

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Generación de Palabras a Partir de la Lengua de Señas
Mexicana**

presentada por
Ing. Luis Obed Romero Nájera

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Máximo López Sánchez

Codirector de tesis
Dr. Juan Gabriel González Serna

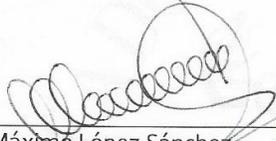
Cuernavaca, Morelos a 10 de enero del 2017
OFICIO No. DCC/001/2017

Asunto: Aceptación de documento de tesis

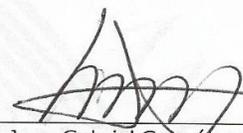
DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **Ing. Luis Obed Romero Nájera**, con número de control M14CE069, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Generación de palabras a partir de la Lengua de Señas Mexicana**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS


Dr. Máximo López Sánchez
Doctor en Ciencias de la
Computación
7498547

CO-DIRECTOR


Dr. Juan Gabriel González Serna
Doctor en Ciencias de la
Computación
7820329

REVISOR 1


Dr. Noé Alejandro Castro Sánchez
Doctor en Ciencias de la
Computación
08701806

REVISOR 2


Dr. Dante Mújica Vargas
Doctor en Comunicaciones y
Electrónica
09131756

C.p. M.T.I. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

NACS/Imz

Cuernavaca, Mor., 10 de enero de 2017
OFICIO No. SAC/030/2017

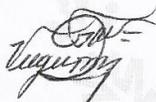
Asunto: Autorización de impresión de tesis

ING. LUIS OBED ROMERO NÁJERA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“Generación de Palabras a Partir de la Lengua de Señas Mexicana”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
“CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO”



DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.T.I. María Elena Gómez Torres .- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

Dedicatoria

A Dios, mis padres y todas esas personas que me apoyaron y creyeron en mí para terminar este trayecto de mi vida.

"[...] Hasta aquí nos ha ayudado Dios."

Agradecimientos

A Dios por todas las bendiciones dadas a mi vida, por permitirme culminar este grado más, por dejarme estar en este lugar y conocer a personas que nunca dejaron de apoyar y que ahora forman parte importante en mi vida.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico que me fue otorgado para desarrollar este trabajo de investigación.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por acobijarme durante el tiempo que duro mi maestría.

A mis padres, hermanos y familia que estuvieron siempre conmigo apoyándome cuando los necesitaba, el apoyo incondicional que siempre recibí estando lejos de ellos, sin mis padres no podría haberlo logrado.

Al Dr. Máximo López Sánchez por su apoyo, consejos, instrucciones, regaños y confianza que deposito en mí para realizar este trabajo de investigación y sobre todo la paciencia, sé que no es fácil pero ha sido un grato trabajar con él. ¡Gracias Dr.!

A mi Co-director el Dr. Juan Gabriel González Serna y mis revisores el Dr. Noé Alejandro Castro Sánchez, el Dr. Dante Mújica Vargas y el Dr. Andrés Blanco Ortega por todas las observaciones, opiniones y consejos recibidos en cada una de las presentaciones para mejorar este trabajo de investigación, muchas gracias por todo.

A mis amigos de la maestría, los de la iglesia y del trabajo, por apoyarme siempre y nunca permitir que me rindiera, siempre han estado allí cuando más se necesitan, muchas gracias a todos.

A las empresas PROSESA MEDICA (*Fer y Metz*) e INEEL (*Yahreli, Felipe, Paty e Issac*), por su confianza que depositaron en mi para poder realizar mis servicios laborarles, ya que sin el apoyo económico que brinda el trabajo me hubiera sido más difícil terminar la maestría, muchas gracias.

A una persona en especial que siempre me dio su apoyo incondicional durante este trayecto de la maestría con sus regaños cuando deseaba rendirme y nunca dejármelo hacer y por su amistad durante todo este tiempo ¡¡Muchas Gracias!!

[...] Hagan todo con humildad, y vean a los demás como mejores a ustedes mismos.

Abstract

Communication is part of the daily life basic activities, it reflects the interaction of the human organism with the environment. The difficulties to interact through language are relevant for the integration of individuals and for their access to various types of services; therefore, an accessible environment favors personal autonomy and participation in civic life, being essential for inclusion in education and health services, as well as the labor market; in negative terms, it introduces barriers that limit both participation and social inclusion of people with this type of disability (INEGI, 2010).

In Mexico in 2010, INEGI stated that 8.3% of the country's population with disabilities have limitations to talk or communicate. Thus, this type of limitation ranks as the fifth in the frequency among the group of people with disabilities, surpassed by mobility, visibility, hearing and mental limitations (INEGI, 2010).

Hearing impairment is the restriction on the perception of sounds, when hearing loss is present and it is not functional for daily life, the acquisition of oral language does not occur naturally, in Mexico the natural language of people with this condition is the Mexican Sign Language (CONADIS, 2012).

In this research work, a web application called "SignMx" was developed. This application recognizes the signs made to an HCI device (Human Computer Interfaces) and then writes them and reproduces them by voice, in this way it is desired to help people with hearing impairment so they can communicate with people who do not have this disability. In order to make the application, it was necessary to develop a methodology which is described in detail in chapter 4.

The application function consists in generating words from made signs. The development of the application was possible with the help of the Leap Motion Controller device because with the LMC, through their cameras and infrared LEDs and in conjunction with the framework LeapTrainer.js, it was able to capture and recognize the made signs and then write them and reproduce them from text to voice.

Resumen

La comunicación forma parte de las actividades básicas de la vida cotidiana, refleja la interacción del organismo humano con el ambiente. Las dificultades para interactuar por medio del lenguaje resultan relevantes para la integración del individuo y para su acceso a varios tipos de servicios; por ello, un entorno accesible favorece la autonomía personal y la participación en la vida cívica, resultando esencial para incorporarse a los servicios de educación y de salud, así como al mercado de trabajo; en términos negativos, introduce barreras que limitan tanto la participación como la inclusión social de las personas que tienen este tipo de discapacidad (INEGI, 2010).

En México en el 2010, declararon tener limitaciones para hablar o comunicarse 8.3% de la población con discapacidad del país. Así, este tipo de dificultad se ubica como la quinta en cuanto a la frecuencia entre el colectivo de personas con discapacidad, superada por las limitaciones de movilidad, de ver, de oír y las mentales (INEGI, 2010).

La discapacidad auditiva es la restricción en la función de la percepción de los sonidos, cuando se tiene la pérdida auditiva y esta no es funcional para la vida diaria, la adquisición de la lengua oral no se da de manera natural, en México la lengua natural de las personas con esta condición es la Lengua de Señas Mexicana (CONADIS, 2012).

En este trabajo de investigación se desarrolló una aplicación web llamada “*SignMx*” la cual reconoce las señas realizadas ante un dispositivo HCI (en inglés, *Human Computer Interfaces*) para luego ser escritas y reproducidas a voz, de esta manera se desea ayudar para las personas sordas tengan una comunicación con las personas que no padecen esta discapacidad. Para llevar a cabo la aplicación fue necesario desarrollar una metodología la cual se describe a detalle en el capítulo 4.

La aplicación consiste en generar palabras a partir de las señas realizadas, con la ayuda del dispositivo Leap Motion Controller fue posible realizar la aplicación debido a que el LMC a través de sus cámaras y led infrarrojos y en conjunto con el *framework* LeapTrainer.js fue capaz de capturar y reconocer las señas para luego ser escritas y reproducirse de texto a voz.

Acrónimos, Abreviaturas y Siglas

API: *Application Programming Interface*

ASL: *American Sign Language.*

ASR: *Mexican Automatic Speech Recogniser*

BLS: *British Sign Language*

CIU: *Competitive Intelligence Unit*

CONADIS: Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad.

DTW: *Dynamic Time Warping*

HMM: *Hidden Markov Models*

INEGI: Instituto Nacional de Estadística y Geografía

IR: *InfraredRays*

KSLT: *Kinect Sign Language Translator*

LMC: *Leap Motion Controller.*

LSA: Lengua de Señas Argentino

LSM: Lengua de Señas Mexicana.

MFCC: *Mel Frequency Cepstral Coefficients*

MLP: *Multiplayer Perceptron Neural networks*

MVC: Modelo Vista Controlador

NBC: *Nave Bayes Classifier*

NLP: *Neuro Linguistic Programming*

OMS: Organización Mundial de la Salud.

OTG: *On The Go*

PSLT: *Portable Sign Language Translator*

RGB: *Red, Green, Blue*

SDK: Software Development Kit

TTS: Text-to-Speech

USB: *Universal Serial Bus*

Índice

Capítulo 1	1
Introducción.....	1
1.1 Introducción	2
1.2 Planteamiento del problema	3
1.3 Justificación	4
1.4 Objetivos.....	5
1.4.1 Objetivo general	5
1.4.2 Objetivo específico	5
1.5 Alcances y limitaciones	6
1.5.1 Alcances	6
1.5.2 Limitaciones	6
1.6 Metodología de solución	7
1.7 Organización de la tesis	9
Capítulo 2	10
Marco Conceptual	10
2.1 Lengua de Señas Mexicano.....	11
2.2 Leap Motion Controller.....	13
2.3 Android	14
2.4 Microsoft Windows.....	14
2.5 TTS (Text to Speech).....	14
2.6 LeapTrainer.js	15
Capítulo 3	16
Estado del arte.....	16
3.1 Trabajos relacionados	17
3.1.1 UNI tablet (MotionSavvy, 2015)	17
3.1.2 Google Gesture (Google Gesture, 2014)	17
3.1.3 Kinect Sign Language Translator (KSLT, 2014)	18
3.1.4 Enable Talk (Talk Enable, 2015)	19
3.1.5 ONIS (RedUSERS, 2013).....	19
3.1.6 Portable Sign Language Translator (PSLT, 2014)	20
3.1.7 Descripción de los campos de trabajos relacionados.....	20
3.1.8 Tabla comparativa de trabajos relacionados	21
3.2 Estado del Arte	22
3.2.1 Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado (Chiguano, Rodríguez & Moreno, Díaz, 2011).....	22
3.2.2 Reconocimiento de imágenes del lenguaje de señas mexicano (Priego, Pérez, Fausto, 2012).	23
3.2.3 Transforming Indian Sign Language into Text Using Leap Motion (P.Karthick, N.Prathiba, V.B.Rekha, & S.Thanalaxmi, 2014).	24
3.2.4 Arabic Sign Language Recognition using the Leap Motion Controller (Mohandes, Aliyu, & Deriche, 2014).....	24

3.2.5 Towards the Development of a Mexican Speech-to-Sign-Language Translator for the Deaf Community (Trujillo, Romero & Caballero, Morales, 2012).....	25
3.2.6 Descripción de los campos de trabajos relacionados	26
3.2.7 Tabla comparativa del estado del arte	27
Capítulo 4.....	28
Metodología de solución.....	28
4.1 Estudio y análisis de los algoritmos para el reconocimiento de patrones aplicados a la LSM.....	29
4.1.1 Modelo Oculto de Markov	29
4.1.1.1 Definición de Modelos Ocultos de Markov	29
4.1.1.2 Arquitecturas de Modelos Ocultos de Markov	29
4.1.1.3 Definición de los elementos de un Modelos Ocultos de Markov	30
4.1.1.4 Problemas básicos de los Modelos Ocultos de Markov	30
4.1.2 Máquina de Soporte Vectorial	31
4.1.2.1 SVM lineal con margen máximo	32
4.1.3 Multicapa de Red Perceptrón	34
4.1.3.1 Modelos de una neurona	35
4.1.3.2 Algoritmo de back-propagation	36
4.1.4 Alineamiento Temporal Dinámico	37
4.1.4.1 Algoritmo DTW	38
4.1.4.2 Restricciones	38
4.2 Clasificación manual del LSM a reconocer.....	40
4.2.1 Abecedario	40
4.2.2 Números.....	43
4.2.3 Palabras	44
4.3 Conexión exitosa del Smartphone con Leap Motion Controller	50
4.3.1 Requerimientos para conexión Smartphone – Leap Motion Controller	51
4.4 Aprendizaje de la Lengua de Señas Mexicana	54
4.4.1 Uso del LeapTrainer.js	55
4.4.2 Opciones del Leaptrainer.js.....	55
4.5 Algoritmo usado para la grabación de la Lengua de Seña Mexicana	57
4.5.1 Entrenamiento de las señas	58
4.5.2. Señas reconocidas	61
4.5.3 Total de señas entrenadas y grabadas	62
4.6 Creación de la aplicación SignMx	63
4.7 Generación de texto a voz	65
4.8 Pruebas de la aplicación.	66
Capítulo 5.....	69
Pruebas y resultados	69
5.1 Introducción a las pruebas	70
5.2 LeapTrainer.js	71
5.3 Pruebas	72
5.3.1 Descripción de la prueba.....	72
5.3.2 Prueba número	73

5.3.3 Prueba abecedario.....	73
5.3.4 Prueba palabras	74
5.3.5 Tipos de prueba	75
5.3.6 Ambiente para realizar la prueba	75
5.3.7 Población utilizada para las pruebas	76
5.4 Resultados	77
5.4.1 Grupo no supervisado	77
5.4.1.1 NS_F_1	77
5.4.1.2 NS_M_2.....	78
5.4.1.3 NS_M_3.....	78
5.4.1.4 NS_M_4.....	79
5.4.1.5 NS_M_5.....	79
5.4.1.6 NS_M_6.....	80
5.4.2 Grupo supervisado	81
5.4.2.1 S_M_1	81
5.4.2.2 S_M_2	81
5.4.2.3 S_M_3	82
5.4.2.4 S_M_4	82
5.4.2.5 S_M_5	83
5.4.2.6 S_F_1.....	83
5.4.3 Gráficas.....	84
5.4.3.1 No supervisado	84
5.4.3.2 Supervisado.....	86
5.4.3.3 Comparación de resultados	88
5.4.3.4 Resumen de la prueba.....	90
Capítulo 6.....	91
Conclusiones y trabajos futuros.....	91
6.1 Conclusiones.....	92
6.2 Proyecto y publicación.....	92
6.3 Trabajos futuros	93
Referencias	94
Anexos.....	96
Anexo A – Señas en 3D reconocidas por SignMx.....	97

Índice de Figuras

Figura 1.1 Metodología de solución	8
Figura 2.1 Mamá dactilología (Serafín de Fleischmann & González, 2011)	11
Figura 2.2 Mamá ideograma (Serafín de Fleischmann & González, 2011)	12
Figura 2.3 Alfabeto del LSM (Priego, Pérez, Fausto, 2012)	12
Figura 2.4 Vista esquemática del LMC (Weichert et al., 2013)	13
Figura 2.5 Sistema diestro de coordenadas (LeapMotion, 2015)	13
Figura 2.6 LeapTrainer.js (O'Leary, 2016).....	15
Figura 3.1 Uni Tablet. (MotionSavvy, 2015).....	17
Figura 3.2 Muñequera (Google Gesture, 2014)	18
Figura 3.3 Kinect Sign (KSLT, 2014).....	18
Figura 3.4 Guante (Talk Enable, 2015)	19
Figura 3.5 ONIS (RedUSERS, 2013)	19
Figura 3.6 PSLT (PSLT, 2014)	20
Figura 4.1 Ejemplo de una SVM	32
Figura 4.2 SVM con margen máximo (en negro representados los vectores soporte)	34
Figura 4.3 Perceptrón de Rosenblatt.....	34
Figura 4.4 Modelo no lineal de una neurona	35
Figura 4.5 Gráfico de señales en una neurona de salida.....	36
Figura 4.6 Secuencias A y B.....	37
Figura 4.7 Ejemplo al aplicar el algoritmo DTW	38
Figura 4.8 Formulario Leap Motion Android SDK	50
Figura 4.9 Archivos proporcionados por Leap Motion	51
Figura 4.10 Instalación de la aplicación.....	52
Figura 4.13c LMC detectado y conectado.....	52
Figura 4.13a Aplicación instalada	52
Figura 4.13b LMC no detectado o no conectado	52
Figura 4.14 Pruebas del funcionamiento del LMC.....	53
Figura 4.15 LeapTrainer.js (Rob O'Leary, 2016)	55
Figura 4.16 Opciones LeapTrainer (Rob O'Leary, 2016)	56
Figura 4.17 Letra "a" realizada.....	58
Figura 4.18 Letra "a" aprendida.....	59
Figura 4.19 Letra "a" porcentaje de coincidencia	59
Figura 4.20 JSON de la seña "a"	60
Figura 4.21 Dato JSON de las señas.....	60
Figura 4.22 Estructura SignMx	63
Figura 4.23 Interfaz de usuario SignMx.....	64
Figura 4.24 Compatibilidad (Resposive Voice, 2016)	65
Figura 4.25 Prueba de la aplicación, seña de la letra "a"	66
Figura 4.26 Prueba de la aplicación, seña de la letra "b"	66

Figura 4.27 Información en consola del navegador	67
Figura 4.28 Prueba con el LeapTrainer, seña de la letra "a"	68
Figura 4.29 Prueba con el LeapTrainer, seña de la letra "d"	68
Figura 5.1 LeapTrainer.js (O'Leary, 2016).....	71
Figura 5.2 Porcentaje de reconocimiento LeapTrainer.js (O'Leary, 2016).....	71
Figura 5.3 Prueba números.....	73
Figura 5.4 Prueba abecedario	73
Figura 5.5 Prueba palabras	74
Figura 5.6 Promedio números – no supervisado	84
Figura 5.7 Promedio abecedario - no supervisado	84
Figura 5.8 Promedio palabras - no supervisado.....	85
Figura 5.9 Promedio general de la prueba - no supervisado	85
Figura 5.10 Promedio números - supervisado	86
Figura 5.11 Promedio abecedario - supervisado	86
Figura 5.12 Promedio palabras - supervisado.....	87
Figura 5.13 Promedio general de la prueba - supervisado	87
Figura 5.14 Comparación de resultados prueba números.....	88
Figura 5.15 Comparación de resultados prueba abecedario	88
Figura 5.16 Comparación de resultados prueba palabras.....	89
Figura 5.17 Comparación general de los resultados de las pruebas	89

Índice de Tablas

Tabla 1.1 INEGI, Encuesta Nacional de Ingresos y Gastos de los Hogares (ENIGH 2012)	4
Tabla 3.1 Tabla comparativa de trabajos relacionados	21
Tabla 3.2 Tabla comparativa del estado del arte	27
Tabla 4.1 Requerimientos LMC Android	51
Tabla 4.2 Señas de la LSM	54
Tabla 4.3 Señas reconocidas	61
Tabla 4.4 Total de señas propuestas y reconocidas	62
Tabla 5.1 Tabla grupo no supervisado	76
Tabla 5.2 Tabla grupo supervisado	76
Tabla 5.3 Resultados del grupo no supervisada femenina 1	77
Tabla 5. 4 Resultados del grupo no supervisada masculino 1	78
Tabla 5.5 Resultados del grupo no supervisada masculino 2	78
Tabla 5.6 Resultados del grupo no supervisada masculino 3	79
Tabla 5.7 Resultados del grupo no supervisada masculino 4	79
Tabla 5.8 Resultados del grupo no supervisada masculino 5	80
Tabla 5.9 Resultados del grupo supervisado masculino 1	81
Tabla 5.10 Resultados del grupo supervisado masculino 2	81
Tabla 5.11 Resultados del grupo supervisado masculino 3	82
Tabla 5. 12 Resultados del grupo supervisado masculino 4	82
Tabla 5.13 Resultados del grupo supervisado masculino 5	83
Tabla 5.14 Resultados del grupo supervisado femenina 1	83
Tabla 5.15 Resultados finales SignMx	90

Capítulo 1

Introducción

En este capítulo se muestra la información que permitirá al lector tener un panorama más general del tema de investigación que se desarrolló en este trabajo de tesis. Al final del capítulo se muestra la organización de la tesis.

1.1 Introducción

La comunicación forma parte de las actividades básicas de la vida cotidiana, refleja la interacción del organismo humano con el ambiente. Las dificultades para interactuar por medio del lenguaje resultan relevantes para la integración del individuo y para su acceso a varios tipos de servicios; por ello, un entorno accesible favorece la autonomía personal y la participación en la vida cívica, resultando esencial para incorporarse a los servicios de educación y de salud, así como al mercado de trabajo; en términos negativos, introduce barreras que limitan tanto la participación como la inclusión social de las personas que tienen este tipo de discapacidad (INEGI, 2010).

La comunicación es el corazón de las dificultades y de los éxitos, asociados con la sordera [...] y la sordera no se refiere a la audición. Se refiere a la comunicación. Al final de cuentas, la comunicación no está limitada al intercambio de ideas o de información, o inclusive emociones y experiencias, la comunicación se relaciona con el conocerse el uno al otro, y el conocer es empezar a entender (Ogden, 2002).

La Ley General para la Inclusión de las Personas con Discapacidad (Texto vigente publicado en el Diario Oficial de la Federación el 30 de mayo de 2011) dice:

"La lengua de una comunidad de sordos, que consiste en una serie de signos gestuales articulados con las manos y acompañados de expresiones faciales, mirada intencional y movimiento corporal, dotados de función lingüística, forma parte del patrimonio lingüístico de dicha comunidad y es tan rica y compleja en gramática y vocabulario como cualquier lengua oral." (Gobernación, 2012).

En el 2010 declararon tener limitaciones para hablar o comunicarse 8.3% de la población con discapacidad del país. Este tipo de dificultad se ubica como la quinta en cuanto a la frecuencia entre el colectivo de personas con discapacidad, superada por las limitaciones de movilidad, de ver, de oír y las mentales (INEGI, 2010).

1.2 Planteamiento del problema

Con base en la Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (mejor conocida como CIF), la discapacidad es un término que engloba deficiencias, limitaciones a la actividad y restricciones a la participación (CONADIS, 2012).

La discapacidad auditiva es la restricción en la función de la percepción de los sonidos, cuando se tiene la pérdida auditiva y esta no es funcional para la vida diaria, la adquisición de la lengua oral no se da de manera natural, en México la lengua natural de las personas con esta condición es la Lengua de Señas Mexicana (CONADIS, 2012).

De acuerdo con la Encuesta Nacional de Ingresos y Gastos de los Hogares 2012, en México 6.6% de la población presenta dificultad (discapacidad) para realizar al menos una de las actividades medidas. La INEGI muestra la discapacidad de hablar o comunicarse como el 8.6% y la discapacidad de escuchar cómo el 16.5% los cuales fueron arrojados por su Encuesta Nacional de Ingresos y Gastos de los Hogares 2012 (INEGI, 2013).

Por otro lado, se puede observar que en las investigaciones realizadas del estado del arte referidas a la detección de la lengua de señas para las personas que tiene la discapacidad escuchar y hablar; se puede apreciar que existen sistemas que detectan las señas, pero en algunas solo detectan el abecedario (Chiguano, Rodríguez & Moreno, Díaz, 2011), (Priego, Pérez, Fausto, 2012), (Mohandes et al., 2014) y solo uno de ellos está realizado para la lengua de señas mexicana (Priego, Pérez, Fausto, 2012), sin embargo el problema que se presenta en los sistemas mencionados es que solo existe una para la Lengua de Señas Mexicana, la portabilidad que existe es otro problema ya que algunos deben estar en un ambiente controlado, algunos es necesario tenerlo en un lugar fijo y ninguno de ellos cuentan con alguna aplicación móvil.

La solución planteada a dicho problema es: Proponer un medio que facilite la comunicación entre personas con discapacidad de escuchar y hablar con personas que no padecen esta discapacidad.

1.3 Justificación

Las personas sordas muchas veces tienen dificultades para contratar y capacitar a intérpretes, en particular en las comunidades rurales o aisladas. Una encuesta sobre la situación de los derechos humanos de las personas sordas reveló que 62 de los 93 países que respondieron tienen servicios de interpretación de lengua de señas, 43 tienen algún tipo de capacitación para intérpretes de lengua de señas y 30 tienen como máximo 20 intérpretes de lengua de señas debidamente capacitados, entre ellos, en Iraq, Madagascar, México, la República Unida de Tanzania, el Sudan, y Tailandia (OMS, 2011).

De acuerdo con la Encuesta Nacional de Ingresos y Gastos de los Hogares 2012, en México 6.6% un aproximado de 7,751,677 personas presentan dificultad (discapacidad), los problemas para caminar son el tipo de discapacidad de mayor presencia (57.5%), seguido de las dificultades para ver (32.5%), oír (16.5%), hablar o comunicarse (8.6%), mental (8.1%), atender el cuidado personal (7.9%) y, finalmente, poner atención (6.5 por ciento) (Tabla 1.1)

Sexo y grupo de Edad	Tipos de Discapacidad						
	Caminar	Ver	Hablar o Comunicarse	Escuchar	Atender el cuidado personal	Poner atención o aprender	Mental
Total	57.5	32.5	8.6	16.5	7.9	6.5	8.1
Hombres	54.8	29	9.7	17.4	7.5	7.3	10.7
Mujeres	59.9	35.8	7.7	15.6	8.3	5.8	5.9
Niños	30.4	25.7	23.6	10.2	7.7	22.1	18.6
Jovenes	23.5	36.2	20.5	12.0	5.0	16.5	20.9
Adultos	49.0	33.8	5.7	10.8	4.7	4.1	12.3
Adultos Mayores	71.9	32.1	6.7	21.8	10.6	4.4	2.1

Tabla 1.1 INEGI, Encuesta Nacional de Ingresos y Gastos de los Hogares (ENIGH 2012)

Una de las opciones para ayudar a realizar la comunicación es el uso de tecnología, entre ellas está la tecnología móvil. En México al final del segundo trimestre de 2014 se alcanzaron 102.9 millones de líneas móviles, de las cuales 42.3% son Smartphone, equivalente a 43.5 millones en manos de los consumidores mexicanos y 51% más con respecto al mismo periodo del año pasado (CIU, 2014). En lo que se refiere a sistemas operativos el software de plataforma de código abierto Android sigue siendo adoptado por distintas marcas de celulares, esto ha provocado un crecimiento acelerado de dicho sistema operativo, al cierre de 2013 se ubicaba en primer lugar, con 52.3% de la participación del mercado, en segundo lugar se encuentra a BlackBerry con 16.3% y en el tercer lugar se encuentra iOS con 13.4% (CUI, 2013).

1.4 Objetivos

1.4.1 Objetivo general

EL objetivo principal de este trabajo de investigación es el desarrollar una aplicación para dispositivos móviles inteligentes para el reconocimiento de la Lengua de Señas Mexicana y traducirlo a palabras.

1.4.2 Objetivo específico

Los objetivos específicos de este trabajo de investigación se indican a continuación:

- Desarrollar un algoritmo que pueda ser aplicado en dispositivos móviles inteligentes para el reconocimiento de la Lengua de Señas Mexicana (LSM).
- Implementar un algoritmo para el reconocimiento de la Lengua de Señas Mexicana (LSM).
- Desarrollar e implementar una interfaz de usuario.
- Realizar la conexión exitosa del *Smartphone* hacia el *Leap Motion Controller (LCM)*.
- Definir y utilizar un algoritmo para guardar las señas reconocidas por el *Leap Motion Controller (LMC)*.
- Realizar el diseño e implementación de una base de conocimiento para señas guardadas.

1.5 Alcances y limitaciones

1.5.1 Alcances

- La aplicación detectará un determinado número de palabras.
- Se podrá mostrar en el *Smartphone* el texto de lo que la seña significa.
- Se mostrará de modo audible el texto de la palabra generada por las señas.
- Se mostrará la seña en pantalla cuando está sea escrita con el teclado del dispositivo inteligente.
- Se podrá usar como medio de enseñanza para las personas que no saben la LSM.
- Se generarán palabras tanto del modo de seña ideogramas.
- Se podrán crear palabras con las letras del abecedario (modo dactilología).

1.5.2 Limitaciones

- La generación de palabras será únicamente para la LSM.
- No se podrán guardar nuevas señas.
- El *Smartphone* deberá de ser gama media – alta debido a que el LMC generará un modelo en 3D de la seña.
- Las señas que utilicen gestos faciales no serán detectadas por las limitaciones que el LMC presenta.
- Algunas señas que utilizan el gesto corporal no serán detectadas por las limitaciones que el LMC presenta.
- La aplicación será desarrollada para *Android* o *Windows* dependiendo las prestaciones que ofrezca LMC.

1.6 Metodología de solución

Con la finalidad de dar solución a la problemática que se presenta, se desarrolló una metodología de solución (Figura 1.1), la cual se compone de 11 procesos distribuidos en 4 fases.

A continuación se muestra una descripción de las actividades.

Fase 1. Estudio y Análisis

1. **Estudio y análisis de los algoritmos para el reconocimiento de patrones aplicados a la LSM:** En este proceso se realizó un estudio de los algoritmos existentes para el reconocimiento de patrones y se analizó la precisión que cada uno presenta para así elegir el algoritmo que fue usado para la detección de las señas.
2. **Clasificación manual del LSM a reconocer:** Con el visualizador que incluye el instalador del *Leap Motion* se realizó una selección manual de las señas que fueron reconocidas por el *Leap Motion Controller* y se clasificaron las posibles señas a reconocer.
3. **Conexión exitosa del Smartphone con LMC:** En este proceso con el cable USB OTG (*On-The-Go*) se conectó el *Leap Motion Controller* con el Smartphone para tener una comunicación entre dispositivos.

Fase 2. Desarrollo

4. **Aprendizaje del LSM:** En este proceso se aprendió las señas que usa la Lengua de Señas Mexicana a partir de la clasificación manual, al tener esta clasificación fue de ayuda para aprender únicamente las señas que fueron candidatas a reconocer.
5. **Generación de ID por el LMC de la seña:** En este proceso se generó ID únicos para cada seña reconocida y así tener un mejor control de todas las señas a detectar.
6. **Desarrollo del algoritmo de reconocimiento de patrones seleccionado al LSM:** En este proceso se implementó un algoritmo para el reconocimiento de las señas que fueron seleccionadas.

Fase 3. Codificación

7. **Creación de la base de conocimiento:** En este proceso se generó una base de conocimiento de todas aquellas señas detectadas por el LMC.
8. **Detección y reconocimiento del LSM con LMC:** En este proceso se desarrolló la aplicación la cual detectó y reconoció las señas realizadas para ser visualizado en texto.
9. **Generación del texto a voz:** En este proceso se generó la codificación para reproducir en audio de la seña reconocida y escrita en el dispositivo.

Fase 4. Pruebas

1. **Pruebas de la aplicación:** En este proceso se realizó la prueba de la aplicación terminada para encontrar errores en la detección de las señas, diseño de la interfaz y la conectividad entre dispositivos
2. **Documentación de las pruebas:** En este proceso se realizó el plan de pruebas que se llevaría a cabo para valorar la aplicación desarrollada, de igual manera se anexa la documentación de las pruebas realizadas.

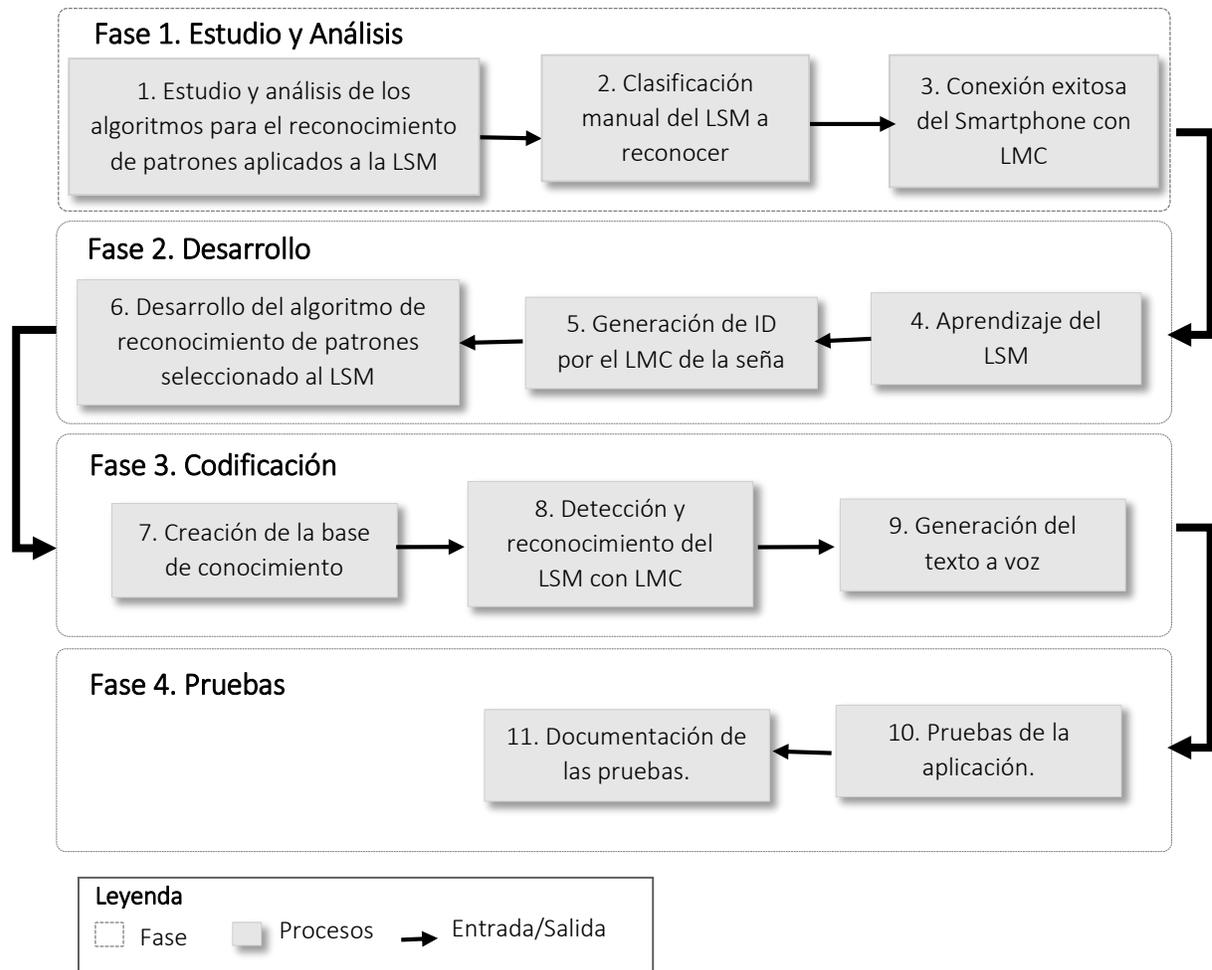


Figura 1.1 Metodología de solución

1.7 Organización de la tesis

La estructura en la que se encuentra organizado este trabajo de investigación es la siguiente:

- Capítulo 2.** En este capítulo se presenta el marco conceptual donde se describen y se definen los conceptos utilizados en este trabajo de investigación que de igual manera le servirá al lector para la facilidad de la lectura del documento.
- Capítulo 3.** En este capítulo se presenta un panorama general del estado del arte en las áreas de investigación que se consideraron importantes para esta investigación. En la Sección 3.1 presenta los trabajos relacionados y en la Sección 3.2 el estado del arte.
- Capítulo 4.** En este capítulo se presenta la metodología de solución propuesta, el desarrollado de cada una de las actividades y la implementación de cada fase de la metodología para así resolver la problemática planteada en este trabajo de investigación.
- Capítulo 5.** En este capítulo se presentan las pruebas que fueron realizadas para valorar la metodología de solución y de igual manera se muestran los resultados obtenidos.
- Capítulo 6.** En este capítulo se presenta la conclusión general de este trabajo de investigación, así también los trabajos futuros que se pueden desarrollar.

Capítulo 2

Marco Conceptual

En este capítulo se aborda el marco conceptual donde se describen y se definen los conceptos más relevantes utilizados en este trabajo de investigación que de igual manera servirá al lector para la facilidad de la lectura del documento.

2.1 Lengua de Señas Mexicano

La Lengua de Señas Mexicana (LSM) es el idioma de las comunidades de sordos, por ello el estudio de esta lengua contribuye a su reconocimiento social y a su revaloración para la enseñanza en la escuela. Una de las características de la lengua de señas, en general, es su forma de transmisión visual, lo cual representa el primer obstáculo para su socialización, si se considera que la LSM es la lengua materna de sólo un 5% de la población de niños sordos (Acosta, 1997). Estas lenguas han surgido en las comunidades de sordos de manera espontánea, son parte de su cultura y reflejo de su visión del mundo; además se encuentran en un proceso productivo constante, como sucede con cualquier otra lengua natural (DIESEME, 2004).

La Lengua de Señas Mexicana tiene algunas reglas particulares que se deben conocer. Por ejemplo: no es de mala educación señalar a la gente. Si una persona se dedica a la enseñanza debe vestirse adecuadamente, es decir, utilizar un color liso que contraste con su piel, para que su interlocutor pueda observar sus manos con claridad. La joyería debe ser discreta, las uñas no deben ir pintadas. Además, no se debe mascar chicle, ya que esto distrae al interlocutor pues en ocasiones los movimientos de los labios corresponden a las palabras que se están interpretando, y un movimiento ajeno distrae y confunde. En el caso de los hombres es recomendable usar chaleco si traen corbata, para evitar que las manos choquen con ella (Serafín de Fleischmann & González Pérez, Manos con Voz, 2011)

La lengua de señas mexicana está compuesta de la **dactilología** y los **ideogramas**.

Dactilología: Lo que bien podría ser el deletreo en la lengua oral y está representada en este diccionario sobre todo con el abecedario, cada palabra se puede representar con la articulación de mano correspondiente de cada letra que la conforma. Por ejemplo *mamá* puede representarse con cada una de sus letras (Figura 2.1).

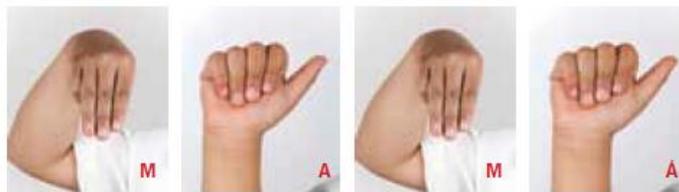


Figura 2.1 Mamá dactilología (Serafín de Fleischmann & González, 2011)

Ideogramas: Representan una palabra con una o varias configuraciones de la mano. De acuerdo con esta realización, *mamá* se articula con la letra *m* de ideograma sobre los labios con la que se golpean varias veces los labios (Figura 2.2).



Figura 2.2 Mamá ideograma (Serafín de Fleischmann & González, 2011)

El alfabeto del LSM (Figura 2.3) que se produce mediante la ejecución de señas usando una mano base y una mano dominante. La mano dominante es la izquierda para las personas zurdas y la derecha para las personas diestras (Priego, Pérez, Fausto, 2012)

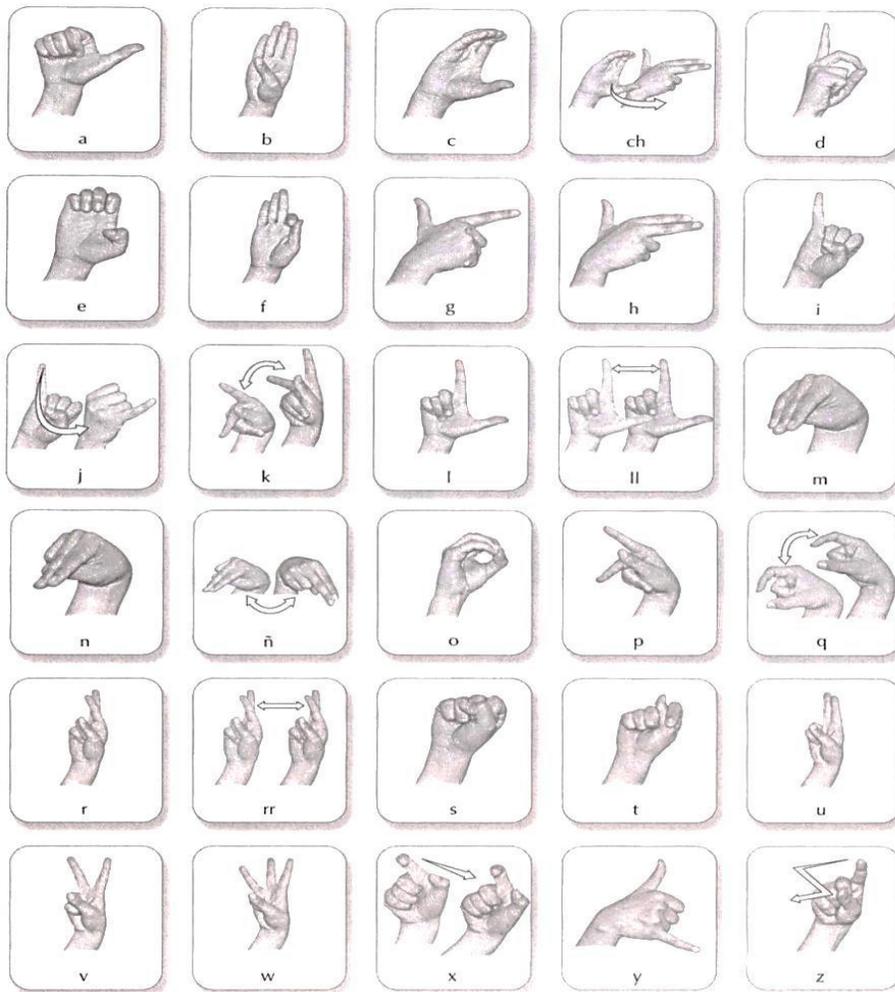


Figura 2.3 Alfabeto del LSM (Priego, Pérez, Fausto, 2012)

2.2 Leap Motion Controller

El *Leap Motion Controller* (LMC), es un dispositivo periférico USB que detecta y lee movimientos dentro de un espacio de interacción en 3D, para su interacción de precisión utiliza un software de control en una computadora. (LeapMotion, 2015). El controlador consiste de 3 luces infrarrojas emitidas y 2 cámaras Infrarrojas (Figura 2.4) (Weichert, Bachmann, Rudak, & Fisseler, 2013).

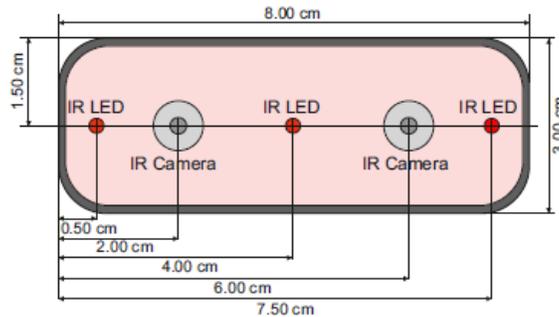


Figura 2.4 Vista esquemática del LMC (Weichert et al., 2013)

El sistema *Leap Motion* reconoce y rastrea las manos, los dedos y las herramientas parecidas a dedos. El dispositivo funciona en una proximidad íntima con alta precisión, velocidad de fotogramas de seguimiento y reporta posiciones discretas, gestos y movimiento. Los sensores están dirigidos a lo largo del eje y el cual va hacia arriba cuando el dispositivo está en su posición de funcionamiento estándar y su campo de visión es aproximadamente de 150° grados. El rango efectivo del *Leap Motion Controller* se extiende desde aproximadamente de 25 a 600 milímetros por encima del dispositivo (1 pulgada a 2 pies) (LeapMotion, 2015).

El sistema *Leap Motion* ocupada un sistema diestro de coordenadas cartesianas (Figura 2.5). El origen se centra en la parte superior del *Leap Motion Controller*. Los ejes X y Z se encuentran en el plano horizontal, el eje X corre en paralelo a lo largo del dispositivo. El eje Y es vertical con los valores positivos crecientes hacia arriba mientras que el eje Z tiene valores positivos crecientes hacia el usuario (LeapMotion, 2015)

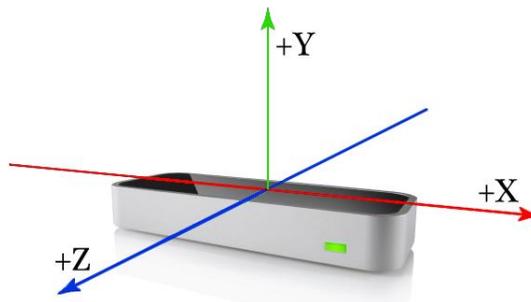


Figura 2.5 Sistema diestro de coordenadas (LeapMotion, 2015)

El *Leap Motion Controller* cuenta con un apartado para los desarrolladores, con este apartado podrán bajar desde la página oficial del *Leap Motion* el *Software Development Kit* (conocido como SDK) que se encuentra disponible para las plataformas de sistemas

operativos Windows, OSX y Linux. De igual manera se encuentra en fase beta la versión SDK de Android, para obtener esta versión del SDK el desarrollador tendrá que llenar una encuesta para que se le sea liberada la versión a su correo.

2.3 Android

Android es un sistema operativo y una plataforma software, está basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, laptops, reproductores de música e incluso para computadores de escritorio. Android permite programar en un entorno de trabajo (*framework*) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos es que cualquier persona que sepa programar puede crear nuevas aplicaciones, *widgets*, o incluso, modificar el propio sistema operativo, dado que Android es de código libre (Báez et al., 2010).

2.4 Microsoft Windows

Microsoft Windows es un sistema operativo, es decir, un conjunto de programas que posibilita la administración de los recursos de una computadora. Este tipo de sistemas empieza a trabajar cuando se enciende el equipo para gestionar el hardware a partir desde los niveles más básicos. Es importante tener en cuenta que los sistemas operativos funcionan tanto en las computadoras como en otros dispositivos electrónicos que usan microprocesadores (teléfonos móviles, reproductores de DVD, etc.). En el caso de Windows, su versión estándar funciona con computadoras, aunque también existe una versión para teléfonos (Windows Mobile). Microsoft domina el mercado de los sistemas operativos con comodidad, ya que Windows está instalado en más del 90% de las computadoras con acceso a Internet en todo el mundo (Definicion.de , 2015)

2.5 TTS (*Text to Speech*)

En la informática existe la denominada síntesis de voz o TTS (*Text-to-Speech*, en inglés), que como bien dicen sus siglas no es más que un sistema de lectura de frases (salida de texto a voz), que pasa un texto escrito, a un formato de audio leído, idóneo para usuarios con problemas de visión, conducción en carretera, lectura de *ebooks* o simplemente cuando no se apetece leer. El sistema, generalmente, incorpora un motor TTS instalado por defecto, con el cual las aplicaciones (o el propio sistema) pueden leer textos en voz alta (Manz, 2015).

2.6 LeapTrainer.js

LeapTrainer.js es un *framework* para el *Leap Motion Controller* el cual es usado para el reconocimiento y aprendizaje de gestos y postura (Figura 2.6). El *framework* soporta codificación de alta y baja resolución de gestos, comparación de plantillas geométricas, correlación cruzada y reconocimiento de gestos y postura a través de las redes neurales.

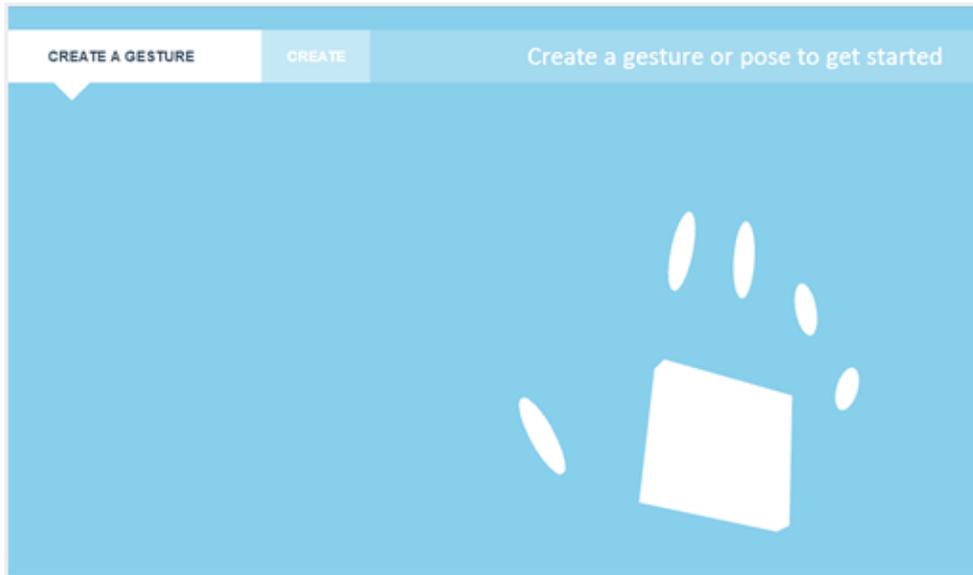


Figura 2.6 LeapTrainer.js (O'Leary, 2016)

Capítulo 3

Estado del arte

En este capítulo se presenta un panorama general del estado del arte en las áreas de investigación que se consideraron importantes para esta investigación. En la **Sección 3.1** presenta los trabajos relacionados y en la **Sección 3.2** el estado del arte.

3.1 Trabajos relacionados

3.1.1 UNI tablet (MotionSavvy, 2015)

Lengua: Lengua de Señas Americano (*American Sign Language* – ASL, en inglés).

Descripción: La empresa *MotionSavvy* desarrolló una carcasa compuesta por dos *Leap Motion Controller* para la detección de las manos para una *Tablet Dell Venue Pro 8* con *Windows 8* (Figura 3.1). La *Tablet* con la ayuda del software desarrollado por la empresa es capaz de leer por el momento alrededor de 300 señas. *UNI Tablet* fue comenzada a ser comercializada en otoño de 2015, realizó una recaudación de fondos en línea *Indiegogo* para comenzar los pre-pedido. El costo para pre-ordenar fue de 499 dólares, aunque se planea que para su venta final tenga un costo de 799 dólares, de igual manera también existirá una suscripción adicional de 20 dólares por mes, que incluyen diccionarios actualizados. Por el momento no está disponible en otras lenguas de señas y no está disponible para otras *tablet*, aunque la empresa está trabajando para ser utilizada en Android, y iOS.

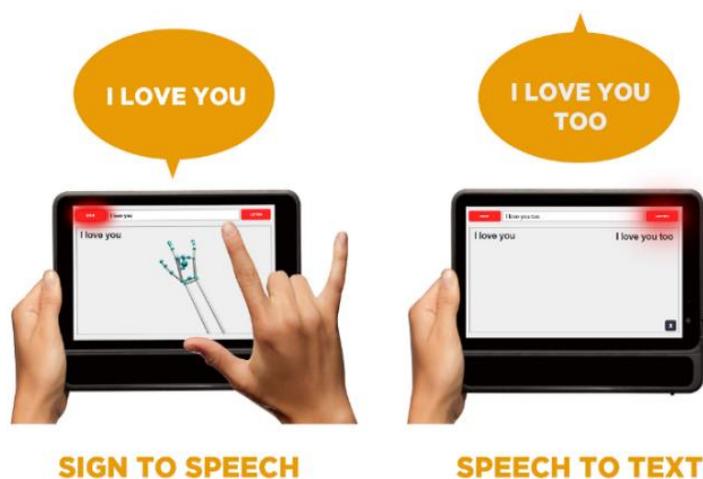


Figura 3.1 Uni Tablet. (MotionSavvy, 2015)

3.1.2 Google Gesture (Google Gesture, 2014)

Lengua: Lengua de Señas Americano (ASL).

Descripción: Estudiantes de la Escuela de Comunicaciones de Berghs, en Suecia se han aliado con Google para desarrollar una tecnología que traduce a voz la lengua de señas de las personas sordomudas utilizando el Smartphone. La persona que se comunica a través de la lengua de señas debe ponerse unas muñequeras electrónicas (Figura 3.2) que recogen el movimiento de los músculos del brazo y leen los impulsos nerviosos mediante electromiografía, la cual esta a su vez con la ayuda del Smartphone transmite en audio lo que se quiere dar a entender a través de las bocinas.



Figura 3.2 Muñequera (Google Gesture, 2014)

3.1.3 Kinect Sign Language Translator (KSLT, 2014)

Lengua: Lengua de Señas Chino y Lengua de Señas Americano (ASL).

Descripción: El traductor utiliza una computadora y una cámara del Kinect que esta a su vez podrá detectar los movimientos de las manos y se podrá mostrar a la otra persona; de igual manera si un usuario que no sea sordo, este puede hablar o escribir palabras en el traductor de Kinect y mediante una persona virtual se mostrara las señas (Figura 3.3). Hasta el momento se tienen identificada 300 señas de la lengua de señas chinas de un total de 4000.



Figura 3.3 Kinect Sign (KSLT, 2014)

3.1.4 *Enable Talk* (Talk Enable, 2015)

Lengua: Lengua de Señas Americano (ASL).

Descripción: El invento consiste en unos guantes (Figura 3.4) utilizados por el usuario encargado de comunicar con las manos. Estos han sido dotados en cada mano de 14 sensores, un micro controlador y un trasmisor Bluetooth. Los sensores se encargan de identificar los movimientos para luego mandarlos a la aplicación móvil, que a su vez se encarga de convertir las señales digitales en voz. Haciendo posible mantener una conversación solo utilizando las manos. El primer prototipo cuesta alrededor de 150 dólares en ser fabricado pero con algo de financiación y desarrollo se podría bajar esa cifra a 75 dólares.



Figura 3.4 Guante (Talk Enable, 2015)

3.1.5 *ONIS* (RedUSERS, 2013)

Lengua: Lengua de Señas Argentino (LSA).

Descripción: Es un software para el sistema operativo Windows que utiliza la cámara de Kinect (una consola de videojuego) para captar las señas y llevarlas a un texto escrito, mientras que también puede hacerlo de modo inverso: al escribirse el texto se representan las señas mediante imágenes (Figura 3.5). Para favorecer la comunicación entre personas sordas con personas que no manejen la lengua de señas. El software se programado en Microsoft Visual Studio 2010 C#.



Figura 3.5 ONIS (RedUSERS, 2013)

3.1.6 *Portable Sign Language Translator (PSLT, 2014)*

Lengua: Lengua de Señas Británico (BSL).

Descripción: El *Portable Sign Language Translator* es un traductor de la lengua de señas el cual es portátil, reconoce y traduce la lengua de señas en texto, cuenta una cámara y un software para el reconocimiento de imágenes (Figura 3.6). El PSLT reconoce a través de la captura de la cámara y lo transforma al lenguaje natural el evalúa las probabilidades de contexto y uso de palabras.



Figura 3.6 PSLT (PSLT, 2014)

3.1.7 Descripción de los campos de trabajos relacionados

En la Tabla 3.1 se muestra una comparativa de los trabajos relacionados y la tesis planteada. A continuación se describen los criterios que se utilizaron para realizar la comparativa.

- **Nombre del trabajo:** Nombre del dispositivo creado o por crear.
- **Lengua de señas:** Para que lengua de señas es creado.
- **Sistema operativo:** El sistema operativo en que la aplicación es ejecutada.
- **Hardware:** Dispositivos usados para la detección y reconocimiento de la lengua de señas.
- **Portátil:** Indica si el dispositivo es posible de llevarlo a todos lados de una manera cómoda.
- **Audio:** Indica si la aplicación proporciona de forma audible la seña detectada.
- **Texto:** Indica si la aplicación proporciona de forma escrita la seña detectada.

3.1.8 Tabla comparativa de trabajos relacionados

Tabla 3.1 Tabla comparativa de trabajos relacionados

Nombre del trabajo	Lengua de señas	Sistema operativo	Hardware	Portátil	Audio	Texto
Uni Tablet	Americano	Windows	Leap Motion	*	*	*
Google Gesture	Americano	Android	Muñequera Electrónica	*	*	*
Kinect Sign Language Translator	Americano, Chino	Windows	Kinect		*	*
Enable Talk	Americano	Android	Sensores, Arduino, Transmisor Bluetooth	*	*	
Onis	Argentino	Windows	Kinect			*
PSLT (Portable Sign Language Translator)	Británico	Windows	Cámara			*

3.2 Estado del Arte

3.2.1 Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado (Chiguano, Rodríguez & Moreno, Díaz, 2011).

Descripción: Este sistema interpreta la lengua de señas realizado con las manos y después las convierte a texto y voz; también cuenta con un entrenador el cual va mostrando en lengua de señas para que uno aprenda, de esta manera uno se puede comunicar con las demás personas de manera exterior y viceversa. La desarrollaron en diferentes etapas. Etapa de Adquisición, Procesamiento digital de imágenes, *Vision Assistant* de Labview.

Al ser en un ambiente controlado el sistema fue realizado en tres fases, primero se escogió con qué fondo se podría ver mejor; usaron un fomix negro mate ya que este no genera brillo en su superficie. La segunda fase fue el de la iluminación, para esta fase usaron 3 tipos de focos incandescente, leds y fluorescente, los 3 focos fueron probados atrás de la cámara y la que mejor se adaptó a las necesidad fue el fluorescente ya que esta no generaba mucho calor como la incandescente y daba la luz necesaria. La última fase fue la elegir la cámara correcta, al realizarlo en Labview el procesamiento de imágenes se tenía que usar una cámara con entrada USB ya que dicho programa cuenta con una librería para leer cámaras en USB.

Para la detección de las señas de las manos utilizan una función llamada "*Lineprofile.vi*", dicha función proporciona valores de pixeles dentro del área de trabajo y con los valores obtenidos se calcula la media y muestra una gráfica en donde se observa la distribución de estos pixeles, el análisis de la imagen se realiza en el espacio de color de RGB 32 bits. Luego utilizan otra función llamada "*ExtractSingleColorPlane.vi*", con ella pasan la imagen del RGB a una escala a de grises para luego calibrar el contraste y brillo y así tener una imagen más clara. Otra función llamada "*IMAQ AutoBthreshold2*" se utiliza en la binarización de la imagen para hacerla resaltar más.

Ya tenidas las imágenes las guardaron en bases de datos, tuvieron que realizar 6 muestras por cada letra del alfabeto, usando "*Vision Assistant*" y usaron los método de clasificación k-vecino más cercano y distancia mínima para identificar la letra a partir de las 6 muestras de 1 letra ya que existen algunas pequeña diferencias entre una mano y otra. En los resultados obtenidos de la detección de las 24 letras del alfabeto fue la siguiente. Se generaron 9 bases de datos con los métodos mencionados con anterioridad, el margen de error fue del 17.85% y de los 3 métodos desarrollados se eligió el de k-vecino más cercano debido a que este método obtuvo un mayor porcentaje de acertamiento de las señas detectadas.

3.2.2 Reconocimiento de imágenes del lenguaje de señas mexicano (Priego, Pérez, Fausto, 2012).

Descripción: El Sistema que desarrollaron permite generar las 25 palabras del alfabeto de la Lengua de Señas Mexicana utilizando el dispositivo Kinect, este sistema cuenta con 2 módulos principales los cuales son: aprendizaje y reconocimiento.

El reconocimiento se da cuando el Kinect toma la captura de la imagen y esta la compara con las que se encuentran en el base de datos, mientras que la de aprendizaje es cuando detecta un nuevo patrón, es decir que exista por lo menos un 90% de similitud entre la imagen almacenada en la base datos y la tomada por el Kinect.

La cámara infrarroja del dispositivo genera un umbral el cual toma 2 parámetros, la distancia frontal y la distancia del fondo y así realizar la segmentación de la imagen, si ningún objeto se encuentra en la imagen no se realizan cálculos. Utilizan matrices evolutivas para almacenar los patrones y hacer posible la identificación mediante el uso de la computadora, los procesos de la matriz evolutiva son: Proceso de aprendizaje – reconocimiento con y sin reforzamiento, proceso de reforzamiento y proceso de olvido.

En el proceso de aprendizaje sin reforzamiento no incluye los factores de coincidencias en sus elementos y en el proceso de reconocimiento los vectores de entrada con los patrones almacenados en la base de datos se comparan a través de un criterio de semejanza, se calcula obteniendo la distancia utilizada, porcentaje de semejanza, factor de semejanza y un criterio de decisión para detectar la semejanza; por ejemplo si una imagen no satisface el umbral es almacenada en la matriz evolutiva, se aprende el nuevo patrón y se agrega a la base de conocimientos, aquí se podrían tener varios patrones con el mismo significado.

En el proceso de aprendizaje con reforzamiento para el reconocimiento del vector de entrada se tiene que comparar con alguna que ya se encuentre almacenada en la matriz evolutiva, se utiliza el mismo proceso de distancia utilizada, factor de semejanza y el criterio de decisión. Cuando se tiene pequeñas variaciones de un mismo patrón se utiliza un proceso llamado reforzamiento, para realizar este proceso se hace sumando los elementos de la clase de matriz almacenada con el valor del elemento correspondiente a la imagen entrada e incrementando en una unidad el valor de incidencia.

La parte del olvido lo que hace es eliminar aquellos patrones los cuales se consideren irrelevantes, que son los que tiene resultados negativos y solo deja los que si son relevantes. Para facilitar el proceso de reconocimiento lo que proponen es que se personalice la base de conocimientos de acuerdo a cada usuario que la utilizara, para que así se elimine la ambigüedad entre patrones de diferentes personas.

3.2.3 *Transforming Indian Sign Language into Text Using Leap Motion* (P.Karthick, N.Prathiba, V.B.Rekha, & S.Thanalaxmi, 2014).

Descripción: El sistema implementa un algoritmo llamado DTW (*Dynamic Time Warping*) en combinación del otro algoritmo IS (*Intelligence Sense*) para convertir los gestos de la mano en un texto apropiado, ayudado por el dispositivo de Leap Motion que consta de un cámara integrada y 2 sensor de IR (*InfraredRays*) para capturar las señales de la mano. La *Neuro Linguistic Programming* (NLP), es una división de Inteligencia Artificial que incluye el Procesamiento del Lenguaje Natural y Redes Neuronales. La IS invoca un disparador cuando el entorno actual cambia dinámicamente, el DTW maneja transformación gesto mapeado con patrones similares.

El gesto reconocido a través de visualizador lo capturaran junto con el texto, calculado con el algoritmo DTW el cual encuentra una coincidencia óptima entre dos series de tiempo. En las dos series de tiempo los datos se deforman de manera no lineal de tal manera que las regiones similares están alineadas, obteniendo así una distancia mínima entre ellos y le añaden a la base de conocimiento que contiene el cómputo métrico de cada gesto. Cuando detectan una mano, dedo, herramienta, o un gesto, el Leap Motion le asigna un ID único. El ID sigue siendo el mismo, siempre y cuando esa entidad permanece visible dentro del campo de campo del dispositivo. Si el seguimiento se ha perdido y recuperado, el Leap Motion puede asignar un nuevo ID (el software puede no saber que la mano o el dedo es el mismo que el anterior visible), de igual manera el sistema descarta un gesto inapropiado que no coincide con la base de conocimientos.

3.2.4 *Arabic Sign Language Recognition using the Leap Motion Controller* (Mohandes, Aliyu, & Deriche, 2014).

Descripción: Lo que proponen es un enfoque para el reconocimiento de la Lengua de Señas Árabe, el cual fue desarrollado en Java con el IDE de NetBeans. Usan un LMC (*Leap Motion Controller*) para la detección de las manos, dedos y de esta manera el LMC les proporcionará los datos y la información de los movimientos.

En la adquisición de los datos, el sistema incluye un estado de pre-procesamiento, un estado de extracción y un estado de clasificación. Utilizaron 2 clasificadores para el desarrollo de este sistema, el primero fue MLP (*Multiplayer Perceptron Neural networks*) mientras que el segundo es NBC (*Nave Bayes Classifier*). Con la ayuda del LMC extraen la información de las manos y este mismo lo etiqueta con un ID único, en dado caso que no se detecte la seña el LMC le asignará un nuevo ID.

Para el desarrollo del sistema se tomaron 10 muestras por cada letra y cada muestra tienen 10 cuadros de datos, lo cual implica un total de 100 cuadros de datos por cada letra, sabiendo que el alfabeto tiene 28 letras; en su sistema ello tiene 2800 cuadros

de datos, toda esa información fue recolecta e importada a MATLAB para el análisis y procesamiento.

Con el uso del MLP y con la ayuda de las redes neuronales artificiales se van obteniendo los pesos de las neuronas y se ajustan durante el proceso de entrenamiento para así lograr la relación de entrada y salida, todo eso fue ayudado con el algoritmo de *back-propagation*. En los resultados que obtuvieron fueron que de los 2800 muestras y la precisión en el reconocimiento de las señas de los clasificadores fue la siguiente: para el NBC fue de 98.3% y el MLP fue de 99.1%.

3.2.5 *Towards the Development of a Mexican Speech-to-Sign-Language Translator for the Deaf Community* (Trujillo, Romero & Caballero, Morales, 2012).

Descripción: Lo que en este artículo muestran es una aplicación hacia el desarrollo de un traductor de voz a lengua de señas del español Mexicano para poder asistir a personas sin discapacidad a interactuar con personas sordas.

El ASR (*Mexican Automatic Speech Recogniser*) es el módulo principal del traductor y de igual manera cuenta con el módulo de *Text Interpreter* y *MSL Database*. Para tener un buen rendimiento del ASR fue necesario el uso de un corpus, el que usaron fue el *Mexican Spanish Corpus DIMEx100* el cual fue desarrollado por Instituto de Investigaciones en Matemáticas Aplicadas y Sistema IIMAS de la UNAM.

Este corpus tiene por objetivo hacer posible la construcción de modelos acústicos y diccionarios de pronunciación para la creación de sistemas computacionales para el reconocimiento del español hablado en México. Seis personas 3 hombres, 3 mujeres de diferentes edades y lugares son los que fueron elegidos para realizar la prueba del traductor. La definición de los fonemas del idioma Español Mexicano se obtuvo con la herramienta *TranscribeMex* que fue desarrollada para etiquetar fonéticamente el corpus DIMEx.

Las muestras deben ser etiquetadas en los niveles ortográficos y fonéticos para realizar entrenamiento supervisado de los modelos acústicos del sistema ASR. El etiquetado ortográfico se realizó manualmente con el software *Wavesurfer*. El algoritmo que usaron fue el HMM (*Hidden Markov Models*) y la implementación de una librería HTK. Para el entrenamiento supervisado, el corpus del habla lo codificaron dentro del *Mel Frequency Cepstral Coefficients* (MFCC). El algoritmo *Viterbi* fue el encargado de encontrar la secuencia de palabras que mejor responde a la señal de voz.

En el módulo de *Text Interpreter* y *MSL Database*, si la palabra es encontrada el intérprete procesa en la pantalla la secuencia de movimiento de MSL asociada con las palabras y si en dado caso no se encontrara en la base de datos, la palabra es “deletreada” y descrita con la representación del LSM asociada para cada letra que forman la palabra.

El vocabulario está formado por 25 palabras (Hola, Adiós, Hoy, Ayer, Mañana, Noche, Alegre, Feliz, Triste, Temor, Enojo, Mamá, Papá, Hijo, Niño, Hermano, Blanco, Rojo, Azul, Casa, Silla, Mesa, Cama, Habitación, Gracias) y 23 letras del alfabeto (menos las letras J, K, Z) y se obtuvo una precisión del 97.2% en una sesión de 400 palabras usadas en total.

3.2.6 Descripción de los campos de trabajos relacionados

En la Tabla 3.2 se resume el estado del arte.

A continuación se describen los criterios que se utilizaron para realizar la comparativa.

- **Nombre del trabajo:** Nombre del trabajo o publicación.
- **Año:** Año de publicación.
- **Lengua de Señas:** Para que lengua de señas fue creado.
- **Hardware:** Dispositivos utilizados para realizar la aplicación.
- **Técnicas o Métodos:** Técnicas o métodos utilizados para solucionar la problemática.
- **Resultados:** Resultados obtenidos.
- **Numero de señas:** Cantidad de señas que son reconocidas.
- **Precisión:** Precisión con la que cuenta al reconocer las señas.

3.2.7 Tabla comparativa del estado del arte

Tabla 3.2 Tabla comparativa del estado del arte

Nombre del trabajo	Año	Lengua de Señas	Hardware	Técnicas o Métodos	Resultado	Número de señas	Precisión
Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado.	2011	Ecuatoriano	Cámara, Foco, PC Escritorio	K vecino más cercano, Distancia mínima, Librerías de LabView	Traduce texto y voz	24	88.15%
Reconocimiento de imágenes del lenguaje de señas mexicano.	2012	Mexicana	Kinect, PC Escritorio	Matrices evolutivas (proceso de aprendizaje, reforzamiento y olvido), redes neuronales	Reconocimiento de las señas	25	90%
Transforming Indian Sign Language into Text Using Leap Motion.	2014	India	Leap Motion, PC Escritorio	Alineamiento Temporal Dinámico (<i>Dynamic Time Warping DTW</i>), Sentido de Inteligencia (<i>Intelligence Sense IS</i>), Base de Conocimiento	Traduce a texto	-	-
Arabic Sign Language Recognition using the Leap Motion Controller.	2014	Árabe	Leap Motion, PC Escritorio	Perceptron Multicapa (<i>Multiplayer Perceptron MLP</i>), (<i>Nave Bayes Classifier NBC</i>)	Reconocimiento de las señas	28	NBC 98.3% MLP 99.1%
Towards the Development of a Mexican Speech-to-Sign-Language Translator for the Deaf Community.	2012	Mexicana	PC Escritorio	Corpus DIMEx100, Reconocimiento del Habla Automático (<i>Automatic Speech Recogniser ASR</i>), Modelo Oculto de Márkov (<i>Hidden Markov Models HMM</i>), Base de datos MSL	Convierte a lengua de señas lo que escucha	48	97.2%



Capítulo 4.

Metodología de solución

En este capítulo se presenta la metodología de solución propuesta, el desarrollo de cada una de las actividades y la implementación de cada fase de la metodología para así resolver la problemática planteada en este trabajo de investigación.

4.1 Estudio y análisis de los algoritmos para el reconocimiento de patrones aplicados a la LSM

4.1.1 Modelo Oculto de Markov

Un Modelo Oculto de Markov o HMM (por sus siglas en inglés, *Hidden Markov Model*) es un proceso estocástico que consta de un proceso de Markov no observado (oculto) $q = \{q_t\} t \in N$ y un proceso observado $O = \{o_t\} t \in N$ cuyos estados son dependientes estocásticamente de los estados ocultos, es decir, es un proceso bivariado (q, O) . Los HMMs se pueden considerar también como sistemas generativos estocásticos, los cuales se emplean en la modelación de series de tiempo.

Formalmente, una cadena de Markov se define como (Q, A) , donde $Q = \{1, 2, \dots, N\}$ son los posibles estados de la cadena y $A = (a_{ij})_{n \times n}$ es una matriz de transición de estados en el modelo. Si $A(t) = (a_{ij})_{n \times n}$ es independiente del tiempo entonces el proceso se llama homogéneo y las probabilidades de transición de estados son de la forma $a_{ij}(t) = P[q_t = j | q_{t-1} = i]$ con las siguientes propiedades:

- I. $0 \leq a_{ij} \leq 1, \quad 1 \leq i, j \leq N$
- II. $\sum_j a_{ij} = 1, \quad 1 \leq i \leq N$

La condición fundamental de que sea una cadena de Markov establece que las probabilidades de transición y emisión dependen solamente del estado actual y no del pasado, esto es, $P[q_t = j | q_{t-1} = i, q_{t-2} = k, \dots] = P[q_t = j | q_{t-1} = i] = a_{ij}(t)$.

4.1.1.1 Definición de Modelos Ocultos de Markov

Un modelo oculto de Markov es una cadena de q junto con un proceso estocástico que toma valores en un alfabeto Σ y el cual depende de q .

Estos sistemas evolucionan en el tiempo pasando aleatoriamente de estado a estado y emitiendo en cada momento al azar algún símbolo del alfabeto Σ . Cuando se encuentra en el estado $q_{t-1} = i$, tiene la probabilidad a_{ij} de moverse al estado $q_t = j$ en el siguiente instante y la probabilidad $b_j(k)$ de emitir el símbolo $o_t = v_k$ en el tiempo t .

Solamente los símbolos emitidos por el proceso q son observables, pero no la ruta o secuencia de estados q , de ahí el calificativo de "oculto" de Markov, ya que el proceso de Markov q es no observado.

4.1.1.2 Arquitecturas de Modelos Ocultos de Markov

Un HMM puede ser representado como un grafo dirigido de transiciones/emisiones. La arquitectura específica que permita modelar de la mejor forma posible las propiedades observadas depende en gran medida de las características del problema. Las arquitecturas más usadas son:

- a) Ergódicas o completamente conectadas en las cuales cada estado del modelo puede ser alcanzado desde cualquier otro estado en un número finito de pasos
- b) Izquierda-derecha, hacia adelante o *Bakis* las cuales tienen la propiedad de que en la medida que el tiempo crece se avanza en la secuencia de observación asociada \mathbf{O} , y en esa misma medida el índice que señala el estado del modelo permanece o crece, es decir, los estados del sistema van de izquierda a derecha
- c) Izquierda-derecha paralelas, son dos arquitecturas izquierda-derecha conectada entre sí.

4.1.1.3 Definición de los elementos de un Modelos Ocultos de Markov

Formalmente, un HMM discreto de primer orden se define como una cinco-tupla

$$(4.1) \quad \lambda = (\Sigma, Q, A, B, \pi)$$

donde

- I. $\Sigma = \{v_1, v_2, \dots, v_M\}$ es un alfabeto o conjunto discreto finito de M símbolos.
- II. $Q = \{1, 2, \dots, N\}$ es un conjunto finito de N estados.
- III. $A = (a_{ij})_{N \times N}$ es una matriz de probabilidades de transición donde a_{ij} es la probabilidad de transición desde el estado i al estado j , para todo $i, j \in N$.
- IV. $B = (b_j(o_t))_{N \times M}$ es un vector de probabilidades de emisión de símbolos, uno por cada estado, donde $b_j = (b_{j1}, b_{j2}, \dots, b_{jM})$ es la probabilidad de emisión del símbolo v_k del alfabeto en el estado j .
- V. $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ es un vector de probabilidades del estado inicial q_0 en Q .

4.1.1.4 Problemas básicos de los Modelos Ocultos de Markov

Existen tres problemas básicos relacionados con los HMMs:

1. Calcular eficientemente $P(\mathbf{O}|\lambda)$ la probabilidad de la secuencia de observación \mathbf{O} dado el modelo $\lambda = (A, B, \pi)$ y la secuencia de observación $\mathbf{O} = (o_1 o_2 \dots o_T)$.
2. Encontrar la trayectoria más probable $q = (q_1 q_2 \dots q_T)$ dado el modelo λ y la secuencia de observación $\mathbf{O} = (o_1 o_2 \dots o_T)$, es decir, $q = \arg r_{r \in Q^*} \{ \max_{r \in Q^*} P(r) \}$.
3. Ajustar los parámetros A, B, π para maximizar $P(\mathbf{O} | \lambda)$

4.1.2 Máquina de Soporte Vectorial

Las Máquinas de Soporte Vectorial o SVM (por sus siglas en inglés *Support Vector Machine*) surgen como un método de clasificación basado en la teoría de Vladimir Vapnik y de su equipo de AT&T de minimización de riesgo estructural. Teniendo un conjunto de datos de muestra o ejemplos de entrenamiento etiquetamos sus clases y entrenamos una Máquina SVM construyendo un modelo que será capaz de predecir la clase de los nuevos datos que le introduzcamos. Intuitivamente la SVM representa en un eje de coordenadas los vectores de entrenamiento, separando las clases presentes en los ejemplos por un espacio lo mayor posible, cuando introducimos nuevos datos se colocan sobre el mismo eje y en función de su proximidad a uno de los grupos antes separados son clasificados en una u otra clase.

El primer paso, antes de poder clasificar, es realizar una etapa de aprendizaje. Consiste en encontrar el hiperplano (frontera de decisión) $h(x) = 0$ que mejor separe un conjunto de datos $X \in \mathcal{R}^d$ según la clase $Y \in \{-1, 1\}$ a la que pertenecen. Dicho hiperplano es el que maximiza la distancia al punto más próximo de cada clase, por lo tanto, estará a la misma distancia de los ejemplos más cercanos de cada categoría.

Según Vapnik, el separador lineal que maximiza el margen (el doble de la distancia al punto más próximo en cada clase) es el que da la mayor capacidad de distinguir características comunes de los datos de cada clase que permitan clasificar datos que no sean los del conjunto de entrenamiento. Para hallar dicho separador, se resuelve un problema de optimización empleando técnicas de programación cuadrática.

A los datos empleados para hallar el hiperplano, se les conoce como vectores de aprendizaje o de entrenamiento, al igual que en otros algoritmos de aprendizaje supervisado. Estos vectores son los que permiten crear los modelos con los que trabaja la SVM para clasificar los nuevos datos que le introduzcamos. A partir de unos datos de entrada x_i , las SVM nos proporcionarán su clase según la regla de clasificación $f(x_i) = \text{signo}(h(x_i))$.

En la Figura 4.1 se representan datos de dos clases (cuadrados azules y círculos verdes) separados por el hiperplano que maximiza la distancia entre ellos. Esta distancia es la marcada como margen, que es máxima para el hiperplano obtenido en este caso, cualquier otro hiperplano presentaría un margen de separación de clases menor y por lo tanto sería menos adecuado.

Tras el aprendizaje se comprueba el error cometido tomando una nueva muestra de datos (conjunto de test o validación) y comparando la salida que obtenemos con su clase real. De cada muestra de datos se suele tomar un 75% como vectores de aprendizaje y el 25% de los vectores restantes se utilizan para poder contrastar la fiabilidad de la frontera de decisión obtenida. La calibración del modelo es fundamental para que dichos modelos sean sólidos y recreen con fiabilidad la situación deseada. En caso de que no consigamos el resultado deseado se deberán aumentar los datos de entrenamiento hasta que obtengamos una clasificación correcta

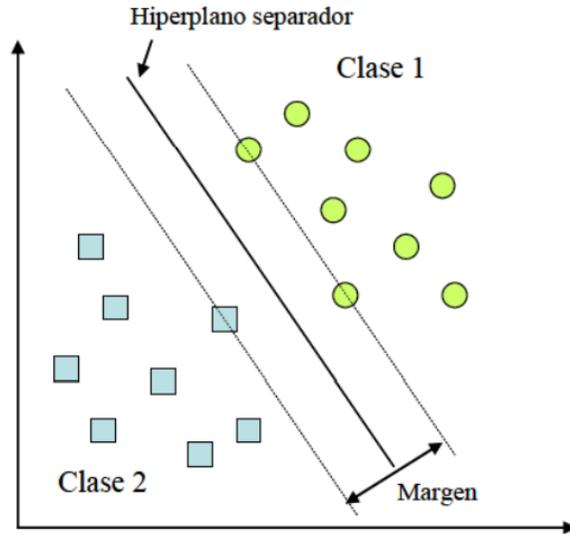


Figura 4.1 Ejemplo de una SVM

La situación más cómoda para resolver con SVM en cuanto al tipo de datos que tengamos, es que dichos datos sean linealmente separables pero las entradas es muy posible que sean no separables linealmente o que exista un cierto nivel de ruido en las medidas que las distorsione.

4.1.2.1 SVM lineal con margen máximo

Solo se deberían emplear cuando los datos son separables linealmente, es decir, se puede usar como frontera de decisión un hiperplano $h(x_i)$ tal que:

$$(4.2) \quad h(x) = w^T x + b = 0$$

donde w y $x \in \mathbb{R}^d$, siendo d la dimensión del espacio de entrada.

La resolución para dicho caso sería suponer que se tiene un conjunto de n datos separables linealmente $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ donde $x_i \in \mathbb{R}^d$ e $y_i \in \{-1, 1\}$.

Se cumplirá, según el lado en el que se encuentren respecto del hiperplano:

$$(4.3) \quad \omega^T x_i + b > 0, \text{ para } y_i = 1, i = 1, \dots, n$$

$$(4.4) \quad \omega^T x_i + b < 0, \text{ para } y_i = -1, i = 1, \dots, n$$

Las ecuaciones definen las dos clases presentes en nuestro problema que al no estar mezcladas, esto es, son linealmente separables, nos permiten hallar con sencillez a nivel matemático el hiperplano que las separa con margen máximo.

Las expresiones antes halladas las podemos reducir a una sola:

$$(4.5) \quad y_i(\omega^T x_i + b) > 0, \text{ para } i = 1, \dots, n$$

Para resolver el problema, se considera que los vectores soporte (los puntos más cercanos al hiperplano), cumplen:

$$(4.6) \quad h(x_i) = 1, \text{ para } y_i = 1$$

$$(4.7) \quad h(x_i) = -1, \text{ para } y_i = -1$$

En la Figura 4.2 los vectores soporte están representados con relleno en color negro. Al ser una aproximación sencilla los vectores soporte son fácilmente identificables a nivel gráfico.

No puede haber datos del conjunto de aprendizaje dentro del margen, por definición del método, por lo que la ecuación $y_i(w^T x_i + b) > 0$ queda:

$$(4.8) \quad y_i(w^T x_i + b) \geq 1, i = 1, \dots, n$$

La distancia $dist(h, x)$ de un punto al hiperplano es:

$$(4.9) \quad dist(h, x) = \frac{|h(x)|}{\|w\|}$$

Como los puntos más próximos al hiperplano cumplen $|h(x)| = 1$, su distancia al hiperplano sería:

$$(4.10) \quad dist(h, x) = \frac{1}{\|w\|}$$

Para encontrar los valores de ω y b hay que resolver un problema de optimización que consiste en maximizar la distancia $dist(h, x)$ entre el hiperplano y el punto de entrenamiento más próximo:

Maximizar:

$$(4.11) \quad \frac{1}{\|w\|}$$

Sujeto a:

$$(4.12) \quad y_i(w^T x_i + b) \geq 1, i = 1, \dots, n$$

Que es la condición de que ningún vector de entrenamiento quede dentro del margen que separa a las dos clases.

En la Figura 4.2 queda planteado el problema con sus elementos característicos: las dos clases (idénticas a las de la Figura 4.1), el hiperplano a hallar y el margen máximo.

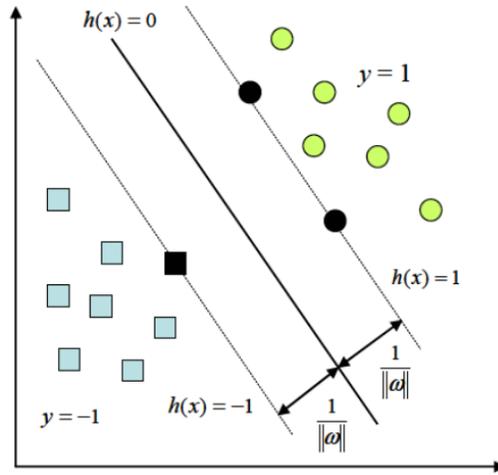


Figura 4.2 SVM con margen máximo (en negro representados los vectores soporte)

4.1.3 Multicapa de Red Perceptrón

En 1958 Rosenblatt publica sus primeros trabajos acerca de su modelo neuronal denominado *perceptrón* junto a su algoritmo de convergencia. Este modelo, como veremos, está formado por tres elementos o niveles de organización: las unidades sensoriales, las unidades asociativas y las unidades de respuesta. Las conexiones entre las dos primeras se definen en forma de pesos fijos y entre las dos últimas mediante pesos variables. Las unidades asociativas actúan como pequeños preprocesadores diseñados para extraer información a partir de los ejemplos presentados.

En la Figura 4.3 se ilustra un *perceptrón* monocapa o simplemente *perceptrón* con una única neurona, estando por ello limitado su uso a la clasificación de patrones en dos clases o categorías. Si expandimos esta capa de salida incluyendo más de una neurona en ella podremos clasificar correctamente más de dos clases con la limitación demostrada por Minsky y Papert (1969) de que estas clases deben ser separables linealmente.

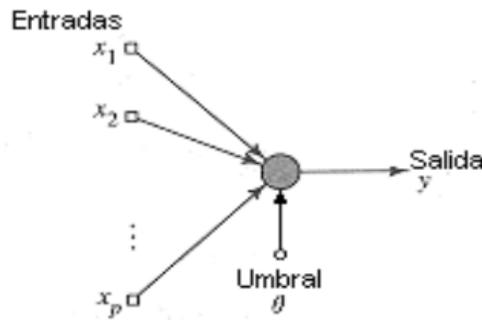


Figura 4.3 Perceptrón de Rosenblatt.

4.1.3.1 Modelos de una neurona

Una neurona es una unidad de proceso de la información, constituyente fundamental de las redes neuronales que puede seguir varios modelos matemáticos, pero que deberá contar esencialmente con las siguientes partes:

1. Una *colección de sinapsis o conexiones pesadas*. Una señal x_j a la entrada de la sinapsis j conectada a la neurona k será multiplicada por el peso w_{kj} . Este peso será positivo si la sinapsis asociada es *excitadora*, y negativo si la sinapsis es *inhibitoria*.
2. Un *sumador* para las señales de entrada pesadas.
3. Una *función de activación*. Su función es la de limitar la amplitud de la salida de la neurona.
4. Una entrada fija que multiplica a un peso adicional llamado *umbral* θ_k y que tiene el efecto de rebajar la entrada de la red al rango de la función de activación.

En la Figura 4.4 vemos el modelo típico de una neurona propuesto por McCullochPitts y formado por los bloques descritos anteriormente.

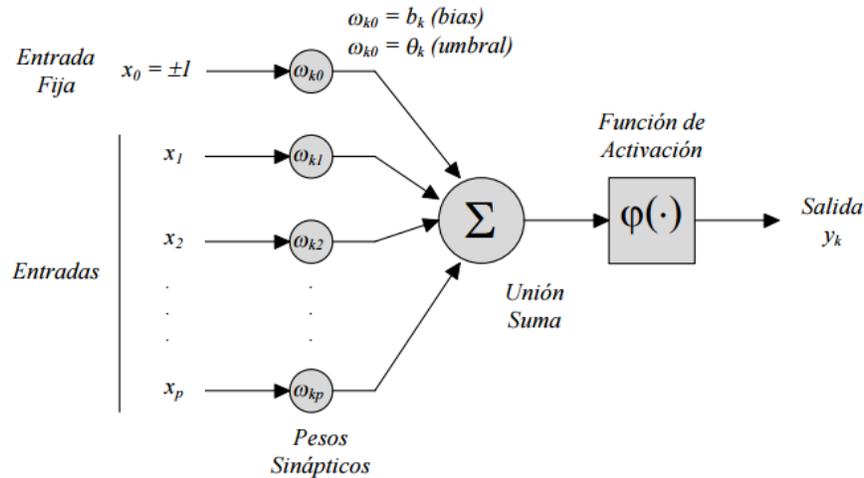


Figura 4.4 Modelo no lineal de una neurona

A partir de las definiciones anteriores podemos describir matemáticamente una neurona k como sigue:

$$(4.13) \quad u_k = \sum_{j=1}^p w_{kj} x_j$$

$$(4.14) \quad v_k = \varphi(u_k - \theta_k)$$

Redefiniendo $v_k = u_k - \theta_k$, obtenemos $y_k = \varphi(v_k)$ y si además, añadimos una nueva sinapsis, cuya entrada sea de valor fijo ± 1 y cuyo peso sea $w_{k0} = \theta_k$, obtendremos los dos modelos de neurona no lineal representados en la figura anterior.

4.1.3.2 Algoritmo de back-propagation

La idea central de esta regla o algoritmo se encuentra en calcular los errores para las unidades de las capas ocultas a partir de los errores de las unidades de la capa de salida siendo propagados capa tras capa hacia la entrada.

La señal de error en la neurona de salida j en la iteración n (presentación del n –ésimo patrón de entrenamiento) viene definida así:

$$(4.15) \quad e_j(n) = d_j(n) - y_j(n)$$

donde j es un nodo de salida, la suma de los errores cuadráticos de la red se puede definir como:

$$(4.16) \quad E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

donde el conjunto C incluye todos los nodos en la capa de salida de la red. La Figura 4.5 muestra la neurona j alimentada por un conjunto de señales procedentes de una capa de neuronas a su izquierda.

El número de patrones o ejemplos forman el conjunto de entrenamiento o *Training set* y lo denotamos con N . Así, el Error Cuadrático Medio (E_{av}) es obtenido sumando $E(n)$ para todas las iteraciones y normalizando respecto N :

$$(4.17) \quad E_{av} = \frac{1}{N} \sum_{n=1}^N E(n)$$

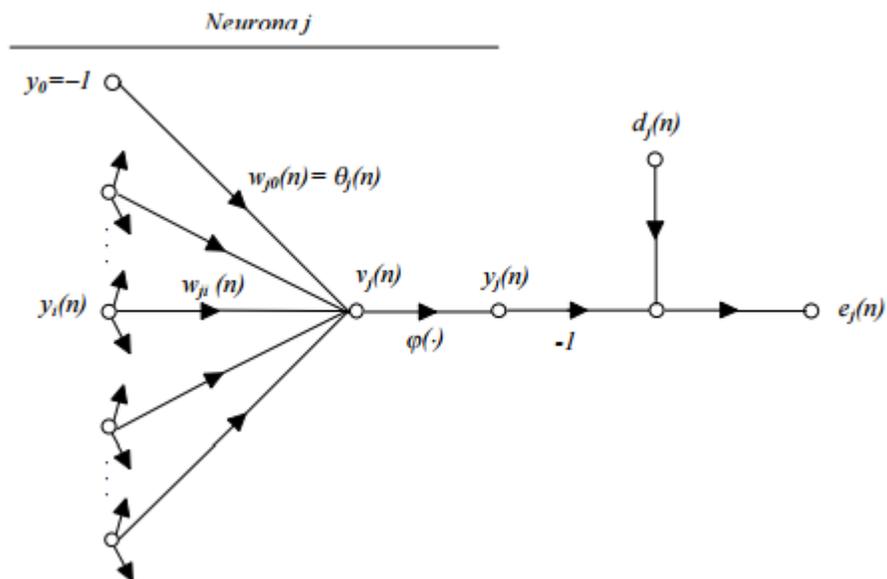


Figura 4.5 Gráfico de señales en una neurona de salida

Para un determinado conjunto de entrenamiento, E_{av} representa una medida del aprendizaje de la RNA denominada *Función Costo*. El objetivo de este proceso de aprendizaje es minimizar esta función de error ajustando los parámetros libres que tenemos (pesos, constante de adaptación).

4.1.4 Alineamiento Temporal Dinámico

El algoritmo DTW, en español llamado “Alineamiento Temporal Dinámico”, sirve para medir la similitud entre dos secuencias temporales que pueden variar en el tiempo o la velocidad. Originalmente se usó para el reconocimiento de voz, pero también se ha aplicado a otras áreas como las secuencias temporales de vídeo o datos gráficos. De hecho, cualquier dato que se puede convertir en una secuencia lineal puede ser analizado mediante el DTW.

En general, este algoritmo calcula una coincidencia óptima entre dos secuencias dadas. Estas secuencias no tienen por qué tener la misma longitud.

$$(4.18) \quad A = a_1, a_2, a_3, \dots, a_i, a_n$$

$$(4.19) \quad B = b_1, b_2, b_3, \dots, b_i, b_m$$

Para entenderlo mejor usaremos un ejemplo sencillo. En la Figura 4.6 tenemos 2 secuencias A y B

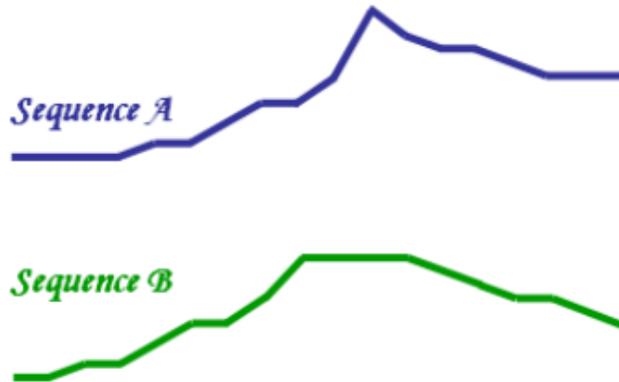


Figura 4.6 Secuencias A y B

Estas 2 secuencias se dibujaran en los lados de una cuadrícula, uno en la parte superior y otro en la parte izquierda. Dentro de cada celda se coloca la diferencia comparando los elementos correspondientes de las dos secuencias. Para encontrar la mejor coincidencia o alineación entre estas dos secuencias se tiene que encontrar un camino a través de la cuadrícula que minimiza la distancia total entre ellos. En la Figura 4.7 podemos ver el camino mínimo en forma de puntos rojos de la secuencia A y B.

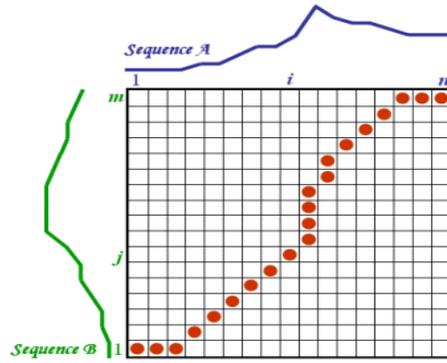


Figura 4.7 Ejemplo al aplicar el algoritmo DTW

4.1.4.1 Algoritmo DTW

El algoritmo tiene dos parámetros de entrada:

(1) Secuencia de datos Q

$$(4.20) \quad Q = \{q_1, q_2, \dots, q_n\}$$

(2) Cadena de búsqueda C

$$(4.21) \quad C = \{c_1, c_2, \dots, c_m\}$$

Para realizar la alineación se construye una matriz D de $n \times m$ elementos, donde cada elemento d_{ij} contiene la distancia

$$(4.22) \quad D(q_i, c_j) = (q_i - c_j)^2$$

entre los elemento q_i y c_j . A partir de esta matriz se define el camino de deformación ω . Éste es continuo, y cada elemento K^{th} de W está definido como $w_k = (i, j)_k$ donde $W = w_1, w_2, \dots, w_k$ y $\max(m, n) \leq K < m + n - 1$.

4.1.4.2 Restricciones

Este camino está marcado por las siguientes restricciones:

- **Condiciones de contorno:** $w_1 = (1, 1)$ y $w_k = (m, n)$ son el comienzo y final del camino, siendo k la longitud del camino de deformación.
- **Continuidad:** Tomando un elemento del camino $w_i = (a, b)$ entonces $w_{i-1} = (a', b')$ donde $a \leq a' \leq a + 1$ y $b - b' \leq 1$. Es decir las celdas son adyacentes.
- **Monotonía:** Tomando un elemento del camino $w_i = (a, b)$ entonces $w_{i-1} = (a', b')$ donde $a - a' \geq 0$ y $b - b' \geq 0$. Obliga a que los puntos de W estén espaciados en el tiempo de forma continua.

Hay varios caminos que satisfacen estas condiciones. La mejor solución va a ser la que cumpla la siguiente ecuación.

$$(4.23) \quad DTW(Q, C) = \sqrt{\sum_{k=1}^K w_k / K}$$

4.2 Clasificación manual del LSM a reconocer

Con la ayuda del libro manos con voz diccionario de lengua de señas mexicana (Serafín de Fleischmann & González Pérez, Manos con Voz - Diccionario de Lengua de Señas Mexicana, 2011) se obtuvieron las señas que serán utilizadas durante este trabajo de investigación; se recolectaron un total de 84 señas.

A continuación se presentas las señas con su imagen y la forma de realizar de cada una de ellas.

4.2.1 Abecedario



A

Con la mano cerrada, se muestran las uñas y se estira el dedo pulgar hacia un lado. La palma mira al frente.



B

Los dedos índice, medio, anular y meñique se estiran bien unidos y el pulgar se dobla hacia la palma, la cual mira al frente.



C

Los dedos: índice, medio, anular y meñique se mantienen bien unidos y en posición cóncava; el pulgar también se pone en esa posición. La palma mira a un lado.



D

Los dedos medio, anular, meñique y pulgar se unen por las puntas y el dedo índice se estira. La palma mira al frente.



E

Se doblan los dedos completamente, y se muestran las uñas. La palma mira al frente.



F

Con la mano abierta y los dedos bien unidos, se dobla el índice hasta que su parte lateral toque la yema del pulgar. La palma mira a un lado.



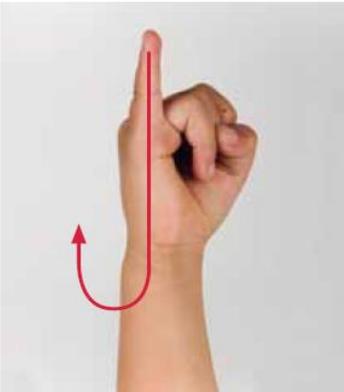
G
Se cierra la mano y los dedos índice y pulgar se estiran. La palma mira hacia usted.



H
Con la mano cerrada y los dedos índice y medio bien estirados y unidos, se extiende el dedo pulgar señalando hacia arriba. La palma mira hacia usted.



I
Con la mano cerrada, el dedo meñique se estira señalando hacia arriba. La palma se pone de lado.



J
Con la mano cerrada, el dedo meñique bien estirado señalando hacia arriba y la palma a un lado dibuja una j en el aire.



K
Se cierra la mano con los dedos índice, medio y pulgar estirados. La yema del pulgar se pone entre el índice y el medio. Se mueve la muñeca hacia arriba.



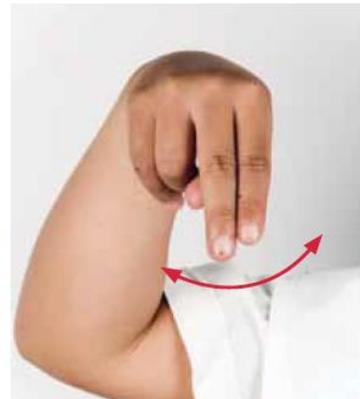
L
Con la mano cerrada y los dedos índice y pulgar estirados, se forma una l. La palma mira al frente.



M
Con la mano cerrada, se ponen los dedos índice, medio y anular sobre el pulgar.



N
Con la mano cerrada, se ponen los dedos índice y medio sobre el pulgar.



Ñ
Con la mano cerrada, se ponen los dedos índice y medio sobre el pulgar. Se mueve la muñeca a los lados.



O
Con la mano se forma una letra o. Todos los dedos se tocan por las puntas.



P
Con la mano cerrada y los dedos: índice, medio y pulgar estirados, se pone la yema del pulgar entre el índice y el medio.



Q
Con la mano cerrada, se ponen los dedos índice y pulgar en posición de garra. La palma mira hacia abajo, y se mueve la muñeca hacia



R
Con la mano cerrada, se estiran y entrelazan los dedos índice y medio. La palma mira al frente.



S
Con la mano cerrada, se pone el pulgar sobre los otros dedos. La palma mira al frente.



T
Con la mano cerrada, el pulgar se pone entre el índice y el medio. La palma mira al frente.



U
Con la mano cerrada, se estiran los dedos: índice y medio unidos. La palma mira al frente.



V
Con la mano cerrada, se estiran los dedos: índice y medio separados. La palma mira al frente.



W
Con la mano cerrada, se estiran los dedos índice, medio y anular separados. La palma mira al frente.



X
Con la mano cerrada, el índice y el pulgar en posición de garra y la palma dirigida a un lado, se realiza un movimiento al frente y de regreso.



Y
Con la mano cerrada, se estira el meñique y el pulgar. La palma mira hacia usted.



Z
Con la mano cerrada, el dedo índice estirado y la palma al frente, se dibuja una letra z en el aire.

4.2.2 Números



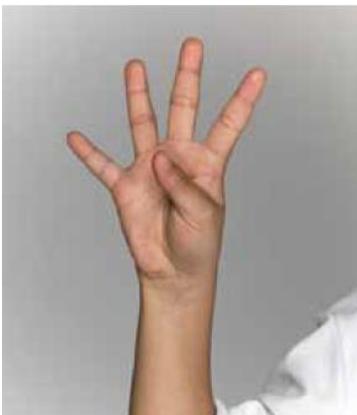
UNO
Se estira el índice con la palma al frente.



DOS
Se estiran los dedos índice y medio.



TRES
Se estiran los dedos índice, medio y anular con la palma al frente.



CUATRO
Se estiran los dedos: índice, medio, anular y meñique.



CINCO
Se abre la mano con la palma al frente y los dedos separados.



SEIS
Se hace la letra a con el pulgar hacia arriba.



DIEZ
Se hace el número 5 con la palma hacia arriba, y se rota la mano rápidamente hacia abajo.

4.2.3 Palabras



CALDO
Se hace una c con los dedos ligeramente separados, y se coloca sobre el dorso del puño, tocándolo con la punta de los dedos.



CARNE
Con los dedos índice y pulgar de una mano se pellizca el dorso de la otra.



PAY
Se colocan las puntas de los dedos: índice y medio estirados en la parte lateral del dedo índice de la otra mano.



TORTA
Se colocan las manos en posición cóncava, una sobre la otra por las palmas, señalando al frente.



CISNE
Con una mano en capullo, se coloca el codo de ese brazo sobre la palma de la otra mano.



JIRAFÁ
Se estiran el índice y el meñique, mientras las yemas del medio, anular y pulgar se unen. El codo de este brazo se coloca sobre la palma de la otra mano.



VACA

Los dedos: medio, anular y pulgar se unen por las yemas, mientras el índice y el meñique se estiran.



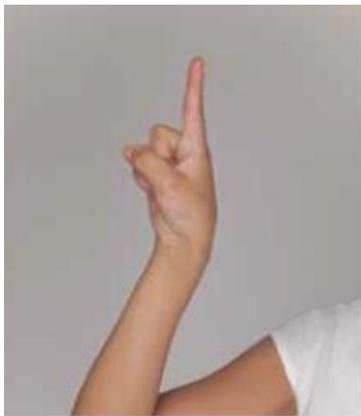
ABAJO

Se estira el índice, y se señala hacia abajo.



ADENTRO

Se coloca el índice dentro de la otra mano cerrada.



ARRIBA

Con el índice estirado se señala hacia arriba.



MENOS

Se estira el índice con la palma hacia abajo, señalando a un lado.



MUCHO

Se pone la mano en capullo con la palma hacia arriba.



BOTE

Se coloca una palma sobre la otra sin tocarse, simulando tener un bote entre las manos.



CANDELERO

Se simula tomar el mango de un candelero con una mano y detenerlo con la otra mano abierta.



CASA

Se unen las manos abiertas por las yemas de los dedos y se mantienen las palmas separadas.



COMAL

Se hace la letra c con ambas manos colocadas frente a frente, y se simula rodear el comal.



COPA

Se coloca la mano como si se tomara una copa.



HOGAR

Se hace una letra h con ambas manos, unidas por las yemas de los dedos índice y medio con las palmas separadas.



MESA

Se coloca la mano abierta con los dedos unidos debajo de la palma de la otra mano, y la toca con la punta de los dedos.



MANO

Se muestra la mano abierta a la persona con quien se habla.



PULGAR

Con el índice se señala el dedo pulgar de la otra mano.



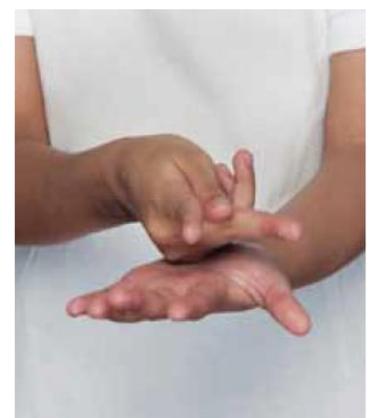
UÑA

Con el índice y el pulgar tocándose por las yemas, se rodea la uña del índice de la otra mano.



ESCUADRA

Se coloca una letra v sobre el índice de la otra mano, tocándolo con la punta de los dedos



PREMIO

Se coloca una p sobre la palma de la otra mano, tocándola con el dorso.



ESPOSO

Se unen las manos por las palmas.



MADRE

Se coloca una letra m sobre el dorso de la otra mano, tocándolo con la punta de los dedos.



PADRE

Se coloca una letra p sobre el dorso de la otra mano.



SEÑORITA

Se coloca una letra t sobre la frente, tocándola con la parte lateral del dedo medio, y se pone esa mano sobre el puño de la otra mano.



ALMEDRA

Se estiran el índice y el pulgar y casi se juntan, sin tocarse, al frente.



CEREZA

Se estiran el índice y el pulgar, y se juntan por las puntas, simulando sostener una cereza del rabito.



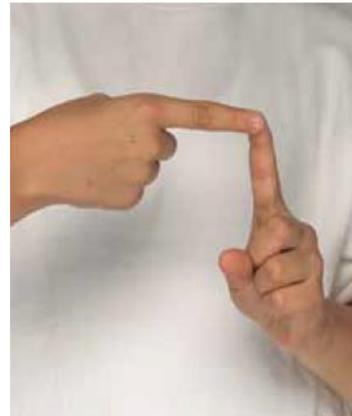
CHAMPIÑON

Se coloca la mano abierta con la palma hacia abajo sobre el dedo índice de la otra mano.



COLIFLOR

Se colocan las manos en posición cóncava, una sobre la otra, sin tocarse.



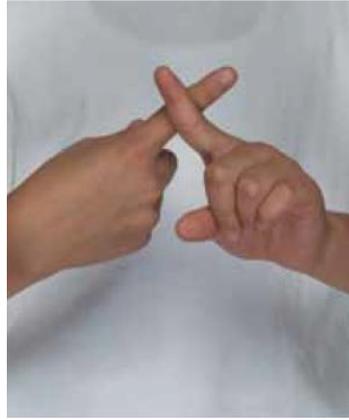
DIVISION

Con la punta del índice estirado, y la palma hacia usted en posición horizontal, se toca la punta del índice de la otra mano, que está estirado y con la palma al frente.



ESFERA

Se hace la letra c con ambas manos, tocándose por la punta de los dedos.



MULTIPLICACIÓN

Se cruzan los índices de ambas manos, tocándose por el dorso.



RAÍZ CUADRADA

Con la punta del índice estirado, se toca la punta del índice de la otra mano, que hace la letra v.



RESTA

Se coloca el índice estirado con la palma hacia usted en posición horizontal a la altura del pecho.



SUMA

Se cruzan los índices de ambas manos, tocándose por los dorsos de los dedos, simulando un signo de suma.



AHÍ

Con el índice se señala el lugar de que se trata.



AQUÍ

Con la mano cerrada y el índice estirado se señala el piso.



BALÓN

Con las manos en posición cóncava, una frente a la otra, se simula sostener un balón.



ESQUINA

Con la punta del índice se toca la punta del otro índice, formando una esquina.



PESOS

Se colocan el índice y el pulgar en posición de garra.



PISTOLA

Se hace una letra g señalando hacia usted.



SOSTENER

Con las manos abiertas y las palmas hacia arriba, se simula sostener un objeto.



TOCAR

Con la yema del dedo medio de la mano abierta, se toca el dorso de la otra.



MUÑECA

Con la mano abierta se rodea la muñeca de la otra mano.

4.3 Conexión exitosa del Smartphone con *Leap Motion Controller*

Después de haber elegido las señas que serían candidatas a formar parte de la aplicación se realizó una conexión entre el Smartphone y el LMC, con esta conexión se podrá ver que tan factible es utilizar el Smartphone como intermediario entre la persona señante y el oyente.

Actualmente el LMC se encuentra para diferentes lenguajes de programación tales como: JavaScript, Unity, C#, C++, Java, Python, Objective-C, Unreal. La empresa de Leap Motion proporciona librerías y bibliotecas para que los desarrolladores utilicen y puedan desarrollar aplicaciones usando su dispositivo el LMC.

Sin embargo no es posible utilizar el LMC en un Smartphone con Sistema Operativo Android. Para poder usarlo es necesario llenar un formulario que la empresa de Leap Motion proporciona y de esta manera es liberado el SDK de Android y pueda ser utilizando en cualquier dispositivo Android.

En la Figura 4.8 se puede observar el cuestionario o bien ingresando a través de la siguiente URI:

https://docs.google.com/forms/d/1R2sqQ_cdiMPC7u-RA5bP98hldJmNFdVb73M3131Ss/viewform

Leap Motion Android SDK

Welcome to our Android SDK private alpha signup form. In order to join the alpha, you'll need to have a Leap Motion developer account (which involves signing our SDK terms and conditions). If you don't have an account, register at <https://developer.leapmotion.com>.

***Obligatorio**

What's your name? *

What's the email address you used to register as a Leap Motion developer? *

What would you like to create with the Android SDK? *

What type of mobile hardware (including processor) do you want to build with? Please select all that apply, up to 3. *

Note that the alpha currently only provides full support for Snapdragon 800+ devices. If you don't have the necessary hardware, we'll be in touch once the SDK supports one of your top choices, or enters public beta.

- Galaxy Note 4, Exynos 5433
- Galaxy Note 4, Snapdragon 805 (Gear VR)
- Galaxy S5, Exynos 5422
- Galaxy S5, Snapdragon 805
- Galaxy Tab S 8.4, Snapdragon 800

Figura 4.8 Formulario Leap Motion Android SDK

Después de haber llenado el formulario, la empresa Leap Motion envía un correo con la dirección URL para descargar los archivos que son necesarios para que el LMC sea reconocido en un Smartphone, el correo es enviado a la dirección de correo electrónico que se utilizó para registrarse con Leap Motion. El tiempo de respuesta es de 1 a 2 días. En la Figura 4.9 se puede apreciar los archivos que fueron compartidos

Nombre	Fecha de modifica...	Tipo	Tamaño
old	30/09/2015 10:39 a...	Carpeta de archivos	
GearVR	18/11/2015 01:19 ...	Carpeta de archivos	
command-line daemon	30/09/2015 10:39 a...	Carpeta de archivos	
LeapMotionCoreAsset_2.3.2_Android	09/10/2015 09:21 ...	Unity package file	223,621 KB
leap_android_unity_examples	17/04/2014 01:14 ...	M4V Video File	5,354 KB
README-2.3.2+35031.txt	23/10/2015 01:59 ...	Documento de Go...	1 KB
LeapDeveloperKit_develop_android_2.3.2_35031	25/11/2015 11:28 a...	Archivo WinRAR Z...	266,023 KB
Robits.apk	14/05/2014 02:15 a...	Archivo APK	10,373 KB
RiggedHandVisualizer.apk	02/10/2015 04:58 ...	Archivo APK	20,203 KB
RagdollThrower.apk	14/05/2014 02:11 a...	Archivo APK	12,782 KB
Playground.apk	02/10/2015 10:23 ...	Archivo APK	38,695 KB
LeapSimple-release.apk	25/09/2015 06:12 ...	Archivo APK	839 KB
LeapDaemon-release-2.3.2+35031.apk	13/10/2015 11:27 ...	Archivo APK	31,532 KB
ImageHands.apk	07/10/2015 09:09 ...	Archivo APK	26,185 KB
AllHandModels.apk	05/10/2015 07:19 ...	Archivo APK	31,163 KB
Framerate	10/12/2014 12:39 a...	Archivo	2,854 KB
GearVR_hardware	17/10/2015 09:14 ...	Adobe Acrobat D...	4,941 KB

Figura 4.9 Archivos proporcionados por Leap Motion

4.3.1 Requerimientos para conexión Smartphone – Leap Motion Controller

Para realizar la conexión exitosa del LMC con el Smartphone es necesario primeramente contar con los requisitos del sistema que la empresa menciona en su correo que envía, las cuales son:

Tabla 4.1 Requerimientos LMC Android

Dispositivo:	Dispositivo Android que cuente arquitectura ARM
Interfaz:	USB OTG
Sistema Operativo:	Android 4.4+
Software Adicional:	Android SDK
Cables:	Adaptador Micro USB OTG
Dispositivo Leap Motion:	Leap Motion Controller periférico o un módulo incrustado aprobado.

Una vez que se verificó que el Smartphone cuenta con los requisitos recomendados se procedió a instalar la aplicación **LeapDaemon-release-2.3.2+35031.apk** (Figura 4.10), esta aplicación tiene la función de emular al LMC para que se pueda realizar cualquier actividad con el LMC y el Smartphone.

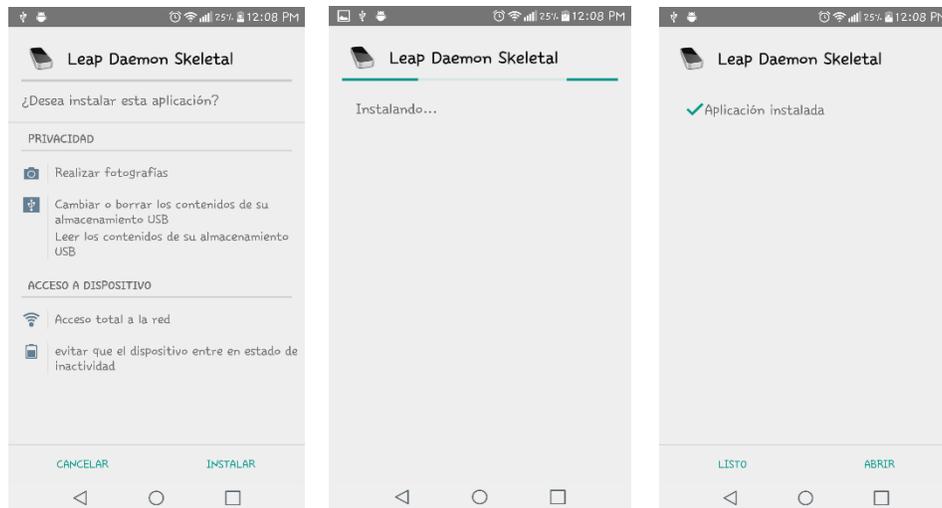


Figura 4.10 Instalación de la aplicación

Una vez que se instaló la aplicación, con el cable Micro USB-OTG se conectó el LMC al Smartphone, se buscó la aplicación con el nombre de *Leap Motion Skeletal* (Figura 4.13a) y la abrimos, cuando no detecta al LMC conectado al Smartphone aparece con una equis en rojo como se puede ver en la Figura 4.13b en caso contrario cuando lo detecta aparecen unos *check* de color verde como se puede apreciar en la Figura 4.13c.



Figura 4.13a Aplicación instalada



Figura 4.13b LMC no detectado o no conectado



Figura 4.13c LMC detectado y conectado

Luego se procedió a instalar una aplicación para probar el funcionamiento que contendría el LMC con el Smartphone, se utilizó la aplicación **Chickens.apk**, en la Figura 4.14 se puede apreciar la aplicación funcionando y la detección de las manos, llevando a cabo una conexión exitosa entre el LMC y el Smartphone.

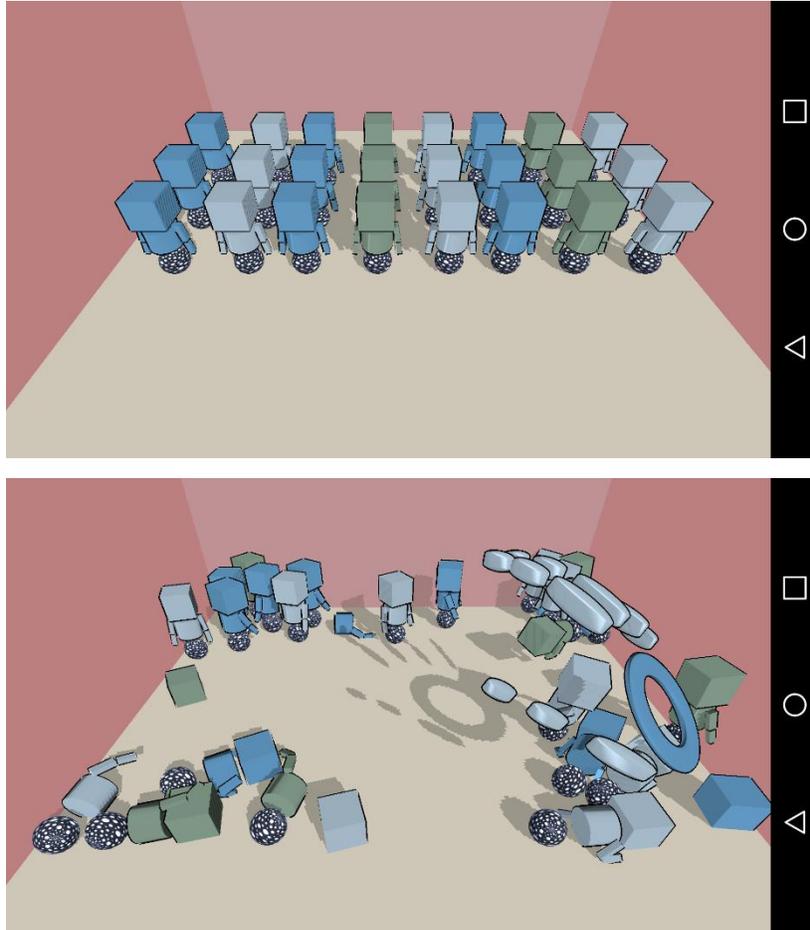


Figura 4.14 Pruebas del funcionamiento del LMC

4.4 Aprendizaje de la Lengua de Señas Mexicana

Para el aprendizaje de la Lengua de Señas Mexicana las señas fueron catalogadas en 2 tipos de clases.

- **Estáticas:** Señas que no requieren movimiento de la mano.
- **Dinámicas:** Señas que requieren movimiento de la mano.

Con la finalidad de enfocarse primeramente con las señas estáticas por las limitaciones que el LMC presenta cuando se realiza algún gesto que lleve movimiento.

Tabla 4.2 Señas de la LSM

Abecedario	Tipo	Números	Tipo	Palabras	Tipo	Palabras	Tipo
A	Estática	1	Estática	Caldo	Estática	Resta	Estática
B	Estática	2	Estática	Jirafa	Estática	Esquina	Estática
C	Estática	3	Estática	Menos	Estática	Torta	Estática
D	Estática	4	Estática	Comal	Estática	Adentro	Estática
E	Estática	5	Estática	Mano	Estática	Candelero	Estática
F	Estática	6	Estática	Esposo	Estática	Mesa	Estática
G	Estática	7	Estática	Cereza	Estática	Escuadra	Estática
H	Estática	8	Estática	Multiplicación	Estática	Señorita	Estática
I	Estática	9	Dinámica	Aquí	Estática	División	Estática
J	Dinámica	10	Dinámica	Sostener	Estática	Suma	Estática
K	Dinámica			Carne	Estática	Peso	Estática
L	Estática			Vaca	Estática	Cisne	Estática
M	Estática			Mucho	Estática	Arriba	Estática
N	Estática			Copa	Estática	Casa	Estática
Ñ	Dinámica			Pulgar	Estática	Muñeca	Estática
O	Estática			Madre	Estática	Premio	Estática
P	Estática			Champiñones	Estática	Almendra	Estática
Q	Dinámica			Raíz Cuadrada	Estática	Esfera	Estática
R	Estática			Balón	Estática	Allí	Estática
S	Estática			Tocar	Estática	Pistola	Estática
T	Estática			Pay	Estática		
U	Estática			Abajo	Estática		
V	Estática			Bote	Estática		
W	Estática			Hogar	Estática		
X	Dinámica			Uña	Estática		
Y	Estática			Padre	Estática		
Z	Dinámica			Coliflor	Estática		

4.4.1 Uso del *LeapTrainer.js*

Para poder realizar la grabación de las señas se utilizó el *framework LeapTrainer.js*. El *framework* necesita del software del *Leap Motion Controller*, por tal motivo es necesario tener instalado el software antes de usar el *framework*. Una vez que se tiene instalado el software se conecta al puerto USB el LMC y se descarga el proyecto desde la URI menciona en el Capítulo 2.

Al ejecutar el archivo *trainer-ui.html* se puede apreciar la interfaz del *framework* (Figura 4.15).

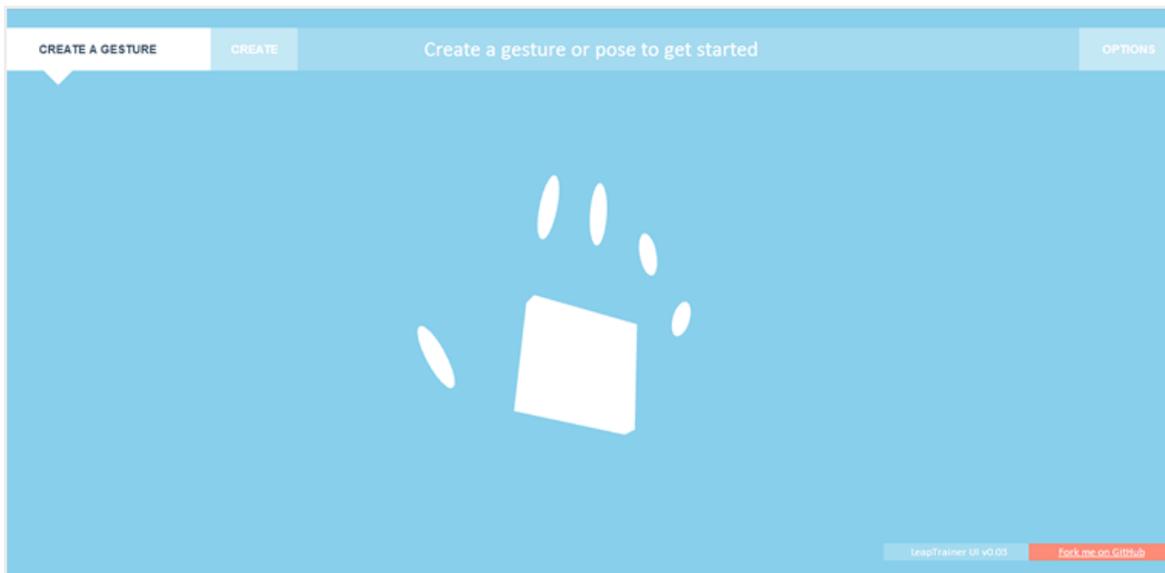


Figura 4.15 *LeapTrainer.js* (Rob O'Leary, 2016)

El LMC detecta la transmisión de los datos haciendo uso del API JavaScript de Leap Motion. Los datos que el API retorna son analizados por el *framework* de *LeapTrainer* y estos son usados para aprender gestos disparando eventos cuando el gesto y pose son detectados.

4.4.2 Opciones del *Leaptrainer.js*

El *LeapTrainer* cuenta con el apartado de opciones en el cual el usuario podrá hacer uso de ellas. Las opciones que cuenta son las siguientes.

- *Recording trigger*: Forma por la cual será grabada la seña.
- *Gesture encoding*: Gesto usado para la codificación.
- *Recognition Strategy*: Algoritmo usado para el reconocimiento de la seña.
- *Min. gesture velocity*: Velocidad mínima que necesita un *frame* para activar la grabación de un gesto (300 *valor por defecto*).

- *Max. pose velocity*: La velocidad máxima que un *frame* puede medir y activar la grabación de una pose (30 *valor por defecto*).
- *Min. gesture frames*: El número mínimo de *frames* que puede contener un gesto reconocido (5 *valor por defecto*).
- *Min. pose frames*: El número mínimo de *frames* que necesita para alcanzarse como grabable antes de grabar la pose que actualmente está activa (75 *valor por defecto*).
- *Hit threshold*: El valor de retorno de la función reconocimiento por encima de un gesto que es considerado reconocido (0.7 *valor por defecto*).
- *Training gestures*: Número de gestos entrenados que requiere para llevar a cabo el modo entrenamiento (1 *valor por defecto*).
- *Convolution factor*: El factor por el cual las muestras de entrenamiento se circunvolucionan sobre una distribución gaussiana con el fin de ampliar los datos de entrenamiento de entrada (0 *valor por defecto*).
- *Down-time*: Número de milisegundos antes de que un gesto sea identificado y después de que otro ciclo de gesto pueda iniciar a grabar (1000 *valor por defecto*).

El usuario puede modificar los valores que vienen por defecto y colocar el valor que desee, esto dependerá de la opción a modificar como se puede apreciar en la Figura 4.16; estos cambios se verán reflejados en los gestos y poses que se deseen grabar y aprender.

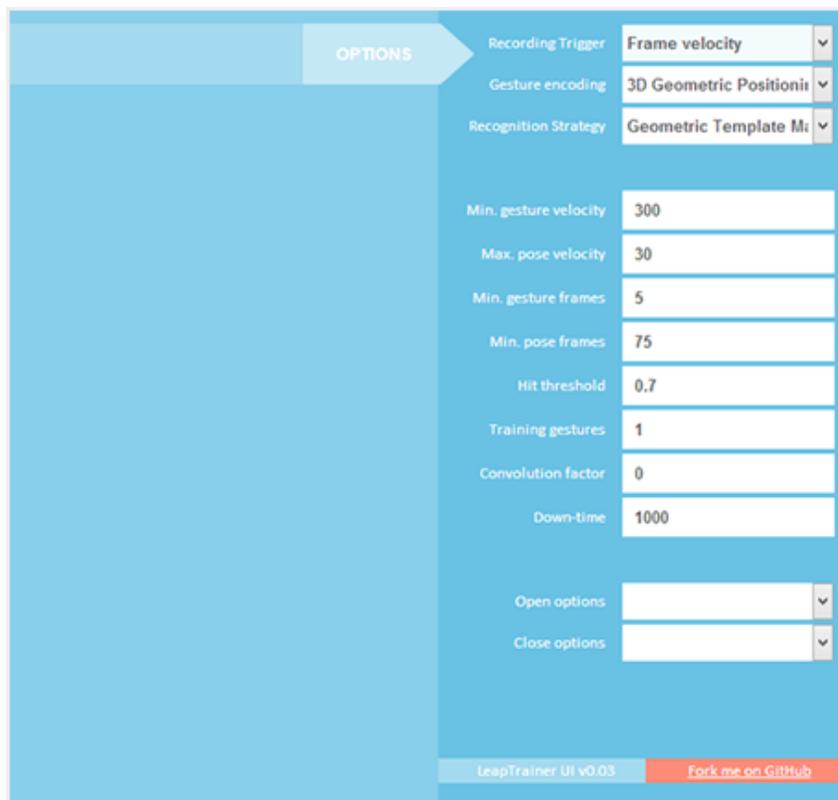


Figura 4.16 Opciones LeapTrainer (Rob O'Leary, 2016)

4.5 Algoritmo usado para la grabación de la Lengua de Seña Mexicana

El *LeapTrainer* cuenta con 3 implementaciones para el reconocimiento de gestos o seña realizada frente al *LMC*:

- *Geometric Template Matching*: Implementa un algoritmo que está basado en el reconocimiento de puntos de nube (*\$P\$ Point-Cloud Recognizer*).
- *Artificial Neural Networks*: Implementa reconocimiento de gestos y postura a través de las redes neurales.
- *Cross-correlation*: Se realiza un correlación cruzada (*cross-correlation*) para detectar similitudes entre gestos de entrada y aprendido.

Geometric Template Matching fue la implementación a usar para el reconocimiento de la Lengua de Señas Mexicana. La implementación está basado en el diagrama matemático para el reconocimiento de nube de puntos (*\$P\$ Point-Cloud Recognizer*) (O'Leary, 2016). Es utilizado para reconocimientos de gestos en 2-D diseñados para el prototipo rápido de interfaces de usuarios basados en gestos. En términos de máquinas de aprendizaje *\$P\$ Point-Cloud* es una instancia basada en el clasificador con funciones de puntuación euclidiana del vecino cercano. *Geometric Template Matching* implementa un método para el reconocimiento de gestos que es basado en nube de puntos (en inglés, *point cloud*). La nube de puntos es un conjunto de vértices en un sistema de coordenadas tridimensional, habitualmente se identifican como coordenadas X, Y, Z que son representaciones de la superficie externa de un objeto (Vatavu, Anthony, & Wobbrock, 2012).

Para el reconocimiento de la nube de puntos se definen la coincidencia entre dos nubes de puntos \mathcal{C} y \mathcal{T} como un función de \mathcal{M} que asocia cada punto $C_i \in \mathcal{C}$ exactamente con un punto $T_j \in \mathcal{T}, T_j = \mathcal{M}(C_i)$. Si \mathcal{C} y \mathcal{T} han sido remuestreados en el mismo número de puntos n , entonces la coincidencia también consistirá exactamente en n pares de puntos. La fórmula (numero) muestra la mejor coincidencia para \mathcal{M} como la suma de la distancia Euclidianas para todos los pares de puntos de \mathcal{M}

$$(4.24) \quad \sum_{i=0}^n \| C_i - T_j \| = \sum_{i=1}^n \sqrt{(C_i \cdot x - T_j \cdot x)^2 + (C_i \cdot y - T_j \cdot y)^2}$$

El reconecedor de nube de puntos necesita buscar la distancia de coincidencia mínima ("La mejor") entre \mathcal{C} y \mathcal{T} de todas las alineaciones de $n!$ posibles, la solución para resolver la distancia mínima es la del uso del algoritmo Húngaro; este algoritmo ofrece el rendimiento de igualación de coste mínimo ideal, pero en una complejidad de tiempo alto.

4.5.1 Entrenamiento de las señas

Una vez catalogadas las señas como dinámicas y estáticas, se procedió al entrenamiento de las señas, el proceso utilizado fue el siguiente:



Conexión *Leap Motion Controller* con la computadora

Para que el dispositivo funcione es necesario tener instalado el software para hacer uso del *Leap Motion Controller*, por otro lado la computadora debe de cumplir con los requisitos mínimos.

Los requisitos mínimos requeridos son:

- Windows® 7+ or Mac® OS X 10.7+
- AMD Phenom™ II or Intel® Core™ i3/i5/i7 processor
- 2 GB RAM
- Puerto USB 2.0
- Conexión a internet

Grabación de la seña

Para la grabación de las señas se inició con el abecedario, luego números y finalizando con días de la semana.

1. Primero se realizó la seña a grabar quedando inmóvil si fuera estática o realizando el movimiento si fuera dinámica (Figura 4.17).

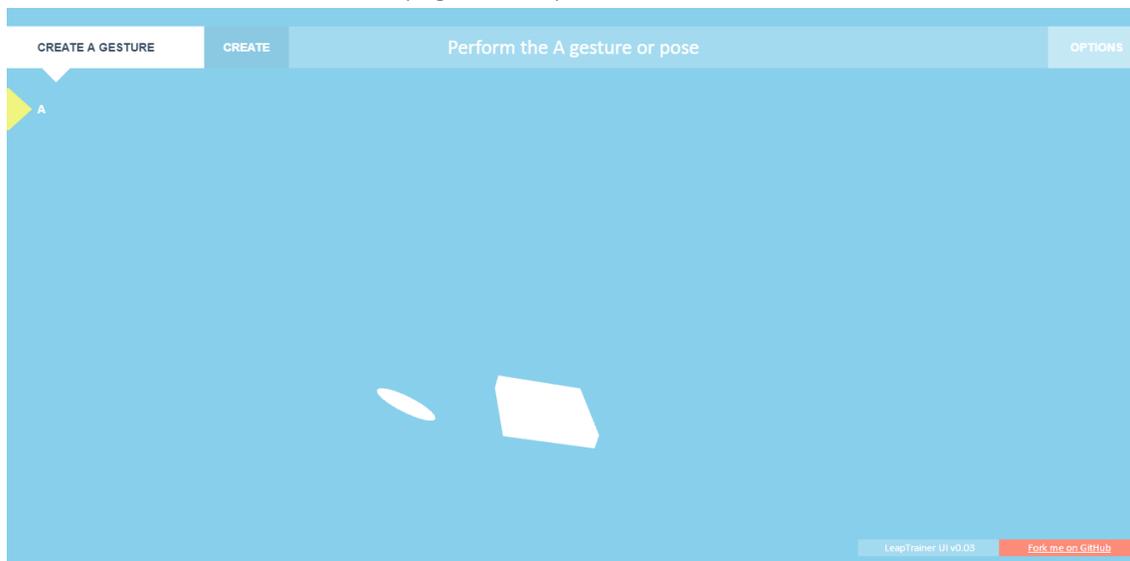


Figura 4.17 Letra "a" realizada

2. La seña ha sido aprendida cuando aparezca la leyenda *pose learned* (Figura 4.18).

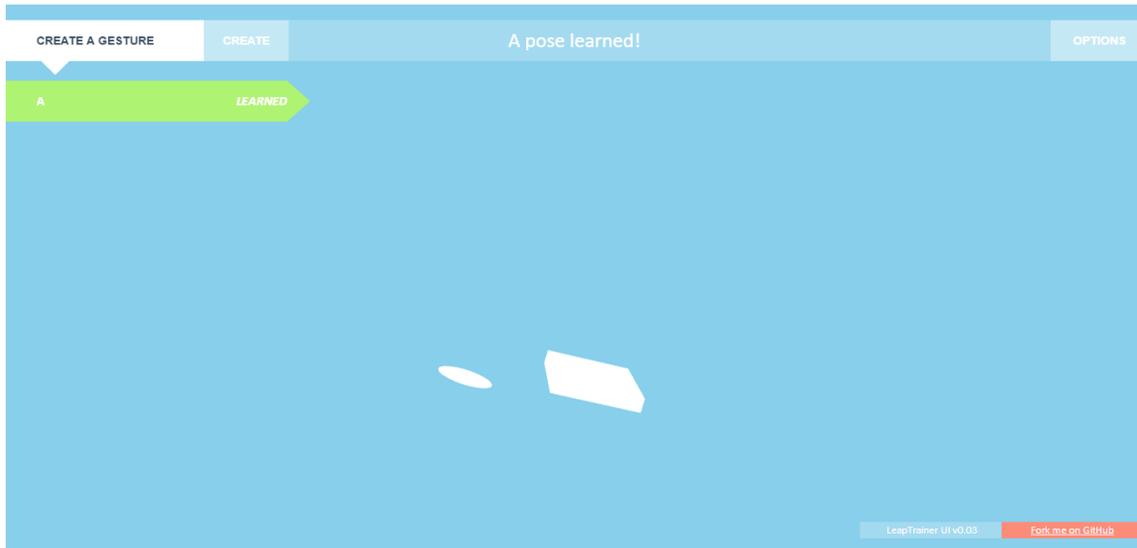


Figura 4.18 Letra "a" aprendida

3. Una vez aprendida la seña el *LeapTrainer* proporciona un porcentaje de reconocimiento para la seña realizada en dicho momento (Figura 4. 19).

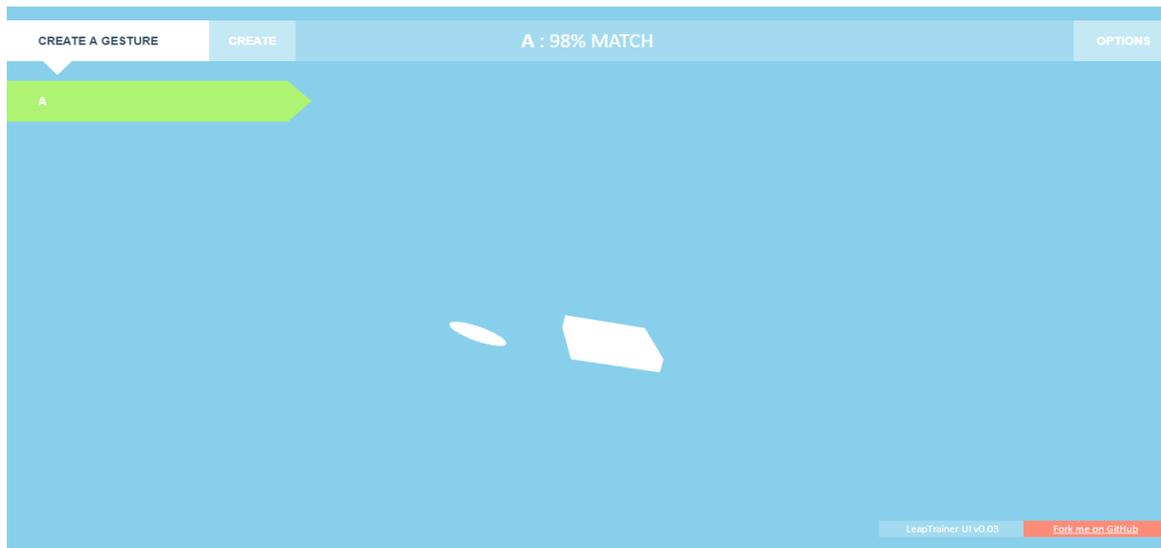


Figura 4. 19 Letra "a" porcentaje de coincidencia

Así se fue haciendo con cada una de las señas que se describen en la Tabla 4.2 Señas de la LSM.

Generación de JSON de seña entrenada

LeapTrainer genera un JSON el cual nos proporciona los datos entrenados de la seña realizada (Figura 4.20). Con este dato JSON y usando algunos de los archivos de JavaScript que contiene el *framework LeapTrainer* se realizó la creación de la aplicación *SignMx*.

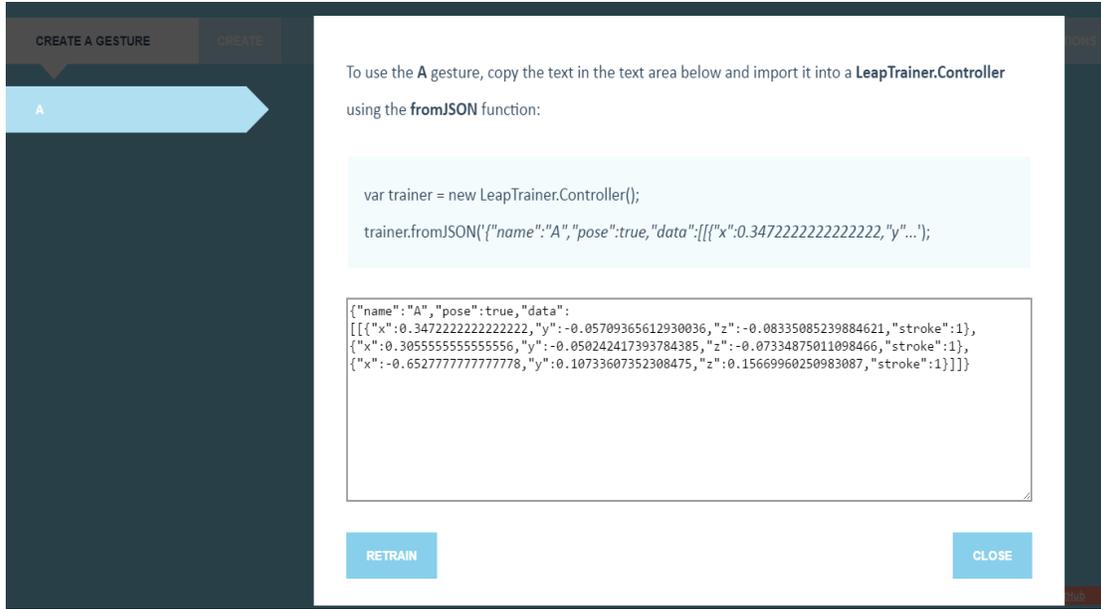


Figura 4.20 JSON de la seña "a"

Guardar dato JSON

Una vez entrenadas y grabadas las señas realizadas se almacenaron en un archivo de texto de manera temporal (Figura 4.21) para posteriormente guardarlas en un archivo *JavaScript* el cual es usado en la aplicación *SignMx*.

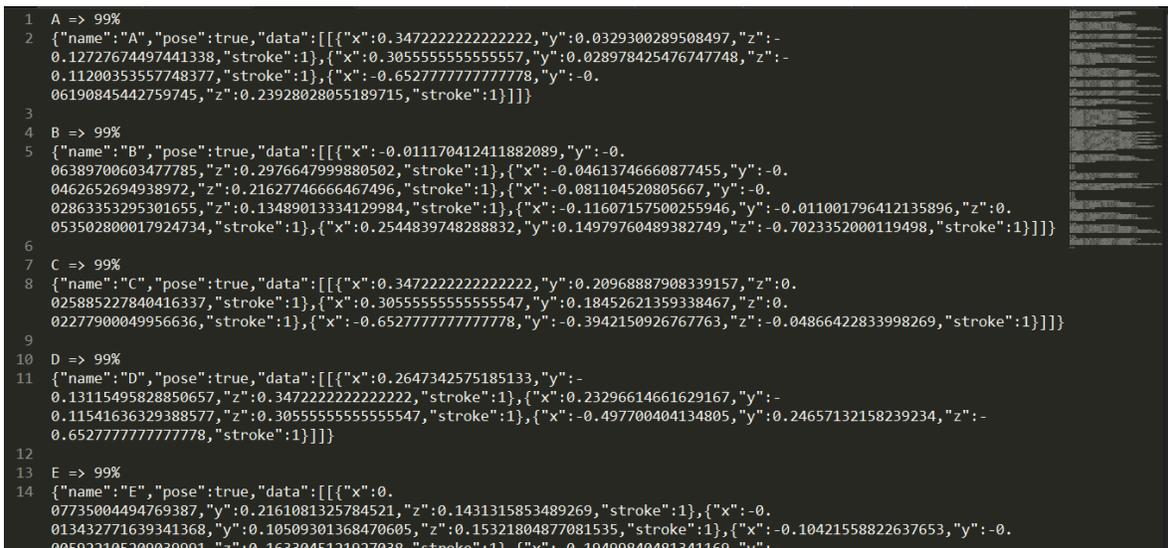


Figura 4.21 Dato JSON de las señas.

4.5.2. Señas reconocidas

Se realizó el proceso de entrenamiento de todas las señas descritas en el apartado 4.2 **Clasificación manual del LSM a reconocer**. A continuación se muestra la Tabla 4.3 la cual contiene las señas que fueron reconocidas; se ocupó la Tabla 4.2 solo con la diferencia en que las señas reconocidas se encuentran en **negritas** y color beige, mientras las que no se reconocieron están en *cursiva* y color celeste.

Tabla 4.3 Señas reconocidas

Abecedario	Tipo	Números	Tipo	Palabras	Tipo	Palabras	Tipo
A	Estática	1	Estática	Caldo	Estática	Resta	Estática
B	Estática	2	Estática	Jirafa	Estática	Esquina	Estática
C	Estática	3	Estática	Menos	Estática	Torta	Estática
D	Estática	4	Estática	Comal	Estática	Adentro	Estática
E	Estática	5	Estática	Mano	Estática	Candelero	Estática
F	Estática	6	Estática	Esposo	Estática	Mesa	Estática
G	Estática	7	Estática	Cereza	Estática	Escuadra	Estática
H	Estática	8	Estática	Multiplicación	Estática	Señorita	Estática
I	Estática	9	Dinámica	Aquí	Estática	División	Estática
J	Dinámica	10	Dinámica	Sostener	Estática	Suma	Estática
K	Dinámica			Carne	Estática	Peso	Estática
L	Estática			Vaca	Estática	Cisne	Estática
M	Estática			Mucho	Estática	Arriba	Estática
N	Estática			Copa	Estática	Casa	Estática
Ñ	Dinámica			Pulgar	Estática	Muñeca	Estática
O	Estática			Madre	Estática	Premio	Estática
P	Estática			Champiñones	Estática	Almendra	Estática
Q	Dinámica			Raíz Cuadrada	Estática	Esfera	Estática
R	Estática			Balón	Estática	Allí	Estática
S	Estática			Tocar	Estática	Pistola	Estática
T	Estática			Pay	Estática		
U	Estática			Abajo	Estática		
V	Estática			Bote	Estática		
W	Estática			Hogar	Estática		
X	Dinámica			Uña	Estática		
Y	Estática			Padre	Estática		
Z	Dinámica			Coliflor	Estática		

En el Anexo A se muestran las imágenes en 3D de las señas reconocidas, estas señas serán utilizadas en la aplicación *SignMx*. Las imágenes fueron tomadas del *Leap Motion Visualizer* complemento integrado en la instalación del software *Leap Motion*.

4.5.3 Total de señas entrenadas y grabadas

Cuando se finalizó el proceso que se utilizó en el apartado **4.5.1 Entrenamiento de las señas** y utilizando las señas descritas en la **Tabla 4.2 Señas de la LSM** se obtuvo la Tabla 4.4 la cual nos proporciona un número total de señas propuestas y un total de señas reconocidas y las que no reconocidas.

Tabla 4.4 Total de señas propuestas y reconocidas

Señas	Propuestos	Reconocidas	No reconocidas
Abecedario	27	17	10
Números	10	8	2
Palabras	47	20	27
TOTAL	84	45	39

Muchas de las señas no fueron reconocidas por las limitaciones que el *Leap Motion Controller* presenta tales como:

- Pérdida de visión cuando se juntan las yemas de los dedos.
- Pérdida de visión cuando se aproximan demasiado una mano con la otra.
- No reconoce gesto cuando transponen una mano encima de la otra.

4.6 Creación de la aplicación *SignMx*

La aplicación *SignMx* (en Español *Señas Mexicana*) es el conjunto de todo lo realizado en este trabajo de investigación. Fue desarrollada para versión web utilizando el lenguaje HTML para crear la vista, JavaScript el cual contiene las señas aprendidas y reconocidas, texto a voz, 3D de las manos entre otras y los CSS para los estilos de la vista.

La estructura que conforma la aplicación se encuentra descrita en la Figura 4.22.



Figura 4.22 Estructura *SignMx*

La interfaz de *SignMx* cuenta con unas opciones para realizar alguna función que se desea ejecutar, dichas opciones están en la Figura 4.23, las opciones son las siguientes:

1. **Panel de escritura:** En este panel serán escritas las letras o palabras de las señas que sean reconocidas.
2. **Reproducir:** Este botón reproduce en audio lo que está escrito en el panel de escritura.
3. **Detener:** Este botón detiene la reproducción del audio.
4. **Limpiar:** Este botón limpia el último carácter que exista en el panel de escritura.
5. **Borrar:** Este botón borra todo lo que contenta el panel de escritura.
6. **Panel de seña:** En este panel se visualizan las señas que el usuario realice a través del *Leap Motion Controller*.



Figura 4.23 Interfaz de usuario SignMx

4.7 Generación de texto a voz

Se utilizó la librería para generar en audio el texto que generan las señas que son reconocidas. Esta librería llamada **Responsive Voice** se encuentra de forma gratuita y se encuentra disponible para descargar desde su sitio oficial. <http://responsivevoice.org/>

Para la versión gratuita simplemente es necesario copiar el código archivo HTML que será llamado cuando se encuentre activo.

```
<script src='https://code.responsivevoice.org/responsivevoice.js'></script>
```

Compatibilidad

ResponsiveVoice es compatible de forma nativa en Chrome para escritorio, Safari e iOS (Figura 4.24 Compatibilidad . Chrome en Android tiene soporte en el sistema operativo, pero este soporte no es accesible en el navegador, por lo cual *ResponsiveVoice* usa la voz de reserva española. Esta voz de reserva se utiliza también en navegadores que no soportan de manera nativa la síntesis de voz, como Internet Explorer, Firefox y Opera (Responsive Voice, 2016).

 Chrome	 Safari	 Internet Explorer	 Firefox	 Opera	 Android	 Apple iOS
						
Native	Native	Audio Stream	Audio Stream	Audio Stream	Native	Native

Figura 4.24 Compatibilidad (*Responsive Voice*, 2016)

Calidad de Voz

- La voz principal es femenina ("*Spanish Female*") y es compatible con Chrome para escritorio, Safari para OSX, y Chrome y Safari para iOS.
- La voz de reserva es femenina y se utiliza en el navegador por defecto de Android, Internet Explorer, Firefox y Opera.

Las dos voces femeninas suenan muy similar, por lo que los usuarios pueden tener una experiencia consistente en plataformas que gozan del soporte nativo, sino también en aquellas que no lo tienen.

4.8 Pruebas de la aplicación.

Cuando la aplicación fue terminada se realizaron pruebas para probar cada una de las funciones que esta contendría como, el reconocimiento correcto de las señas, que el significado de la seña realizada fuera la que se escribiera, la reproducción correcta del texto escrito, borrado de letras, etc.

Se realizaron pruebas del reconocimiento de la mano en 3D y que el significado de la seña realizada correspondiera a un elemento de la lengua de señas, por ejemplo en la Figura 4.25 se realizó la seña de la letra "a" para corroborar que la seña en 3D se mostrara dentro del panel de seña y que la letra se escribiera dentro el panel de escritura.



Figura 4.25 Prueba de la aplicación, seña de la letra "a"

De igual manera se realizó para concatenar más letras como se puede apreciar en la Figura 4.26.



Figura 4.26 Prueba de la aplicación, seña de la letra "b"

En la Figura 4.27 se muestra información de salida de los datos, esta parte se realizó con la finalidad para corroborar que la aplicación si se encuentre reconociendo las señas, porque existían casos en lo que no se mostraban la mano en 3D pero se escribía su significado o en caso contrario se reconocía la mano en 3D pero no se escribía el significado o daba un significado que no correspondía.

```

THREE.WebGLRenderer 71 three.js:17834
Optimized for desktop usage. leap-0.6.4.js:222
ResponsiveVoice r3 responsive-voice.js:19
***** ENTRO A BORRAR - ESPACIO - LIMPIAR ***** senasNumeros.js:5
***** ENTRO A NUMEROS ***** senasNumeros.js:36
***** ENTRO AL ABECEDARIO ***** senasNumeros.js:115
***** ENTRO AL DIAS DE LA SEMANA ***** senasNumeros.js:281
RV: Voice support ready responsive-voice.js:240
false senasNumeros.js:243
Función V ----- Entro al 2 ----- senasNumeros.js:244
2 senasNumeros.js:253
false senasNumeros.js:125
Función B ----- Entro al 4 ----- senasNumeros.js:126
4 senasNumeros.js:135
false senasNumeros.js:125
Función B ----- Entro al 4 ----- senasNumeros.js:126
4 senasNumeros.js:135
2 ----- Entro al D ----- senasNumeros.js:148
2 ----- Entro al 1 ----- senasNumeros.js:40
----- Entro al A ----- senasNumeros.js:120
A senasNumeros.js:243
true senasNumeros.js:244
Función V ----- Entro al V----- senasNumeros.js:248
V senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:68
true senasNumeros.js:69
false senasNumeros.js:244
Función V ----- Entro al 2 ----- senasNumeros.js:253
2 senasNumeros.js:125
false senasNumeros.js:126
Función B ----- Entro al 4 ----- senasNumeros.js:135
4 senasNumeros.js:125
false senasNumeros.js:126
Función B ----- Entro al 4 ----- senasNumeros.js:135
2 ----- Entro al D ----- senasNumeros.js:148
2 ----- Entro al 1 ----- senasNumeros.js:40
----- Entro al A ----- senasNumeros.js:120
A senasNumeros.js:243
true senasNumeros.js:244
Función V ----- Entro al V----- senasNumeros.js:248
V senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:125
true senasNumeros.js:126
Función B ----- Entro al B----- senasNumeros.js:130
B senasNumeros.js:68
true senasNumeros.js:69
Función B ----- Entro al B----- senasNumeros.js:73
B senasNumeros.js:68
true senasNumeros.js:69
Función B ----- Entro al B----- senasNumeros.js:73
2 ----- Entro al Y ----- senasNumeros.js:272
----- Entro al limpiar ----- senasNumeros.js:32
----- Entro al borrar ----- senasNumeros.js:15

```

Figura 4.27 Información en consola del navegador

Por otro lado se realizaron pruebas con el *framework LeapTrainer*, con la finalidad de comprobar el porcentaje de similitud de la seña realizada y la seña almacenada.

En la Figura 4.28 se muestra el porcentaje de similitud de la letra “a”, mientras que en la Figura 4.29 se muestra el porcentaje de similitud de la letra “d” y así se realizaron para cada una de las señas almacenadas.

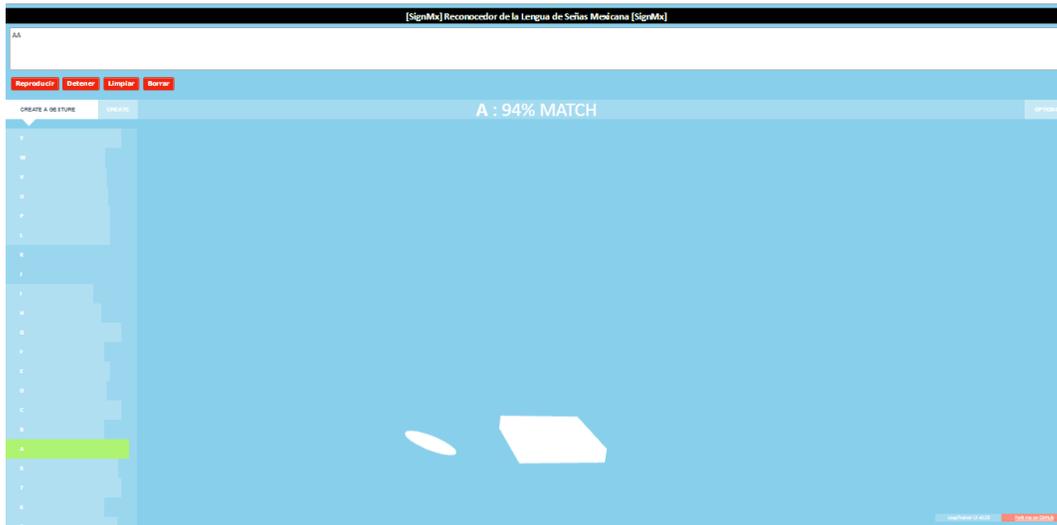


Figura 4.28 Prueba con el LeapTrainer, seña de la letra "a"



Figura 4.29 Prueba con el LeapTrainer, seña de la letra "d"

Capítulo 5.

Pruebas y resultados

En este capítulo se presentan las pruebas que fueron realizadas para valorar la metodología de solución y de igual manera se muestran los resultados obtenidos.

5.1 Introducción a las pruebas

La aplicación está diseñada primordialmente para las personas que son sordomudas y por tal motivo las pruebas deben de ser realizadas por ellas, pero por diversos motivos no fueron realizadas por la comunidad sorda, en su caso la realizaron persona que desconocen la Lengua de Señas, aunque se debe mencionar que este trabajo no es limitativo a ese conglomerado de personas sordomudas, sino que sirve como apoyo de comunicación para todo individuo en general.

Las pruebas fueron diseñadas para medir el porcentaje de similitud entre la seña almacenada y la seña que se realiza frente al dispositivo *Leap Motion Controller*. Para poder medir dicho porcentaje se utilizó el *framework LeapTrainer.js* el cual se describió en el apartado 2.6 LeapTrainer.js.

Hay que considerar de igual manera las limitaciones que el *Leap Motion Controller* contiene, en vista que algunas señas propuestas para este trabajo de investigación no fueron reconocidas por el dispositivo, algunas de ellas son:

- Reconocimiento erróneo ante una seña con movimiento.
- Pérdida de visión cuando se juntan las yemas de los dedos.
- Pérdida de visión cuando se aproximan demasiado una mano con la otra.
- No reconoce gesto cuando transponen una mano encima de la otra.

Debido a estas limitaciones el universo de señas que se reconoció es limitado; las señas reconocidas son las que pertenecen a este trabajo de investigación y de igual manera fueron utilizadas para realizar las pruebas.

5.2 LeapTrainer.js

Como se describió en el apartado 2.6 LeapTrainer.js, es un *framework* para el *Leap Motion Controller* el cual es usado para el reconocimiento y aprendizaje de gestos y poses (Figura 5.1). El *framework* soporta codificación de alta y baja resolución de gestos, comparación de plantillas geométricas, correlación cruzada y redes neuronales basadas en gestos para el reconocimiento de poses.

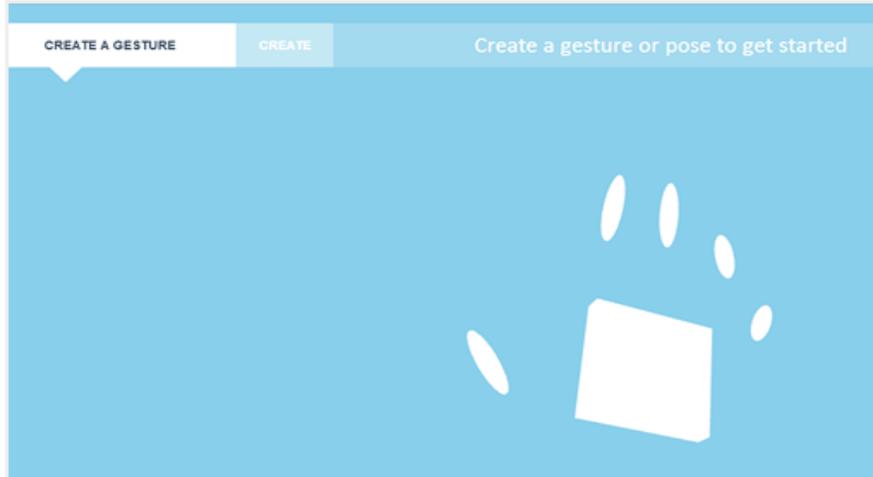


Figura 5.1 LeapTrainer.js (O'Leary, 2016)

Al grabar un gesto o pose, LeapTrainer.js proporciona un JSON de los datos ya grabados y entrenados, a su vez en su interfaz proporciona un porcentaje de reconocimiento del gesto o pose que se grabó (Figura 5.2), este porcentaje será usado en las pruebas para calcular el promedio de porcentaje de similitud de la seña realizada y así mismo obtener el promedio de porcentaje de similitud de la prueba en general.

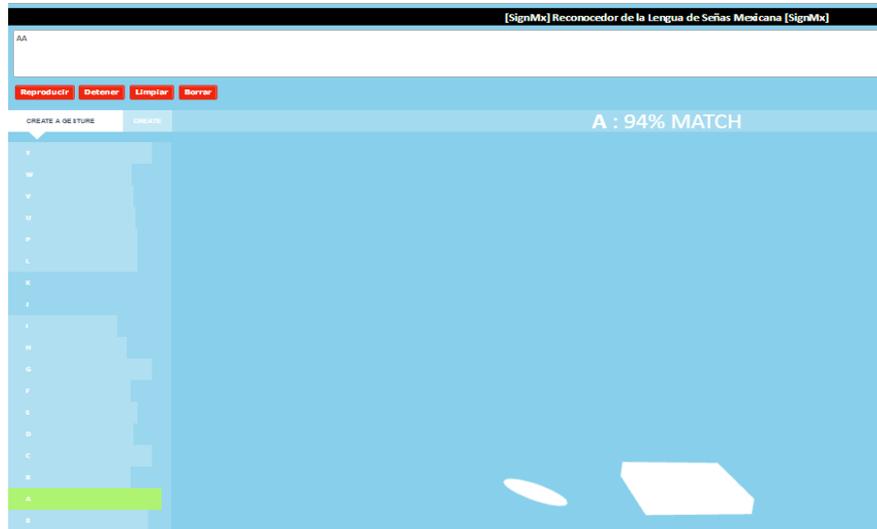


Figura 5.2 Porcentaje de reconocimiento LeapTrainer.js (O'Leary, 2016)

5.3 Pruebas

Las pruebas realizadas tuvieron como objetivo evaluar el rendimiento del sistema, en específico medir el porcentaje de similitud entre la seña almacenada y la seña que se realiza frente al dispositivo *Leap Motion Controller*.

5.3.1 Descripción de la prueba

Las pruebas se realizaron con ayuda del *framework LeapTrainer* debido a que éste nos proporciona el porcentaje de similitud entre la seña que realiza el usuario y la seña que se encuentra almacenada dentro de la aplicación.

Las pruebas se dividieron en tres tipos:

1. Prueba número.
2. Prueba abecedario.
3. Prueba palabras.

El proceso para la realización de las pruebas consistió en:

- Ofrecerle a los usuarios una breve introducción del tema para ponerlos en el contexto de lo que se espera realice durante la prueba.
- Demostrar como realiza las señas una persona sordomuda.
- Enseñar el modo de realizar las señas para que el dispositivo las reconozca.

Después de conocer la forma en que se realizaban las señas, el participante realizó diversos intentos en el *framework LeapTrainer* y únicamente los tres porcentajes de similitud para la seña realizada, en ese momento fueron los que se tomaron en cuenta, en dado caso que el participante obtuviera una seña errónea se optó por colocar un 0% de reconocimiento en la prueba, este mismo criterio se aplicó para los otros intentos y para las otras pruebas.

Es importante mencionar que las señas contenidas en cada prueba estarán incluidas en la aplicación final *SignMx*.

5.3.2 Prueba número

La prueba número especifica aquellas señas que la aplicación final *SignMx* reconoce; en las celdas de intentos se escribió el porcentaje de similitud de la seña, el cual es proporcionado por *LeapTrainer*, al final de cada seña se obtuvo un promedio por seña y de igual manera un promedio general de todas las señas de la prueba en ejecución (Figura 5.3). Estas señas fueron propuestas en el reporte de avances del 3er semestre, no todas las señas fueron reconocidas por las limitaciones del *Leap Motion Controller*.

Números	Intentos			Total
	1	2	3	
1				
2				
3				
4				
5				
6				
7				
8				

Figura 5.3 Prueba números

5.3.3 Prueba abecedario

La prueba abecedario especifica aquellas letras del abecedario que la aplicación final *SignMx* reconoce; en las celdas de intentos se escribió el porcentaje de similitud de la seña, el cual es proporcionado por *LeapTrainer*, al final de cada seña se obtuvo un promedio por seña y de igual manera un promedio general de todas las señas de la prueba en ejecución (). Estas señas fueron propuestas en el reporte de avances del 3er semestre, no todas las señas fueron reconocidas por las limitaciones del *Leap Motion Controller*.

ABC	Intentos			Total
	1	2	3	
A				
B				
C				
D				
E				
F				
G				
H				
I				
J				
K				
L				
P				
U				
V				
W				
Y				

Figura 5.4 Prueba abecedario

5.3.4 Prueba palabras

La prueba palabras especifica los elementos que la aplicación final *SignMx* reconoce; en las celdas de intentos se colocaron el porcentaje de similitud de la seña, el cual es proporcionado por *LeapTrainer*, al final de cada seña se obtuvo un promedio por seña y de igual manera un promedio general de todas las señas de la prueba en ejecución (Figura 5.5). Estas señas fueron propuestas en el reporte de avances del 4to semestre, no todas las señas fueron reconocidas por las limitaciones del *Leap Motion Controller*.

Palabras	Intentos			Total
	1	2	3	
Mano				
Torta				
Vaca				
Abajo				
Arriba				
Menos				
Casa				
Comal				
Copa				
Hogar				
Pay				
Escuadra				
Suma				
Cereza				
Raíz Cuadrada				
Pistola				
Almendra				
División				
Multiplicación				
Sostener				

Figura 5.5 Prueba palabras

5.3.5 Tipos de prueba

Las pruebas fueron divididas por dos tipos: **No supervisado** y **Supervisado**.

- **No supervisado:** En esta prueba se consideró un grupo formado por una mujer y cinco hombres. Se les compartió una pequeña introducción sobre la aplicación que se desarrolló y su funcionamiento; de igual manera se les explicó de forma rápida y sencilla como se realizan la prueba y como realizan las señas las personas sordomudas.
- **Supervisado:** En esta prueba se consideró un grupo formado por una mujer y cinco hombres. Se proporcionó una atención personal a cada miembro del grupo, a su vez, se compartió una pequeña introducción sobre la aplicación y se les explicó cómo realiza las señas una persona sordomuda y como deben de llevar a cabo las señas cuando utilicen el *Leap Motion Controller*.

La finalidad de tener estos dos tipos fue:

- Obtener el nivel de reconocimiento de las señas.
- Evaluar la manera de realizar las señas ante el *Leap Motion Controller*.
- Que tan fácil o difícil fueron realizar las señas ante la aplicación final.
- Dificultades o problemas presentados.
- Comprensión y entendimiento de la interfaz *SignMx*.
- Conclusiones finales de cada usuario.

5.3.6 Ambiente para realizar la prueba

El ambiente en la cual se realizaron las pruebas fueron las siguientes.

- **No supervisado:** Las pruebas fueron realizadas dentro de un aula de clase con la iluminación artificial.
- **Supervisado:** Algunas de las pruebas fueron realizadas dentro de una habitación de una casa con iluminación artificial y una prueba fue realizada al aire libre con iluminación natural, esto con la finalidad de observar la funcionalidad del *Leap Motion Controller* ante la exposición de un ambiente no controlado.

5.3.7 Población utilizada para las pruebas

La población de personas utilizada para la prueba fue de un total de 12 personas de las cuales 2 fueron mujeres y 10 hombres, a continuación se presentan dos tablas de los grupos no supervisado Tabla 5.1 y supervisado Tabla 5.2.

- **Clave:** Clave compuesta para identificar a cada persona.
 - Grupo al que pertenece (NS = No supervisado, S = Supervisado)
 - Género del participante (F = Femenino, M = Masculino)
 - Número de persona (1,2,3,4,5,...,n)
- **Sexo:** Género del participante para determinar la población de hombres y mujeres que realizaron la prueba.
- **Edad:** Edad del participante para determinar el rango de edad que realizaron la prueba.
- **Escolaridad:** Escolaridad del participante para determinar la escolaridad promedio que realizaron la prueba.

No supervisado

Tabla 5.1 Tabla grupo no supervisado

Clave	Sexo	Edad	Escolaridad
NS_F_1	Femenino	24	Maestría
NS_M_2	Masculino	25	Maestría
NS_M_3	Masculino	26	Maestría
NS_M_4	Masculino	25	Maestría
NS_M_5	Masculino	33	Maestría
NS_M_6	Masculino	36	Maestría

Supervisado

Tabla 5.2 Tabla grupo supervisado

Clave	Sexo	Edad	Escolaridad
S_M_1	Masculino	25	Maestría
S_M_2	Masculino	17	Preparatoria
S_M_3	Masculino	21	Licenciatura
S_M_4	Masculino	25	Maestría
S_M_5	Masculino	25	Maestría
S_F_6	Femenino	24	Licenciatura

5.4 Resultados

Los resultados obtenidos se muestran por tipos de prueba y grupo, primeramente aparecerán los del grupo **no supervisado** y posteriormente los **supervisado**, para luego finalizar con el análisis de los todos los datos recolectados.

5.4.1 Grupo no supervisado

5.4.1.1 NS_F_1

Resultados obtenidos de la persona NS_F_1 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.3).

Tabla 5.3 Resultados del grupo no supervisada femenina 1

Números	Intentos			TOTAL
	1	2	3	
1	90	92	89	90.33
2	89	94	88	90.33
3	92	90	89	90.33
4	89	94	93	92.00
5	90	88	92	90.00
6	89	79	84	84.00
7	88	84	84	85.33
8	83	86	84	84.33
Promedio Prueba:				88.33

#	abc	Intentos			Total
		1	2	3	
1	A	88	92	87	89.00
2	B	82	83	86	83.67
3	C	95	94	96	95.00
4	D	93	92	94	93.00
5	E	0	0	0	0.00
6	F	88	88	87	87.67
7	G	92	91	95	92.67
8	H	88	84	0	57.33
9	I	86	88	93	89.00
10	J	77	73	72	74.00
12	L	98	92	95	95.00
13	P	0	0	0	0.00
14	U	95	0	0	31.67
15	V	0	95	0	0.00
16	W	0	0	0	0.00
17	Y	94	89	0	61.00
Promedio Prueba:					59.31

#	Palabras	Intentos			Total
		1	2	3	
1	Mano	92	92	93	92.33
2	Torta	82	82	0	54.67
3	Vaca	95	92	87	91.33
4	Abajo	88	97	97	94.00
5	Arriba	0	0	0	0.00
6	Menos	91	91	90	90.67
7	Casa	0	0	0	0.00
8	Comal	0	0	0	0.00
9	Copa	92	90	88	90.00
10	Hogar	0	0	0	0.00
11	Pay	91	95	88	91.33
12	Escuadra	93	0	0	31.00
13	Suma	88	91	89	89.33
14	Cereza	96	92	86	91.33
15	Raíz Cuadrada	0	0	0	0.00
16	Pistola	87	94	88	89.67
17	Almendra	92	96	98	95.33
18	División	87	0	0	29.00
19	Multipliación	87	0	0	29.00
20	Sostener	88	0	0	29.33
Promedio Prueba:					54.42

5.4.1.2 NS_M_2

Resultados obtenidos de la persona NS_M_2 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5. 4).

Tabla 5. 4 Resultados del grupo no supervisada masculino 1

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	90	89	88	89.00	1	A	88	89	89	88.67	1	Mano	90	89	91	90.00
2	88	90	90	89.33	2	B	91	91	87	89.67	2	Torta	0	0	0	0.00
3	92	89	89	90.00	3	C	0	0	90	30.00	3	Vaca	91	93	87	90.33
4	93	94	94	93.67	4	D	92	90	89	90.33	4	Abajo	97	97	97	97.00
5	91	90	92	91.00	5	E	75	87	81	81.00	5	Arriba	95	96	89	93.33
6	84	84	95	87.67	6	F	91	89	94	91.33	6	Menos	97	94	95	95.33
7	89	90	89	89.33	7	G	0	0	0	0.00	7	Casa	0	0	0	0.00
8	85	88	88	87.00	8	H	93	85	86	88.00	8	Comal	0	0	0	0.00
Promedio Prueba:				89.63	9	I	87	97	88	90.67	9	Copa	88	0	0	29.33
					10	J	70	72	77	73.00	10	Hogar	0	0	0	0.00
					12	L	89	92	94	91.67	11	Pay	88	95	94	92.33
					13	P	89	86	85	86.67	12	Escuadra	0	0	88	29.33
					14	U	0	0	0	0.00	13	Suma	0	0	0	0.00
					15	V	94	96	95	95.00	14	Cereza	92	87	87	88.67
					16	W	98	0	0	32.67	15	Raíz Cuadrada	90	92	93	91.67
					17	Y	86	92	94	90.67	16	Pistola	85	85	86	85.33
					Promedio Prueba:				69.96	17	Almendra	83	82	83	82.67	
										18	División	0	0	0	0.00	
										19	Multiplicación	88	89	89	88.67	
										20	Sostener	93	0	0	31.00	
										Promedio Prueba:				53.22		

5.4.1.3 NS_M_3

Resultados obtenidos de la persona NS_M_3 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.5).

Tabla 5.5 Resultados del grupo no supervisada masculino 2

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	0	0	0	0.00	1	A	96	95	90	93.67	1	Mano	91	92	87	90.00
2	0	83	87	56.67	2	B	98	98	0	65.33	2	Torta	0	0	0	0.00
3	0	91	91	60.67	3	C	0	0	0	0.00	3	Vaca	0	0	0	0.00
4	94	93	94	93.67	4	D	91	92	93	92.00	4	Abajo	83	86	86	85.00
5	90	94	90	91.33	5	E	86	88	0	58.00	5	Arriba	89	90	85	88.00
6	81	0	0	27.00	6	F	0	90	0	30.00	6	Menos	95	97	93	95.00
7	87	0	91	59.33	7	G	0	0	0	0.00	7	Casa	0	0	0	0.00
8	83	86	85	84.67	8	H	0	0	0	0.00	8	Comal	81	83	0	54.67
Promedio Prueba:				59.17	9	I	0	0	0	0.00	9	Copa	0	0	0	0.00
					10	J	0	0	0	0.00	10	Hogar	0	0	0	0.00
					12	L	94	96	95	95.00	11	Pay	0	0	0	0.00
					13	P	87	0	0	29.00	12	Escuadra	0	0	0	0.00
					14	U	0	0	0	0.00	13	Suma	92	92	92	92.00
					15	V	94	94	93	93.67	14	Cereza	83	84	0	55.67
					16	W	98	98	98	98.00	15	Raíz Cuadrada	92	0	0	30.67
					17	Y	0	0	0	0.00	16	Pistola	0	0	0	0.00
					Promedio Prueba:				40.92	17	Almendra	92	93	92	92.33	
										18	División	0	0	0	0.00	
										19	Multiplicación	0	0	86	28.67	
										20	Sostener	0	0	0	0.00	
										Promedio Prueba:				36.59		

5.4.1.4 NS_M_4

Resultados obtenidos de la persona NS_M_4 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.6).

Tabla 5.6 Resultados del grupo no supervisada masculino 3

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	88	83	86	85.67	1	A	90	92	86	89.33	1	Mano	0	0	0	0.00
2	88	89	0	59.00	2	B	81	81	81	81.00	2	Torta	0	0	0	0.00
3	86	83	87	85.33	3	C	0	0	0	0.00	3	Vaca	89	0	0	29.67
4	84	88	91	87.67	4	D	88	92	92	90.67	4	Abajo	79	87	86	84.00
5	89	89	89	89.00	5	E	88	92	92	90.67	5	Arriba	88	89	87	88.00
6	84	83	81	82.67	6	F	88	89	88	88.33	6	Menos	90	0	0	30.00
7	89	86	85	86.67	7	G	88	0	0	29.33	7	Casa	0	0	0	0.00
8	88	89	88	88.33	8	H	0	0	0	0.00	8	Comal	0	80	80	53.33
Promedio Prueba:				83.04	9	I	0	0	0	0.00	9	Copa	0	0	0	0.00
					10	J	66	75	68	69.67	10	Hogar	0	0	0	0.00
					12	L	94	96	95	95.00	11	Pay	0	0	0	0.00
					13	P	87	0	0	29.00	12	Escuadra	89	89	91	89.67
					14	U	90	94	93	92.33	13	Suma	89	91	89	89.67
					15	V	93	94	0	62.33	14	Cereza	98	90	95	94.33
					16	W	0	92	0	30.67	15	Raíz Cuadrada	81	0	0	27.00
					17	Y	0	0	0	0.00	16	Pistola	0	89	0	29.67
					Promedio Prueba:					53.02	17	Almendra	91	85	88	88.00
											18	División	0	0	0	0.00
											19	Multiplicación	0	0	0	0.00
											20	Sostener	89	90	87	88.67
											Promedio Prueba:			44.84		

5.4.1.5 NS_M_5

Resultados obtenidos de la persona NS_M_5 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.7).

Tabla 5.7 Resultados del grupo no supervisada masculino 4

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	94	93	93	93.33	1	A	91	90	92	91.00	1	Mano	92	92	91	91.67
2	91	90	90	90.33	2	B	94	91	92	92.33	2	Torta	84	84	83	83.67
3	85	96	94	91.67	3	C	0	0	0	0.00	3	Vaca	89	93	91	91.00
4	85	87	86	86.00	4	D	88	93	90	90.33	4	Abajo	96	98	92	95.33
5	91	90	90	90.33	5	E	0	0	0	0.00	5	Arriba	89	83	87	86.33
6	93	84	80	85.67	6	F	0	0	0	0.00	6	Menos	0	0	0	0.00
7	91	90	88	89.67	7	G	94	94	93	93.67	7	Casa	0	0	0	0.00
8	90	92	95	92.33	8	H	0	0	0	0.00	8	Comal	88	88	88	88.00
Promedio Prueba:				89.92	9	I	85	86	85	85.33	9	Copa	0	0	0	0.00
					10	J	72	68	78	72.67	10	Hogar	0	0	0	0.00
					12	L	93	89	89	90.33	11	Pay	0	0	0	0.00
					13	P	86	0	0	28.67	12	Escuadra	89	89	90	89.33
					14	U	0	0	0	0.00	13	Suma	84	85	84	84.33
					15	V	0	0	0	0.00	14	Cereza	93	92	95	93.33
					16	W	0	0	0	0.00	15	Raíz Cuadrada	0	0	0	0.00
					17	Y	96	95	0	63.67	16	Pistola	0	0	0	0.00
					Promedio Prueba:					44.25	17	Almendra	91	94	93	92.67
											18	División	88	89	92	89.67
											19	Multiplicación	0	0	0	0.00
											20	Sostener	0	0	0	0.00
											Promedio Prueba:			42.29		

5.4.1.6 NS_M_6

Resultados obtenidos de la persona NS_M_6 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.8).

Tabla 5.8 Resultados del grupo no supervisada masculino 5

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	90	89	93	90.67	1	A	94	93	88	91.67	1	Mano	92	93	93	92.67
2	94	94	90	92.67	2	B	90	89	90	89.67	2	Torta	78	79	73	76.67
3	94	90	91	91.67	3	C	91	90	91	90.67	3	Vaca	96	91	92	93.00
4	92	94	0	62.00	4	D	95	96	90	93.67	4	Abajo	89	84	95	89.33
5	95	91	92	92.67	5	E	80	79	79	79.33	5	Arriba	84	83	84	83.67
6	79	88	96	87.67	6	F	91	91	87	89.67	6	Menos	94	88	83	88.33
7	88	88	89	88.33	7	G	93	92	92	92.33	7	Casa	83	82	0	55.00
8	94	92	93	93.00	8	H	96	83	84	87.67	8	Comal	82	82	87	83.67
Promedio Prueba:				87.33	9	I	89	81	80	83.33	9	Copa	86	81	82	83.00
					10	J	67	68	68	67.67	10	Hogar	0	0	0	0.00
					12	L	95	89	95	93.00	11	Pay	0	0	0	0.00
					13	P	87	88	0	58.33	12	Escuadra	82	82	87	83.67
					14	U	87	88	87	87.33	13	Suma	90	90	92	90.67
					15	V	93	94	95	94.00	14	Cereza	82	89	94	88.33
					16	W	99	98	99	98.67	15	Raíz Cuadrada	0	0	0	0.00
					17	Y	88	87	87	87.33	16	Pistola	89	91	88	89.33
					Promedio Prueba:				86.52	17	Almendra	94	94	97	95.00	
										18	División	84	88	91	87.67	
										19	Multiplicación	88	91	92	90.33	
										20	Sostener	88	0	0	29.33	
										Promedio Prueba:				66.90		

5.4.2 Grupo supervisado

5.4.2.1 S_M_1

Resultados obtenidos de la persona S_M_1 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.9).

Tabla 5.9 Resultados del grupo supervisado masculino 1

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total	
	1	2	3				1	2	3				1	2	3		
1	94	98	95	95.67	1	A	92	94	98	94.67	1	Mano	91	88	90	89.67	
2	93	94	93	93.33	2	B	93	93	92	92.67	2	Torta	0	0	0	0.00	
3	89	93	95	92.33	3	C	88	87	87	87.33	3	Vaca	90	89	89	89.33	
4	96	98	96	96.67	4	D	94	92	90	92.00	4	Abajo	78	84	83	81.67	
5	90	90	93	91.00	5	E	86	85	87	86.00	5	Arriba	0	0	0	0.00	
6	95	93	95	94.33	6	F	90	92	90	90.67	6	Menos	92	95	89	92.00	
7	90	88	87	88.33	7	G	96	96	93	95.00	7	Casa	0	0	0	0.00	
8	93	90	89	90.67	8	H	85	90	96	90.33	8	Comal	88	93	87	89.33	
				Promedio Prueba:	92.79	9	I	92	88	90	90.00	9	Copa	0	0	0	0.00
						10	J	71	71	69	70.33	10	Hogar	0	0	0	0.00
						11	L	92	90	90	90.67	11	Pay	0	0	0	0.00
						12	P	88	88	90	88.67	12	Escuadra	85	84	88	85.67
						13	U	94	94	94	94.00	13	Suma	93	89	91	91.00
						14	V	96	94	94	94.67	14	Cereza	90	89	91	90.00
						15	W	98	0	0	32.67	15	Raíz Cuadrada	89	88	89	88.67
						16	Y	96	93	98	95.67	16	Pistola	89	88	88	88.33
										Promedio Prueba:	86.58	17	Almendra	93	91	93	92.33
												18	División	82	89	90	87.00
												19	Multiplicación	0	0	0	0.00
												20	Sostener	0	0	0	0.00
																Promedio Prueba:	52.12

5.4.2.2 S_M_2

Resultados obtenidos de la persona S_M_2 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.10).

Tabla 5.10 Resultados del grupo supervisado masculino 2

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total	
	1	2	3				1	2	3				1	2	3		
1	88	90	90	89.33	1	A	97	96	94	95.67	1	Mano	90	89	90	89.67	
2	91	90	92	91.00	2	B	93	92	93	92.67	2	Torta	0	0	0	0.00	
3	90	94	92	92.00	3	C	92	93	92	92.33	3	Vaca	92	88	90	90.00	
4	96	96	97	96.33	4	D	91	91	90	90.67	4	Abajo	88	87	90	88.33	
5	94	97	95	95.33	5	E	89	88	89	88.67	5	Arriba	90	90	88	89.33	
6	92	97	95	94.67	6	F	93	91	91	91.67	6	Menos	94	91	91	92.00	
7	93	93	93	93.00	7	G	95	93	98	95.33	7	Casa	0	0	0	0.00	
8	90	90	91	90.33	8	H	92	91	91	91.33	8	Comal	86	88	90	88.00	
				Promedio Prueba:	92.75	9	I	90	86	85	87.00	9	Copa	89	90	86	88.33
						10	J	71	68	65	68.00	10	Hogar	90	94	95	93.00
						11	L	88	85	87	86.67	11	Pay	0	0	0	0.00
						12	P	95	90	93	92.67	12	Escuadra	88	90	91	89.67
						13	U	95	95	96	95.33	13	Suma	90	91	90	90.33
						14	V	93	93	91	92.33	14	Cereza	92	90	91	91.00
						15	W	90	88	90	89.33	15	Raíz Cuadrada	93	90	94	92.33
						16	Y	94	94	94	94.00	16	Pistola	90	90	92	90.67
										Promedio Prueba:	90.23	17	Almendra	91	90	93	91.33
												18	División	92	93	90	91.67
												19	Multiplicación	0	0	0	0.00
												20	Sostener	0	0	0	0.00
																Promedio Prueba:	69.18

5.4.2.3 S_M_3

Resultados obtenidos de la persona S_M_3 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.11).

Tabla 5.11 Resultados del grupo supervisado masculino 3

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	93	89	89	90.33	1	A	95	96	98	96.33	1	Mano	91	90	92	91.00
2	91	92	94	92.33	2	B	92	94	93	93.00	2	Torta	0	0	0	0.00
3	94	94	95	94.33	3	C	90	90	89	89.67	3	Vaca	90	92	94	92.00
4	91	91	90	90.67	4	D	88	89	89	88.67	4	Abajo	97	94	92	94.33
5	88	87	89	88.00	5	E	87	84	88	86.33	5	Arriba	88	89	89	88.67
6	82	88	92	87.33	6	F	88	93	90	90.33	6	Menos	98	97	95	96.67
7	93	92	91	92.00	7	G	94	92	95	93.67	7	Casa	0	0	0	0.00
8	96	88	89	91.00	8	H	89	86	85	86.67	8	Comal	96	89	90	91.67
Promedio Prueba:				90.75	9	I	90	88	89	89.00	9	Copa	89	88	88	88.33
					10	J	67	70	70	69.00	10	Hogar	96	95	94	95.00
					12	L	85	88	85	86.00	11	Pay	93	0	0	31.00
					13	P	89	84	87	86.67	12	Escuadra	92	90	90	90.67
					14	U	97	94	92	94.33	13	Suma	95	93	94	94.00
					15	V	95	96	95	95.33	14	Cereza	92	89	90	90.33
					16	W	90	95	90	91.67	15	Raíz Cuadrada	91	85	90	88.67
					17	Y	88	96	92	92.00	16	Pistola	88	87	87	87.33
					Promedio Prueba:				89.29	17	Almendra	94	97	90	93.67	
										18	División	87	90	86	87.67	
										19	Multiplicación	0	0	0	0.00	
										20	Sostener	0	0	0	0.00	
										Promedio Prueba:				71.65		

5.4.2.4 S_M_4

Resultados obtenidos de la persona S_M_4 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5. 12).

Tabla 5. 12 Resultados del grupo supervisado masculino 4

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	94	90	89	91.00	1	A	93	90	90	91.00	1	Mano	90	89	88	89.00
2	90	90	95	91.67	2	B	92	91	85	89.33	2	Torta	0	0	0	0.00
3	96	97	95	96.00	3	C	93	92	90	91.67	3	Vaca	90	87	88	88.33
4	84	95	94	91.00	4	D	95	96	94	95.00	4	Abajo	80	78	81	79.67
5	90	90	90	90.00	5	E	90	85	88	87.67	5	Arriba	88	86	85	86.33
6	93	90	87	90.00	6	F	90	87	89	88.67	6	Menos	89	88	88	88.33
7	87	90	90	89.00	7	G	91	95	96	94.00	7	Casa	92	0	0	30.67
8	95	92	91	92.67	8	H	87	85	86	86.00	8	Comal	91	84	85	86.67
Promedio Prueba:				91.42	9	I	90	92	89	90.33	9	Copa	92	88	88	89.33
					10	J	72	67	69	69.33	10	Hogar	95	94	90	93.00
					12	L	90	89	90	89.67	11	Pay	85	88	87	86.67
					13	P	89	86	86	87.00	12	Escuadra	88	93	92	91.00
					14	U	98	97	97	97.33	13	Suma	88	86	87	87.00
					15	V	97	98	98	97.67	14	Cereza	91	89	86	88.67
					16	W	98	92	92	94.00	15	Raíz Cuadrada	87	89	94	90.00
					17	Y	98	94	96	96.00	16	Pistola	83	92	95	90.00
					Promedio Prueba:				90.29	17	Almendra	94	95	95	94.67	
										18	División	91	93	92	92.00	
										19	Multiplicación	0	0	0	0.00	
										20	Sostener	0	0	0	0.00	
										Promedio Prueba:				74.94		

5.4.2.5 S_M_5

Resultados obtenidos de la persona S_M_5 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.13).

Tabla 5.13 Resultados del grupo supervisado masculino 5

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	89	85	93	89.00	1	A	95	92	95	94.00	1	Mano	88	88	92	89.33
2	93	93	93	93.00	2	B	89	90	89	89.33	2	Torta	0	0	0	0.00
3	93	93	91	92.33	3	C	92	88	91	90.33	3	Vaca	90	90	93	91.00
4	92	91	93	92.00	4	D	93	92	96	93.67	4	Abajo	87	87	87	87.00
5	86	88	95	89.67	5	E	90	89	86	88.33	5	Arriba	87	86	87	86.67
6	77	88	95	86.67	6	F	0	0	0	0.00	6	Menos	93	91	90	91.33
7	92	92	92	92.00	7	G	91	90	92	91.00	7	Casa	0	0	0	0.00
8	91	92	92	91.67	8	H	91	0	0	30.33	8	Comal	86	0	90	58.67
Promedio Prueba:				90.79	9	I	90	94	93	92.33	9	Copa	0	0	0	0.00
					10	J	78	76	75	76.33	10	Hogar	90	94	95	93.00
					12	L	88	87	88	87.67	11	Pay	0	86	0	28.67
					13	P	90	90	87	89.00	12	Escuadra	92	90	91	91.00
					14	U	94	94	94	94.00	13	Suma	86	94	91	90.33
					15	V	92	92	93	92.33	14	Cereza	91	91	93	91.67
					16	W	93	95	97	95.00	15	Raíz Cuadrada	95	95	91	93.67
					17	Y	87	92	93	90.67	16	Pistola	90	85	82	85.67
					Promedio Prueba:				80.90	17	Almendra	87	95	83	88.33	
										18	División	92	93	82	89.00	
										19	Multiplicación	0	0	0	0.00	
										20	Sostener	0	0	0	0.00	
										Promedio Prueba:				63.24		

5.4.2.6 S_F_1

Resultados obtenidos de la persona S_F_1 al realizar las señas frente al *Leap Motion Controller* y del valor de medición obtenido con el *framework LeapTrainer* (Tabla 5.14).

Tabla 5.14 Resultados del grupo supervisado femenina 1

Números	Intentos			TOTAL	#	abc	Intentos			Total	#	Palabras	Intentos			Total
	1	2	3				1	2	3				1	2	3	
1	88	95	95	92.67	1	A	87	92	91	90.00	1	Mano	89	84	88	87.00
2	91	92	90	91.00	2	B	94	89	90	91.00	2	Torta	0	0	0	0.00
3	98	95	96	96.33	3	C	0	0	0	0.00	3	Vaca	95	91	90	92.00
4	95	94	93	94.00	4	D	91	89	90	90.00	4	Abajo	83	85	81	83.00
5	92	90	91	91.00	5	E	93	88	84	88.33	5	Arriba	87	88	83	86.00
6	88	90	92	90.00	6	F	85	87	91	87.67	6	Menos	93	94	89	92.00
7	89	89	89	89.00	7	G	97	95	95	95.67	7	Casa	0	0	0	0.00
8	94	94	92	93.33	8	H	86	87	91	88.00	8	Comal	91	82	0	57.67
Promedio Prueba:				92.17	9	I	88	89	88	88.33	9	Copa	91	87	87	88.33
					10	J	66	66	67	66.33	10	Hogar	0	0	0	0.00
					12	L	91	88	90	89.67	11	Pay	96	97	93	95.33
					13	P	90	84	88	87.33	12	Escuadra	0	0	0	0.00
					14	U	97	92	97	95.33	13	Suma	85	0	0	28.33
					15	V	99	97	97	97.67	14	Cereza	87	86	88	87.00
					16	W	94	92	94	93.33	15	Raíz Cuadrada	96	95	93	94.67
					17	Y	97	88	88	91.00	16	Pistola	87	94	91	90.67
					Promedio Prueba:				83.73	17	Almendra	96	84	92	90.67	
										18	División	86	86	85	85.67	
										19	Multiplicación	0	0	0	0.00	
										20	Sostener	0	0	0	0.00	
										Promedio Prueba:				57.61		

5.4.3 Gráficas

Con los datos recolectados se obtuvieron las medias por cada prueba para luego ser categorizadas en un solo porcentaje por prueba y en general, considerando lo que realizó el participante.

5.4.3.1 No supervisado

En el promedio de números (Figura 5.6) se puede observar que cinco de los participantes del grupo logró obtener una buena realización de las señas debido a que sus valores se encuentra por encima del promedio general y únicamente uno de los participantes obtuvo una valor por debajo del promedio de 82.02.

PROMEDIOS NÚMEROS	
Persona 1	88.33
Persona 2	89.63
Persona 3	59.17
Persona 4	83.04
Persona 5	89.92
Persona 6	87.33
Promedio Gral	82.02

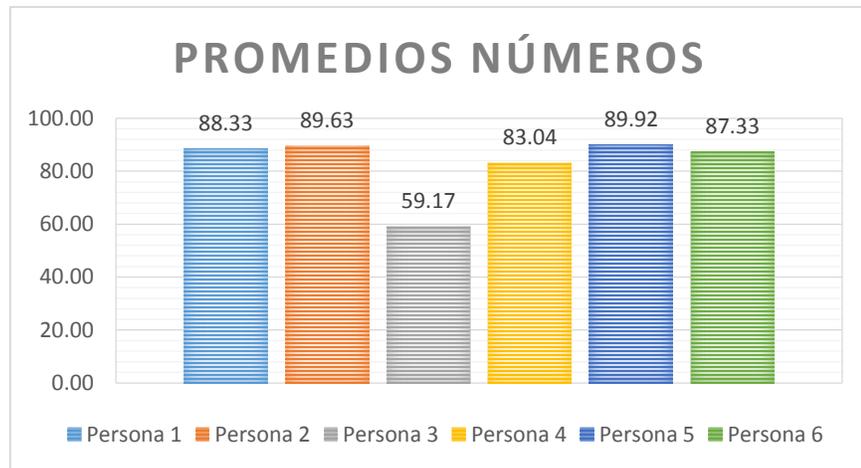


Figura 5.6 Promedio números – no supervisado

En el promedio de abecedario (Figura 5.7) los promedios por personas se encuentran variados, tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 53.49.

PROMEDIOS ABECEDARIO	
Persona 1	59.31
Persona 2	69.96
Persona 3	40.92
Persona 4	53.02
Persona 5	44.25
Persona 6	86.52
Promedio Gral	53.49

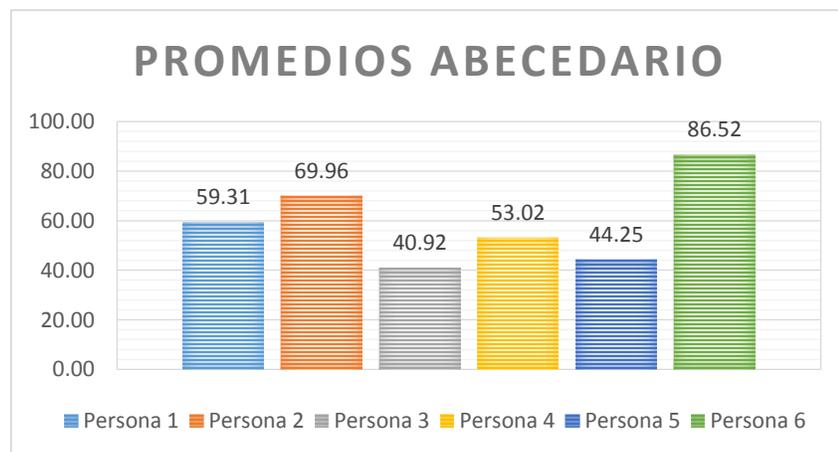


Figura 5.7 Promedio abecedario - no supervisado

En el promedio de palabras (Figura 5.8) los promedios por personas se encuentran variados, tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 46.27.

PROMEDIOS PALABRAS	
Persona 1	54.42
Persona 2	53.22
Persona 3	36.59
Persona 4	44.84
Persona 5	42.29
Persona 6	66.90
Promedio Gral	46.27

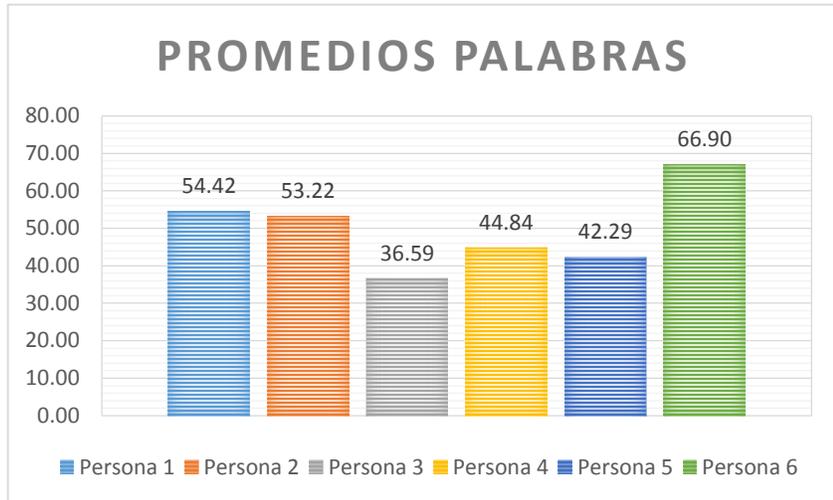


Figura 5.8 Promedio palabras - no supervisado

En el promedio de palabras (Figura 5.9) tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 60.59.

PROMEDIO GENERAL PRUEBA	
Persona 1	67.35
Persona 2	70.93
Persona 3	45.56
Persona 4	60.30
Persona 5	58.82
Persona 6	80.25
Promedio Gral	60.59

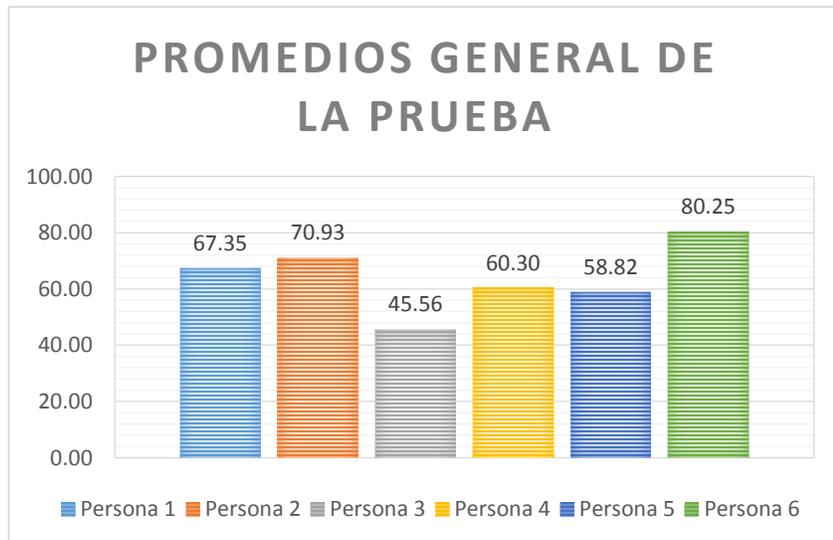


Figura 5.9 Promedio general de la prueba - no supervisado

5.4.3.2 Supervisado

En el promedio de números (Figura 5.10) tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 91.70, cabe denotar que los porcentaje obtenidos de esta prueba son muy aceptables.

PROMEDIOS NUMEROS	
Persona 1	92.79
Persona 2	92.75
Persona 3	90.75
Persona 4	91.42
Persona 5	90.79
Persona 6	92.17
Promedio Gral	91.70

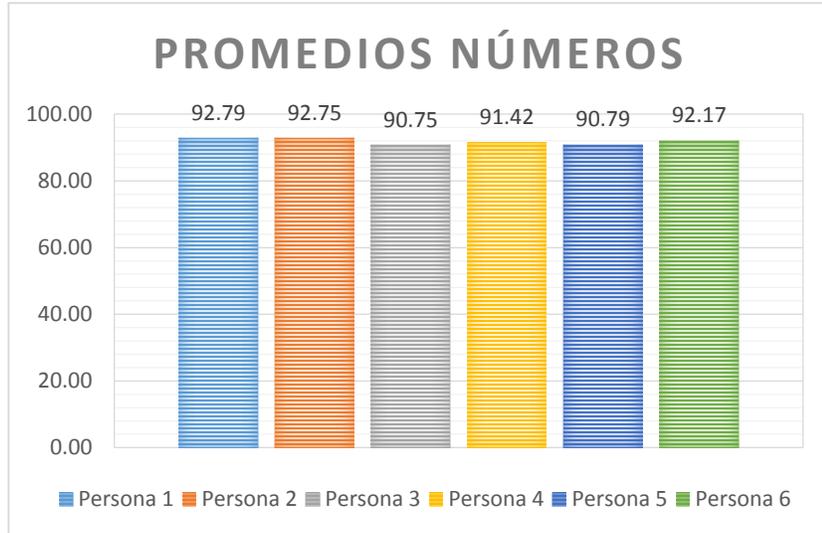


Figura 5.10 Promedio números - supervisado

En el promedio de abecedario (Figura 5.11) tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 87.46, de igual manera con la prueba de números el porcentaje de esta prueba es un buen valor obtenido.

PROMEDIOS ABECEDARIO	
Persona 1	86.58
Persona 2	90.23
Persona 3	89.29
Persona 4	90.29
Persona 5	80.90
Persona 6	83.73
Promedio Gral	87.46

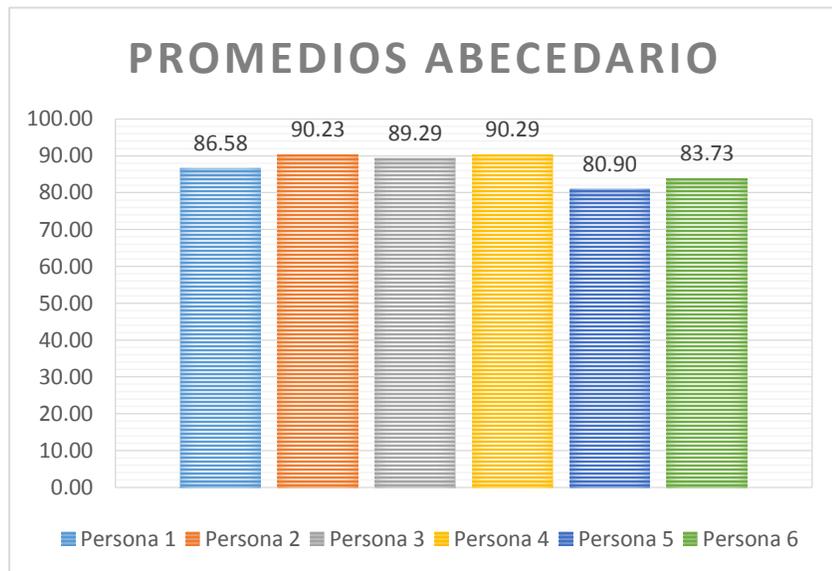


Figura 5.11 Promedio abecedario - supervisado

En el promedio de palabras (Figura 5.12) tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 66.22.

PROMEDIOS PALABRAS	
Persona 1	52.12
Persona 2	69.18
Persona 3	71.65
Persona 4	74.94
Persona 5	63.24
Persona 6	57.61
Promedio Gral	66.22

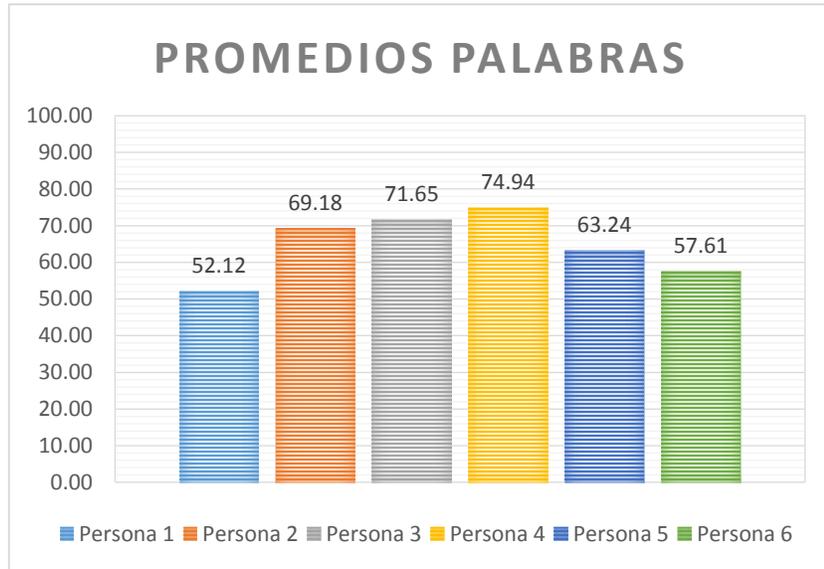


Figura 5.12 Promedio palabras - supervisado

En el promedio general de la prueba (Figura 5.13) tres de los participantes obtuvieron arriba del promedio de la prueba, mientras que los otros tres se quedaron debajo del promedio 81.70.

PROMEDIO GENERAL PRUEBA	
Persona 1	77.16
Persona 2	84.05
Persona 3	83.90
Persona 4	85.55
Persona 5	78.31
Persona 6	77.83
Promedio Gral	81.70

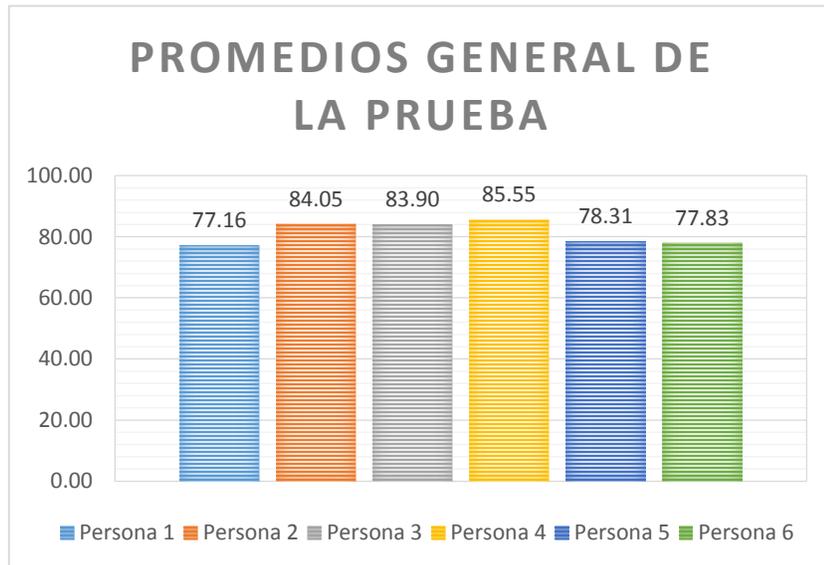


Figura 5.13 Promedio general de la prueba - supervisado

5.4.3.3 Comparación de resultados

Con los resultados obtenidos se compararon de acuerdo al nombre de la prueba realizada. Se compararon los resultados obtenidos en la prueba números (Figura 5.14) con las personas que pertenecen a cada grupo No supervisado y supervisado, de igual manera se realizaron para las pruebas abecedario (Figura 5.15) y la de palabras (Figura 5.16).

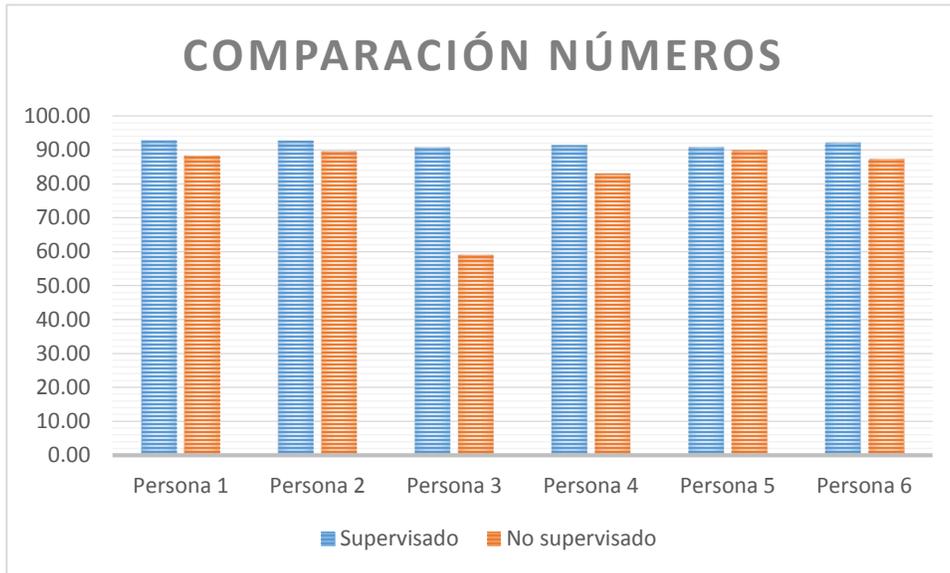


Figura 5.14 Comparación de resultados prueba números

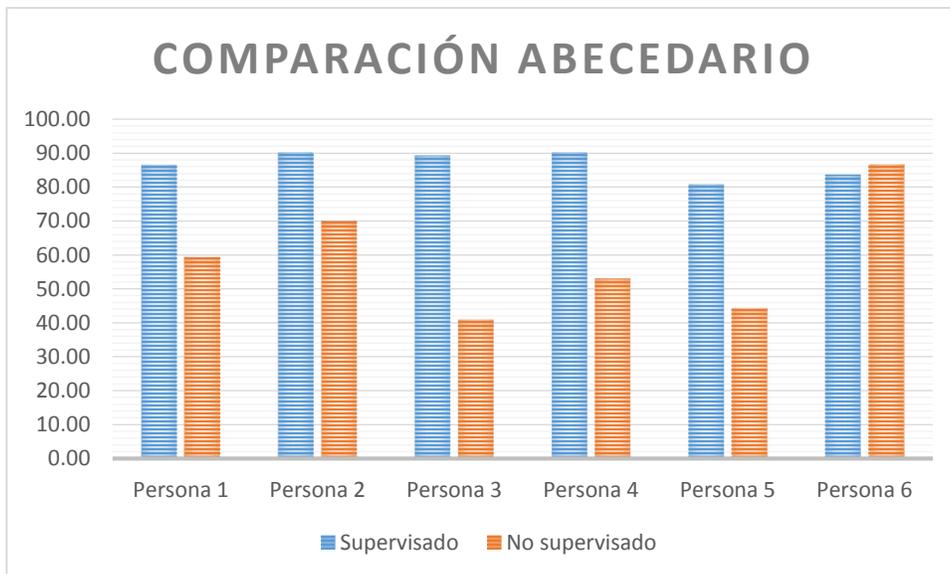


Figura 5.15 Comparación de resultados prueba abecedario

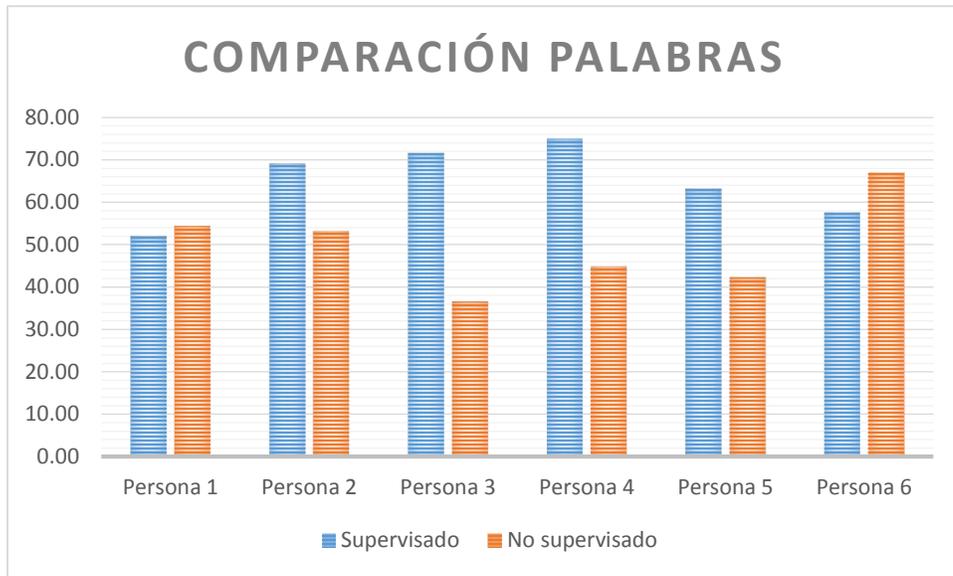


Figura 5.16 Comparación de resultados prueba palabras

De igual manera se realizó una comparación en general en cada tipo de prueba no supervisado y supervisado, en la Figura 5.17 se puede apreciar que existe un porcentaje considerable entre uno tipo de prueba con el otro, con esta comparación se puede concluir que la atención personal que se tuvo con las personas del tipo de prueba supervisado fue satisfactoria y se puede ver reflejada en la gráfica.

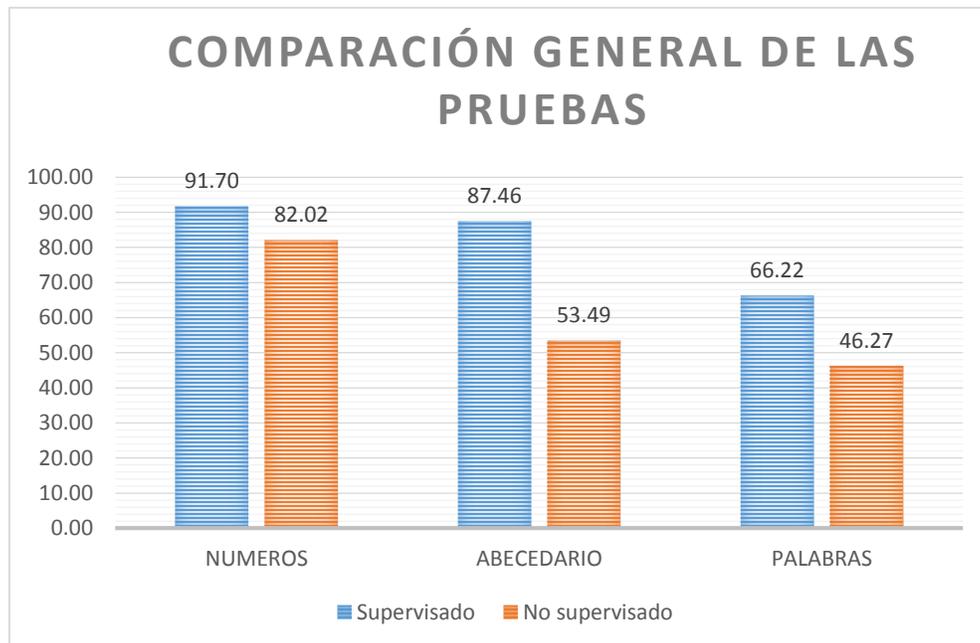


Figura 5.17 Comparación general de los resultados de las pruebas

5.4.3.4 Resumen de la prueba

Con los resultados obtenidos del grupo no supervisado se puede apreciar que es necesario ser más explícito al momento de decirle al usuario final como debe de realizar las señas frente al dispositivo *Leap Motion Controller* ya que ellos obtuvieron un porcentaje bajo, 60.59% de la prueba en general, mientras que el grupo supervisado obtuvieron un porcentaje mayor, 81.70% de la prueba en general, esto debido a que se mantuvo una atención personal con cada participante puesto que se les dijo como se deberían de realizar las señas. Por lo que es importante implementar una fase de capacitación previa al uso de la aplicación.

Mientras mayor sea el universo de señas que se desea reconocer y menor número de señas reconocidas, el nivel de precisión decae mucho, por tal motivo es necesario implementar otro método para el reconocimiento de las señas ya sea con otro algoritmo, utilizar algún dispositivo, integrar otro dispositivo de ayuda al *Leap Motion Controller*, etc.

Para finalizar se muestra la Tabla 5.15 la cual contiene:

- El total de señas propuestas durante el trabajo de investigación.
- El total de señas reconocidas.
- El total de señas no reconocidas.
- El porcentaje de precisión de la aplicación *SignMx*.

Tabla 5.15 Resultados finales SignMx

3 Semestre				
	Propuestas	Reconocidas	Sin Reconocer	% Reconocimiento
Abecedario	27	17	10	62.96 %
Numero	10	8	2	80.00 %
Total	37	25	12	67.57 %
Anexo 4 Semestre				
Palabras	47	20	27	42.55 %
			Total	55.06 %

Capítulo 6.

Conclusiones y trabajos futuros

En este capítulo se presenta la conclusión general de este trabajo de investigación, así también los trabajos futuros que se pueden desarrollar.

6.1 Conclusiones

En el mundo actual se observa la inclusión de las personas discapacitadas, pero a muy pocas se les da ese apoyo que realmente necesitan; en este caso con las personas sordomudas que no pueden comunicarse de manera verbal con las personas que no padecen dicha discapacidad, por tal motivo en este trabajo de investigación se buscó la manera en que ellos puedan tener una mejor comunicación haciendo uso de lo que la tecnología nos ofrece.

Se desarrolló una metodología para poder desarrollar esta ayuda para ellos, cada paso de la metodología influyó para obtener como producto final una aplicación la cual interpreta las señas y las traduce a voz. Para esto se utilizó el dispositivo *Leap Motion Controller* la cual rastrea la mano y la interpretará mostrando en 3D la seña realizada, para luego proceder a ser reproducida a una voz sintetizada y de esta manera producirse la comunicación.

Con los resultados obtenidos en las pruebas se puede deducir que es necesario capacitar a las personas que usarán la aplicación debido a que las señas deberán de ser realizadas de una manera en específico. Por otro lado por las limitación que presenta el *Leap Motion Controller* no se pudieron clasificar un mayor número de señas, pero este trabajo podría ser de utilidad para trabajos futuros con la misma tecnología pero usando otro algoritmo para el reconocimiento de las señas o utilizar dispositivo extras.

6.2 Proyecto y publicación

- Durante la estancia en la institución se participó en el Evento Nacional de Innovación Tecnológica (ENIT) con el proyecto PARPAVOX en la fase regional y nacional, el cual obtuvo el primer lugar en la fase regional mientras que en la nacional no se obtuvo uno de los tres primeros lugares.
- Publicación y presentación oral del artículo "*Recognition of Mexican Sign Language Through the Leap Motion Controller*" Pag.147-150 ISBN 1-60132-430-8, en Las Vegas, Nevada en el *World Congress in Computer Science, Computer Engineering & Applied Computing: WORLDCOMP 2016*, los días del 25 al 28 de julio de 2016.

6.3 Trabajos futuros

A continuación se muestran los trabajos futuros relacionados con este trabajo de investigación.

- Mejorar o buscar un nuevo algoritmo para el reconocimiento de las señas en vista de los valores obtenidos.
- Mejorar la interfaz de *SignMx* y utilizar el patrón de arquitectura de software MVC.
- Ampliar la investigación que se realizó y buscar nuevas maneras para la inclusión de las personas sordomudas.
- Cargar imágenes estáticas para hacer mejor el reconocimiento de las señas a la hora de comparación la seña realizada con la almacenada.
- Usar pictogramas para formar oraciones y estas ser reproducidas.
- En vista el *Leap Motion Controller* puede grabar una seña en 3D para luego ser visualizada y repetida n veces, se podría crear una aplicación como medio de enseñanza para aquellas personas que deseen conocer como realizan las señas las personas sordomudas.

Referencias

- Acosta, L. (1997). *Comunicación y enseñanza de la lectura en niños sordos e hipoacúsicos*. Universidad de Minnesota.: Tesis de Maestría.
- CIU, T. C. (29 de Agosto de 2014). *Dinamismo en el Mercado de Smartphones en México*. Obtenido de http://www.the-ciu.net/nwsltr/289_2Distro.html
- CONADIS. (2012). *Glosario de términos sobre discapacidad*. México, D.F.: CONADIS.
- CUI, T. C. (2013). *Mercado de Smartphones en México en 2013*. Obtenido de http://the-ciu.net/nwsltr/237_1Distro.html
- Definicion.de . (28 de Abril de 2015). *Definicion.de* . Obtenido de <http://definicion.de/windows/>
- DIEELSEME. (2004). *Estudio introductorio al léxico de la LSM*. Mexico D.F.: Dirección de Educación Especial.
- Enable Talk, E. (16 de Febrero de 2015). *Enable Talk project*. (Enable Talk) Recuperado el 26 de Febrero de 2015, de <http://www.enabletalk.com>
- f. (2). f. f. f.
- Gobernación, S. d. (30 de Noviembre de 2012). *Diario Oficial de la Federación*. Recuperado el 2015 de Marzo de 02, de http://dof.gob.mx/nota_detalle.php?codigo=5281002&fecha=30/11/2012
- Google Gesture. (03 de Julio de 2014). *Google Gesture*. (ADD BERGHS) Recuperado el 26 de Febrero de 2015, de <http://add.berghs.se/case/google-gesture>
- INEGI. (2010). Las personas con discapacidad en México: una visión censal. En I. N. Geografía, *Las personas con discapacidad en México, una visión al 2010* (pág. 282). México: INEGI.
- INEGI. (2013). *Estadísticas a propósito del día internacional de las personas con discapacidad*. México, D.F.: Instituto Nacional de Estadísticas y Geografía.
- KSLT. (16 de Diciembre de 2014). *Kinect Sign Language Translator expands communication possibilities - Microsoft Research*. (Microsoft) Recuperado el 26 de Febrero de 2015, de <http://goo.gl/23KTmi>
- LeapMotion. (27 de Abril de 2015). *Leap Motion Developer Portal*. Obtenido de Leap Motion Developer Portal: https://developer.leapmotion.com/documentation/java/devguide/Leap_Overview.html
- LeapMotion. (27 de Abril de 2015). *Leap Motion Developer Portal*. Obtenido de Leap Motion Developer Portal: https://developer.leapmotion.com/Leap_Motion_App_Store_Distribution_Agreement_100914.pdf
- Manz. (28 de Abril de 2015). *Emezeta blog*. Obtenido de <http://www.emezeta.com/articulos/10-sintetizadores-de-voz-tts-para-android>

- MORENO, D. N., & CHIGUANO, R. E. (2011). *Diseño e implementación de un sistema traductor de lenguaje de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado*. Quito, Ecuador: Tesis Licenciatura.
- MotionSavvy. (2015 de Febrero de 2015). *MotionSavvy » Two way communication tool for the deaf*. (Motionsavvy) Recuperado el 26 de Febrero de 2015, de <http://www.motionsavvy.com>
- Ogden, P. (2002). En *El jardín silencioso. Criando a su hijo sordo*. Oregon, Estados Unidos: Butte Publications, Inc.
- O'Leary, R. (04 de Noviembre de 2016). *Github - Roboleary*. Obtenido de <https://github.com/roboleary/LeapTrainer.js/>
- OMS. (2011). *Informe mundial sobre la discapacidad*. Ginebra, Suiza: OMS.
- PSLT. (18 de Diciembre de 2014). *Home - PSLT*. (Technabling Ltd) Recuperado el 26 de Febrero de 2015, de <http://www.pslt.org>
- RedUSERS. (30 de Octubre de 2013). *Estudiantes crean un sistema con Kinect para traducir el lenguaje de señas argentino*. (RedUSERS) Recuperado el 26 de Febrero de 2015, de <http://goo.gl/bfVczH>
- Resposive Voice. (02 de Junio de 2016). *Resposive Voice*. Obtenido de Resposive Voice: <http://responsivevoice.org/text-to-speech-languages/texto-a-voz-en-espanol/>
- Roboshop. (27 de Abril de 2015). *Roboshop*. Obtenido de Putting robotics at your service™: <http://www.robotshop.com/media/files/pdf/datasheet-leap-motion-controller.pdf>
- Saussure, F. d. (1945). Objeto de la lingüística. En F. d. Saussure, *Curso de lingüística general* (pág. 37). Buenos Aires: Losada S.A.
- Serafín de Fleischmann, M. E., & González Pérez, R. (2011). *Manos con voz*. Mexico, DF: Libre Acceso A.C.
- Serafín de Fleischmann, M. E., & González Pérez, R. (2011). *Manos con Voz*. México, D.F.: Libre Acceso A.C.
- Serafín de Fleischmann, M. E., & González Pérez, R. (2011). *Manos con Voz - Diccionario de Lengua de Señas Mexicana*. Mexico, D.F.
- Talk Enable. (16 de Febrero de 2015). *Enable Talk project*. (Enable Talk) Recuperado el 26 de Febrero de 2015, de <http://www.enabletalk.com>
- Trujillo Romero, F., & Caballero Morales, S. O. (2012). Towards the Development of a Mexican Speech-to-Sign-Language Translator for the Deaf Community. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*, pp. 83-89.
- Vatavu, R. D., Anthony, L., & Wobbrock, J. O. (2012). Gesture as Point Clouds: A \$P Recognizer for User Interface Prototypes. *ACM*, 273-279.



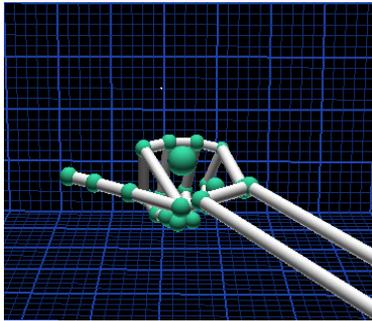
Anexos

Anexo A – Señas en 3D reconocidas por *SignMx*

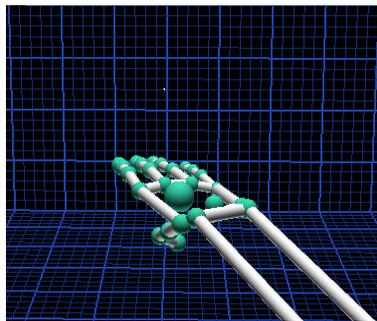
Las señas que la aplicación *SignMx* reconoce, se pueden apreciar en la tabla x, dichas imágenes fueron obtenidas desde el *Leap Motion Visualizer*.

Señas 3D del abecedario

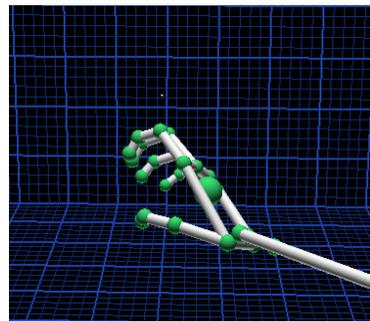
Abecedario



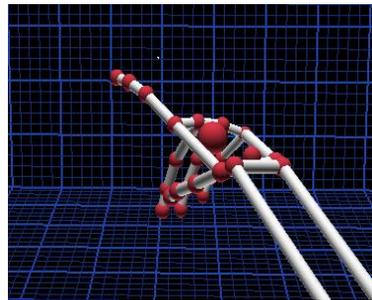
A



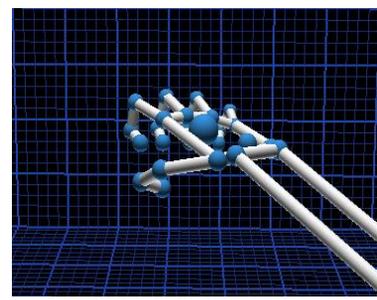
B



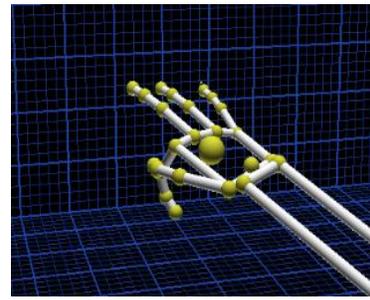
C



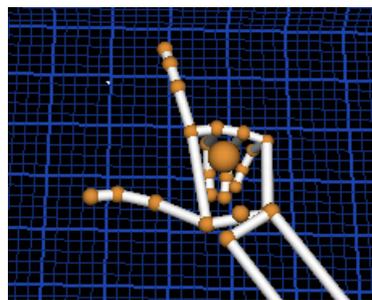
D



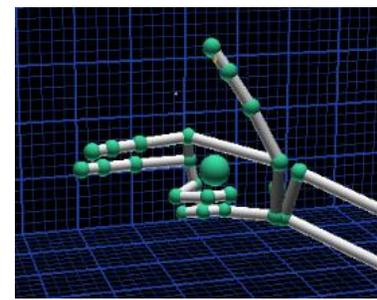
E



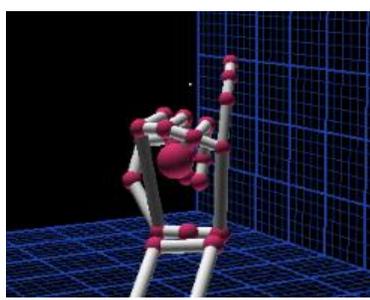
F



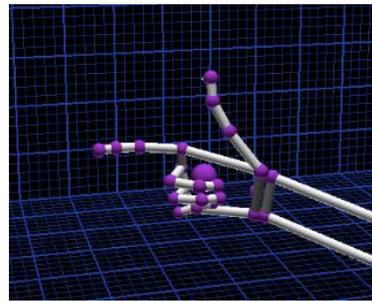
L



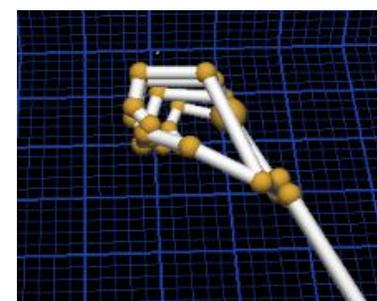
H



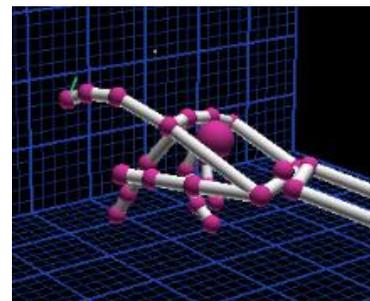
I



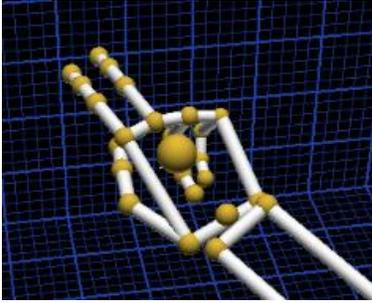
G



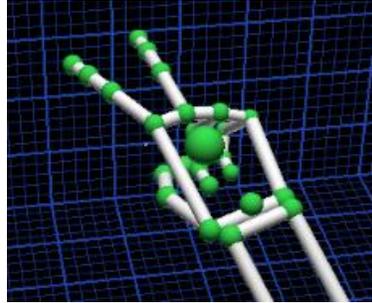
O



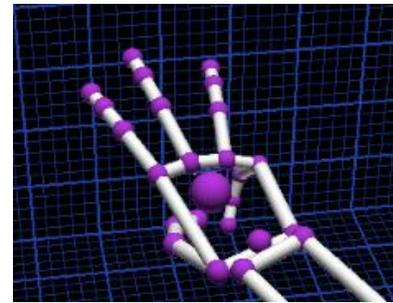
P



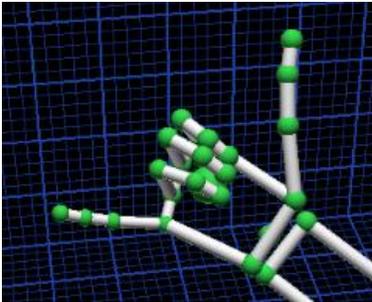
U



V



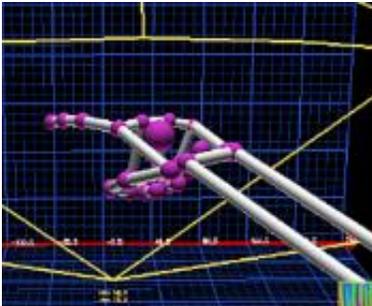
W



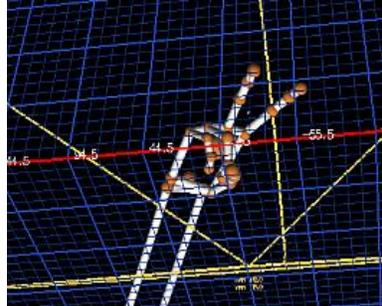
Y

Señas 3D de los números

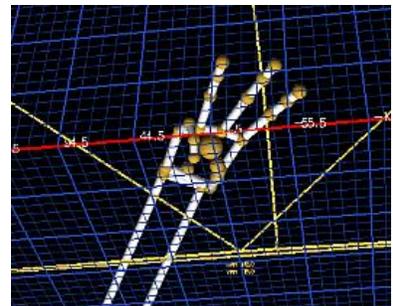
Números



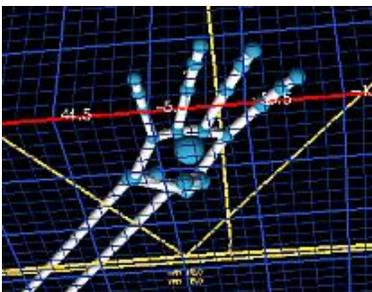
1



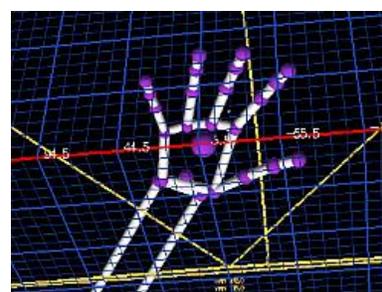
2



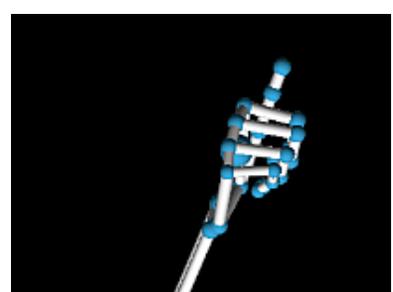
3



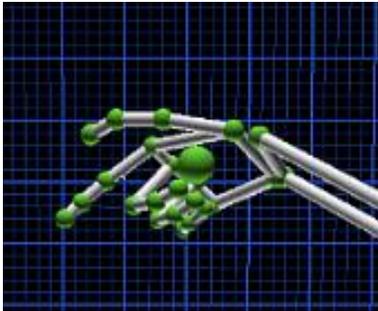
4



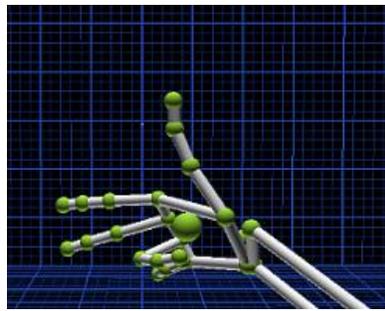
5



6



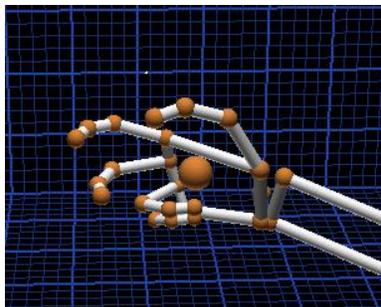
7



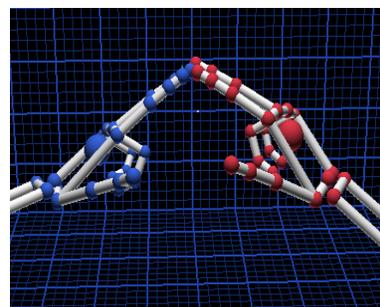
8

Señas 3D de las palabras

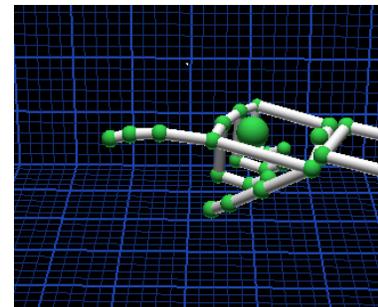
Palabras



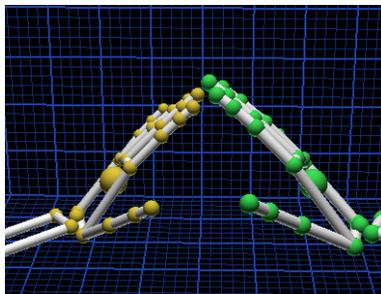
COPA



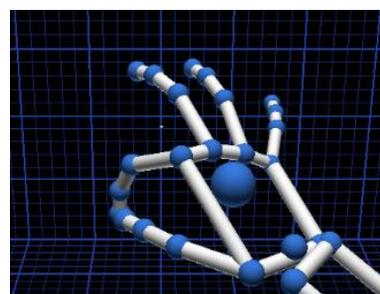
HOGAR



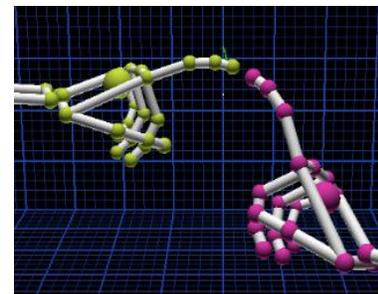
MENOS



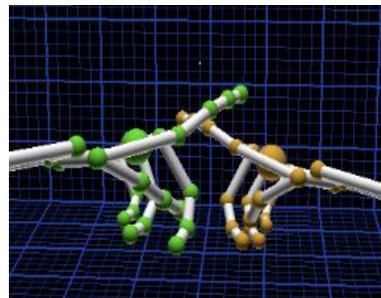
CASA



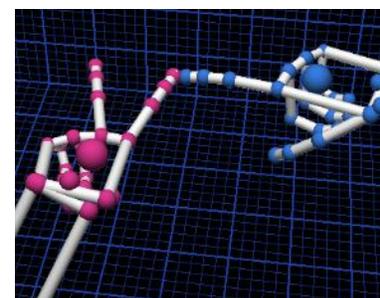
CEREZA



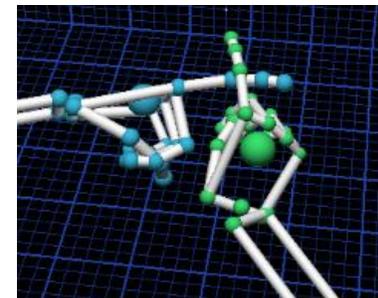
DIVISIÓN



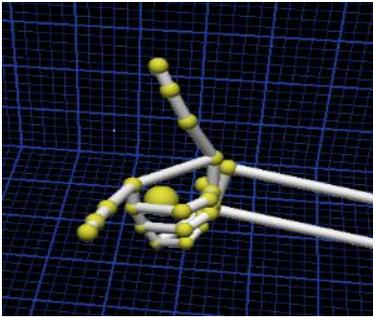
MULTIPLICACIÓN



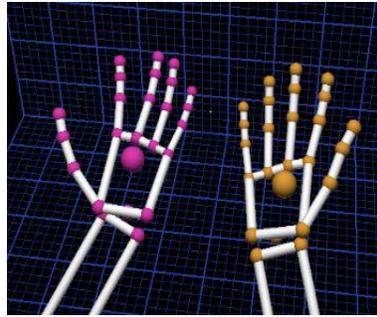
RAÍZ CUADRADA



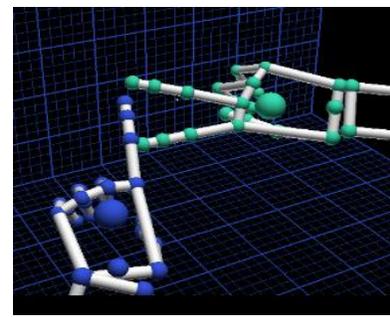
SUMA



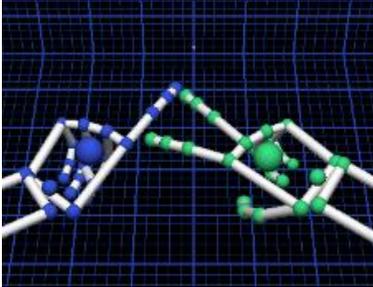
PISTOLA



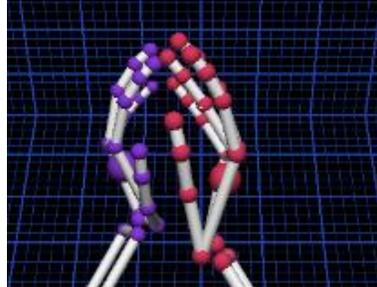
SOSTENER



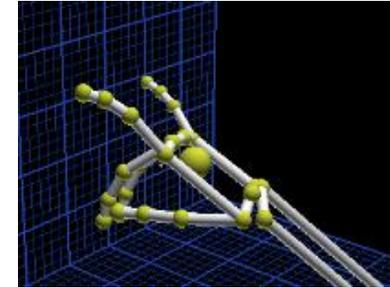
ESCUADRA



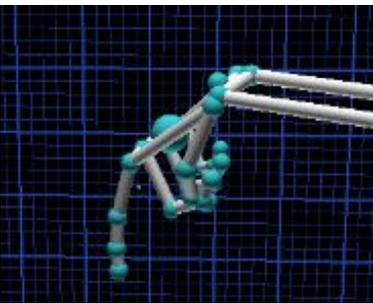
PAY



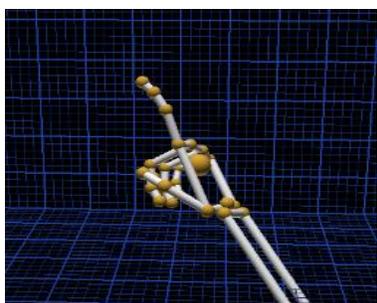
TORTA



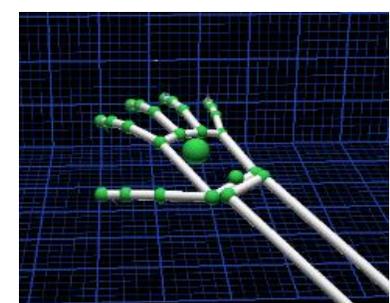
VACA



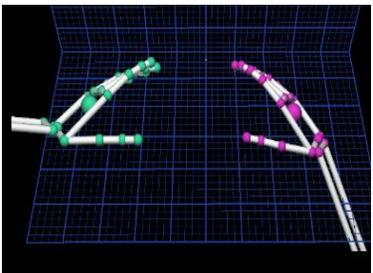
ABAJO



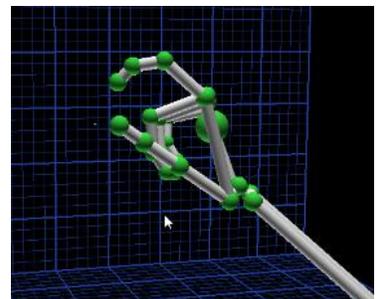
ARRIBA



MANO



COMAL



ALMENDRA

