Realización de encuestas de prueba*

En esta fase el aplicativo ya era capaz de detectar señales, con lo cual se concluyó la implementación del aplicativo ya que dio resultados positivos y permitió identificar las letras del alfabeto (con excepción de la J y la Z) tal y como puede verse en la ilustración 33 (ver fig 33).



*Figura 33: Prueba etapa 3 (aplicación final)*

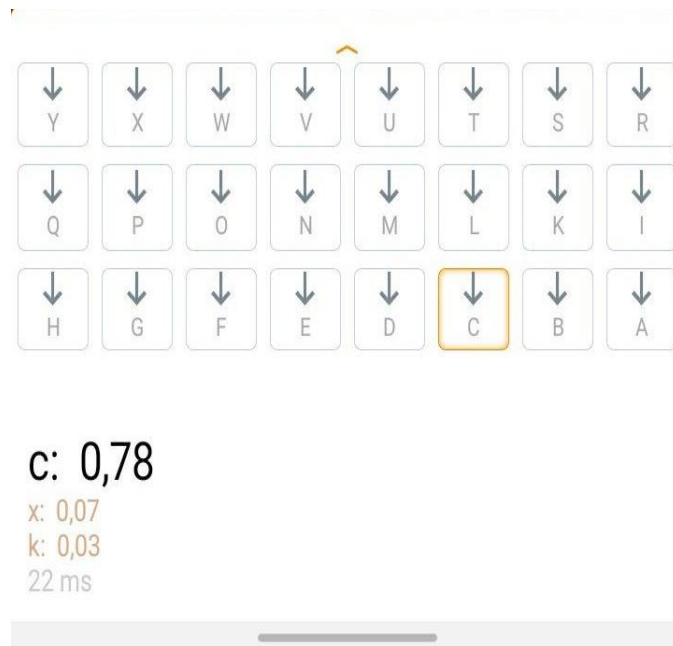
#### 5.4. Funcionamiento de la aplicación

El funcionamiento de la aplicación es muy básico, simplemente se instala la aplicación en un dispositivo Android, se ejecuta con un tap en el icono de la aplicación y por ultimo se enfoca la cámara en la mano en donde se quiere detectar la seña tal y como se observa en la figura 34 (ver fig 34).



*Figura 34: Toma de la seña con cámara móvil*

Luego de esto, la aplicación mostrara en el teclado, la letra correspondiente a la seña y las demás posibles letras como se puede ver en la figura 35 (ver fig 35).



*Figura 35: Teclado indicador de letra.*

## 6. Resultados

Los resultados se obtuvieron según los datos generados por las imágenes y datos de prueba, estos pueden ser vistos a continuación en una tabla que muestra los resultados que tiene cada imagen sobre los datos de entrenamiento extrayendo cuantas imágenes tiene cada clase en los datos de prueba (ver tabla 10),

*Tabla 10: Tabla de resultados de prueba sobre clases*

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
Class 0	0	0	0	331
Class 1	0	0	0	432
Class 2	0	0	0	310
Class 3	0	0	0	245
Class 4	0	0	0	498
Class 5	0	0	0	247
Class 6	0	0	0	348
Class 7	0	0	0	436
Class 8	0	0	0	288
Class 10	0	0	0	331
Class 11	0	0	0	209
Class 12	0	0	0	394
Class 13	0	0	0	291
Class 14	0	0	0	246
Class 15	0	0	0	347
Class 16	0	0	0	164
Class 17	0	0	0	144
Class 18	0	0	0	246
Class 19	0	0	0	248
Class 20	0	0	0	266
Class 21	0	0	0	346
Class 22	0	0	0	206
Class 23	0	0	0	267

Ademas de los resultados en la tabla anterior, cambien se puede observar en la siguiente gráfica de calor, como la aplicación y el modelo desarrollado responde a las necesidades dadas, esto puede verse a continuación (ver fig 36).

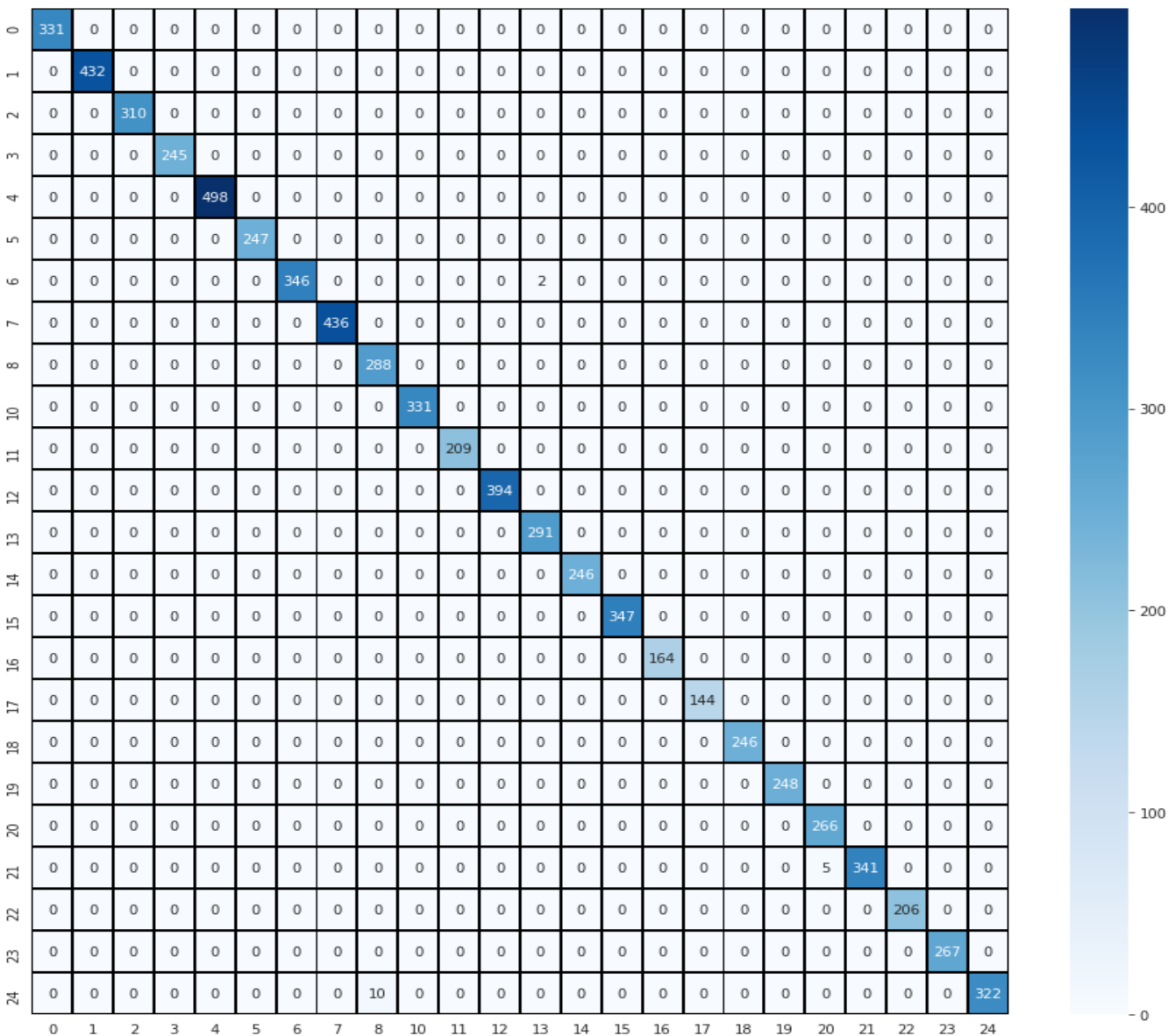
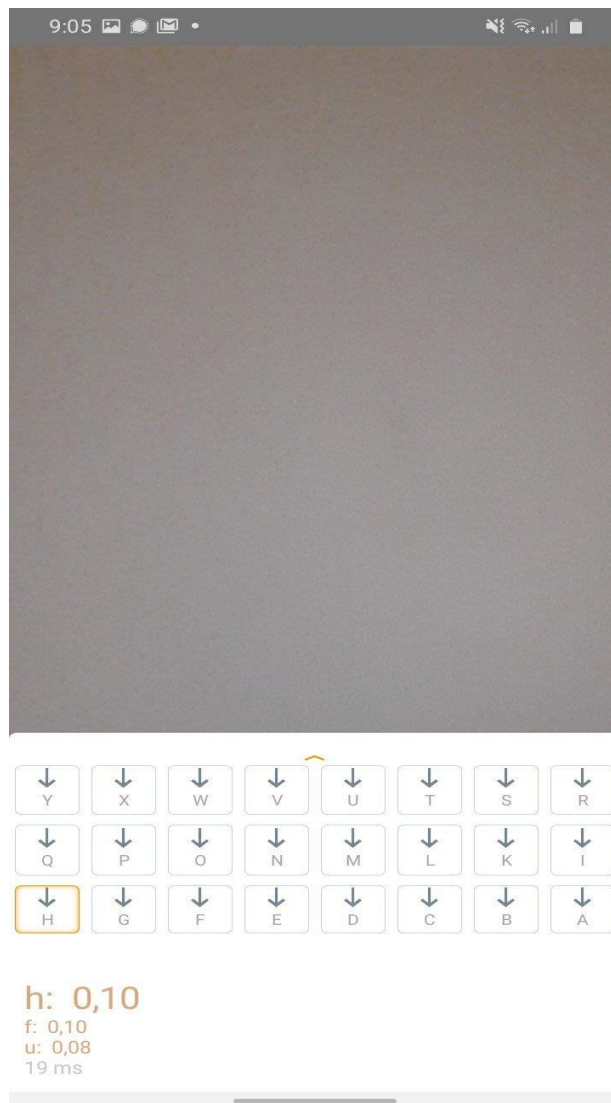


Figura 36: Mapa de calor, casos vs pruebas



Todo esto resulto en un modelo capaz de detectar las imágenes necesarias y en una aplicación fácil de utilizar, económica y portable tal y como se podrá observar en la siguiente ilustración (ver fig 37). Cabe decir que todo el funcionamiento y los resultados obtenidos con la aplicación podrán ser vistos en las conclusiones.



*Figura 37: Aplicación final*

## 7. Conclusiones

Como conclusión, se obtuvo una aplicación sencilla de utilizar por cualquier persona que tenga acceso a un dispositivo móvil, además de que se obtiene una alternativa económica, fácil de usar y accesible para todo aquel tipo de persona que necesite traducir las letras del lenguaje de señas colombiano al lenguaje español, todo esto a través de técnicas modernas de programación y haciendo uso de conceptos de machine learning y es que esta aplicación no solamente nos sirve para traducir señas al español, sino que también puede servir como soporte a un desarrollo más comercial o a un desarrollo con enfoque didáctico en la enseñanza del lenguaje de señas colombiano. En otras palabras, con este desarrollo se cumplieron los objetivos propuestos al inicio del proyecto como lo son:

- Objetivo general: Este objetivo se cumplió a cabalidad debido a que se pudo desarrollar el diseño de un prototipo que permite detectar la mayoría de las letras del abecedario a través de un algoritmo de machine learning, como se demostró en el capítulo 5.
- Objetivos específicos: Para el cumplimiento de estos se generó una investigación basándose en el estado del arte, se desarrolló un modelo y se generó una aplicación final con lo que puede decir que estos también fueron llevados a cabo en su totalidad.

Además de los objetivos cumplidos, también se obtuvo como resultado destacable el desarrollo de la interfaz de la aplicación móvil y su adaptación a las necesidades de los usuarios a los cuales está destinado el producto final ya que en su desarrollo estas fueron tenidas en cuenta como se puede observar en el capítulo 5.1.4, también se destaca el uso de algunos patrones de diseño de interfaces móviles humanas como lo es el material design creado por Google, el cual se enfoca a que el producto final sea limpio, fácil de entender y agradable visualmente; características encontradas en el resultado final que puede verse en el capítulo anterior (capítulo 6).

Cabe destacar que en este proyecto solo se señala el proceso mediante el cual se diseñó la aplicación enfocándose en que el resultado final sea fácil de utilizar por un usuario promedio y no se enfocó como tal en el desarrollo del modelo mediante deep learning, ya que si bien el modelo propuesto funciona, probablemente en un futuro sea mejor utilizar técnicas como LSTM para que los resultados sean más precisos y para que puedan hacerse sobre videos y no sobre imágenes como sucedió en el proyecto planteado anteriormente.

También hay que decir que se destaca como buen resultado el uso de herramientas y técnicas de análisis de imágenes, inteligencia artificial y machine learning en campos diferentes a las ciencias de la computación, con lo cual cada día se logra que la brecha entre diferentes comunidades se vea estrechada haciendo el uso de la tecnología.

Por último se resalta que este tipo de aplicaciones son posibles gracias al enorme soporte de las comunidades detrás de las herramientas utilizadas como lo es TensorFlow, Keras, Kotlin, Pycharm y del conocimiento que se adquirió a través de múltiples fuentes de información que hacen posible que cada día la brecha entre personas

con discapacidades y personas que carecen de estas pero que están dispuestas a mejorar el mundo.

## **8. Recomendaciones y trabajos futuros**

Para la continuación del trabajo se recomienda volver a la literatura mostrada en este documento con el fin de mejorar el modelo de aprendizaje de uno basado en CNN a LSTM para que los resultados finales sean mejores, basados en captura de vídeos o imágenes 3D para que ya no solo puedan ser detectado señas sino palabras completas, señas con movimiento como las encontradas en la letra J y Z o incluso oraciones completas. También se hace necesario una mejora en la aplicación agregando mas funciones como la de grabación, la de lectura de texto o Text To Speach (TTS), además de una mejora en la interfaz que permita que se cumplan objetivos de ergonomia los cuales no fueron evaluados en este trabajo.

## 9. Bibliografía

- [1] I. N. para Sordos, "CONTEXTO GENERAL DE LA PROBLACION SORDA EN COLOMBIA," Grupo Observatorio Social, 2016. [Online]. Available: [http://www.insor.gov.co/observatorio/download/Infog\\_pan\\_sordos\\_Col\\_sept2016.pdf](http://www.insor.gov.co/observatorio/download/Infog_pan_sordos_Col_sept2016.pdf).
- [2] I. N. para Sordos, "Indicadores educación 2015", Insor.gov.co, 2017. [Online]. Available: [http://www.insor.gov.co/observatorio/download/indicadores\\_educacion\\_2015.xlsx](http://www.insor.gov.co/observatorio/download/indicadores_educacion_2015.xlsx). [Accessed: 22- Aug- 2020].
- [3] I. N. para Sordos, "Indicadores de trabajo 2017", Insor.gov.co, 2020. [Online]. Available: [http://www.insor.gov.co/observatorio/download/indicadores\\_trabajo\\_2017.xlsx](http://www.insor.gov.co/observatorio/download/indicadores_trabajo_2017.xlsx). [Accessed: 22- Aug- 2020].
- [4] B. Berra-Novoa, R. Gonzalez-Valenzuela, and P. Shiguihara-Juarez, "Peruvian sign language recognition using low resolution cameras," in 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2018, pp. 1-4, doi: 10.1109/INTERCON.2018.8526408.
- [5] B. Berra-Novoa, R. Gonzalez-Valenzuela, and P. Shiguihara-Juarez, "Peruvian sign language recognition using low resolution cameras," in 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2018, pp. 1-4, doi: 10.1109/INTERCON.2018.8526408.
- [6] R. H. Sampieri, C. F. Collado, P. B. Lucio, and M. de la L. C. Pérez, Metodología de la investigación, vol. 1. Mcgraw-hill México, 1998.
- [7] Unesco, "Sordomudez", [Vocabularies.unesco.org](http://vocabularies.unesco.org), 2019. [Online]. Available: <http://vocabularies.unesco.org/browser/thesaurus/es/page/concept7458>. [Accessed: 23- Aug- 2020].
- [8] "Sordomudez | Definición de Sordomudez por Oxford Dictionaries | Spanish. [Online]. Available: <https://www.lexico.com/es/definicion/sordomudez>. [Accessed: 23- Aug- 2020].
- [9] R. David Reyes Pastor, "¿Cómo se detecta? | Discapacidad auditiva", [Descargas.pntic.mec.es](http://descargas.pntic.mec.es). [Online]. Available: [http://descargas.pntic.mec.es/cedec/atencion\\_diver/contenidos/nee/discapacidadauditiva/cmo\\_se\\_detecta.html](http://descargas.pntic.mec.es/cedec/atencion_diver/contenidos/nee/discapacidadauditiva/cmo_se_detecta.html). [Accessed: 23- Aug- 2020].
- [10] L. Amador Hidalgo, Inteligencia artificial y sistemas expertos. Córdoba: Servicio de Publicaciones de la Universidad de Córdoba, 1996.
- [11] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Malaysia: Pearson Education Limited, 2016.
- [12] C. Bishop, Pattern Recognition and Machine Learning, 1st ed. Springer, 2006.

- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [14] M. Y. Javed, M. M. Gulzar, S. T. H. Rizvi, M. J. Asif, and Z. Iqbal, "Implementation of image processing based Digital Dactylology Converter for deaf-mute persons," in *2016 International Conference on Intelligent Systems Engineering (ICISE)*, 2016, pp. 14-18, doi: 10.1109/INTELSE.2016.7475155.
- [15] B. Sai and T. Sasikala, "Object Detection and Count of Objects in Image using Tensor Flow Object Detection API", *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019. Available: 10.1109/icssit46314.2019.8987942 [Accessed 24 August 2020].
- [16] N. Ketkar, *Deep learning with Python*, 1st ed. Springer, 2017, pp. 97-111.
- [17] J. Sánchez, "En busca del Diseño Centrado en el Usuario (DCU): definiciones, técnicas y una propuesta", *Nosolousabilidad.com*, 2011. [Online]. Available: [http://www.nosolousabilidad.com/articulos/dcu.htm?utm\\_source=iNeZha.com&utm\\_medium=im\\_robot&utm\\_campaign=iNeZha](http://www.nosolousabilidad.com/articulos/dcu.htm?utm_source=iNeZha.com&utm_medium=im_robot&utm_campaign=iNeZha). [Accessed: 25- Aug- 2020].
- [18] D. Garrizo and C. Ortiz, "Modelos del proceso de educación de requisitos: Un mapeo sistemático," *Ing. y Desarrollo*, vol. 34, no. 1, pp. 184-203, 2016, doi: 10.14482/indc.33.2.6368.
- [19] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision", *2018 IEEE International Conference on Big Data (Big Data)*, 2018. Available: <https://ieeexplore.ieee.org/document/8622141>. [Accessed 24 August 2020].
- [20] B. Neel Kamal, V. Y and R. G. N, "Indian Sign Language Gesture Recognition using Image Processing and Deep Learning", in *International Conference on Digital Image Computing: Techniques and Applications*, Bengaluru, 2019.
- [21] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs", *Neural Computing and Applications*, vol. 32, no. 12, pp. 7957-7968, 2020. Available: 10.1007/s00521-019-04691-y [Accessed 24 August 2020].
- [22] P. Krishnan and P. Balasubramanian, "Detection of Alphabets for Machine Translation of Sign Language Using Deep Neural Net", *2019 International Conference on Data Science and Communication (IconDSC)*, 2019. Available: 10.1109/icondsc.2019.8816988 [Accessed 24 August 2020].
- [23] P. Loke, J. Paranjpe, S. Bhabal and K. Kanere, "Indian sign language converter system using an android app", *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, 2017. Available: 10.1109/iceca.2017.8212852 [Accessed 24 August 2020].
- [24] H. Plattner, "Mini guía: una introducción al Design Thinking." [Online]. Available: <https://dschool-old.stanford.edu/sandbox/groups/designresources/wiki/31fbd/attachments/027aa>

/GUÍA DEL PROCESO CREATIVO.pdf?sessionID=8af88fee76ecd1fb7879c915073461486c425622.

[25] I. B. M. IBM, “Manual CRISP-DM de IBM SPSS Modeler,” IBM Corp., p. 56, 2012.

[26] B. Berra-Novoa, R. Gonzalez-Valenzuela, and P. Shiguihara-Juarez, “Peruvian sign language recognition using low resolution cameras,” in 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2018, pp. 1-4, doi: 10.1109/INTERCON.2018.8526408.