

TRADUCTOR DE SÍMBOLOS DE ALFABETO DEL LENGUAJE DE
SIGNOS
COLOMBIANO AL LENGUAJE ESCRITO



KELLY VANESSA MONSALVE PINEDA
JULIÁN ENRIQUE POLO ÁLVAREZ

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
TULUÁ
2016

TRADUCTOR DE SÍMBOLOS DE ALFABETO DEL LENGUAJE DE
SIGNOS
COLOMBIANO AL LENGUAJE ESCRITO



KELLY VANESSA MONSALVE PINEDA
JULIÁN ENRIQUE POLO ÁLVAREZ

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO
DE
INGENIERO EN SISTEMAS

DIRECTOR:
ALBEIRO APONTE, MSC

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
TULUÁ
2016

NOTA DE ACEPTACIÓN

Firma del Evaluador

Firma del Director

Tuluá, Valle del Cauca __/__/__

DEDICATORIAS

*La razón por la cual nunca renuncié a la idea de un día culminar mi
proceso académico
Fuiste tú, Saray Ocampo Monsalve, viviste todo este proceso conmigo,
fueron muchos los momentos en que sentí que no podría lograrlo sin
embargo continué, necesito contarte esta historia en un futuro para que
entiendas que aunque las cosas se vuelvan difíciles y casi imposible, se
puede lograr lo que uno se propone.*

*Dedico este trabajo a tres personas más que fueron fundamentales en
este logro, mi esposo Iván Ocampo de quien aprendí mucho en todo este
tiempo, mi madre Blanca Nidia Pineda quien fue mi consejera y amiga en
este proceso y a mi suegra María Consuelo Rivera por su apoyo
incondicional.*

Kelly Vanessa Monsalve Pineda

*Les dedico este trabajo a todas las personas que tienen dificultad para
hablar y que son discriminados por esto hoy en día.*

A mi familia por ayudarme en todo en mi carrera

*Y a todos quienes estuvieron detrás de mí apoyándome en todo
momento.*

Julián Enrique Polo Álvarez

AGRADECIMIENTOS

La realización del presente proyecto titulado: “Traductor de símbolos del alfabeto del lenguaje de signos Colombiano (LSC) al lenguaje escrito”, requirió de gran esfuerzo pero no habría sido posible sin el apoyo incondicional de personas que siempre creyeron en nosotros.

Agradecemos primeramente a Dios por brindarnos la oportunidad de obtener formación profesional, a nuestras familias porque su respaldo y comprensión nos dieron las bases para afrontar los momentos difíciles en nuestra carrera.

A nuestros profesores por brindarnos sus conocimientos durante este proceso y en especial, a nuestro director Albeiro Aponte por nunca faltar al compromiso que adquirió con nosotros, por su constante ayuda y por brindarnos las herramientas para culminar este trabajo.

Resaltamos la colaboración del compañero Álvaro Andrés Loaiza, gracias por su ayuda desinteresada en dudas acerca del Lenguaje de Señas Colombiano y en el proceso de pruebas.

A nuestros compañeros por todo el apoyo brindado y todas aquellas personas que de una u otra forma aportaron en la realización y culminación de este proyecto.

TABLA DE CONTENIDO

	<i>Pág.</i>
RESUMEN	1
1. INTRODUCCIÓN	2
1.1 Descripción general	2
1.2 Planteamiento del problema	2
1.3 Objetivos	4
1.3.1 Objetivo general	4
1.3.2 Objetivos específicos	4
1.4 Estructura del documento	4
2. MARCO REFERENCIAL	5
2.1 Marco teórico	5
2.1.1 Procesamiento Digital de Imágenes	5
2.1.2 Sistema de Reconocimiento de Patrones	7
2.1.3 DTW (Dynamic Time Warping)	8
2.1.4 Abecedario del lenguaje de señas Colombiano (LSC)	10
2.1.5 Librería Opencv	11
2.2 Marco legal	11
2.2.1 Ley 324 de 1996	11
2.2.2 Ley estatutaria 1618 del 27 febrero 2013	12
2.3 Marco conceptual	12
2.3.1 Visión Computacional o Visión Artificial	12
2.3.2 Patrón	12
2.3.3 Clase de Patrones	12
2.3.4 Detección de Bordes	12
2.3.5 Diversidad Funcional (Comunicativa)	12
2.3.6 Series de Tiempo o Series Temporales	12
2.3.7 Distancia Euclidiana	13
2.3.8 Contorno de una Imagen	13

2.4 Estado del arte	13
2.4.1 Antecedentes	13
2.4.2 Desarrollos en Colombia	13
2.4.3 Desarrollos en otros países	14
3. DESARROLLO DEL PROYECTO	15
3.1 Análisis	15
3.1.1 Procesamiento digital de imagen	16
3.1.2 Creación de patrones	16
3.1.3 Clasificador	17
3.2 Diseño	17
3.2.1 Arquitectura	17
3.2.2 Diagrama de Componentes	18
3.2.3 Diagrama de Clases	19
3.2.4 Modelo de Datos	21
3.3 Implementación	22
3.3.1 Codificación	22
3.3.2 Instalación	22
3.3.3 Comportamiento de la aplicación	22
4. PRUEBAS Y DISCUSIÓN DE RESULTADOS	28
4.1 Prueba No. 1	28
4.2 Prueba No. 2	28
4.2.1 Comportamiento del sistema	30
4.2.2 Análisis de cada señal de la tabla de resultados	32
4.2.3 Porcentaje de acertación del sistema	33
4.3 Prueba No.3	34
4.4 Prueba No. 4	35
5. CONCLUSIONES Y PROYECCIONES	37
5.1 Conclusiones	37
5.2 Proyecciones	39

REFERENCIAS	40
ANEXOS	44

LISTA DE TABLAS

	<i>Pág.</i>
Tabla1: Participación porcentual de la población sorda	3
Tabla 2: Sujeto modelo	28
Tabla 3: Resultados de 15 sujetos de prueba al evaluar cada letra del abecedario	29
Tabla 4: Prueba de deletreo 1	35
Tabla 5: Prueba de deletreo 2	35
Tabla 6: Prueba de deletreo 3	35
Tabla 7: Prueba de deletreo 4	35
Tabla 8: Prueba de deletreo 5	35
Tabla 9: Prueba de deletreo 6	35
Tabla 10: Prueba de deletreo 7	35
Tabla 11: Prueba de deletreo 8	36
Tabla 12: Prueba de deletreo 9	36

LISTA DE FIGURAS

	<i>Pág.</i>
Figura 1: Esquema del Procesamiento Digital de Imágenes	5
Figura 2: Alineación temporal de dos secuencias temporales	8
Figura 3: Abecedario del LSC	10
Figura 4: Abecedario del lenguaje de señas	15
Figura 5: Arquitectura	18
Figura 6: Diagrama de componentes	18
Figura 7: Diagrama de clases	20
Figura 8: Estructura de archivo	21
Figura 9: Modelo de datos	21
Figura 10: Comportamiento del sistema	23
Figura 11: Segmentación de la imagen	24
Figura 12: Extracción de características	26
Figura 13: Y vs I	30
Figura 14: B vs E	31
Figura 15: M vs N	31
Figura 16: V vs W	32
Figura 17: Porcentaje de acertación de cada seña del abecedario LSC	32
Figura 18: Porcentaje de acertación de 15 sujetos de prueba	33
Figura 19: Evidencia fotográfica de la prueba realizada a 15 personas	44

RESUMEN

El presente trabajo de grado consiste en el reconocimiento de las señas estáticas del abecedario del Lenguaje de Señas colombiano (LSC) con el fin de promover la inclusión social de las personas con diversidad funcional en el habla por medio de la tecnología. El proyecto se desarrolló bajo la metodología de Proceso Unificado Ágil (AUP), adoptando una arquitectura Modelo Vista Controlador (MVC) y un paradigma de programación orientado a objetos en la codificación. En su implementación se utilizan técnicas de procesamiento digital de imágenes para la segmentación, el mejoramiento de la imagen y proyección de contornos para la extracción de características. La clasificación de la seña se realiza por medio del algoritmo Dynamic Time Warping (DTW). Como resultado, el sistema logro reconocer en promedio el 79% de las señas estáticas con una desviación estándar del 6%.

ABSTRACT

This graduation work is the recognition of static sign alphabet Colombian language (LSC) in order to promote social inclusion of people with disabilities in speech through technology. The project was developed under the methodology of Agile Unified Process (AUP), adopting architecture Model View Controller (MVC) and coding object oriented programming paradigm. In its implementation techniques for digital image processing segmentation, image enhancement and contour projection for feature extraction they are used. The classification of the signal is performed by Dynamic Time Warping algorithm (DTW). As a result, the system will recognize achievement on average 79% of static signals with a standard deviation of 6%.

1. INTRODUCCIÓN

1.1 Descripción general

El Lenguaje de Señas Colombiano LSC, es la manera que utilizan las personas con dificultad en el habla para comunicarse con sus semejantes y con individuos que no presenten la mencionada dificultad. El cual parte de símbolos que son representados por medio de los dedos de las manos y facciones del rostro; estos símbolos caracterizan las letras del abecedario, palabras específicas y los números arábigos.

El Lenguaje de Señas Colombiano LSC no es conocido por todas las personas del país y en consecuencia se dificulta la comunicación entre los individuos que tienen diversidad funcional en el habla hacia aquellos que no la poseen. Para tal fin, se pretende desarrollar una herramienta tecnológica que sea un prototipo de lo que podría ser un traductor del abecedario del lenguaje de señas que muestre la utilización de las TICs en el apoyo del proceso de comunicación entre oyentes y personas sordas fomentando de esta manera su inclusión social en todas las áreas de la sociedad, iniciando así un camino hacia el desarrollo de herramientas tecnológicas que faciliten una comunicación interactiva entre todos los miembros de una comunidad sin que sea impedimento las diferencias comunicativas que existe entre el lenguaje hablado y el lenguaje de señas.

1.2 Planteamiento del problema

El lenguaje de señas no es un lenguaje universal, por lo tanto, el uso de los símbolos varían según el contexto cultural en donde se encuentre la persona que aprende dicha forma de comunicación, en pocas palabras, no es el mismo lenguaje de señas de Colombia al de otros países[1].

Si bien, quienes mayor uso hacen de este lenguaje son las personas con diversidad auditiva y en el habla; existen individuos que no poseen tales limitaciones, que se interesan por aprender tal forma de comunicación (intérpretes), sin embargo, son minoría y esto hace que los sujetos que por necesidad deben utilizar el sistema de señas, se encuentren en una incomprensión (comunicativa, claro está) frente a las personas hablantes.

Frente a la situación mencionada, actualmente el gobierno colombiano con la ley estatutaria 1618 del 2013 que garantiza el pleno ejercicio de los derechos de las personas con incapacidad, tiene un firme paso respecto a la inclusión social de todas aquellas personas sordas o sordomudas, y en conjunto con las TICs (Tecnologías de la Información y la Comunicación) pretende integrar a las personas con o sin discapacidad en un mismo entorno. En lo que respecta a Colombia, equivale a la integración de 455.718 personas con limitaciones para oír puesto que según el DANE (Departamento Administrativo Nacional de Estadística) hasta el año 2005 se contó la participación porcentual de la población sorda, referenciada en

la tabla 1[2]:

Tabla1: Participación porcentual de la población sorda

Población sin ninguna limitación	41.468.384	93,08%
Población con alguna discapacidad	2.624.898	5,89%
Población con limitaciones para oír	455.718	1,02%
Total Población Colombiana en 2005	44.549.000	100,00%

[Fuente:www.insor.gov.co/observatorio/participacion-porcentual-de-la-poblacion-sorda/]

Con relación a lo mencionado, las TIC se ubican como alternativa para la interpretación del lenguaje que utilizan los sujetos con diversidad funcional en el habla, y como evidencia de esto se encuentran algunas aplicaciones que intentan facilitar la comunicación para estas personas; sin embargo dichas aplicaciones son sistemas unidireccionales que se basan únicamente en traducir texto del lenguaje hablado a signos en la lengua de señas como lo son: Signslator capaz de traducir al instante el español al lenguaje de signos[34], un sistema de traducción de lenguaje de dactilológico basado en redes neuronales desarrollado por la escuela de ingeniería de Antioquia - Universidad CES[31] y un traductor de español a lenguaje de señas online que funciona a través de un intérprete animado llamado Iris disponible en la página web HETAH[33]. Del mismo modo existen investigaciones en el campo informático que traducen simbología del alfabeto de la lengua de señas a texto, pero su alcance no va más allá de reconocer letras del abecedario ya que no le dan sentido a ese conjunto de letras traducido, por consiguiente no es posible transmitir una idea. De no contar con herramientas tecnológicas que contribuyan a este fin, se hace difícil lograr una inclusión social en todos los ámbitos de la vida de las personas que no manejan el lenguaje hablado.

Por lo tanto y teniendo en cuenta lo anteriormente descrito, se hace necesario desarrollar aplicaciones informáticas que permitan a las personas sordomudas transmitir oraciones a individuos que no comprendan su lengua en situaciones cotidianas, aplicaciones que formen palabras a partir de los signos traducidos, de esta manera posibilitar una comprensión del lenguaje de señas por parte de la persona que se comunica con el lenguaje hablado.

- **Formulación del problema**

¿Cómo apoyar el proceso de comunicación de las personas con diversidad funcional en el habla hacia los individuos hablantes por medio de las Tecnologías de la Información y la Comunicación (TIC)?.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una aplicación de escritorio que traduzca ciertas letras del abecedario del LSC (Lenguaje de Señas Colombiano) al lenguaje escrito formando la palabra que se desea comunicar.

1.3.2 Objetivos específicos

- Determinar el abecedario del Lenguaje de Señas Colombianos para interpretarlo.
- Extraer información de la imagen por medio de los métodos de mejoramiento y segmentación basados en PDI.
- Crear un modelo de datos de 10 letras del abecedario del LSC a traducir.
- Clasificar el conjunto de caracteres capturados para su interpretación.
- Validar la aplicación con un conjunto de imágenes.

1.4 Estructura del documento

El presente documento se puede resumir en 4 apartados en los que básicamente se describe y plantea el problema y se muestra la solución propuesta con sus respectivos resultados. Estos son:

En el capítulo 1 se muestra una descripción general del contexto en el que se desarrolla el trabajo, la formulación del problema que se quiere solucionar y se presentan los objetivos del trabajo.

En el capítulo 2 se muestran los antecedentes y situación actual de las soluciones propuestas para tratar el problema presente entre la comunicación de las personas sordas y/o mudas hacia las personas hablantes y se presenta el marco teórico sobre los temas de estudios en los que se fundamenta el trabajo.

En el capítulo 3 se describe la metodología empleada durante el desarrollo del trabajo, se realiza el diseño del sistema, utilizando los conocimientos de los capítulos anteriores, aquí se definen las partes que constituyen este sistema y todos los algoritmos aplicados en cada uno.

En el capítulo 4 se efectúa la evaluación del sistema, mediante los experimentos realizados al mismo. Los datos obtenidos serán analizados para posteriormente realizar las conclusiones.

2. MARCO REFERENCIAL

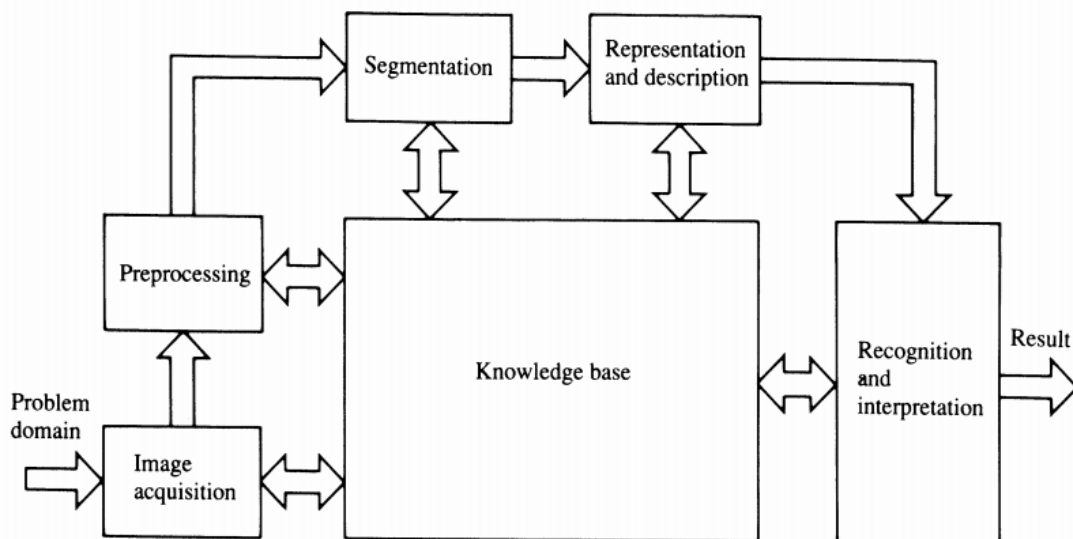
2.1 Marco teórico

El marco teórico del trabajo se formula desde cinco referencias teóricas: la teoría del PDI, la teoría del sistema de reconocimiento de patrones, el algoritmo DTW (Dynamic Time Warping), el abecedario del lenguaje de señas Colombiano y la teoría sobre la librería OpenCV, que se integraron y se desarrollaron para la construcción de este proyecto en beneficio de las personas con dificultad en el habla.

2.1.1 Procesamiento Digital de Imágenes

En la figura 1 se muestra de manera general, el procedimiento paso a paso del procesamiento de imágenes.

Figura 1: Esquema del Procesamiento Digital de Imágenes



[Fuente: http://users.dcc.uchile.cl/textasciitilde{}jsaavedr/libros/dip_gw.pdf]

- **Adquisición de la Imagen:** La operación básica en los sistemas de adquisición de imágenes y video es la especificación de la intensidad (color) de la imagen en un arreglo regular de puntos en el espacio, y en el tiempo [3]. La adquisición de la imagen está a cargo de algún transductor o conjunto de transductores que mediante la manipulación de la luz o de alguna u otra forma de radiación que es emitida o reflejada por los cuerpos, se logra formar una representación del objeto dando lugar a la imagen [4].

- Pre procesamiento de la Imagen: Mejorar la imagen para aumentar la posibilidad de éxito en las etapas siguientes [5]. En este caso se hace uso de las operaciones de mejora de calidad, algunas cuales son:
 - Brillo: se define como la intensidad de la luz en cada píxel de una imagen digital, al visualizar una imagen con variaciones en el brillo, esta se verá mucho más clara o más oscura de acuerdo a qué tanto brillo tendrá esta. Cuando se varía el brillo en una imagen todos los píxeles de la imagen modifican su luminosidad en igual cantidad [6].
 - Contraste: se define como la posibilidad de distinguir dos o más densidades distintas, esto quiere decir que a mayor contraste las partes oscuras de la imagen se diferenciarán con mayor notoriedad de las zonas luminosas [6].
 - Corrección gamma: Define la relación que hay entre el valor numérico de un píxel y su actual luminosidad, sin el gamma las sombras que aparecen cuando se toma una imagen con una cámara digital no serán notadas por nuestros ojos [7].
 - Filtro de la mediana: Dada una imagen $f(i, j)$, el procedimiento consiste en generar una nueva imagen $g(i, j)$ cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles $f(i, j)$ incluidos en un entorno de vecindad predefinido [8].
- Segmentación: El análisis de imágenes comprende todos los métodos y técnicas que se utilizan para extraer información de una imagen. El primer paso para ello lo constituye la segmentación de imágenes que consiste en la división o partición de la imagen en varias zonas o regiones homogéneas y disjuntas a partir de su contorno, su conectividad, o en términos de un conjunto de características de los píxeles de la imagen que permitan discriminar unas regiones de otras [9]. En este proyecto, para segmentar la imagen, se utiliza los siguientes métodos:
 - Imagen en escala de grises: Es la representación de una imagen en la que cada pixel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende entre blanco y negro [10].
 - Umbralización: Es uno de los más importantes métodos de segmentación. El objetivo es convertir una imagen en escala de grises a una nueva con sólo dos niveles, blanco y negro, de manera que los objetos queden separados del fondo [11].
 - Erosión: La erosión de una imagen binaria A por un elemento estructurante B produce una nueva imagen binaria $g = Ab$ con los de todas las ubicaciones (x, y) de origen de un elemento estructurante en la que este elemento estructurante B se adapte a la imagen de entrada A , es decir, $g(x, y) = 1$ si B encaja en A y 0 en caso contrario, se repetirá en todas las coordenadas de píxel (x, y) [12,13].

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

- Dilatación: La dilatación de una imagen A por un elemento estructurante B produce una nueva imagen binaria $g = Ab$ con los de todas las ubicaciones (x, y) de origen de un elemento estructurante en la que este elemento estructurante B se adapta en la imagen A , es decir, $g(x, y) = 1$ si B encaja en A y 0 en caso contrario, se repetirá en todas las coordenadas de píxel (x, y) . Cabe aclarar que la dilatación tiene el efecto opuesto a la erosión, se añade una capa de píxeles a ambos de los límites interior y exterior de las regiones [12,13].

$$A \oplus B = \bigcup_{b \in B} A_b$$

- Representación y descripción: Las técnicas de segmentación producen datos en bruto en forma de píxeles de un contorno o de una región. Aunque algunas veces estos datos se utilizan de esta manera para obtener descriptores, la práctica común es utilizar esquemas que compacten los datos en representaciones que son más útiles en el cálculo de descriptores. Algunas formas de representación son [14]: códigos de Cadena, se utilizan para representar un contorno por medio de una sucesión conexas de segmentos de longitud y dirección especificadas; Aproximaciones Poligonales, el objetivo es captar la esencia de la forma del contorno, con un polígono con el menor número de lados posible; Firmas, es una representación funcional unidimensional de un contorno y se puede generar de varias formas.

2.1.2 Sistema de Reconocimiento de Patrones

El objetivo del reconocimiento de Patrones es asignar un patrón a la clase que pertenece lo más automáticamente posible. El reconocimiento de patrones opera en la segmentación de la imagen de los objetos de nuestro interés, luego clasificarlos con un conjunto definido de patrones [15]. Los métodos usados para clasificar patrones son:

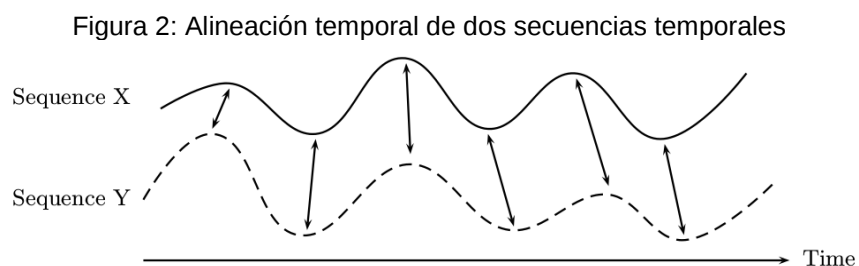
- Métodos basados en alineamiento de características: Estos métodos consisten en la comparación de la muestra tomada con otra almacenada previamente, denominada plantilla. Ejemplos de este tipo de clasificadores son el LSC (Longest Common Subsequence), TWLSC (Time-Warped Longest Common Subsequence), el DTW (Dynamic Time Warping) o lo que es lo mismo el Alineamiento Temporal Dinámico, en este se basan los algoritmos anteriores. En nuestro caso de estudio, es el algoritmo utilizado y sobre el cual profundizaremos más adelante [16].

- Métodos basados en modelos estadísticos: Los métodos basados en modelos estadísticos se basan en el uso de patrones de referencia para elaborar modelos probabilísticos y/o estadísticos. Los métodos más empleados de esta área son el algoritmo HMM (Hidden Markov Models), es decir, los Modelos Ocultos de Markov y el GMM o Modelos de Mezclas de Gaussianas [16].
- Métodos basados en fronteras de decisión: Consiste en crear unas fronteras entre clases en función de un determinado criterio de error, el cual se forja a partir de la relación entre los resultados obtenidos y los deseados. En este campo destacan los árboles de decisión, las redes neuronales y las máquinas de vectores de soporte [16].

2.1.3 DTW (Dynamic Time Warping)

Es una técnica conocida para encontrar un equilibrio óptimo de alineamiento entre dos secuencias dadas, bajo ciertas restricciones. Intuitivamente, las secuencias se deforman de manera no lineal a coincidir entre sí. Originalmente, DTW se ha utilizado para comparar los diferentes patrones del habla en el reconocimiento automático de voz. En campos como la minería de datos y recuperación de información [17]. Posteriormente se descubrieron sus buenas cualidades en el reconocimiento de imágenes, como en el estudio de Sato y Kogure donde se reconocen firmas en la fotografía de un documento [16] o en el reconocimiento de imágenes de documentos que contienen palabras y textos [18].

El algoritmo DTW realiza un alineamiento entre dos secuencias de vectores de distinta longitud mediante programación dinámica. De dicho alineamiento se obtiene una medida de distancia entre los dos patrones temporales. Como se observa en la figura 3.



[Fuente:

http://www.springer.com/cda/content/document/cda_downloaddocument/9783540740476-c1.pdf?SGWID=0-0-45-452103-p173751818.]

Modelo matemático

A continuación se muestra el modelo matemático sobre el cual se basa el algoritmo DTW [19].

Se asume al comparar dos series temporales:

$$X = \{x_1, \dots, x_n\} \wedge Y = \{y_1, \dots, y_m\}$$

Para cada $i = 1 \dots n$ y $j = 1 \dots m$

Se define una función f para cada par de elementos x_i y y_j que deben cumplir

$$d(i, j) = f(x_i, y_j) \geq 0$$

Donde d es la matriz de distancias entre los vectores X y Y , f es la distancia euclidiana entre cada par de elementos x_i y y_j

La técnica de semejanza está dada por:

$$\varphi(k) \text{ con } k = 1 \dots T$$

$$\varphi(K) = \varphi x(k), \varphi y(k)$$

$$\text{con } \varphi x(k) \in \{1 \dots n\} \wedge \varphi y(k) \in \{1 \dots m\}$$

La función de semejanza $\varphi x(k)$ y $\varphi y(k)$ re-asigna el índice de temporal de X y Y respectivamente, dando un φ , computando un promedio acumulado de la distorsión entre las series temporales deformadas de X y Y

$$d\varphi(x, y) = x \\ k = 1 \dots T$$

$$d(\varphi x(k), \varphi y(k)) m\varphi(k) / M\varphi$$

Donde $m\varphi(k)$ es un paso de coeficiente de ponderación y $M\varphi$ es la correspondiente constante normalizadora, que asegura que las distorsiones acumuladas sean comparadas a lo largo de diferentes trayectorias

$$\varphi x(k+1) \geq \varphi x(k)$$

$$\varphi y(k+1) \geq \varphi y(k)$$

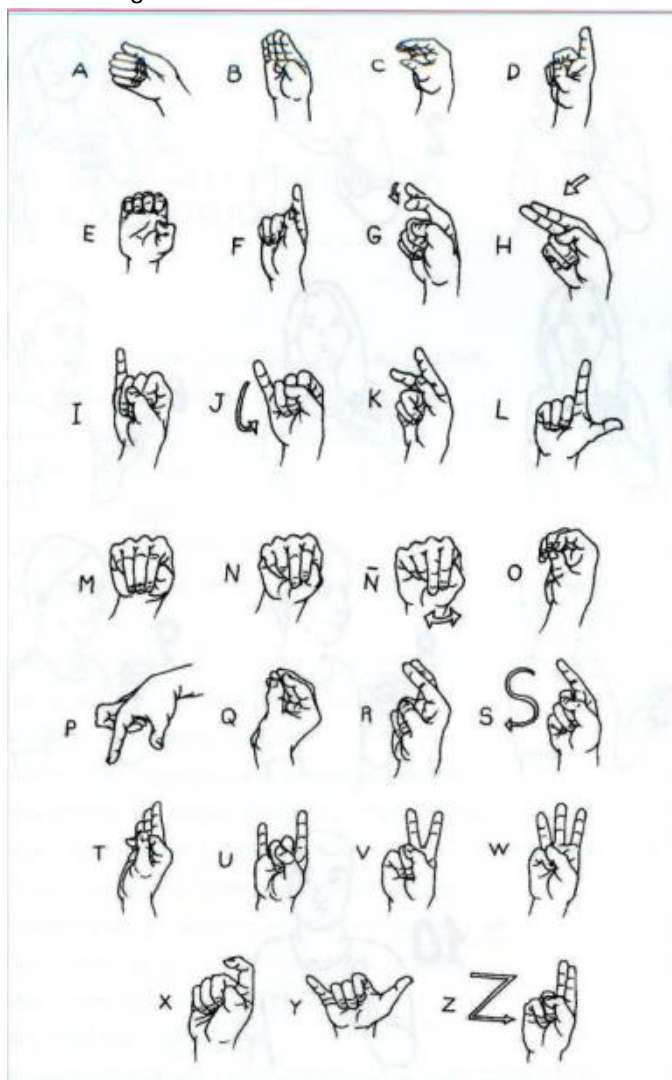
El propósito del DTW es encontrar la alineación óptima φ de

$$D(x, y) = \min d\varphi(x, y)$$

2.1.4 Abecedario del Lenguaje de Señas Colombiano (LSC)

Tanto un individuo común y corriente como los especialistas que se han dado a la tarea de estudiar una lengua, saben que uno de los primeros procesos para iniciarse en este oficio es aprendiendo lo más fundamental de cualquier dialecto: el abecedario. En Colombia, El abecedario de signos se compone de 27 señas que representan sus homólogas en el alfabeto escrito, algunas de estas señas requieren un movimiento extra como lo son las señas de las letras G, J, Ñ, S, Z. En la figura 2 se puede observar el abecedario completo:

Figura 3: Abecedario del LSC



[Fuente:<http://www.lenguasdecolombia.gov.co/sites/lenguasdecolombia.gov.co/files/lengua%20de%20se%C3%B1as.pdf>]

2.1.5 Librería Opencv

Opencv es una librería de computación visual creada por Intel, esta librería está disponible para múltiples plataformas como Windows, Linux, Mac, Android, además cuenta con soporte para diferentes lenguajes como Python, Java, C/C++, entre otros. Opencv puede ser usado bajo licencia BSD para proyectos escolares o comerciales, Opencv puede ser usado en la robótica, análisis de imágenes o vídeo, seguimiento de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos y más [20].

En la realización del presente proyecto, la librería fue de gran utilidad para realizar el procesamiento de la imagen. Entre todas sus funciones se utilizó:

- `cvtColor`: Para convertir la imagen de entrada en RGB a escalas de grises.
- `threshold`: Para realizar la umbralización.
- `medianBlur`: Aplicación del filtro de la mediana.
- `dilate` y `erode`: dilatación y erosión.
- `canny`: es una potente función que tiene la librería para detección de bordes.

Además de estas, se utilizó otros componentes para el ajuste de iluminación y ajuste de la cámara.

2.2 Marco legal

En Colombia existen leyes que promueven la igualdad y la no discriminación hacia personas con diversidad funcional. Algunas de estas son:

2.2.1 Ley 324 de 1996

Por la cual se crean algunas normas a favor de la población sorda. En ésta, el Estado aprueba la lengua de señas como oficial de la comunidad sorda y se plantea la investigación y difusión de la misma, se prevé la introducción de tecnologías y el servicio de intérpretes [21].

2.2.2 Ley estatutaria 1618 del 27 febrero 2013

Por medio de la cual se establecen las disposiciones para garantizar el pleno ejercicio de los derechos de las personas con discapacidad. El objeto de esta ley es garantizar y asegurar el ejercicio efectivo de los derechos de las personas con discapacidad, mediante la adopción de medidas de inclusión, acción afirmativa y de ajustes razonables y eliminando toda forma de discriminación por razón de discapacidad [22].

2.3 Marco conceptual

2.3.1 Visión Computacional o Visión Artificial: Es darle a una máquina la facultad del ojo humano y que ella pueda interpretar la imagen captada dando un sentido apropiado. Es el estudio de los procesamiento de imágenes, para entenderlos y construir máquinas con capacidades iguales a las humanas [23].

2.3.2 Patrón: Conjunto de características discriminantes de un objeto [24].

2.3.3 Clase de Patrones: Conjunto de Patrones similares [24].

2.3.4 Detección de Bordes: Usado principalmente en la detección de la forma descrita por un objeto en una imagen; El funcionamiento de los algoritmos para la detección de bordes consisten principalmente en ver la relaciones de intensidad entre los píxeles vecinos que rodean a un determinado píxel, esta relación está dada por una determinada máscara de convolución [25].

2.3.5 Diversidad Funcional (Comunicativa): Es un término alternativo al de discapacidad que ha comenzado a utilizarse en España por iniciativa de los propios afectados. El término fue propuesto en el Foro de Vida Independiente, en enero de 2005, y pretende sustituir a otros cuya semántica puede considerarse peyorativa, tales como “discapacidad” o “minusvalía”. Se propone un cambio hacia una terminología no negativa, no rehabilitadora, sobre la diversidad funcional [26].

2.3.6 Series de Tiempo o Series Temporales: Es una sucesión de datos numéricos en un orden dado que ocurren en un intervalo uniforme [27].

2.3.7 Distancia Euclidiana: Es la Medida de la cercanía o separación de los atributos de dos entidades x y y [28].

$$D = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

2.3.8 Contorno de una Imagen: Es definido como las líneas que permiten trazar los límites de una imagen [29].

2.4 Estado del arte

2.4.1 Antecedentes

En Colombia fue reconocida oficialmente la lengua de señas en el año 1996 mediante la ley 324, pero en años atrás la preocupación por establecer un lenguaje de comunicación entre personas sordas, se empezó a manifestar desde el año 1984 cuando se empezaron a conformar comunidades de personas sordas y hablantes con el propósito de enseñar y divulgar la lengua, en el año 1993 la Federación Nacional de Sordos de Colombia (FENASCOL) publicó las primeras cartillas acerca de la simbología del lenguaje y posteriormente fueron surgiendo e integrándose a este fin otras organizaciones como INSOR (Institutos Nacional para Sordos), hasta que en 2006 se publica el primer diccionario básico de la lengua de señas en Colombia [30].

2.4.2 Desarrollos en Colombia

En pro de la sistematización de dicho lenguaje, en Colombia, se han hecho investigaciones y desarrollos buscando la manera en que sea posible establecer una conversación informal entre una persona hablante y una persona sordo o sordomudo en una situación cotidiana. Por ejemplo, en la escuela de ingeniería de Antioquia - Universidad CES, se desarrolló un sistema de traducción del lenguaje dactilológico, está compuesto por un sistema inalámbrico adherido a un guante (hardware) y software basado en redes neuronales que hace las capturas de los signos e identifica el tipo de signo que es representado [31]. En la Universidad Tecnológica de Pereira (UTP) se diseña e implementa un servicio web, como trabajo de grado, que permite crear y consultar un vocabulario del lenguaje de señas y visagrafía partiendo de una palabra del español [32]. El ingeniero de sistemas Jorge Enrique Leal crea un traductor de español a lenguajes de señas online que funciona a través de un intérprete animado llamado Iris y se encuentra disponible en la página web de la fundación HETAH [33]. Estos son algunos trabajos desarrollados en el área, pero encontramos muchos más que son orientados a la inclusión social en Colombia de personas con dificultad en el habla.

2.4.3 Desarrollos en otros países

En todos los países, las personas con dificultad en el habla, es motivo para desarrollar investigaciones y aplicaciones que fomenten su inclusión social. Otras aplicaciones orientadas a este fin son: Signslator, una aplicación capaz de traducir al instante el español al lenguaje de signos, creada por la agencia de publicidad TBWA\España para la Asociación para la Normalización del Lenguaje de Signos. Orientado al lenguaje de signos español y fue uno de los trabajos más premiados en el festival de comunicación iberoamericana celebrado en Miami [34]. En la universidad de Barcelona, como trabajo de grado, se crea un sistema capaz de interpretar los símbolos de los alfabetos del Lenguaje de Signos Americano (ASL) y mostrarlos por pantalla [35]. También existen diversos trabajos orientados a un lenguaje de signos específico, donde todos coinciden en que como desarrollo futuro se pueda adaptar a todos los países del mundo.

Todos estos proyectos han sido desarrollados con el fin de integrar a personas con dificultad en el habla y aquellos que no, para que puedan interactuar sin ningún problema. El objetivo de este proyecto fue orientarlo hacia tal fin, partiendo del concepto que se viene trabajando en Colombia: la inclusión social, y para el desarrollo del mismo, se tomó como referencia el último trabajo mencionado.

3. DESARROLLO DEL PROYECTO

La metodología bajo la cual se desarrolló el proyecto fue la AUP (Proceso Unificado Ágil), es una versión simplificada del Proceso Unificado de Rational (RUP), se eligió porque a diferencia de las otras metodologías ágiles como la XP, esta integra conceptos tradicionales bajo técnicas ágiles. Por ejemplo, el levantamiento de requerimientos en vez de historias de usuario, lo que se ajusta a la necesidad del proyecto debido a que la interacción entre el usuario y la aplicación se limita a dos funciones haciéndose necesario mayores especificidades en los requerimientos del sistema.

3.1 Análisis

Esta etapa del proyecto se divide en dos fases, la primera consiste en determinar las señas del LSC que se van a interpretar y la segunda, en el levantamiento de requerimientos.

En la primera fase se estudió el abecedario del lenguaje de signos Colombiano y se determinó las señas para interpretar. Estas señas son representadas por medio de imágenes estáticas, por ende, sólo se tuvieron en cuenta aquellas que no requieren movimiento en su interpretación. A continuación se muestran en la figura 4 las letras del abecedario con su respectiva seña que se agregó al modelo.

Figura 4: Abecedario del lenguaje de señas



[Creación propia]

La segunda fase del análisis del proyecto se divide en tres módulos:

3.1.1 Procesamiento digital de imagen: Consta de cinco requerimientos del sistema que describen las modificaciones que se le debe realizar a la imagen de entrada para finalmente extraer su vector característico. Estos son:

- **Requerimiento 1:** El Sistema debe ser capaz de capturar la señal realizada por medio de una cámara webcam.
- **Requerimiento 2:** El Sistema debe ser capaz de cambiar el contraste, brillo y gama, de la imagen capturada.
- **Requerimiento 3:** El sistema debe ser capaz de cambiar el espacio de color de la imagen de la señal de RGB a escala de grises para un posterior mejoramiento.
- **Requerimiento 4:** El Sistema debe ser capaz de segmentar la señal del resto de la escena.
- **Requerimiento 5:** El sistema debe ser capaz de extraer las características de la imagen segmentada.

3.1.2 Creación de patrones: Consta de cuatro requerimientos del sistema que describen las acciones del mismo necesarias para garantizar el correcto almacenamiento de los patrones en el modelo de datos. Estos son:

- **Requerimiento 6:** El sistema debe consolidar en un vector las características de la imagen modelo que represente una señal del abecedario del LSC.
- **Requerimiento 7:** El Sistema debe almacenar de manera permanente el vector característico de cada imagen que representa a cada señal del abecedario del LSC formando un modelo del patrón de cada señal.
- **Requerimiento 8:** El sistema debe corresponder cada vector característico del modelo de patrones con la letra del lenguaje escrito a la cual representa.
- **Requerimiento 9:** El sistema debe permitir que el usuario elija agregar una nueva señal al modelo de patrones.

3.1.3 Clasificador: Consta de cuatro requerimientos del sistema que especifican las condiciones que debe tener el clasificador para realizar una exitosa interpretación de la seña. Estos son:

- **Requerimiento 10:** El Sistema debe tener almacenado los modelos de los patrones de las señas, para hacer la comparación con la seña captada por la webcam.
- **Requerimiento 11:** El Sistema debe ser capaz de diferenciar entre una seña y otra, usando los modelos de patrones almacenados en el sistema.
- **Requerimiento 12:** El Sistema debe ser capaz de mostrar la letra referente a su seña asociada.
- **Requerimiento 13:** El sistema debe permitir que el usuario elija interpretar una seña.

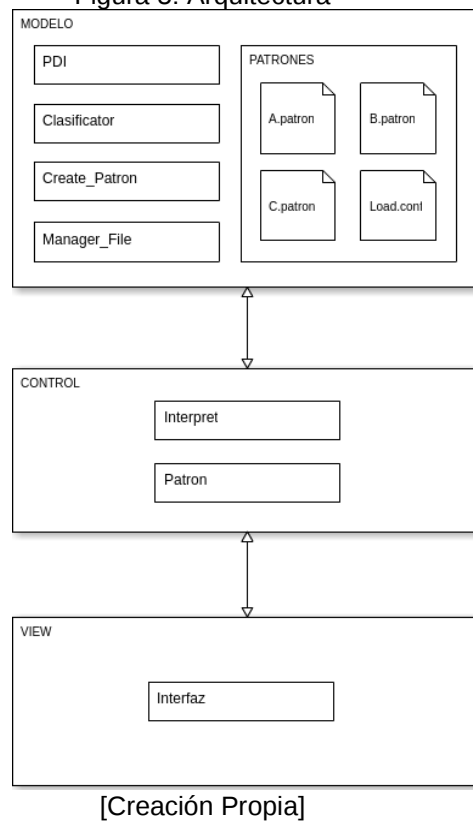
3.2 Diseño

En esta sección se muestran los diseños realizados durante el desarrollo del proyecto.

3.2.1 Arquitectura

La arquitectura MVC (Modelo, Vista y Controlador) consiste en dividir el sistema en tres capas, donde el modelo contiene el núcleo de la funcionalidad (dominio) de la aplicación; el controlador reacciona a las peticiones del cliente, ejecutando la acción adecuada y la vista es aquella interfaz con la que interactúa directamente el usuario. Esta lógica fue aplicada al diseño de la aplicación figura 5 ya que permite la escalabilidad, es una característica importante porque se puede acceder a cada capa de manera independiente y ser modificadas, sí se requiere, sin alterar ninguna otra parte del código. Este proyecto es solo un prototipo y como trabajos futuros, se puede volver más robusto, optimizar, ajustar y/o adaptar a diferentes funcionalidades sin mayor complejidad gracias a que fue programado de manera modular.

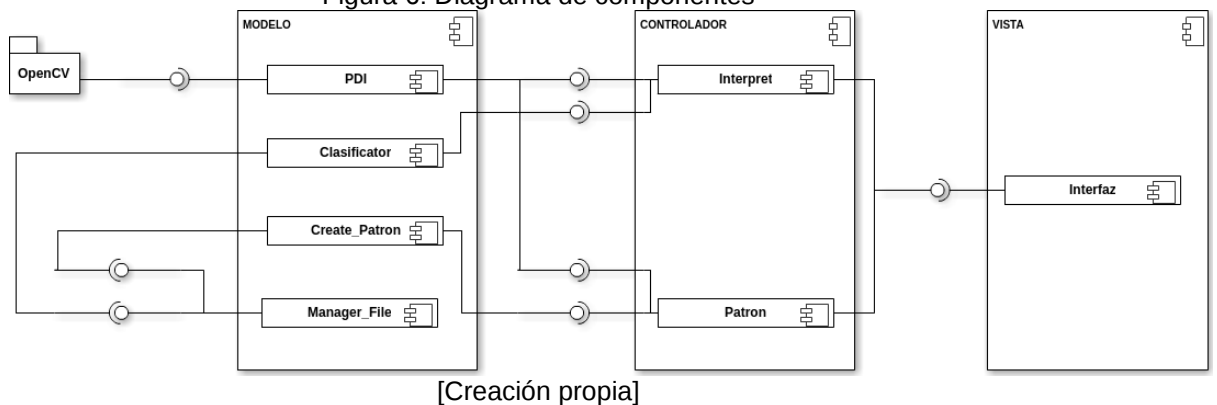
Figura 5: Arquitectura



3.2.2 Diagrama de Componentes

El diagrama de componente, figura 6, muestra de forma global cómo interactúan cada una de las partes con otras, se muestra la división en sus tres capas, MVC, se observa la forma modular de cómo se programó la aplicación, donde cada módulo es independiente a otro, pero permite usar funcionalidades de otro módulo o paquete, como es la librería OpenCV o el componente Manager_File, el diagrama muestra de forma global como sería las interacciones entre los componentes, dando una primera idea a su funcionamiento.

Figura 6: Diagrama de componentes



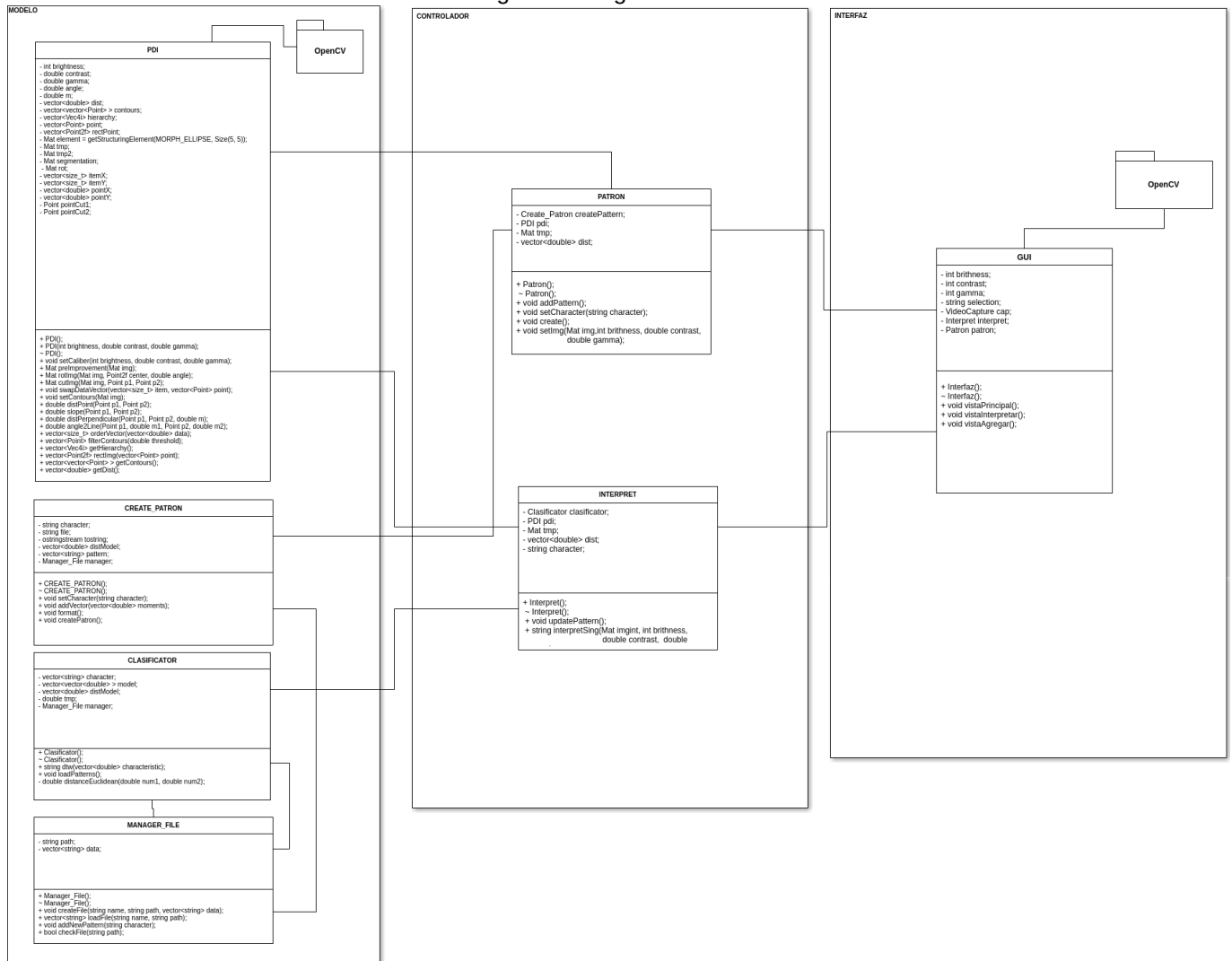
3.2.3 Diagrama de clases

Los módulos se distribuyen en forma de bloques, como se puede observar en el diagrama de clases.

En el módulo o capa principal, modelo, se encuentra la clase PDI la cual es la clase principal del sistema, es la encargada del procesamiento de la imagen y extracción de características, procedimientos que son aplicados en ambas funcionalidades del sistema, tanto para agregar una seña como para interpretarla. Las clases create_Patron y Clasificador, son las encargadas de cumplir con cada funcionalidad respectiva, agregar una seña al modelo o crear un nuevo patrón e interpretar una seña o clasificarla. Ambas clases hacen uso de Manager_File, el administrador del modelo de datos. En la capa llamada controlador, se encuentra solo dos clases Patron e Interpret, estas se encargan de invocar las funciones de las clases del modelo para hacer efectiva cada una de las funcionalidades del sistema. La clase Patron interactúa con la clase Create_Patron y la clase Interpret con la clase Clasificador, ambas clases hacen uso de la clase PDI ya que reciben las imágenes capturadas.

Finalmente en la capa Interfaz, se encuentra la clase Interfaz, encargada de brindar al usuario la información las funcionalidades del sistema y es donde se inicia el proceso de captura de la imagen y control de los eventos de teclado. En la figura 7 se observa el mencionado diagrama.

Figura 7: Diagrama de clases



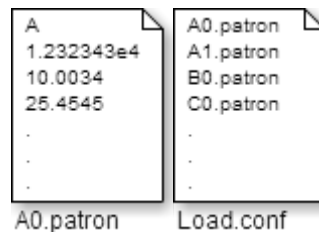
[Creación propia]

3.2.4 Modelo de Datos

La necesidad de tener archivos auto contenidos, para facilitar tanto el mantenimiento como su instalación fue necesario desarrollar nuestro propio Modelo de Datos para el almacenamiento de cada modelo de las señas, las cuales se van a usar al momento de la clasificación. Este modelo de datos consta de un archivo primario “Load.conf” encargado de tener el orden de carga de cada uno de los patrones almacenados como modelo, la manera en cómo se almacena es de forma descendente y se lee de igual manera, los archivos secundarios como “A0.patron” contiene en su primera línea la letra correspondiente a la seña y en sus demás líneas las características ya extraídas, de igual forma se almacena de forma descendente para facilitar su lectura, para poder almacenar varios modelos de una misma seña el programa enumera de forma sucesiva los archivos como: A0.patron, A1.patron, ..., An.patron.

En la figura 8 se muestra de forma gráfica cómo sería la estructura del archivo principal y de los archivos secundarios.

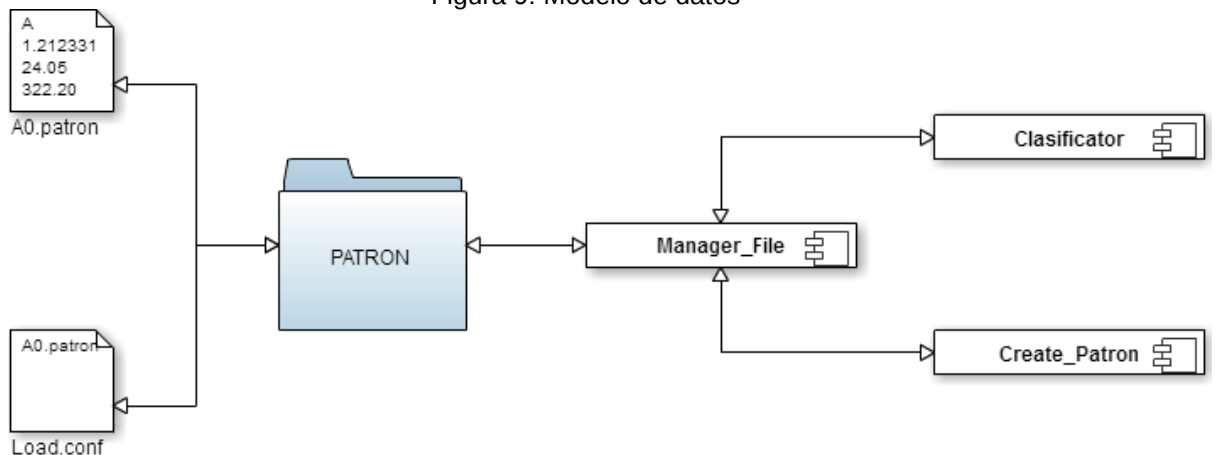
Figura 8: Estructura de archivo



[Creación Propia]

Tanto la lectura como la escritura de los archivos se hace por medio de nuestro administrador de archivos “Manager_File”, el cual es usado por las clases “Clasificator” y “Create_Patron”. En la figura 9 se ejemplifica el modelo de datos que se describió.

Figura 9: Modelo de datos



[Creación propia]

3.3 Implementación

En esta sección se menciona el software y hardware que fue utilizado para la implementación del programa y el comportamiento del mismo, donde se explica cómo fue desarrollado.

3.3.1 Codificación

El programa se cargó en el repositorio GitHub para facilitar el trabajo en equipo. Se codificó bajo el Sistema Operativo GNU Linux Mint en la versión 17.1, con el lenguaje de programación C++ en la versión 4.9 y se usó la librería OpenCV en la versión 2.4.11 para el procesamiento de la imagen. La implementación se realizó en el editor de texto Sublime Text 3 y se empleó scripts para la compilación, ejecución y carga del programa en el repositorio. El modelo de datos de datos se diseñó bajo archivos de texto facilitando su administración.

Se evitó trabajar con IDEs para eludir la instalación de agregados que interfieran en la ejecución y/o en la estructura del modelo MVC y prevenir problemas de versionamiento.

3.3.2 Instalación

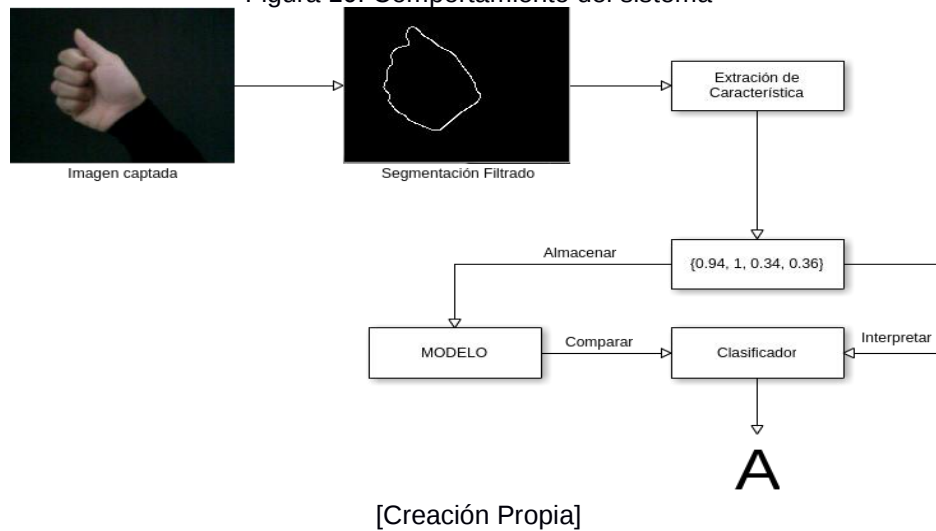
El programa se ejecutó en un computador portátil marca Acer, con un procesador Intel Core i3, una memoria RAM de 3GB y un sistema operativo de 64 bits. Se utilizó una webcam de 8 Mpx. Para ver el proceso completo de instalación y ejecución diríjase al manual del usuario.

3.3.3 Comportamiento de la aplicación

El sistema implementado se compone de las siguientes etapas (Ver figura 10):

- Obtención de las imágenes a partir de capturas sucesivas.
- Filtrado y segmentación de las imágenes para su posterior análisis.
- Obtención del vector característico de cada una de las imágenes, una vez el sistema realiza este paso, se puede acceder a cualquiera de las dos opciones: almacenar un patrón o interpretar una señal.
- Creación del modelo de patrones, si se desea almacenar una señal, el vector característico se guarda en el modelo de patrones.
- Clasificación de los datos con la clase correspondiente, si se desea interpretar una señal, el clasificador accede al modelo de patrones para comparar un vector característico de entrada que corresponde a la señal que se desea interpretar con los almacenados, y finalmente retorna un resultado.

Figura 10: Comportamiento del sistema



A continuación se explica cada una de las etapas del sistema:

- **Obtención de la imagen**

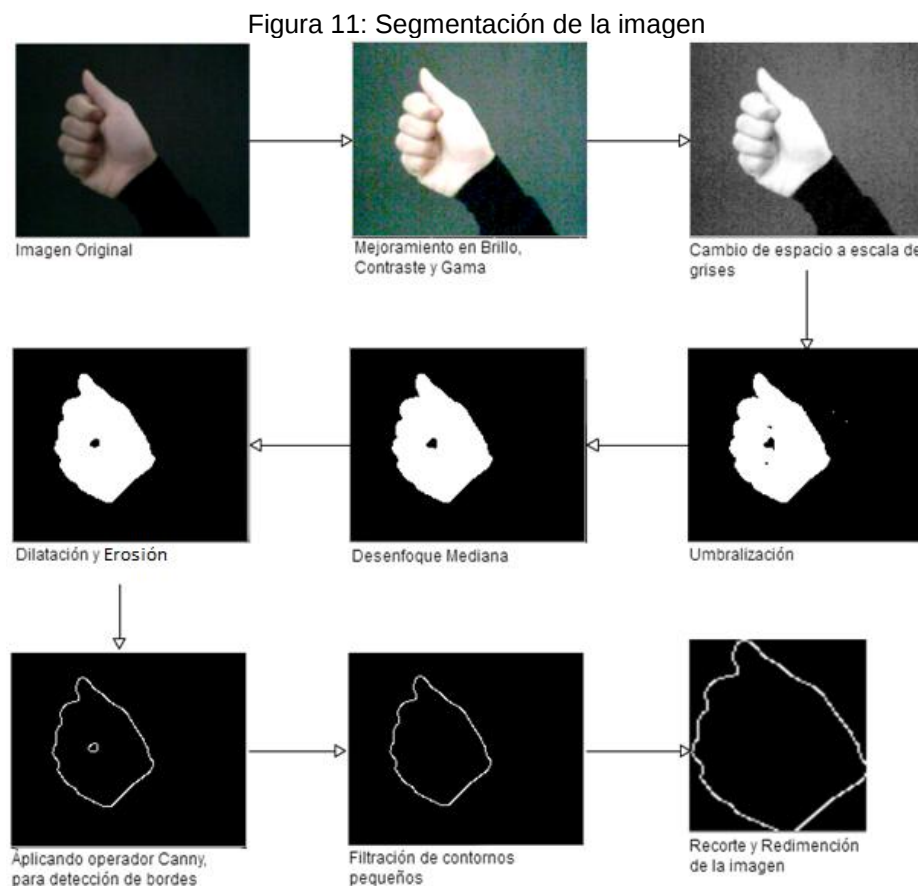
Antes de realizar las capturas, es necesario establecer un escenario propicio para que la aplicación funcione correctamente. Para ello, se debe tener control de la iluminación ambiente y otras fuentes de luz que pueden llegar a generar sombras y/o brillos no deseados, se debe tener una iluminación frontal sobre la mano, un fondo oscuro y cubrir el antebrazo hasta la muñeca, con una manga larga de color oscuro con el fin de que la aplicación capture solo la parte de la mano. Este paso es importante ya que si se toma parte del antebrazo esto puede interferir en el contorno, entorpeciendo la clasificación.

- **Segmentación**

Antes de realizar la captura de la imagen, el sistema permite configurar los aspectos generales de la imagen (Brillo, Contraste y Gamma), además muestra cómo está quedando la imagen resultante, para que el usuario pueda apreciar los resultados de los cambios realizados por él y realizar la captura al momento que considere pertinente. Una vez obtenida la imagen se envía a su procesamiento:

- Se convierte a escala de grises.
- Se umbraliza la imagen con un valor de 128, lo que significa que, los píxeles con mayor valor los convierte en 255 (blanco) y los de menor valor los convierte en 0 (negro).
- Se aplica el filtro de la mediana.
- Se aplica una erosión y dilatación.
- Se aplica canny, función de OpenCV, para extracción del contorno de la imagen.
- Se realiza un filtrado de contornos donde se eliminan aquellos que tengan una longitud menor o igual al valor asignado.

En la figura 11 se muestra paso a paso el proceso de segmentación mencionado.



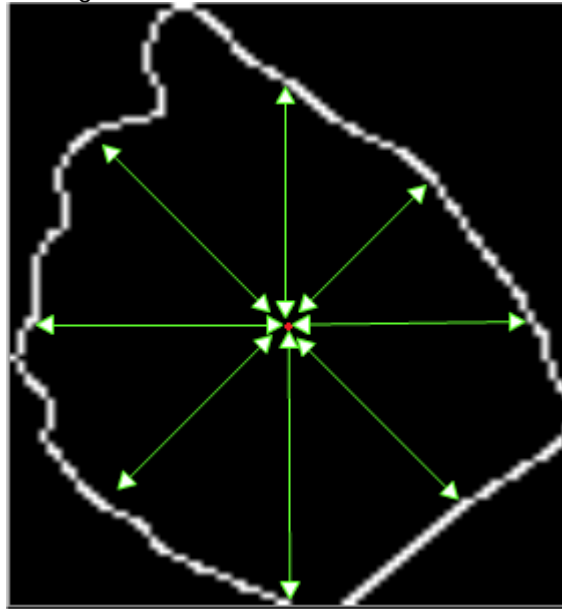
[Creación Propia]

- **Extracción de características**

Al inicio del proyecto, se exploraron diferentes métodos de extracción de características como: Surf (Speeded Up Robust Features), Sift (Scale Invariant Feature Transform), momentos Zernike y momentos invariantes HU. De estos se eligió el método de los momentos HU para implementar y realizar pruebas iniciales ya que, en teoría, era el más adecuado por ser invariante a la rotación, traslación y escalación, pero no dio buenos resultados ya que es muy sensible a los cambios de iluminación, mismo efecto en los métodos mencionados anteriormente, y no permitía discriminar entre una señal y otra, arrojando datos incongruentes al operar en la misma señal. Debido a esto se optó por otro método basado en descripción de contornos, el cual surgió como una idea propia fundamentada en el método de extracción de características descrito en la sección 2-2 del trabajo de grado: “Reconocimiento automático del lenguaje de signos: Lenguaje ASL” [35].

El método se desarrolló de la siguiente manera: en primera instancia, era necesario garantizar que la escalabilidad de la imagen no afectara el resultado que se obtiene en esta etapa, para ello, se recorta la imagen abarcando todo su contorno y se redimensiona a un valor de 300 píxeles. Una vez realizado esto, se encuentra el centro de la imagen y se recorre todo su contorno desde este punto, calculando la distancia que hay entre el centro y cada punto que contenga contorno; el recorrido se inicia desde el punto más alto en y, lo que hace que la aplicación sea sensible a la rotación. Cada distancia se guarda en un vector el cual será el vector característico de cada señal representada en la imagen de entrada, este vector se normaliza dividiendo por el número mayor de este cada dato del vector dando como resultados números entre [1, 0], esto se hace para que los vectores tengan un rango determinado haciendo más fácil la clasificación por el DTW. En la figura 12 se muestra de manera gráfica el método descrito.

Figura 12: Extracción de características



[Creación Propia]

El punto rojo indica el centro estimado de la imagen, y las flechas verdes son las distancias de cada punto del contorno al centro de la imagen.

- **Creación del modelo de patrones**

Una vez obtenido el vector característico, se almacena cada dato del vector como un string en un archivo de texto de formato “.patron”, donde el primer dato corresponde a la letra que representa la seña en el lenguaje escrito, y el resto a sus respectivas distancias.

Al momento de almacenar los patrones, se crea un archivo “Load.conf” este contiene el nombre de todos los patrones y por cada patrón nuevo que se ingrese, automáticamente, es agregado en este archivo. Si el modelo está vacío, este se crea con el primer patrón agregado.

- **Clasificador**

Inicialmente se usó el clasificador de distancia mínima, que consta de determinar la menor distancia entre dos vectores de entrada usando la distancia euclidiana. Por la forma de cómo se extraen las características del contorno los vectores tienden a ser de diferente tamaño, por este motivo este método no se pudo aplicar de forma correcta, dando resultados poco congruentes. Por consiguiente, se debió investigar un clasificador que permitiera conocer la semejanza de dos vectores de diferente longitud y se llegó al clasificador DTW que lo permitió.

Para realizar la interpretación de una seña, ante todo, se carga el modelo de patrones en memoria. Se realiza la comparación entre el vector característico de entrada con cada uno de los almacenados,

esto se hace, por medio del algoritmo DTW (ver sección 2.1.3), este arroja como resultado un vector de costos de dicha comparación. Se procede a sacar el costo mínimo, el cual corresponderá al patrón del vector almacenado que más se asemeja al vector de entrada. Retornando la letra del lenguaje escrito que representa la imagen de entrada.

4. PRUEBAS Y DISCUSIÓN DE RESULTADOS

Para realizar la validación del sistema se han efectuado varias pruebas. A continuación se describe cada una de ellas.

4.1 Prueba No. 1

Se evaluó las señas estáticas del abecedario del LSC con el sujeto modelo del cual se guardó los patrones en el modelo de datos, con el fin de examinar el correcto funcionamiento de los algoritmos de PDI y clasificación en el mejor de los casos, en el cual en teoría, debía dar los resultados correctos. En la tabla 2 se muestra sus resultados.

Tabla 2: Sujeto modelo

A	B	C	D	E	F	H	I	K	L	M	N	O	P	Q	R	T	U	V	W	X	Y
A	B	C	D	E,B	F	H	I	K	L	M,N	N	O	P	Q	R	T	U	V	W	X	Y,I

[Creación propia]

Las tendencias en las letras E y B, M y N y Y e I, se daban debido a la rotación e inclinación de la mano, por tanto, se requiere mucha precisión en su realización. Como se puede observar, a pesar de las 3 letras que presentan la confusión, el sistema arrojó los resultados correctos en cada una, lo que concluye que los métodos utilizados para procesar, representar y clasificar la información de la imagen produce los efectos apropiados, en especial, el algoritmo DTW cuya clasificación es óptima al corresponder correctamente cada seña de entrada con la almacenada.

4.2 Prueba No. 2

Esta prueba consistió en probar las señas del abecedario del lenguaje de signos del LSC en 15 personas (ver anexo 1), con el fin de analizar 3 aspectos del sistema, primero medir el comportamiento del mismo al probar con individuos que presentan diferentes características morfológicas de la mano derecha en comparación al modelo guardado; segundo, examinar la dificultad que tiene cada seña para ser acertada y por último, determinar el promedio de acertación que tiene el sistema en general. En la tabla 3 se muestran los resultados obtenidos:

Tabla 3: Resultados de 15 sujetos de prueba al evaluar cada letra del abecedario

Letras/ sujetos	A	B	C	D	E	F	H	I	K	L	M	N	O	P	Q	R	T	U	V	W	X	Y
1	A	X	C	D	E	X	H	I	K	L	M	X	O	P	Q	R	T	U	X	X	X	Y
2	A	B	C	D	X	F	H	I	K	L	X	X	O	P	Q	R	T	U	X	X	X	Y
3	A	X	C	D	X	X	H	I	K	L	X	X	O	P	Q	R	T	U	X	W	X	Y
4	A	B	X	X	E	F	H	I	K	L	M	X	O	P	Q	R	T	U	V	W	X	Y
5	A	B	C	D	X	X	H	I	K	L	M	X	O	P	Q	R	T	U	X	W	X	Y
6	A	B	C	D	E	F	H	I	K	X	X	X	O	P	Q	R	T	U	X	W	X	Y
7	A	B	C	D	X	F	H	I	K	L	M	X	O	P	Q	R	X	U	X	X	X	Y
8	A	B	C	D	X	F	H	I	K	X	X	X	O	P	Q	R	T	U	V	X	X	Y
9	A	X	C	D	E	F	H	I	K	X	X	X	O	P	Q	R	T	U	X	W	X	Y
10	A	B	C	D	X	F	H	I	K	L	M	X	O	P	Q	R	T	U	X	X	X	X
11	A	X	C	D	X	X	H	I	K	L	X	X	O	P	X	R	T	U	V	X	X	Y
12	A	B	C	D	E	X	H	I	K	L	M	X	O	P	Q	R	T	U	V	X	X	Y
13	A	B	C	D	E	F	H	I	K	L	M	X	O	P	Q	R	T	U	V	X	X	Y
14	A	X	C	D	E	X	H	I	K	L	M	X	O	P	A	R	T	U	V	W	X	Y
15	A	X	C	X	E	F	H	I	K	L	X	N	O	P	Q	R	T	U	X	W	X	Y

[Creación Propia]

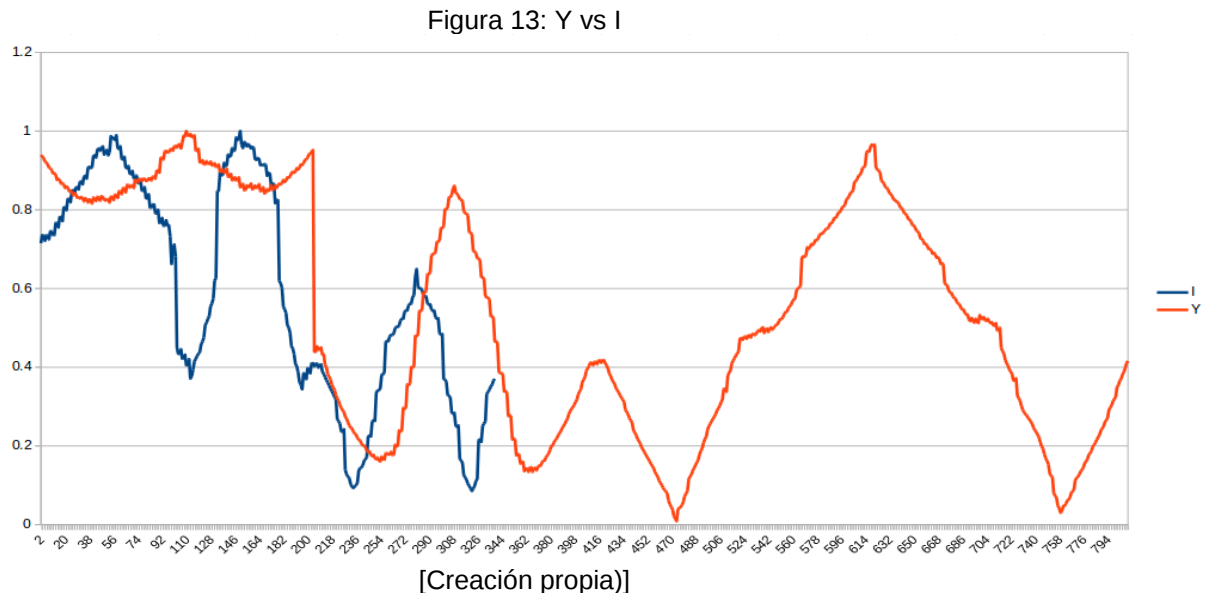
Nota: Las letras de color negro son las señas acertadas, las de color rojo son resultados erróneos que arrojó el sistema.

Como se puede observar en la tabla anterior, los resultados varían entre una persona u otra, esto se debe al método de extracción de características el cual es sensible al tamaño y textura de la mano, propiedades morfológicas que son únicas en cada persona. A continuación se analiza cada aspecto mencionado.

4.2.1 Comportamiento del sistema

Durante la realización de pruebas del sistema, se notó ciertas tendencias de error, esto fue, la confusión ente las letras Y e I, B y E, M y N, W y V. Se procedió a analizar los vectores característicos de estas letras para determinar la razón por la cual el sistema arrojó tales resultados.

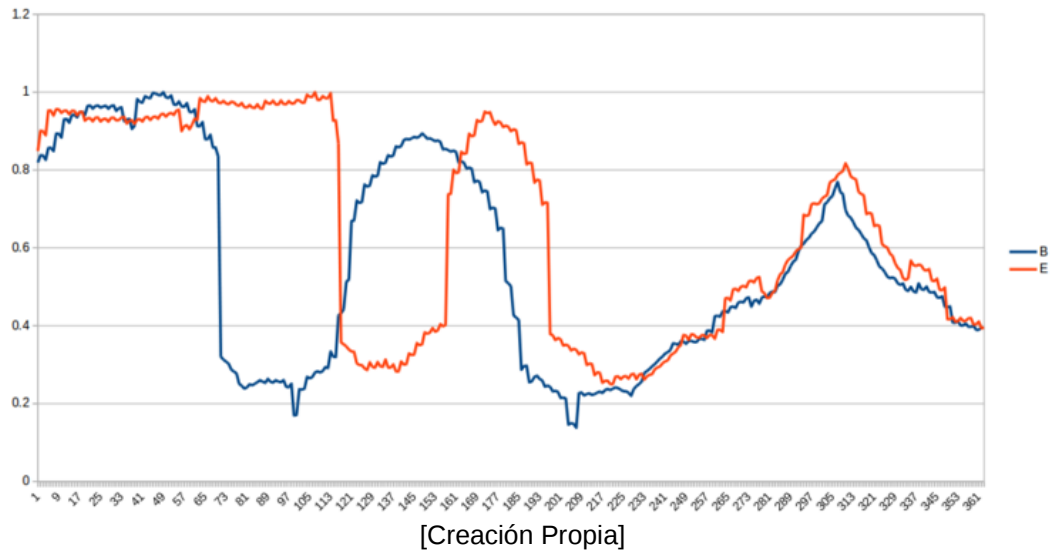
En la figura 13 se muestra la gráfica de los vectores característicos de las letras Y e I.



Como se puede observar estos vectores son semejantes, presentan la misma forma en sus cotas y valles, a pesar de que tienen una gran diferencia en tamaño. Esto ocurre porque el clasificador aproxima el vector de entrada, que puede variar en cada captura, al que más se asemeje de los almacenados en el modelo, según sean los factores externos (características morfológicas de la mano, iluminación del entorno, ubicación de la cámara, rotación de la mano, entre otros) se obtendrá como resultado la letra que corresponde a la seña o sus similares en los datos almacenados de la misma.

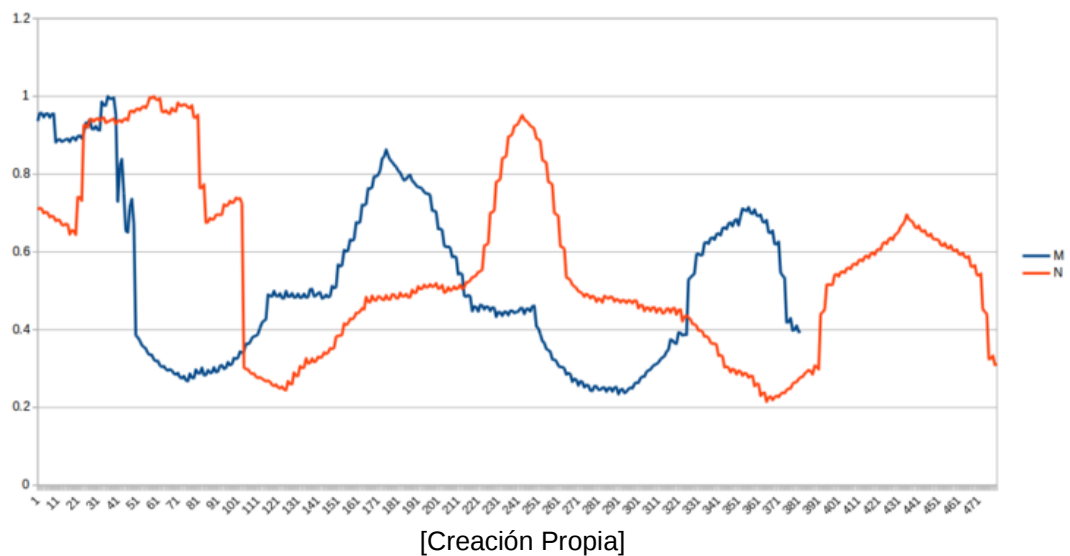
En la figura 14 se muestran los vectores característicos de las letras B y E.

Figura 14: B vs E



Se observa en esta figura como la meseta del comienzo y la cota final tiende a coincidir en ambas series. Razón por la cual el sistema puede confundir una con otra. Otro caso de este tipo, es el presentado en la figura 15, donde se muestra los vectores característicos de las letras M y N.

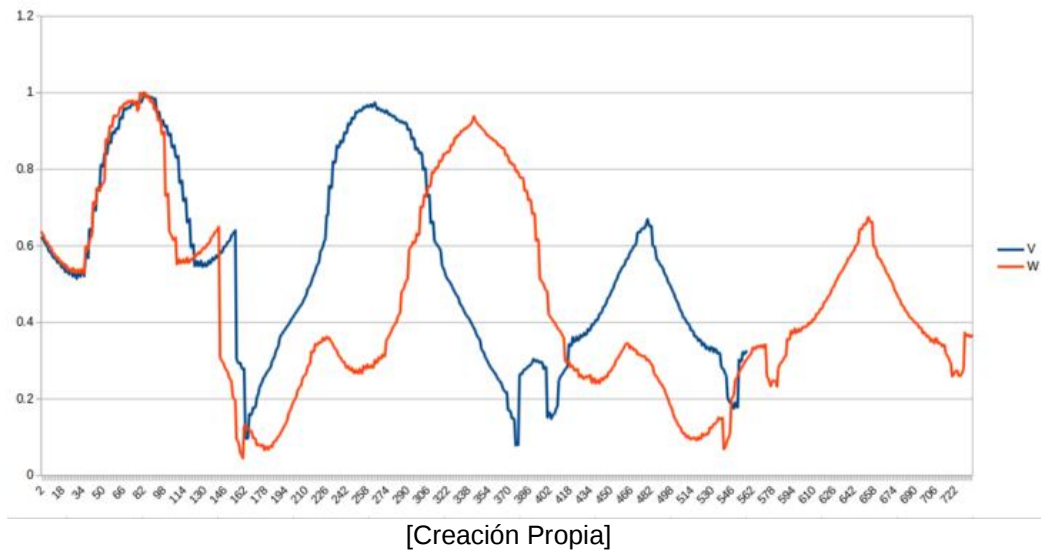
Figura 15: M vs N



En la figura anterior, se puede observar que los valles están desplazados en una serie a otro, aunque las cotas no coinciden en su totalidad, como en la figura 12, esto influye en la clasificación.

En la figura 16 se muestran los vectores característicos de las letras V y W.

Figura 16: V vs W

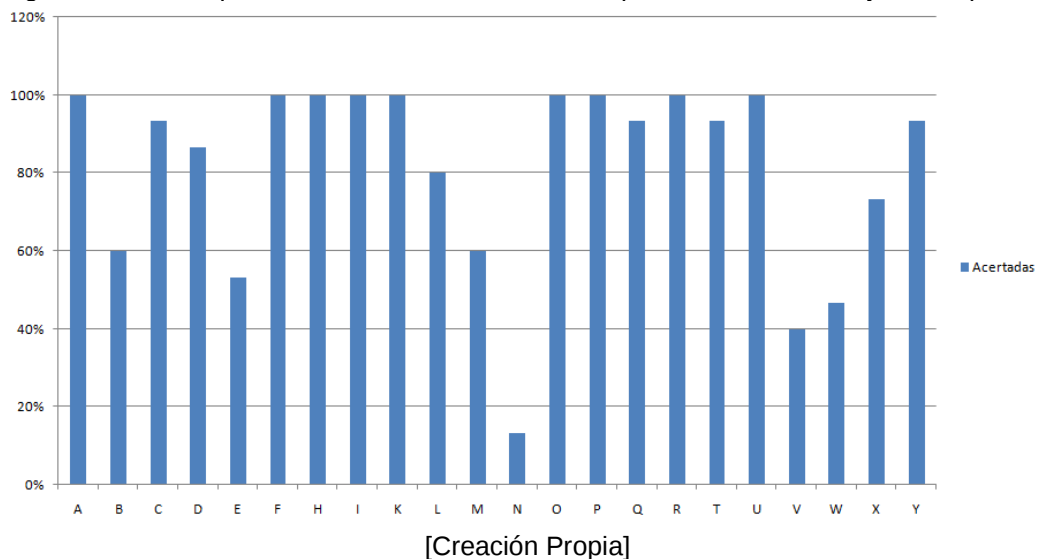


En esta, a pesar de la diferencia de longitud entre ambos vectores, coinciden en el inicio y sus cotas son similares por tanto el sistema también las puede confundir si no tiene la precisión necesaria.

4.2.2 Análisis de cada seña de la tabla de resultados

Se procedió a evaluar el porcentaje de acertación que tuvo cada seña en la población de muestra y se expone el resultado de manera gráfica en la figura 17 para mejor apreciación.

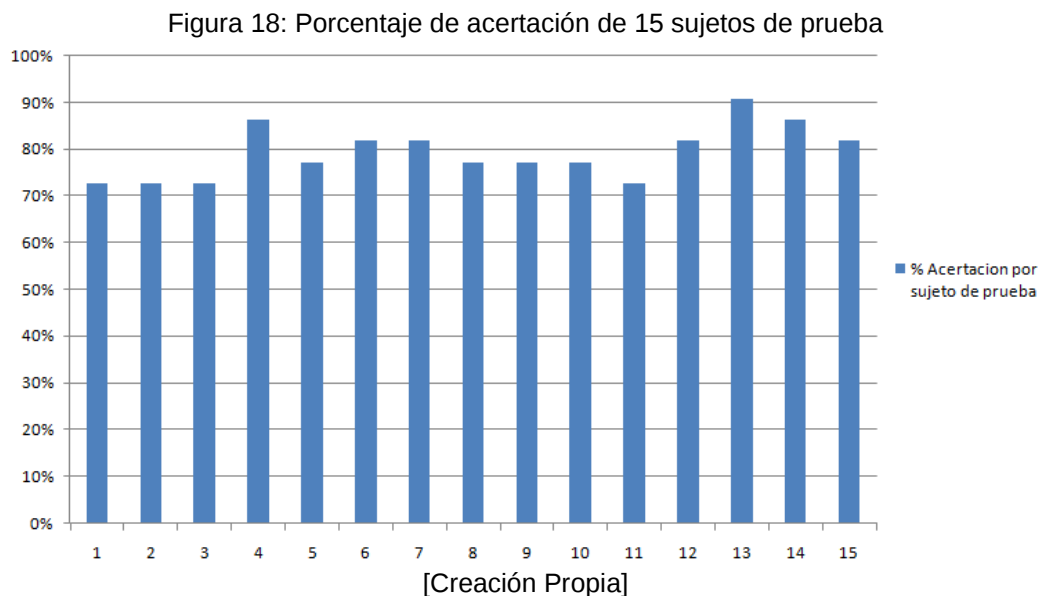
Figura 17: Veces que fue reconocida una letra en la prueba con los 15 sujetos de prueba



Se observa que las señas con mayor porcentaje de acierto son las que representan las letras A, C, D, F, H, I, K, O, P, Q, R, T, U y Y. Estas tienen un porcentaje de acertación por encima del 80% que indica que son menos sensibles a las variables extrañas (factores externos no controlables como morfología de la mano y otros) y por ende se aciertan fácilmente, con solo lograr la precisión necesaria. Las señas con nivel de acertación medio, son aquellas que se encuentran entre el 60% y 80%, requieren de más precisión del contorno de la imagen de entrada con el almacenado, para obtener un resultado correcto, estas son: B, L, y X. Finalmente, las señas con un porcentaje de acertación menor a 60% son difíciles de reconocer correctamente, ya que debe haber mucha precisión por parte del usuario en la realización de la seña y un factor influyente y no controlable, es la textura morfológica de la mano del usuario; lo que en ocasiones causa que a pesar de que la seña sea realizada de la manera correcta no se reconozca adecuadamente.

4.2.3 Porcentaje de acertación del sistema

Finalmente, se estimó la acertación del sistema, para ello, se determinó cuantas señas se reconocieron correctamente en cada individuo de prueba y se muestra gráficamente en la figura 18.



En la gráfica se contempla que aunque la mano derecha de los sujetos de prueba cuenta con una constitución morfológica diferente en cada uno, el sistema fue capaz de dar respuesta a cada individuo con un mínimo porcentaje de acertación del 70% y un máximo del 90%, es decir, que el sistema tuvo aciertos entre 16 y 20 señas de las 22 que se probaron en las 15 personas. De acuerdo a estos datos, la desviación estándar y la media de la gráfica son de 6% y 79% respectivamente, o sea, que la aplicación reconoce en promedio 18 señas del abecedario del LSC cumpliendo con el objetivo del proyecto de reconocer al menos 10.

En conclusión, esta prueba permitió determinar qué:

- El sistema en promedio reconoce el 79% de las señas estáticas del abecedario del Lenguaje de signos Colombiano.
- Las letras que son menos sensibles a las variables extrañas y por ende, tienen mayor posibilidad de acierto son: A, C, D, F, H, I, K, O, P, Q, R, T, U y Y.
- El sistema presenta confusión en las señas Y e I, B y E, M y N, W y V, por lo cual se requiere tener más precisión en la realización de la seña por parte del usuario para que el vector característico de entrada sea más específico conforme al contorno almacenado y retorne la letra correcta.

4.3 Prueba No. 3

Una prueba más fue necesaria para verificar que las señas guardadas en el modelo sean las correctas y correspondan con las señas utilizadas en la cotidianidad de las personas que utilizan el LSC para comunicarse. Para ello se requirió de personas expertas en este lenguaje.

Inicialmente, se hizo un acercamiento con la fundación de sordos y mudos Asortul de Tuluá, en donde se presentó el programa a dos personas: un intérprete y una persona sorda a quienes se mostró las señas del abecedario con las cuales trabaja el sistema, ellos corrigieron las señas T, K y H, las cuales se realizaban con una orientación equivocada y confirmaron la buena realización de las otras.

Uno de los sujetos de la prueba no. 2 fue una persona que ha crecido con dificultad para oír y ha trabajado en problemáticas relacionadas con la inclusión social de personas sordas y/o mudas, por tanto, posee el conocimiento y la experiencia en el lenguaje de señas Colombiano. Él contribuyó al proyecto como asesor y sujeto de prueba en varias ocasiones para perfeccionar la realización de las señas. En la última prueba que se le aplicó, fue acompañada de una breve entrevista (ver anexo 2) en la que concluyó que aunque el sistema tiene límites en el control de variables extrañas como el escenario de pruebas y condiciones específicas de la mano para lograr una interpretación correcta, es un acercamiento de cómo utilizar tecnología básica (computador y cámara de características básicas) que apoye el proceso de inclusión social de personas con dificultad en el habla.

4.4 Prueba No. 4

Otro propósito de este proyecto fue deletrear frases y palabras a partir de las señas almacenadas en el modelo, para evaluar si es posible transmitir oraciones cortas por este medio. En las tablas 4-12 se evidencian dichas pruebas.

Tabla 4: Prueba de deletreo 1

H	O	L	A	M	U	N	D	O	% ACERTACIÓN
H	O	L	A	M	U	B	D	O	88%

[Creación propia]

Tabla 5: Prueba de deletreo 2

A	L	F	A	B	E	T	O	% ACERTACIÓN
A	L	F	A	B	E	T	O	100%

[Creación propia]

Tabla 6: Prueba de deletreo 3

M	I	N	O	M	B	R	E	K	E	L	L	Y	% ACERTACIÓN
M	I	M	O	M	B	R	E	K	E	L	L	I	84%

[Creación propia]

Tabla 7: Prueba de deletreo 4

C	O	M	P	U	T	A	D	O	R	% ACERTACIÓN
C	O	M	P	U	T	A	D	O	R	100%

[Creación propia]

Tabla 8: Prueba de deletreo 5

C	A	M	A	R	A	% ACERTACIÓN
C	A	M	A	R	A	100%

[Creación propia]

Tabla 9: Prueba de deletreo 6

Y	O	T	U	V	E	U	N	A	F	L	O	R	% ACERTACIÓN
Y	O	T	U	V	E	U	N	A	F	L	O	R	100%

[Creación propia]

Tabla 10: Prueba de deletreo 7

C	O	L	O	M	B	I	A	T	I	E	R	R	A	Q	U	E	R	I	D	A	% ACERTACIÓN
C	O	L	O	M	E	I	A	T	I	E	R	R	A	Q	U	E	R	I	D	A	95%

[Creación propia]

Tabla 11: Prueba de deletreo 8

M	I	P	A	P	A	Y	M	I	M	A	M	A	C	O	M	E	N	C	O	C	O	% ACERTACIÓN
M	I	P	A	P	A	I	N	I	N	A	B	A	C	O	M	E	N	C	O	C	O	81%

[Creación propia]

Tabla 12: Prueba de deletreo 9

X	I	L	O	F	O	N	O	% ACERTACIÓN
X	I	L	O	F	O	F	O	87%

[Creación propia]

Tales pruebas son muestras de que es posible comunicar frases cortas, pero el porcentaje de acertación de cada frase o palabra en esta prueba específica no aplica para todas las personas, es decir, estos resultados pueden variar entre un individuo u otro debido a que el sistema funciona de manera diferente con cada una. Sin embargo, todas se permiten comunicar alguna frase ya que los resultados anteriores demuestran que la aplicación reconoce más de 10 señas, suficientes para formar palabras o frases cortas.

5. CONCLUSIONES Y PROYECCIONES

5.1 Conclusiones

En este capítulo se recogen las conclusiones más relevantes a las que se ha llegado durante el desarrollo del presente trabajo y se indican las líneas de trabajo futuras que se derivan de la realización de este trabajo de grado.

Entre las conclusiones más destacadas se tiene:

- Se optó por interpretar las señas estáticas usando procesamiento digital de imágenes, debido a que las señas que tienen movimiento requieren un procesamiento diferente enfocado hacia video.
- En la fase de la extracción de la imagen, el método sobresaliente fue el de segmentación por umbral en un espacio de color en escala de grises y la extracción de características basado en una descripción de contorno, aunque este último es sensible a las características morfológicas de la mano como el tamaño, el grosor de sus dedos, si las falanges presentan alguna desviación, entre otros. Fue el más óptimo de los otros explorados (como los momentos invariantes de HU, Surf, Sift, momentos Zernike).
- Se creó un modelo de datos propio para almacenar las características extraídas de la imagen, el cual contiene 22 patrones que representan a las 22 señas del abecedario del LSC, además esto permitió controlar y acceder al modelo fácilmente, posibilitando su portabilidad.
- Un punto importante dentro del proyecto fue clasificar las señas por medio del algoritmo DTW, debido a que cada movimiento gestual de la mano es diferente, por lo tanto, genera un vector característico distinto y el algoritmo clasificó adecuadamente con vectores característicos de distinto tamaño, ya que cada movimiento gestual en cada intento no se realiza de la misma forma ni de la misma manera, por lo cual, se genera por cada vez un vector característico diferente, el cual fue un inconveniente que se tuvo inicialmente al momento de elegir el método de clasificación, y se solucionó con el mencionado algoritmo porque opera con vectores de distinto tamaño sin ningún problema.

- Se realizaron pruebas de funcionalidad de la aplicación con 15 voluntarios que realizaron el ejercicio de hacer las señas del abecedario del LSC, obteniendo como resultados valores concluyentes, por ejemplo las señas que presentan mayor probabilidad de ser reconocidas son las que representan las letras A, C, D, F, H, I, K, O, P, Q, R, T, U y Y; y las que tienen dificultad para ser interpretadas son B, L, X, E, N, V y W. En promedio el sistema tiene un porcentaje de acertación del 79% que equivale a 18 señas y una desviación estándar del 6%. En cada persona reconoce más de 10 señas cumpliendo con el objetivo planteado.

La aplicación de este trabajo, está sujeto a una serie de restricciones de uso como el fondo elegido sobre el que se van a realizar las capturas, se recomienda fondo oscuro y realizar capturas solo de la mano (sin antebrazo) para disminuir información de entrada; iluminación adecuada, ya que puede variar de un momento a otro durante la prueba, se presenta principalmente si se realiza la prueba en un escenario que contenga luz ambiente; los valores de gamma, brillo y contraste (que pueden afectar el proceso de segmentación y reconocimiento de los contornos); la realización de la seña solo con la mano derecha y un factor influyente para la correcta interpretación, es la contextura morfológica de la mano, rotación e inclinación de la misma, ángulo de captura de la cámara y la correcta realización de los símbolos por parte del usuario.

Concluyendo, la aplicación de escritorio desarrollada funciona y de acuerdo a su promedio de acertación en las señas del abecedario del LSC permite comunicar oraciones cortas. Además sirve de fundamento para ampliar investigaciones que consistan en la creación de aplicativos enfocados a promover la inclusión social de individuos con dificultad para comunicarse verbalmente.

5.2 Proyecciones

Actualmente como se encuentra el proyecto brinda la oportunidad de realizar trabajos futuros, bien sea para mejorarlo o para extenderlo, entre estos trabajos se pueden realizar los siguientes:

- Mejorar la segmentación abarcando fondos complejos .
- Incluir señas dinámicas como las que corresponden a letras que faltaron por incorporar en el modelo de patrones.
- Reconocer señas hechas por ambas manos.
- Extender el reconocimiento de la seña a que abarquen cara y torso.
- Controlar la iluminación de forma automática para diferentes escenarios.
- Implementar un algoritmo de inteligencia artificial para que el sistema aprenda a partir de las señas de cada persona que pruebe el sistema y se haga cada vez más eficiente.
- Crear una interfaz para que los usuarios sean capaces de usar el aplicativo intuitivamente.

REFERENCIAS

- [1] A. A. Belluscio. (2012, Oct. 10) Lenguas de señas: "cada comunidad desarrolló la propia por necesidad" [Online] Disponible: <http://www.innovat.org.ar/lenguas-de-senas-cada-comunidad-desarrollo-la-propia-por-necesidad/>
- [2] INSOR, "Participación porcentual de la población sorda", accedido Mayo 2015. [Online] Disponible: <http://www.insor.gov.co/observatorio/participacion-porcentual-de-la-poblacion-sorda/>
- [3]"Adquisición de imágenes", accedido Mayo 2015. [Online] Disponible: http://iee.fing.edu.uy/ense/assign/codif/material/transparencias/03_adquisicion.pdf
- [4] B. Escalante, (2006, Agosto) "Procesamiento Digital de Imágenes". [Online] Disponible: <http://verona.fi-p.unam.mx/boris/teachingnotes/Introduccion.pdf>
- [5] D. Fernandez, K. Mejia, "preprocesamiento de imágenes digitales a través de su transformada de fourier". Accedido Mayo 2015. [Online] Disponible: <http://www.optica.unican.es/RNO7/Contribuciones/articulospdf/fernandezDM.pdf>
- [6] R. Angelone, (2012, Ago. 12) "Brillo, contraste y gamma". [Online] Disponible: http://blogprocdigimra.blogspot.com.co/2012/08/brillo-el-brillo-se-define-como-la_12.html
- [7] M. Cárdenas, (2012, Ago. 14) "Corrección gamma". [Online] Disponible: <http://procesamientodigitaldeimagenesupb2012.blogspot.com.co/2012/08/correccion-de-gamma-contraste-y-brillo.html>
- [8] (2006) "Reducción de ruido en una imagen digital". [Online] Disponible: <http://www4.ujaen.es/~satorres/practicas/practica2.pdf>
- [9] (2015) "Introducción al Procesamiento Digital de Imágenes" (PDI) [Online] Disponible: https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/717885/mod_resource/content/2/PDI-Clase%201.pdf
- [10] "Escala de grises" accedido Sept. 2015. [Online] Disponible: <http://www.fotonostra.com/glosario/escalagrisegrayscale.htm>
- [11] (2015, Junio) "Umbralización". [Online] Disponible: http://www.lpi.tel.uva.es/textasciitilde{nacho/docencia/ing_ond_1/trabajo_s_03_04/sonificacion/cabroa_archivos/umbralizacion.html

[12] Morphological Image Processing [Online] Disponible: <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm>

[13] Steven W. Smith, Ph.D, "The Scientist and Engineer's Guide to Digital Signal Processing," Chapter 25: Special Imaging Techniques [Online] Disponible: <http://www.dspguide.com/ch25/4.htm>

[14] "Segmentación de imágenes" accedido Oct. 2015 [Online] Disponible: http://www.lcc.uma.es/~textasciitilde{}munozp/documentos/procesamiento_de_imagenes/temas/pi_cap6.pdf

[15] "Representación y Descripción" accedido Oct. 2015 [Online] Disponible: <http://dea.unsj.edu.ar/imagenes/recursos/capitulo6.pdf>

[16] P. Alonso Fuertes, "Análisis del algoritmo DTW para reconocimiento biométrico de personas mediante firma manuscrita on-line" Memoria de título, Ing. de sistemas audiovisuales, Depto. de Tecnología Electrónica, Univ. Carlos III de Madrid, Madrid, España, Sept. 2012.

[17] SAKOE; CHIBA, (1978) "Dynamic Programming Algorithm Optimization for spoken word recognition". [Online] Disponible: https://www.researchgate.net/publication/3176441_Dynamic_Programming_Algorithm_Optimization_for_Spoken_Word_Recognition

[18] M. Rath, R. Manmatha, "Word Image Matching Using Dynamic Time Warping", accedido Febrero 2016. [Online] Disponible: <http://ciir-publications.cs.umass.edu/pdf/MM-38.pdf>

[19] T. Giorgino, "Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package", accedido Febrero 2016. [Online] Disponible: <https://cran.r-project.org/web/packages/dtw/vignettes/dtw.pdf>

[20] (2015, Feb. 25) "OpenCV API Reference" [Online] Disponible: <http://docs.opencv.org/2.4.11/modules/refman.html>

[21] "DANE, Marco legal de la discapacidad" accedido Marzo 2015 [Online] Disponible: https://www.dane.gov.co/files/investigaciones/discapacidad/marco_legal.pdf

[22] (2013, Feb. 27) "Ley estatutaria 1618" [Online] Disponible: <http://wsp.presidencia.gov.co/Normativa/Leyes/Documents/2013/LEY%201618%20DEL%2027%20DE%20FEBRERO%20DE%202013.pdf>

[23] L. Enrique Sucar, G. Gómez, "Visión computacional", accedido Abril 2015. [Online] Disponible: <http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>

- [24] “Introducción al reconocimiento de objetos”, accedido Abril 2015. [Online] Disponible: <http://alojamientos.us.es/gtocom/pid/tema7.pdf>
- [25] R. Kimmel, “Fast Edge Integration”, accedido Abril 2015 [Online]. Disponible en: http://www.cs.technion.ac.il/~ron/PAPERS/Paragios_chapter2003.pdf
- [26] J. Romañach, M. Lobato, (2005, Mayo) “Diversidad funcional, nuevo término para la lucha por la dignidad en la diversidad del ser humano” [Online] Disponible: http://www.asocias.org/vidaindepend/docs/diversidad%20funcional_vf.pdf
- [27] “Time series” accedido Febrero 2016. [Online] Disponible: <http://www.investopedia.com/terms/t/timeseries.asp>
- [28] A. Herrera, (2000) “La clasificación numérica y su aplicación en la ecología” [Online] Disponible: [http://programaecomar.com/CapitulosLC/1\)Introduccion_Pp1a17.pdf](http://programaecomar.com/CapitulosLC/1)Introduccion_Pp1a17.pdf)
- [29] (2008-2016) “Definición: Contorno” [Online] Disponible: <http://definicion.de/contorno/>
- [30] N. R Melo, “La lengua de Señas Colombiana”, accedido Abril 2015 [Online] Disponible: <http://www.lenguasdecolombia.gov.co/content/lengua-de-se%C3%B1as-colombiana>
- [31] D. Betancur, M. Gómez, A. Peña, “Traducción automática del lenguaje dactilológico de sordos y sordomudos mediante sistemas adaptativos,” Rev. ing. Biomédica ISSN 1909-9762 vol. 7 no. 13, pp.18-30, Enero-Junio de 2013.
- [32] M. González, J. Mejía, “Diseño e implementación de un servicio web que permita crear y consultar un vocabulario de la lengua de señas y visagrafía partiendo de una palabra del español” Memoria de título, Ing. De sistemas y computación, facultad de ingenierías eléctrica, electrónica, física y ciencias de la computación, Univ. Tecnológica de Pereira, Pereira, Colombia, 2010
- [33] Corporación Colombia digital, (2013, Ago. 15) “Colombiano desarrolla primer traductor online de lengua de señas” [Online] Disponible: <http://colombiadigital.net/actualidad/noticias/item/5452-colombiano-desarrolla-primer-traductor-online-de-lenguas-a-senas.html>
- [34] E. de la torre Castejón, “Signslator, el primer traductor de lengua de signos española”, accedido Marzo 2015 [Online] Disponible: <http://enpositivo.com/2014/07/signslator-el-primer-traductor-de-lengua-de-signos-espanola/>

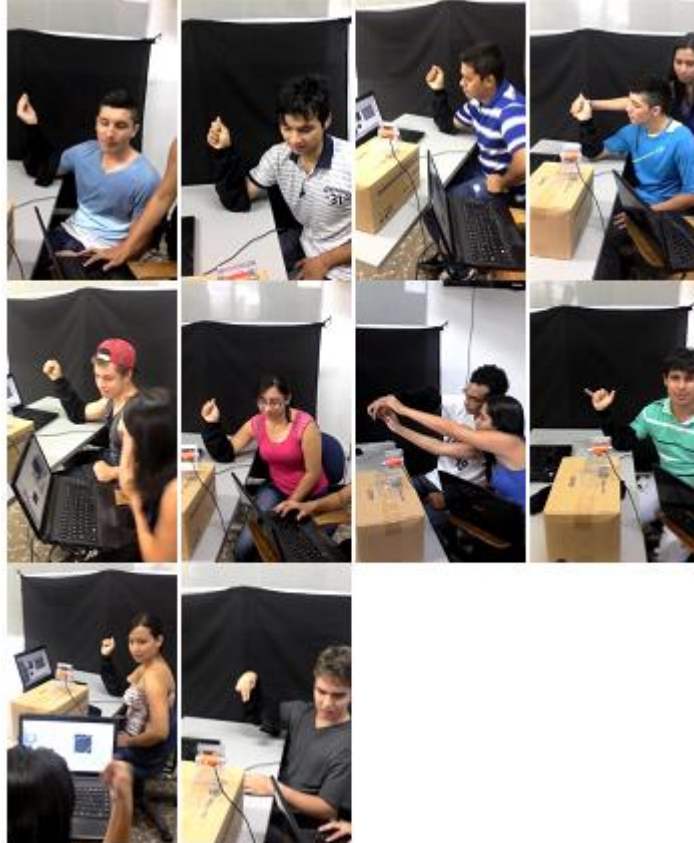
[35] A. López, "Reconocimiento automático de lenguaje de signos: Lenguaje ASL" Memoria de título, Ing. Técnico en informática de sistemas, Facultad de matemáticas, Univ. De Barcelona, Barcelona, España, Julio 20 de 2009

ANEXOS

Anexo 1

Evidencia fotográfica de la prueba realizada a 15 personas figura 19, para la validación del funcionamiento adecuado del sistema. La indumentaria usada fue un caballete que sostiene la tela de color oscuro, la mesa que sirve de apoyo para las personas de prueba y la caja para ubicar la cámara a la altura del mismo nivel de la mano.

Figura 19: Evidencia fotográfica de la prueba realizada a 15 personas



[Creación Propia]

Anexo 2

Entrevista

Esta entrevista fue dirigida al estudiante de ingeniería en sistemas Álvaro Andrés Loaiza, quien presenta pérdida de audición del 60% en su oído izquierdo y del 65% en su oído derecho, trabajó en fundaciones con personas sordas donde interactuó con ellas por medio de lenguaje de señas y está en pro a la inclusión social.

1. ¿Las señas almacenadas en la aplicación del abecedario del LSC corresponde a las utilizadas en la cotidianidad de las personas con dificultad en el habla?

Si corresponde, la única dificultad con la aplicación es que no permitía hacer la seña de manera natural, dependía mucho de la forma de la mano aspectos como la inclinación, que tan separado pueden estar los dedos, que incline un poco aquí, entre otros.

2. ¿Es posible transmitir una oración corta a partir de la respuesta del sistema?

Si es posible, pero de modo lento y pausado, quizás si le logre la posibilidad más rápidamente almacenando en el sistema, de primera mano, imágenes y los contornos con la mano de quien vaya a utilizar, y no almacenado internamente con los de otro usuario, debido a la dificultad de la aplicación de interpretar con contornos distintos.

3. ¿Qué opinión tiene del sistema?

El sistema lo considero muy interesante lo que puede desarrollarse a futuro, permitiéndole desarrollar con más técnicas dentro de la ciencias de la computación tales como inteligencia artificial, desarrollar una interfaz y que se pueda usar con dispositivos móviles y pueda interpretar señas sin necesidad de ambientes adecuados para la aplicación, sino en entornos naturales, además es muy útil para la población sorda poder escribir a través de seña e incluso aprovecharse como intérpretes entre una persona que no tenga conocimiento y poder así comunicarse con los demás, caso paralelo como los traductores instantáneos cuando nos encontramos en un país con un idioma en el cual no tengamos conocimiento.

4. ¿Alguna sugerencia hacia el sistema?

Como principal sugerencia, es permitir realizar la seña de manera natural sin necesidad de realizar correcciones por parte de quienes desarrollan el aplicativo.

Anexo 3

Repositorio de la Aplicación

<https://github.com/Jpolo00/InterpreteLSC>