Universidad de Concepción
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

CONSTRUCCIÓN DE GESTOS EN LENGUA DE SEÑAS CHILENA A PARTIR DE SUB-GESTOS

por

Paula Geraldine Castillo Osses

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Electrónico(a)

Profesor guía

Mario Medina Carrasco

Marzo 2025
Concepción, Chile

Al puerco más oloroso y a mi Flaquirucho.

"Optimism is a strategy for making a better future. Because unless you believe that the future can be better, you are unlike to step up and take responsibility for making it so". Noam Chomsky

Agradecimientos

Para empezar, quisiera agradecer a mis padres M. Hernán Castillo B. y Patricia Osses P. quienes me han apoyado toda mi vida y han sido un ejemplo de esfuerzo y perseverancia. Gracias por siempre creer en mí. A mis hermanos Hernán Castillo O. y Cristián Castillo O. por ayudarme a cumplir con mis objetivos universitarios de diversas maneras, y a mi mami Elvis y mi tata Miguelito que no me alcanzaron a ver terminar, pero sé que me acompañan siempre en donde estén. Espero que se sientan orgullosos de esta personita que ayudaron a criar.

Quiero agradecer a mi mejor amiga Valeria Carrasco V. que en estos 23 años (tuve que sacar la cuenta) de amistad siempre ha estado para mí, siendo un apoyo incondicional a pesar de las distancias. Espero que terminando todo esto podamos juntarnos a compartir unos buenos mates, jugar cartas y escuchar cumbias.

También quiero agradecer a mis amigos que realicé en este largo y doloroso camino llamado universidad, al Gonzalito un fiel compañero de batallas, a Sofi una admirable y adorable pequeña gran mujer, a Carlitos un increíble contador de historias, a Abel por acompañarme tantos años y ayudar a formar parte de lo que soy ahora. Y a las demás personas que han formado parte de mi experiencia en esta vida que no están aquí, pero que llevo en mi corazón, siempre les estaré eternamente agradecida por hacer de mi vida algo mejor.

Quiero agradecer al jefazo Mario Medina, por ser el mejor jefe de carrera que pude tener y por ser un gran ser humano. Su pasión y curiosidad por ciertos temas siempre avivaron la mía y me ayudaron a seguir.

Finalmente, pero no menos importantes, quiero agradecer a mi puerco oloroso, mi perra, mi fiel compañera que llegó a mi vida cuando empecé esta carrera y desde entonces me ha acompañado en las buenas, en las malas y en las olorosas. Y a mi Flaquirucho, David Oscanoa L., quien no ha dejado de apoyarme desde que nos hicimos amigos, me ha enseñado a ser valiente, resiliente, a vivir sin importar el qué, el cómo o dónde, "sin timón y en el delirio", no me alcanzará la vida para agradecer todo lo que ha hecho por mí. Independiente de los resultados, terminando esto, hay que ir por unos buenos completos, como corresponde.

Resumen

Este trabajo presenta una alternativa en la construcción de señas de la lengua de señas chilena mediante sub-gestos (no necesariamente tienen significado por sí solos) que pueden ser compuestos para formar diferentes gestos, aprovechando la estructura de esta lengua. Además, este trabajo presenta la implementación de un programa que permite realizar la conversión de glosas en lengua de señas chilena a animaciones 3D.

Adicionalmente, se realiza una revisión de los métodos de "traducción"de lenguas de señas al idioma hablado, desde intérpretes, reconocimiento de imágenes, video, etc. A nivel país también se han presentado proyectos que tienen estas ideas en su concepción. Sin embargo, no se había profundizado en la forma en la que se construyen las animaciones; este es el objetivo de este trabajo.

Al ser la lengua de señas una lengua natural, posee también características propias de las lenguas orales que podemos aprovechar cuando hablamos de çonstruirüna oración o palabra. Debemos entender que algunos términos serán más bien abstractos que literales, como por ejemplo, al hablar de "palabra"sería más correcto el uso de "glosa". Siendo la "glosa", el método que utilizaremos para establecer las relaciones entre el español chileno y la lengua de señas chilena. Esto, sumado a la disponibilidad de recursos tecnológicos que permiten la animación 3D y que se encuentran a libre disposición como Blender y Unity, programas utilizados para crear las animaciones que representan a cada sub-gesto.

Summary

This work presents an alternative in the construction of Chilean Sign Language signs by the usage of sub-gestures (which do not necessarily have meaning by themselves) that can be composed to form different gestures, taking advantage of the structure of this language. In addition, this work presents the implementation of a program that allows the conversion of Chilean sign language glosses into 3D animations.

Additionally, a review of the "translation"methods from sign languages to the spoken language is made, which includes: interpreters, image recognition, video, etc.. At a national level, projects have also been presented that have these ideas in their conception. However, the way in which the animations are constructed has not been studied in depth; this is the objective of this work.

Since sign language is a natural language, it also has characteristics of oral languages that we can take advantage of when we talk about çonstructing.[a] sentence or word. We must understand that some terms will be abstract rather than literal, for example, when speaking of "word", it would be more correct to use "gloss". The "gloss"method will be used to establish the relationship between Chilean Spanish and Chilean Sign Language. This, added to the availability of technological resources that allow 3D animation and that are freely available such as Blender and Unity, programs used to create the animations that represent each sub-gesture.

Tabla de contenidos

Tabla de figuras

Tabla de secciones de código

1. Introducción a la Lengua de Señas

1.1. Primeros inicios

Las personas sordas han existido probablemente desde los inicios de la humanidad, se considera incluso que las primeras formas de comunicación humanas pudieron darse a través de los gestos y expresiones faciales, antes de que se desarrollara el lenguaje oral como tal y todo lo que conocemos respecto a este hoy en día [1]. Sin embargo, las personas con esta condición no siempre fueron aceptadas dentro de la sociedad o vistas como pares, sus derechos como ciudadanos eran limitados por su condición [2]. En la antigua Grecia, por ejemplo, Aristóteles dijo: *"Aquellos que nacen sordos se vuelven insensatos e incapaces de razonar"*. No podemos asegurar con certeza la intención del autor al mencionar estas palabras, pero tuvieron un impacto en cómo estas comunidades veían y se relacionaban con estas personas.

Por suerte, a lo largo de esta historia no todo ha sido negativo. Los primeros avances en materia de la educación de personas sordas fueron introducidos por la iglesia, la cual anteriormente no incluía a la comunidad sorda al considerar que la fe era un principio que el ser adquiría por medio de la escucha. Fue San Agustín quien señaló que los sordos podían adquirir la fe por otros medios, haciendo referencia a los gestos y movimientos que estos utilizaban, entendiéndolo como una forma de comunicación similar a la hablada [3].

1.2. Primeros pasos en la enseñanza de la Lengua de Señas

Los registros más antiguos de los que se tiene información sobre la enseñanza de las personas sordas son relatos transmitidos de persona a persona. Si bien no se puede determinar con exactitud quién "creó" las señas presentes en los alfabetos manuales, se teoriza que eran utilizadas y enseñadas por monjes quienes realizaban votos de silencio en los monasterios [4].

Dentro de los primeros hombres mencionados en la historia de la enseñanza de la lengua de señas se encuentra el Fray Pedro Ponce de León. Aunque se discute si él fue el primero en desarrollar una metodología sobre cómo enseñar este lenguaje, se sabe que educó a los hijos de Juan Sancho de Tovar y Velasco, un miembro de la nobleza española en el siglo XVI [5] y que para ello empleaba un alfabeto de señas bimanual [6], similar al utilizado hoy en día por las lenguas de señas británica, neozelandesa y australiana (BANZSL) mostrado a continuación en la Figura 1 [7],

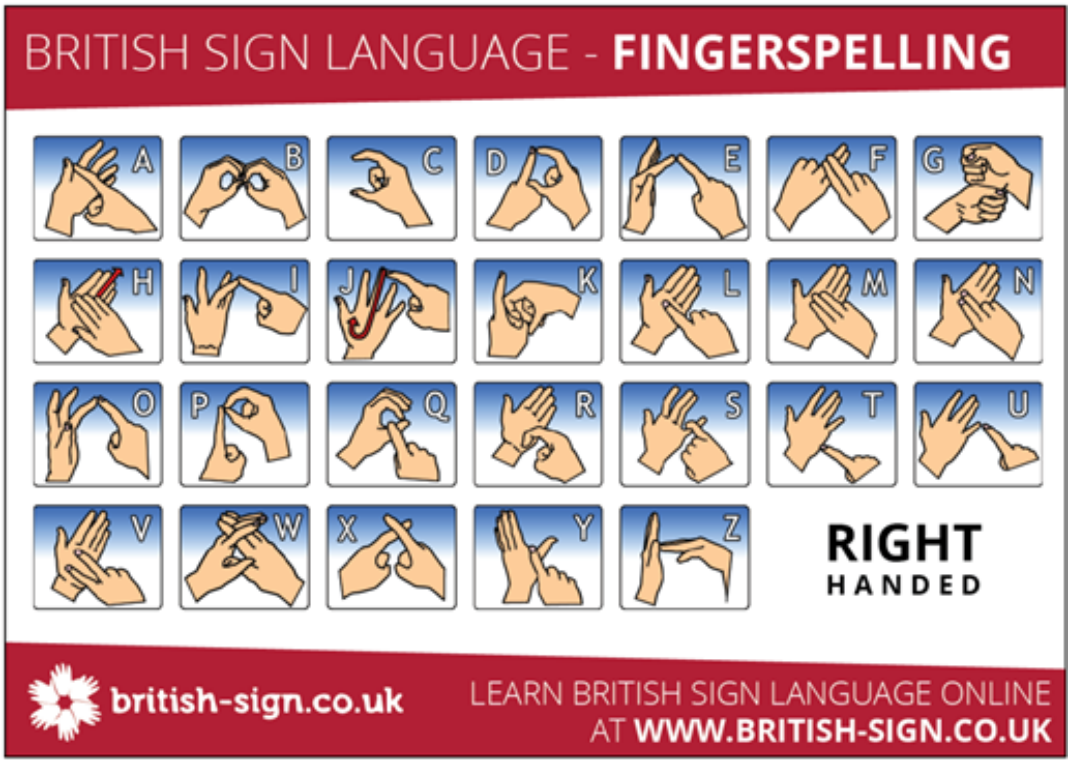


Figura 1: Alfabeto dactilológico bimanual británico.

Otro personaje controversial pero destacable es Juan Pablo Bonet quien plasmó en su libro *"Reducción de las letras y arte de enseñar a hablar a los mudos*[1]*"* (Madrid, 1620) [8], un tratado sobre cómo enseñar la lengua de señas a personas sordas, en la que se incluyen imágenes del alfabeto manual utilizado, el cual fue inspirado en el alfabeto manual del Padre Fray Melchor de Yebra, quien documentó dicho alfabeto en su libro "Refugio de los enfermos" y quien postula como inventor del mismo a San Buenaventura. Este alfabeto en sus inicios fue creado con la intención de ayudar a personas que estaban en su lecho de muerte, que no tenían la capacidad de hablar u oír, a poder confesarse [9]. A continuación, se muestra la captura de las páginas del libro de Bonet donde se pueden apreciar los gestos asociados a las letras del abecedario en lo que sería un alfabeto dactilológico unimanual[8].

[1]Si bien el título del libro hace referencia a "... enseñar a hablar a los mudos", debe considerarse que en la antigüedad la palabra "sordo" no era utilizaba como hoy en día, por lo que en documentación correspondiente a la época se suele encontrar la palabra "mudos" para referirse a personas que presentasen sordera, debido a que estos al no escuchar tampoco podían desarrollar la capacidad de hablar, sobre todo en casos de sordos prelocutivos, es decir, que habían nacido sordos o que quedaron sordos a edades muy tempranas.

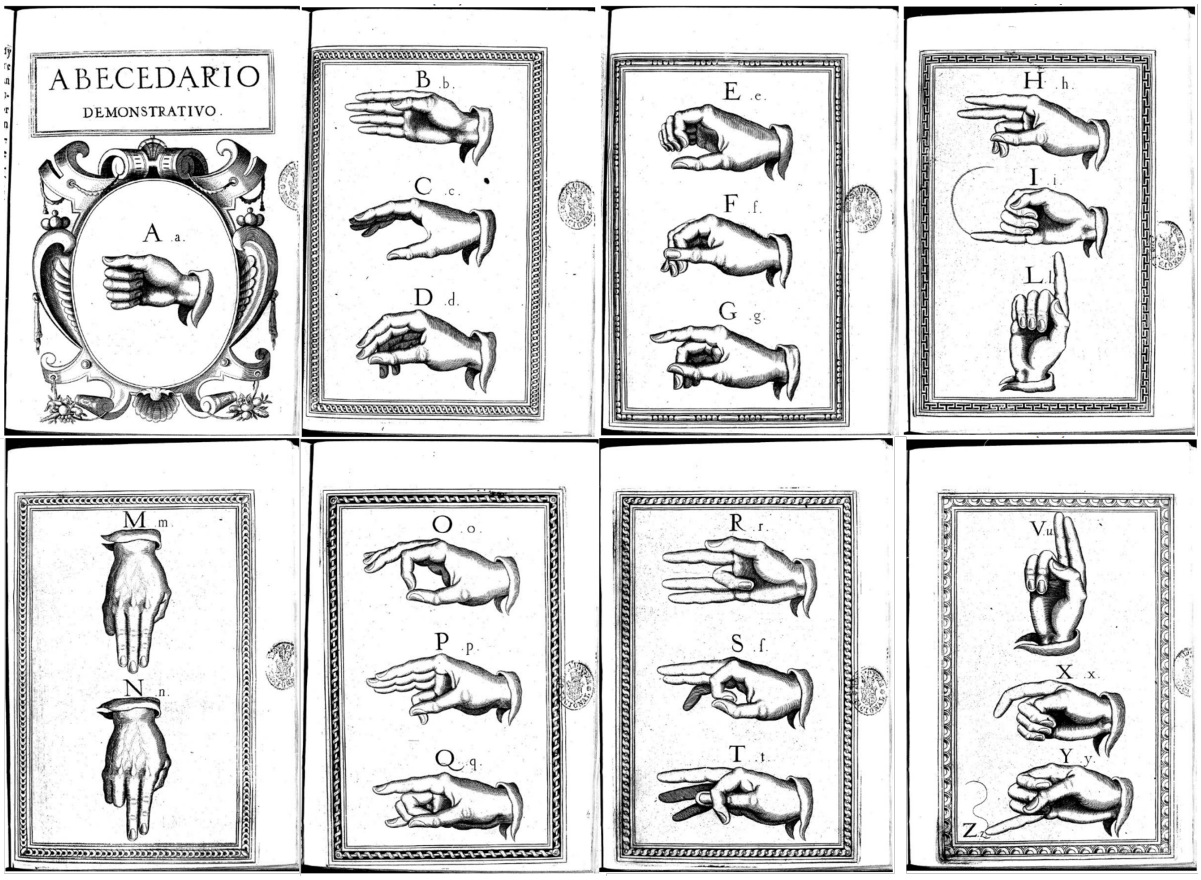


Figura 2: Láminas con el alfabeto dactilológico de las letras del abecedario del libro de Juan Pablo Bonet "Reducción de las letras y arte de enseñar a hablar a los mudos" presentes entre las páginas 130 – 131.

El libro de Juan Pablo Bonet llegó a oídos de otro personaje relevante en esta historia, el abad Charles-Michel de l'Épée, quien diseñó un método de enseñanza de personas sordas mucho más completo que los creados por sus antecesores y también quien fundó la primera escuela gratuita para sordos en Francia en 1771, donde instruyó a sus alumnos a aprender a escribir el francés, deletrear palabras mediante un alfabeto dactilológico, expresar conceptos mediante señas simples y posteriormente inició a estos en materias más complejas, permitiendo a los sordos de Francia alcanzar niveles educativos más elevados. Si bien, L'Épée no fue el creador del lenguaje de señas francés, sí añadió nuevas señas a este. Además, debido a la popularidad que obtuvo por su escuela en Europa, llegaron desde el extranjero personas de diversos países con el propósito de aprender este método educativo, expandiéndose por varios lugares del mundo [10]. Dichas influencias se pueden ver en la Figura 3 [11] un mapa con las principales familias de lenguas de señas registradas en el mundo, donde la hispanofrancesa es la que concentra la mayor cantidad de países.

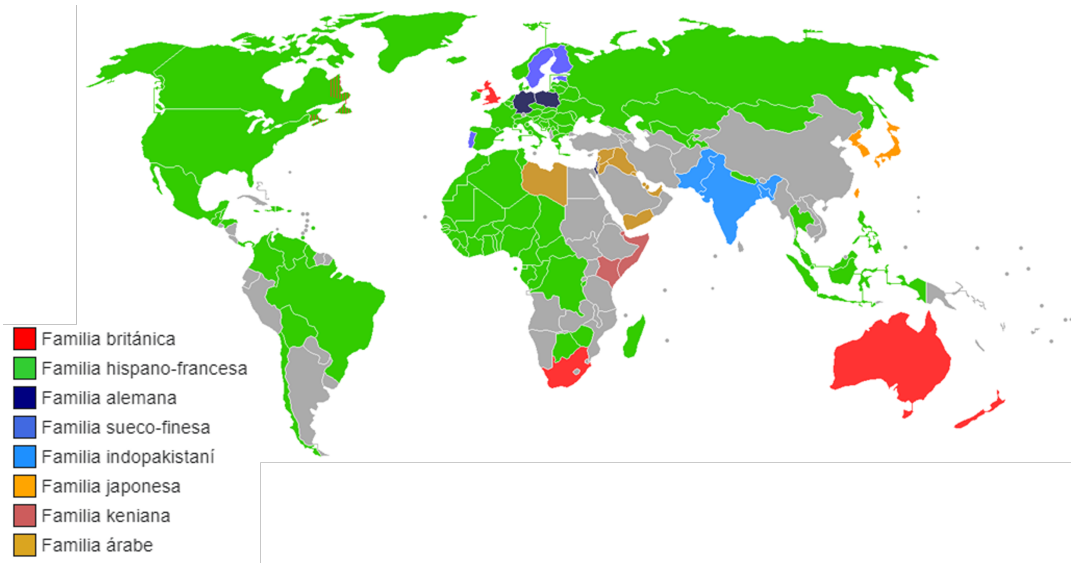


Figura 3: Mapa mundial con las principales familias de lenguas de señas.

1.3. Método Oralista

A la par que se extendía por Europa el método de L'Épée, en Alemania Samuel Heinicke propuso el "método oral". Este se basaba principalmente en que los alumnos sordos aprendieran a leer los labios y eventualmente recibían entrenamiento en la articulación de fonemas que les permitieran "hablar" o formar palabras [12]. Si bien a inicios no tuvo tanto reconocimiento, con la llegada del siglo XIX, en el Segundo Congreso Internacional de Maestros de Sordomudos, celebrado en Milán, Italia, entre el 6 y 11 de septiembre de 1880, se estableció como la principal forma de educación de personas sordas, prohibiéndose el "método manualista" y las señas, por considerarse que se veían afectadas "el habla, la lectura labial y la claridad de los conceptos" (Segunda Resolución del congreso) [13].

Esto supuso un retroceso en los avances que se habían logrado en la educación de las personas sordas. Posteriormente al congreso, los educadores del método manualista fueron desapareciendo en Europa, llegando esta corriente a América, impactando las escuelas del continente, que cambiaron sus formas de enseñanza también [13].

1.4. Lengua de señas en América

A continuación, se hará la breve revisión histórica de 4 países del continente americano y su camino en la enseñanza y adopción de la lengua de señas. Cabe resaltar que las señas siempre han formado parte del sordo, pero muchas de estas instituciones contribuyeron a estandarizar la lengua, formar comunidades y sentar las bases para su posterior desarrollo educativo.

1.4.1. Lengua de señas estadounidense (ASL)

A inicios del siglo XIX de la mano del Dr. Thomas Hopkins Gallaudet, quien, intrigado por la sordera de su pequeña vecina, una niña de 9 años llamada Alice Cogswell, se interesó en buscar un sistema de educación adecuado para personas sordas. Dada la relevancia que obtuvo el método de L'Épée en Europa, Gallaudet viajó al viejo continente para aprender de quienes tenían para ese entonces los mejores métodos educativos dentro de este ámbito [14].

En Europa, Gallaudet se encontró con el abad Sicard, sucesor de L'Épée en el Instituto Nacional para Sordos-mudos; además, conoció a Laurent Clerc y Jean Massieu, dos prestigiosos educadores para sordos exalumnos de Sicard, con quienes estudiaría sobre esta famosa metodología. Al volver a Estados Unidos, Gallaudet le propone a Clerc que lo acompañe con la intención de fundar en el país una escuela para sordos. Así es como nace en 1817 la primera escuela pública gratuita para sordos en Hartford, Connecticut, llamada "Escuela Americana para Sordos". Estos fueron los primeros pasos en la construcción de la lengua de señas americana (ASL) siendo una fusión de señas estadounidenses y las aprendidas por Gallaudet en Francia [14].

Para los años 1863 ya se habían establecido 22 escuelas en el país, muchas de ellas por exestudiantes de Clerc, quienes replicaban las metodologías educativas de Clerc para personas sordas [14].

Posteriormente, en el año 1864, Edward Miner Gallaudet, hijo menor de Thomas Hopkins Gallaudet y profesor de la Escuela Americana para Sordos, presentó la idea de establecer una universidad para sordos al congreso estadounidense, petición que fue aceptada, abriéndose en ese mismo año la división universitaria del Instituto de Columbia, el "Colegio Nacional de Sordos-Mudos", renombrada en 1986 como "Universidad Gallaudet", la cual se mantiene hasta hoy en día, siendo la única universidad para sordos en el mundo [14].

Sin embargo, pasaron décadas para que la lengua de señas y la comunidad sorda volviesen a ser relevantes. Fue en 1960 cuando William Stokoe, lingüista y profesor de la Universidad Gallaudet, quien propuso en su estudio "Sign language structure: An Outline of the Visual

Communication Systems of the American Deaf", que las lenguas de señas poseían estructuras similares a las lenguas verbales y que, en consecuencia, constituyen una forma de lengua natural. Este hito marcó el reconocimiento de la lengua de señas como una lengua natural y, por lo tanto, a partir de aquí se abrió paso a un nuevo campo de investigación dentro de la lingüística, además del reconocimiento de esta, como la principal forma de comunicación de las personas sordas [15].

1.4.2. Lengua de señas mexicana (LSM)

En México en el año 1867, el francés Eduardo Adolfo Huet fundó la Escuela Nacional de Sordomudos. Sin embargo, esta fue cerrada a inicios del siglo XX; se desconocen los motivos por los cuales esto ocurrió. Se cree que la popularidad del "método oralista" aumentó, culminando con el congreso de Milán en el cual se prohibió utilizar y enseñar el método "manualista" [16].

Durante el siglo XX, la dirección de educación especial mexicana buscó enseñar a los sordos el español oral y escrito, con el fin de posteriormente integrarlos en una escuela de educación común. Esto provocó que se dejara de enseñar la lengua de señas mexicana [17].

En la década de los 80's, en las escuelas de educación especial se comenzó a practicar la filosofía de la "Comunicación Total", que permitía que las personas sordas se comunicaran a través de cualquier medio, sea este oral, escrito, dibujos, señas, etc. También fue en esta época donde la Dirección General de Educación Especial publicó "Mis primeras señas: Una introducción al Lenguaje Manual" y posteriormente "Mis primeras señas II", dos cuadernos didácticos dirigidos a los docentes de educación especial, dentro de los cuales se enseñaban señas básicas que permitieron hacer referencia a partes del cuerpo, colores, comida, etc. [17].

Finalmente, en el año 2005, la lengua de señas mexicana es reconocida como una lengua nacional y principal forma de comunicación de las personas sordas en México.

1.4.3. Lengua de Señas Nicaragüense (ISN)

La lengua de señas de Nicaragua constituye un caso particular dentro de la historia de estas lenguas, ya que en la década de los 80's con la Revolución Sandinista el gobierno en sus esfuerzos por suprimir el analfabetismo buscó educar a las personas sordas, mediante la implementación de señas elementales y lectura de labios. Sin embargo, este método de aprendizaje falló rotundamente. Los estudiantes sordos carecían de conceptos complejos pues no habían recibido educación hasta ese entonces y habían desarrollado señas simples que les permitían comunicarse en su entorno familiar [18].

Si bien, la educación de sordos por parte de los profesores no tuvo mayor éxito, en los

pasillos, durante los recreos, los estudiantes comenzaron a construir una lengua propia que les permitía comunicarse con otros alumnos sordos. De esta forma, la lengua de señas nicaragüense es el caso de una lengua que fue creada por una comunidad de sordos, la cual no tiene parentesco con ninguna otra lengua de señas en el mundo [19].

Finalmente, en 2009 se reconoce la lengua de señas nicaragüense como el medio oficial de comunicación de las personas sordas en Nicaragua.

1.4.4. Lengua de Señas Argentina (LSA)

En Argentina, los registros que se obtienen de enseñanza como tal de las personas sordas datan del año 1857, en el cual abriría sus puertas la primera escuela para sordos en Buenos Aires, con el alemán Karl Keil como director de esta. Sin embargo, se desconoce la metodología empleada por la escuela, aunque se cree que, por la nacionalidad de su director, esta pudo haber empleado el método oralista. Años más tarde, en 1871, Keil murió a causa de la fiebre amarilla y, en consecuencia, el establecimiento educacional cerró sus puertas [20].

Los años posteriores se dieron una serie de reformas influenciadas principalmente por Europa, en particular Italia, la cual estaba fuertemente influenciada por el método del oralismo alemán. Para 1897 se creó en el Instituto Nacional una unidad para niñas sordas, separando a los alumnos por sexo. Esto tendría repercusiones más tarde, ya que los niños sordos desarrollaron señas que posteriormente sólo utilizan los hombres, ocurriendo de manera similar para las niñas y mujeres sordas [21].

Finalmente, en 2023 la lengua de señas argentina fue reconocida oficialmente por el gobierno argentino, reconociendo la LSA como la principal forma de comunicación de las personas sordas argentinas.

1.5. Lengua de Señas Chilena

1.5.1. Enseñanza de la lengua de señas en Chile

En cuanto a la educación de sordos se trata, Chile no se quedó atrás. De hecho, fue pionero en la creación de la primera escuela pública para sordos a nivel latinoamericano, fundada en 1852, la cual se mantiene actualmente con el nombre de “Escuela Anne Sullivan”. Esta escuela era exclusiva para varones; dos años después, en 1854, se abrió una escuela para niñas sordas, donde aprenderían diversas materias como lectura, escritura, gramática castellana, aritmética, religión, costura y bordado para las niñas y encuadernación para los niños. Posteriormente, en

1889, se crea el Instituto de Sordos-Mudos, cuya función principal será formar profesores en educación especial y educar a los sordos del país, utilizando el método oralista impuesto en el congreso de Milán de 1880, asumiendo este modelo como "la única opción comunicativa para la comunidad sorda" [22].

Pasaron alrededor de 100 años manteniendo el método oralista, hasta que, en el Congreso de Hamburgo de 1980, este modelo fue cuestionado y se reconoció el derecho de las personas sordas a educarse bajo una lengua accesible para ellos, como lo es la lengua de señas, con la posibilidad de aprender y utilizar las lenguas orales de los lugares en los que vivan. A partir de aquí comenzaron a gestarse los cambios, y la educación de personas sordas empieza un proceso de experimentación, la Universidad Metropolitana de Ciencias de la Educación se suma a la filosofía de la "Comunicación Total". En consecuencia, a estos cambios, comenzaron a implementarse y enseñarse en las aulas la lengua de señas chilena (LSCh) [22].

1.5.2. Lengua de Señas Chilena (LSCh)

Corresponde a la lengua de señas utilizada por las personas Sordas chilenas, reconocida como la lengua natural de esta comunidad por la ley 20.422, artículo 26. En Chile actualmente, se reconocen 4 dialectos situados en Antofagasta, Valparaíso, Santiago y Concepción.

Tiene un alfabeto manual de 27 señas que representan las letras del alfabeto español, el cual se puede ver en la Figura 4, obtenida de millarayconhipoacusia[23]. Este es utilizado para deletrear palabras que no tienen una seña asignada, como nombres propios, palabras nuevas, siglas, etc.

La seña debe entenderse como un conjunto de expresiones y acciones que pueden incluir el movimiento de las manos, expresiones faciales y uso del espacio alrededor del señante. A continuación, se mostrarán algunos ejemplos tomados del Diccionario Lengua de Señas Chilena-Español Tomo I [24] y Tomo II [25].

Se puede ver en la Figura 5 cómo la mujer contrae la cara de manera similar a la reacción que se obtiene al ingerir el zumo de un limón debido al ácido de esta fruta y cómo la mano es llevada a un costado de la cara imitando la acción que se realiza al exprimirla para extraer su jugo.

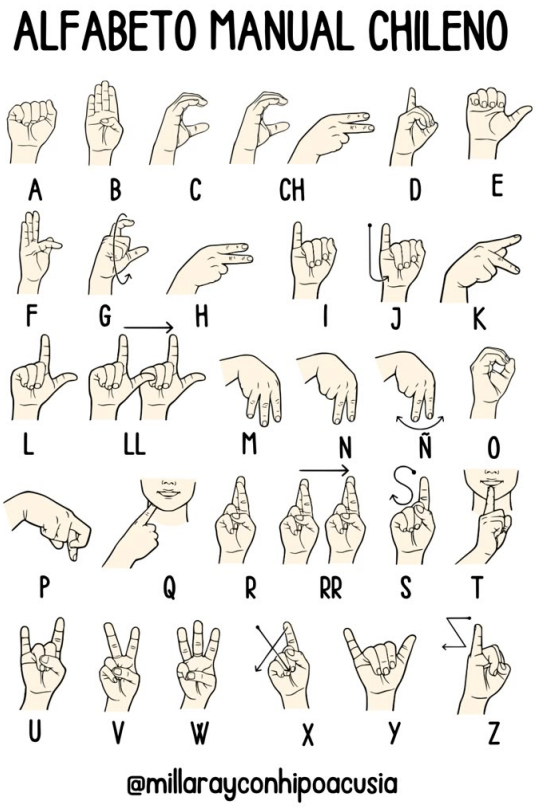


Figura 4: Alfabeto manual chileno.

Figura 5: Mujer realizando la seña correspondiente a LIMÓN.

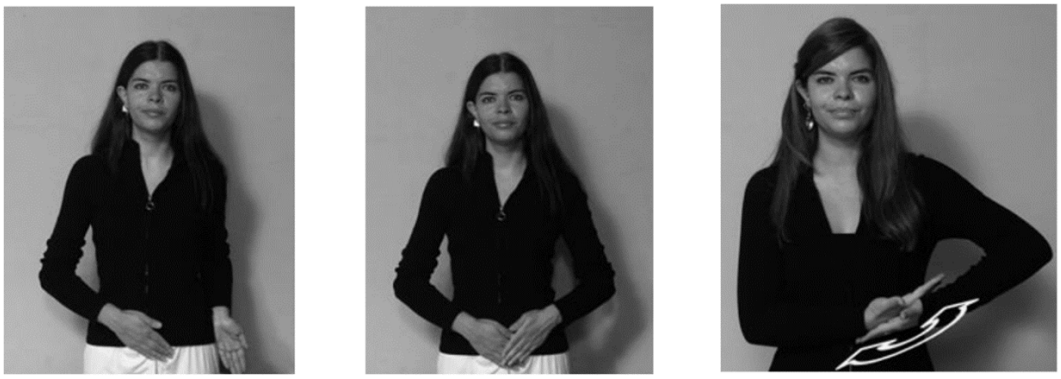


Figura 6: Mujer realizando la seña correspondiente a MUDAR-GUAGUA.

En la Figura 6 podemos ver cómo la seña incluye partes del cuerpo, específicamente la barriga, llevando una mano primero al vientre (primera sub-seña) y luego la otra (segunda sub-seña), de la misma manera en la que una persona cerraría un pañal. Notar que esta seña está compuesta por tres sub-señas, donde la última corresponde a la glosa GUAGUA.

Figura 7: Mujer realizando la seña correspondiente a PINGÜINO.

En la Figura 7 podemos ver una seña que utiliza todo el cuerpo, en la que se imita la forma particular en la que camina esta ave, poniendo los brazos al costado simulando las aletas, el cuerpo rígido y balanceándolo de lado a lado.

Como se pudo ver en las imágenes mostradas anteriormente, la lengua de señas chilena es compleja y abarca un amplio conjunto de expresiones tanto faciales como corporales, que va mucho más allá de simplemente realizar una mímica, aunque en ocasiones haya similitudes con los gestos que los hablantes del español realizamos.

1.5.3. Lingüística

Para la comprensión de la estructura gramatical de la lengua de señas chilena se utilizó como referencia el trabajo realizado por Pablo Saldías [26] donde se establecen algunas características generales que posee dentro de las lenguas viso-gestuales.

Si bien se puede pensar que las señas son la mímica de una palabra en español, realmente poseen una estructura que sigue 5 parámetros básicos: la configuración manual, el movimiento, el lugar en que se ubica, la orientación de la mano y la expresión gestual; es decir, las señas se componen de partes más pequeñas que no tienen un significado en particular, pero que al unirse pueden componer distintas señas, así como en los idiomas hablados se unen fonemas para formar palabras.

a) b)

Figura 8: (a) Mujer realizando la seña VERDAD. (b) Mujer realizando la seña VINO.

En las imágenes de la Figura 8 se puede ver cómo dos gestos que poseen la misma configuración manual, pero con distinto movimiento de brazos, en una el brazo y mano son dirigidos hacia la palma de la mano contraria (a) y en la otra el brazo y la mano son llevados hacia la cara (b) terminan configurando dos señas con significados distintos y no relacionados entre sí.

Dentro de la sintaxis de la lengua de señas chilena no se ha llegado a un consenso. Si bien se tienen investigaciones que indicarían un orden mayoritariamente SVO (sujeto-verbo-objeto), también se han encontrado órdenes del tipo SOV y OSV.

1.6. Definiciones

A continuación, estableceremos algunas definiciones básicas dentro del mundo de las lenguas de señas.

1.6.1. Lengua de señas

Corresponde a una lengua natural, es decir, un sistema de comunicación humana creado de manera espontánea por un grupo de personas con la intención de comunicarse entre sí, específicamente utilizado por personas sordas o aquellos que manejen la lengua. Se caracteriza por utilizar un canal gesto-visoespacial y por no disponer de una representación escrita oficial. Para esta última, en la actualidad se han propuesto diversas formas por parte de la comunidad científica, siendo más aceptada hasta el momento la "glosa" [26].

Si bien pueden existir similitudes entre lenguas de señas, estas presentan variaciones entre países o incluso entre ciudades de un mismo país.

1.6.2. Intérpretes de lengua de señas

Corresponden a la forma más básica de traducción, estas personas ya sea a través de estudios o experiencias personales, cuentan con un dominio de la lengua que les permite realizar traducciones en tiempo real. Son vistos principalmente en programas de televisión estatal, en eventos, entre otros. Sin embargo, la cantidad de intérpretes es acotada y su ubicación geográfica a veces puede dificultar su disponibilidad, además de que al tratarse de una persona, ésta sólo puede atender una situación a la vez.

1.6.3. Videos

Otra forma sencilla de traducción es mediante videos realizados por intérpretes o señantes de la lengua. Esta modalidad tiene la ventaja de que se mantiene el gesto en su totalidad, es decir, a través del video se ven claramente las expresiones faciales, el entorno gestual, además de mantener la fluidez o rapidez del gesto. Sin embargo, almacenar cada video correspondiente a un gesto es algo sumamente costoso en memoria y no toda la información que posee es realmente

importante. Esto también ocurre si se desean realizar análisis sobre las imágenes captadas en video para un tratamiento posterior.

1.6.4. Pose

Este término corresponde a la representación 2D que se obtiene a partir de una imagen de video, corresponde a una simplificación de una figura humana donde se extraen las extremidades y articulaciones como puntos clave (*keypoints*). En este caso, las poses no se utilizan para realizar traducciones directas, pero participan en el proceso intermedio de traducción, presentando ventajas frente a los videos, ya que permiten mantener una fidelidad relativamente alta del gesto a bajo costo en memoria, debido a que elimina la información externa. Como es un paso intermedio en el proceso de traducción, necesita complementarse con otros métodos.

1.6.5. Sistemas de representación escrita

El gesto debe entenderse como la unión de los movimientos corporales, faciales y la configuración espacial que crea el señante con su entorno. Se han propuesto conjuntos de símbolos con la intención de realizar una representación de un gesto, que pueda ser leído e interpretado por computador. Sin embargo, no ha tenido tanta aceptación dentro de las comunidades sordas debido a faltas en su estructura y definición.

1.6.6. Glosa

Este término hace referencia a un concepto asociado a una palabra en español, se suele escribir con letras mayúsculas y no corresponde a una traducción literal del español sino más bien a lo que significa esa expresión para las personas sordas. Así, por ejemplo, la glosa "BUENOS-DÍAS", si bien se compone de dos palabras en español, en LSCh corresponde a una sola palabra [24].

1.7. Etapas en la traducción de las lenguas de señas

Traducir la lengua de señas no suele ser generalmente una tarea sencilla, hay diversos métodos que podemos emplear sobre cómo obtener, por ejemplo, la representación de una glosa.

A continuación, se verán algunas de las etapas que han sido propuestas por diversos investigadores y cómo cada etapa también puede ser dividida en sub-etapas. No necesariamente en un orden específico o estricto, ya que dependerá del camino por el que queramos empezar.

1.7.1. Video a pose

Uno de estos caminos es tomar un video de una glosa, que corresponde a una representación 2D y procesarlo para transformarlo a una pose que corresponde a una representación 2D. En esta representación se establecen puntos clave o *keypoints*, que suelen estar en las articulaciones de las personas o en la cara, específicamente bordeando los ojos, la nariz y la boca. Sin embargo, en este caso se obtiene una imagen plana en la que se pierden algunas características de la lengua, como por ejemplo, la configuración espacial que pueden tener las glosas producidas por el señante. [27]

Se ha probado también la estimación de una pose en 3D, la cual además busca recrear cómo sería la posición del .^esqueleto"si existiese un plano Z. Esta prueba se realizó con OpenPose, que mediante un modelo de redes neuronales convolucionales, puede estimar la pose del cuerpo y manos [28]. De esta manera se obtienen keypoints, una representación de las partes del cuerpo en el espacio del vídeo en un cuadro de este.

Otras se enfocan en reconocer y clasificar a qué corresponde cada parte del cuerpo y cada área de píxeles en los cuadros del video y, posterior a eso, reconstruir la imagen 3D basándose en que estos puntos se ubiquen respetando la anatomía del ser humano [29].

1.7.2. Pose a glosa

En esta etapa tenemos que pasar de la representación de la pose, una secuencia de figuras que corresponde a un esqueleto ya sea bidimensional o tridimensional simplificado representativo de un gesto, a una caracterización textual de una seña. [27]

La cantidad de figuras bidimensionales o tridimensionales que se tengan de las poses será determinante en la calidad de la animación que presenten posteriormente, esto significa que, mientras más figuras se tengan, mejor será la representación de la glosa en cuanto a fluidez.

1.7.3. Video a glosa

Esto consiste en tomar un video donde un señante realice señas y asociarlo con su representación textual. Una de las técnicas realiza esto mediante el entrenamiento de un modelo que reconoce cada seña en cuestión, etiquetándola con la glosa correspondiente. [27]

1.7.4. Glosa a texto

Esto consiste en asociar una glosa con texto en idioma hablado, por ejemplo, el español. Esto supone la dificultad de que ambas estructuras, si bien son similares en el sentido de que las glosas pueden verse como "palabras"del español, asumir esto sería un error y se caería en lo que se suele llamar .ªpañol señado", que no tiene nada que ver con las lenguas de señas latinoamericanas o europeas cercanas. [27]

Una de las técnicas propuestas para realizar esto es mediante el procesamiento del lenguaje natural (NLP, por sus siglas en inglés), donde conviene realizar un análisis sintáctico, es decir, separar la oración en sus componentes gramaticales, lematización, que consiste en buscar la forma canónica de la palabra en español. Por ejemplo, el lema de un verbo sería su infinitivo como: çorriendo.º çorrí.ªs correr y el lema de "perros.º "perra.ªs perro. Además, sería necesario realizar la permutación de algunas palabras o la eliminación de éstas, obteniendo un resultado similar a las oraciones en lengua de señas.

1.7.5. Glosa a animaciones

Sin embargo, en este proyecto se utilizó otra técnica de traducción. En la que se crean animaciones correspondientes a una subanimación de la seña completa. El detalle de cómo se realiza será mencionado en capítulos posteriores.

2. Animaciones

2.1. Inicios de la animación 3D

Debido a que este trabajo utiliza animación 3D, profundizaremos específicamente en esta. La animación 3D representa hoy día cuánto se ha avanzado tecnológicamente en la representación digital de cuerpos con volúmenes por computadora.

2.1.1. Stop-motion

En sus inicios, específicamente para la creación de películas 3D, se utilizaba la técnica del *stop motion*, que corresponde a un método audiovisual de animación en el que se utilizan objetos a los que se les van tomando fotografías, modificando levemente su posición en cada una de estas. Al unir estos fotogramas en una película, se logra dar movimiento al objeto en cuestión.

Las primeras películas en las que se utilizó esta técnica datan de fines del siglo XIX, inicios del siglo XX. A esto se suma una variante en la cual, los "objetos", se construían de materiales como la arcilla o plasticina llamada *claymotion*.

Si bien las películas realizadas con esta técnica poseen un valor estético reconocible, construir los modelos no es tarea sencilla. Las producciones de estas cintas pueden tardar muchos años. Además, para lograr mayor fluidez se requiere un alto número de fotogramas.

2.1.2. CGI (Computer Generated Imagery): Historia y técnicas

Las imágenes generadas por computador, fueron otro avance en la animación 3D. Los inicios de esta tecnología se remontan a los simuladores de vuelo y Edwin Link, que en los años 1927 y 1929 diseñó su primer entrenador de vuelo llamado , sus esfuerzos fueron reconocidos por el ejército estadounidense que comenzó el desarrollo de simuladores más avanzados que incluyeran dispositivos tecnológicos, si bien inicialmente el simulador no incluía retroalimentación visual, un acercamiento a esto fue el ciclorama, que consistió en pintar las paredes de la sala de entrenamiento simulando una cabina. Sin embargo, en 1939 Link posteriormente desarrolló el .[en]trenador de navegación celestial"que permitía el entrenamiento nocturno y la navegación transatlántica. En este se creaba un ambiente realista proyectando unas estrellas sobre una cúpula que se encontraba sobre el simulador y éstas podían reubicarse para imitar los cambios horarios y de ubicación[30].

En 1878 y tras haberse unido a la empresa Singer,fue desarrollado el Singer-Link DIG o Generador de imágenes digitales en español. El cual es considerado dentro de la tecnología CGI, como un dispositivo perteneciente a la primera generación del mundo. Durante la década de los 80, con la guerra fría en curso, Singer-Link creó simuladores para diversos sistemas militares. Durante la década de los 90, desarrollaron simuladores para vuelos comerciales [30].

A la par que esto se desarrollaba, también ocurrían eventos importantes en simultáneo, por ejemplo en 1946 salió una de las primeras súper computadoras programables, ENIAC. El hecho de que éstas computadoras hayan podido programarse y reprogramarse para cumplir funciones diferentes abrió un mundo de posibilidades. Grace Hopper, la programadora de la versión comercial de ENIAC, en 1952 publicó un artículo en el que comentaba los conceptos generales de la traducción de lenguajes y compiladores. Esto creó un entorno que aumentó el universo de aplicaciones informáticas y usuarios.[31]

En 1954, John W. Backus propuso a IBM utilizar el lenguaje de programación FORTRAN, con la intención de facilitar el cálculo científico y el manejo de fórmulas numéricas. En 1964, John G. Kemeny y Thomas E. Kurtz inventaron el lenguaje BASIC, éstos lo compartieron con todo aquel que quisiera aprender a programar. Desde aquí, los programadores comenzarían a desarrollar muchos otros lenguajes.[31]

Sin embargo, todos estos avances no se quedaron sólo en las manos de programadores, científicos y matemáticos. En conjunto se comenzó a gestar un interés por el desarrollo del arte. Las imágenes que se generaban como resultado de las investigaciones de fórmulas complejas resultaron en el interés de grupos de artistas. De hecho, se le atribuye a William Fetter, un diseñador gráfico, como el inventor del término "Gráficos por computadora". Sus trabajos en Boeing Aircraft Co, empresa para la que trabajaba, se centraban en la composición de figuras humanas. Uno de los trabajos más reconocidos y recordados de los inicios de los gráficos por computadora corresponde al "Boeing Man.º "First Man", imagen que podemos ver en la Figura 9, tomada de Proceedings SPIE 166 (1978) [31].

Sólo era cosa de tiempo para que la animación por computadora comenzara a avanzar y terminara convirtiéndose en algo cada vez más común, despertando el interés de diversas personas de distintas áreas de la tecnología y el arte.

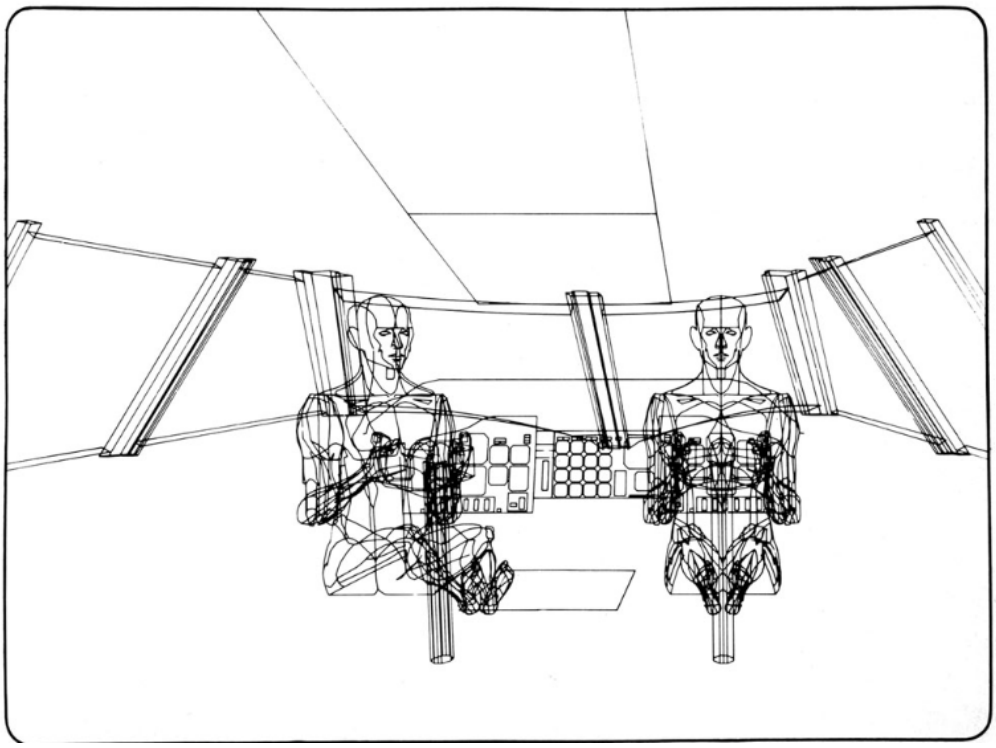


Figura 9: "First Man"diseñada por Fetter en 1968.

2.2. Programas de animación disponibles en el mercado

Entre el software de animación utilizado en esta memoria se consideraron Godot, Blender y Unity. Sin embargo, según un artículo de ESIC [32], hay 10 programas que son utilizados comúnmente:

- Autodesk Maya
- Autodesk 3D Max
- Blender
- Houdini
- Cinema 4D
- Sketchup
- Zbrush

  Sin embargo, muchos de estos programas son de pago, poseen varias opciones en cuánto a modelado, renderizado, etc. que no siempre son sencillas de utilizar o entender para un usuario principiante.

En este trabajo se eligió Blender como aplicación para llevar a cabo la tarea de generar las animaciones, ya que es un software libre y de código abierto que además posee un gran número de tutoriales disponibles en la red.

2.3. Blender

Blender es un software multiplataforma que se ejecuta en Linux, macOS y Windows. Es reconocido principalmente por su utilidad en la generación de animaciones y modelado 3D, pero también tiene herramientas que permiten realizar texturización, renderización, edición de video, creación de animaciones 2D [33]

2.4. Unity

Unity es un motor de videojuegos multiplataforma, disponible para Linux, macOS y Windows. Este software permite crear, diseñar y generalmente hacer funcionar un videojuego. Sin embargo, para este trabajo se utilizó para la automatización de la ejecución de las animaciones.

Debido a las bibliotecas de comunicación por HTTPS, se utilizó además para establecer las conexiones del servidor y el cliente.

3. Introducción general

En la actualidad, diversos avances en la tecnología han permitido que existan más recursos computacionales y de animación. Estas herramientas se pueden poner a disposición de la población para generar mayor integración de los grupos más marginados.

Dentro de estos grupos se encuentran las personas con discapacidad auditiva, parcial o total, que utilizan la lengua de señas chilena como forma principal de comunicación. Diversas son las dificultades que se presentan en la vida de estas personas, impactando negativamente en su entorno social, cultural, educativo, etcétera, por ende, en su calidad de vida.

La lengua de señas chilena posee la misma naturaleza que lenguas como el inglés o el español, es decir, evoluciona con el tiempo, posee su propia estructura gramatical y se generan dialectos entre grupos de poblaciones separadas geográficamente. Sin embargo, debido a la complejidad gramatical que presenta por su naturaleza viso-gestual y la escasez de estudios lingüísticos realizados en el país sobre esta, es difícil generar una caracterización que permita realizar traducciones como las que vemos actualmente en traductores online [26].

Algunos proyectos que realizan traducciones de lengua de señas hechos por otros países almacenan cada gesto correspondiente a una palabra, modificando principalmente la forma en cómo generan el gesto, es decir, si se realiza la animación mediante un programa de animación 3D [34] o utilizando un intérprete con un dispositivo de captura de movimiento [35]. Esta forma, además de utilizar gran cantidad de espacio en memoria, es poco versátil, ya que las señas sólo son utilizadas para una lengua específica.

En esta memoria se busca crear un banco de sub-gestos similar a un "alfabeto viso-gestual" que permita componer gestos a partir de sub-gestos realizados mediante animaciones 3D. Los sub-gestos se obtienen a partir de similitudes encontradas en la lengua que permiten agrupar gestos en base a su significado o la similitud en la que estos se realizan, es decir, mismo movimiento de brazo izquierdo, derecho, repetición de un sub-gesto en un gesto, etc.

3.1. Objetivos generales

Generar un recurso digital que facilite la comunicación con personas Sordas utilizando como metodología sub-gestos compuestos que permitan asociar una glosa en LSCh con una palabra en español.

3.2. Objetivos específicos

- Generar interfaz de usuario (terminal) para solicitar desplegar animaciones
- Generar animaciones para alimentar la base de datos del programa principal.
- Escribir código que genere conjuntos de las animaciones para formar una frase ingresada en la interfaz.
- Clasificar en Json animaciones por sub gestos.

3.3. Alcances

- Se realizará la animación de una "frase simple" compuesta por 1 o 2 sub-gestos.
- Se realizará la animación de una "frase compleja" compuesta por más de 2 sub-gestos.

3.4. Limitaciones

- No se incluye en este informe la animación de expresiones o gestos faciales.
- No se incluye en este informe la animación de miembros inferiores del cuerpo, cómo cadera, piernas y pies.

4. Marco teórico

4.1. Animaciones

Para realizar las animaciones se utilizaron dos programas: Blender y Unity. Los conceptos mencionados a continuación están directamente relacionados con estos y han sido tomados de la docuAutodesk Maya.

4.1.1. Modelado

El modelado de un objeto consiste en crear una representación 3D de un objeto real o ficticio mediante la agrupación de polígonos para recrear la forma del cuerpo. Estos polígonos pueden rotarse, moverse, extenderse y deformarse, lo que permite que se vea un movimiento de este cuerpo en el espacio virtual si estos cambios se realizan durante el tiempo. [36]

Es importante notar que el modelado 3D no sólo implica la forma del cuerpo humano, también incluye atributos como el pelo, ropa, etc. Sin embargo, considerando el caso de este trabajo, los objetos a modelar se tratan de cuerpos humanos y el objetivo es que estos puedan interpretar señas; sólo se considerarán los aspectos relevantes al movimiento de brazos, manos y dedos.

Los *"Bones"* o huesos por su traducción en español, son los elementos base que componen un *"armature"* o esqueleto de un personaje. Un armature en Blender funciona de manera similar a como lo hacen los esqueletos en la vida real. Estos dan soporte al cuerpo y determinan los puntos donde ocurren las flexiones del modelo, como se puede ver en la Figura 10 (b). Por otra parte, los puntos sirven como ejes para las rotaciones de estos huesos. También, el armature es un tipo de objeto utilizado para el rigging en el programa.

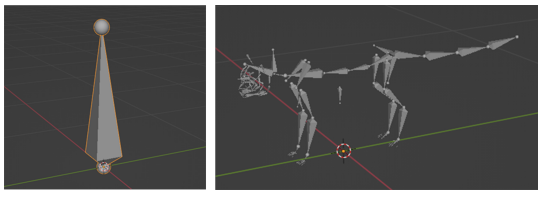


Figura 10: Bones y Armature en Blender. (a) hueso individual básico. (b) esqueleto de un gato. El hueso individual básico también puede considerarse un armature.

Por otra parte,a los tipos de rotaciones que pueden realizar estos huesos sobre un eje, se

les llama *Degrees of Freedom*.

4.1.2. Rigging

Es una técnica de animación en la que un modelo es compuesto por huesos, donde cada hueso es una transformación tridimensional de una pose por defecto. El rig en Blender corresponde a todos los controles que nos permitirán mover a un personaje en una animación. En la Figura 11 se puede ver el rig generado en el esqueleto de un gato. Las líneas en naranja simbolizan las áreas donde se puede realizar un movimiento o la rotación de una articulación.

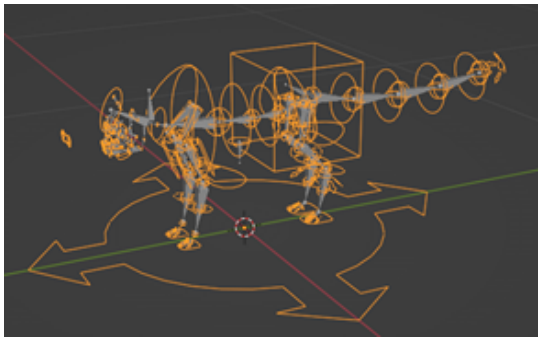


Figura 11: Rig correspondiente a un esqueleto gatuno.

4.1.3. Keyframes

Un *keyframe* o fotograma clave es un marcador de tiempo que almacena el valor de una propiedad, por ejemplo, en Blender, una animación se compone de varios fotogramas en los cuales está guardada la información correspondiente a la ubicación de los huesos. Posteriormente, el programa realiza la interpolación de estos puntos, generando una sensación de movimiento más fluida en el resultado final.

4.1.4. Inverse Kinematics (IK)

La cinemática inversa facilita el trabajo que implica la animación de un esqueleto. Por ejemplo, al mover un hueso padre, los huesos hijos conectados a este deberán moverse siguiendo el movimiento del hueso padre. Sin la cinemática inversa, los huesos de todo el brazo deberían animarse uno a uno de manera manual.

4.1.5. Layers

Las capas o *layers* son una herramienta en Unity que permite manejar máquinas de estado complejas para animaciones de diferentes partes del cuerpo. Por ejemplo, en este trabajo se utilizó junto a una máscara para separar el brazo derecho del brazo izquierdo.

En la Figura 12 se puede ver el brazo izquierdo en verde, lo que significa que sólo está activada la movilidad para esta extremidad. Esta máscara será aplicada sobre la capa que contiene una de las máquinas de estado que conforman la animación.

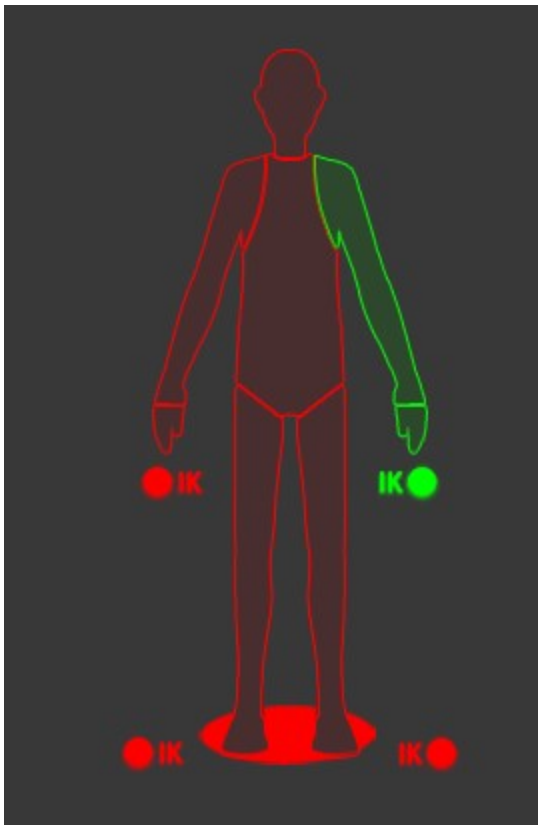


Figura 12: Máscara aplicada al lado izquierdo.

De esta forma, las capas nos permiten separar las acciones que queremos que ocurran en simultáneo, pero permitiéndonos animar cada extremidad del cuerpo de manera diferente.

4.2. JSON

Es un formato de texto, su nombre deriva de sus siglas en inglés JavaScript Object Notation (notación de objeto de JavaScript), se utiliza para el almacenamiento y transmisión de

datos mediante una estructura sencilla y fácil de leer. Los datos se agrupan de la forma ""clave":valor", donde el valor puede ser de tipo cadena de caracteres, números, booleanos, arreglos o null.

4.3. Servidor

Un servidor web corresponde a un programa informático que mantiene comunicación bidireccional o unidireccional con un cliente, realizando cambios sobre una aplicación en base a las peticiones realizadas por dicho cliente. La transmisión de datos se realiza mediante un protocolo donde HTTP corresponde al más común.

Flask, es un framework escrito en Python que permite crear aplicaciones web sencillas basadas en WSGI, una especificación de interfaz que determina cómo van a comunicarse el servidor y la aplicación web.

5. Desarrollo

A continuación, se mencionan las etapas que permitieron llevar a cabo la realización de este proyecto.

5.1. Introducción

En esta sección se detallan los procedimientos, programas y algoritmos utilizados en la realización de este traductor, abarcando la generación de las animaciones 3D que representan los gestos en lengua de señas chilena, los algoritmos que permiten la manipulación del programa de animación y los que componen el servidor y las uniones entre estos.

5.2. Análisis lengua de señas chilena

Es importante recalcar que la lengua de señas chilena no posee una traducción directa al español. El texto que se utiliza en este trabajo, llamado "glosa" y que suele escribirse en mayúsculas, es una representación de una acción, "la seña", que hace referencia a un concepto. La seña puede estar compuesta de varios movimientos realizados principalmente por los brazos, las manos y el rostro.

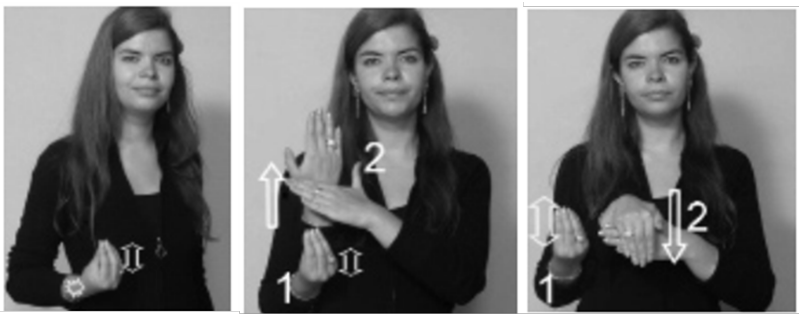
Las señas utilizadas en este trabajo se obtuvieron del diccionario bilingüe lengua de señas chilena - español Tomo I (A-H) y Tomo II (I-Z), los cuales fueron complementados con videos realizados por la Universidad Santo Tomás en sus Talleres DAE [37], por Roberto Muñoz Muñoz [38], por el Centro de Innovación – Mineduc [39] y por Lense Biobío Chile [40].

5.3. Identificación de sub-gestos

Se puede observar en los diccionarios y videos mencionados anteriormente, que una seña se compone de una o más acciones, por ejemplo, realizar cambios en la posición de brazos, manos o dedos, y que estas pueden repetirse en más de un gesto.

En este trabajo, a estos gestos que permiten obtener otros gestos se les denominó sub-gestos, y como se mencionó en el párrafo anterior, un sub-gesto puede formar parte de más de una seña. Dentro del universo de gestos que posee el diccionario, se trabajaron los sub-gestos de dos formas. Una de ellas considera los sub-gestos como movimientos ya sea de brazos o dedos que componen un gesto. Por ejemplo, elevar los brazos y abrir las manos, donde el

primero corresponde a un sub-gesto y el segundo a otro sub-gesto. La segunda forma considera señas que además de ser un gesto con un significado particular unido a otro gesto, cambian el significado del concepto. Por ejemplo, la seña "BUENO/A" existe por sí sola con un significado propio que hace referencia a una persona o algo con cualidades positivas, pero también forma parte de la seña "BUENOS-DIAS" cuyo significado corresponde a un saludo o despedida que se realiza por las primeras horas del día y "BUENAS-TARDES" que igual corresponde a un saludo o despedida que se realiza por las tardes. En la Figura 3.1 se puede ver cómo el gesto BUENO forma parte de las señas BUENOS-DIAS y BUENAS-TARDES simbolizado en (b) y (c) con el número 1.



a) b) c)

Figura 13: (a) Mujer realizando la seña BUENO/A. (b) Mujer realizando la seña BUENOS-DIAS. (c) Mujer realizando la seña BUENAS-TARDES.

5.4. Animaciones

Para crear las animaciones se utilizó un modelo humano diseñado en Blender realizado por Pixel_monster y CGDive [41] en conjunto con una herramienta disponible en el mismo programa llamada human meta-rig (ver en la Figura 14), el cual corresponde a un conjunto de cadenas de huesos relacionadas entre sí que permiten realizar movimientos y establecer puntos de inflexión en la animación simulando un esqueleto humano.

Figura 14: Modelo y human meta-rig antes de ser ajustados.

Ambos objetos fueron superpuestos para llevar a cabo los ajustes de tamaño y ubicación de los huesos en el modelo de manera que calzara acorde a la anatomía humana. Posteriormente, se utilizó la herramienta *generate rig* disponible en Blender, en la cual, el rig es unido al modelo generando las secciones que permiten controlar los movimientos como se puede ver en la Figura 15.

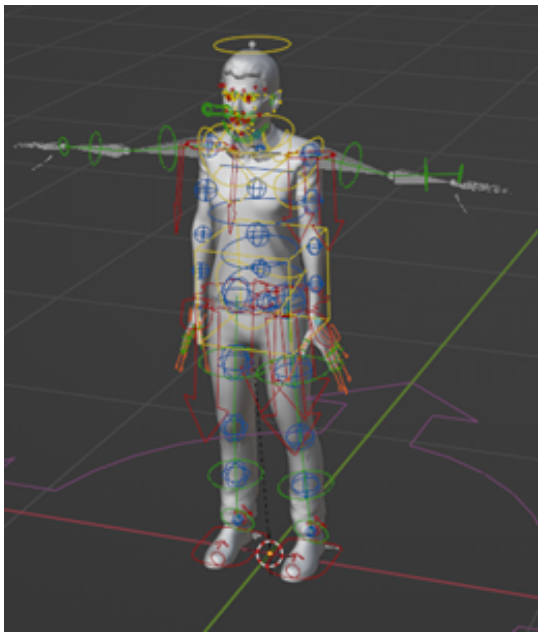


Figura 15: Modelo con el rig generado.

Con este proceso listo se da paso a la generación de las animaciones, aquí se va modificando la posición del modelo guardando los puntos clave que la componen en un *frame*, esto se realiza varias veces hasta completar la seña que se desea animar. Al finalizar, la animación resultante es guardada en un archivo con extensión *fbx*.

Figura 16: Sub-gestos cargados en el programa Unity.

Estos archivos *fbx* se cargan en el programa Unity en la carpeta de *Assets/Resources* como muestran en la Figura 16. Aquí, son utilizados por el programa Unity, que reproduce la animación utilizando el modelo Sophie (ver Figura 17) disponible en Mixamo [42].

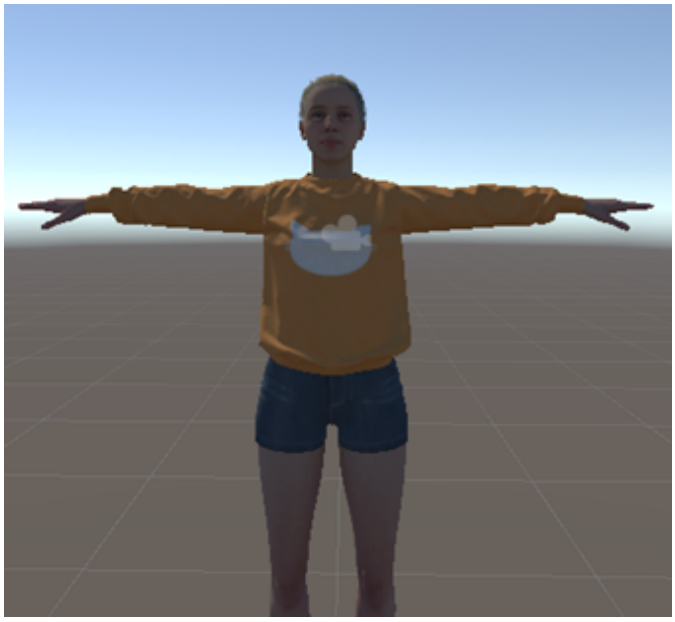


Figura 17: Modelo Sophie en programa Unity.

5.5. Arquitectura

5.5.1. Estructura de las animaciones

Para trabajar con las animaciones se creó una estructura que permita unir múltiples sub-animaciones que representan a los sub-gestos mencionados en la sección anterior y que puedan ser interpretadas por los programas elaborados en este trabajo. Para establecer las diferencias entre cada animación se utiliza como clave el nombre del gesto. Debido a que el gesto está compuesto de sub-gestos, se hizo necesario incluir un arreglo de sub-animaciones con la información de cada una de estas.

Un sub-gesto puede ser realizado por cada brazo, por ejemplo, la mano y el brazo izquierdo hacen un movimiento circular mientras que la mano derecha se mantiene fija al frente; por esta razón se tomó la decisión de separar las animaciones de los movimientos correspondientes a cada brazo. En Unity existen los layer targets que permiten separar las animaciones con la ayuda de una máscara en la que se seleccionan las secciones con las que se va a trabajar, en este caso se utilizaron dos: una para el brazo derecho y otra para el brazo izquierdo.

Para establecer las diferencias entre cada sub-animación se utiliza como clave el nombre del sub-gesto correspondiente a un archivo fbx, que en este trabajo fue creado en Blender.

En Unity las animaciones tienen un tiempo de inicio y duración que depende de la velocidad en la que es reproducido, por lo que también fueron incorporados dentro de la estructura. Considerando lo anterior, se definió la siguiente estructura general para poder reproducir las animaciones con los siguientes campos:

- name: contiene el nombre de la seña.
- subAnimations: contiene un arreglo de objetos con información de las animaciones que lo componen.

  Cada objeto de subAnimations contiene los siguientes campos:

- name: contiene el nombre de la sub-animación.
- layerTarget: indica que máquina de estado del programa Unity va a utilizar.
- startTime: contiene el tiempo en que inicia la animación
- speed: indica la velocidad de reproducción de la animación

Con el objetivo de almacenar los gestos y establecer la secuencia de sub-gestos que los componen, se creó un banco de animaciones en el que, al ingresar la glosa, esta entrega la

animación como un archivo JSON.

```
{
    "name": "buenos-dias",
    "subAnimations": [
        {
            "name": "bueno",
            "layerTarget": ["leftArm","rightArm"],
            "startTime": 0,
            "speed": 1
        },
        {
            "name": "dia_der",
            "layerTarget": ["rightArm"],
            "startTime": 59,
            "speed": 1
        },
        {
            "name": "dia_tarde_Izq",
            "layerTarget": ["leftArm"],
            "startTime": 59,
            "speed": 1
        }
    ]
}
```

Sección de código 1: Contenido de archivo JSON correspondiente a la glosa BUENOS-DIAS.

En la sección de código 1 podemos ver un ejemplo de la estructura que se mencionó anteriormente. Para empezar, tenemos el nombre de la glosa BUENOS-DIAS que está compuesta de tres sub-animaciones: "bueno", que está presente en ambas capas y corresponde a la sub-animación inicial por eso su startTime es cero; "dia_der", que está presente sólo en la capa derecha cuyo inicio es en 59 y "dia_tarde_Izq", que está presente sólo en la capa izquierda y cuyo inicio también es en 59, ya que deben reproducirse al mismo tiempo. El atributo startTime de las dos sub-animaciones que están después de "bueno" se debe a que ambas deben esperar a que esta termine de reproducirse, es decir, el valor numérico 59 corresponde a la duración de la sub-animación "bueno". Los nombres indican los gestos en los que participan y el brazo por el que son realizadas; en el caso de este ejemplo "dia_der" participa en la realización del gesto

DIA y es realizado por la mano derecha. Y para el caso de "dia_tarde_Izq" esta sub-animación participa en la realización del gesto DIA, pero también en el gesto TARDE y es realizada por la mano izquierda. La velocidad de reproducción no es modificada, por lo tanto, el atributo speed se mantiene en 1 para todas las sub-animaciones.

5.6. Servidor

Con el fin de poder automatizar el proceso de reproducción de las animaciones a través del programa Unity, se revisaron las alternativas disponibles, dentro de las que se encuentran utilizar bibliotecas no oficiales para realizar la comunicación por medio de sockets o la biblioteca oficial de Unity llamada Transport. Sin embargo, por simplicidad en este trabajo se optó por usar UnityWebRequest junto con Flask para poder realizar la conexión entre el terminal con el que interactúa el usuario, el programa Unity y el banco de animaciones [43].

El servidor posee dos rutas /serve_gloss y /play_animation, la primera corresponde a la ruta que establece el terminal con el servidor y la segunda es la ruta que se establece entre el servidor y el programa Unity. Cuando se realiza la petición de las animaciones, estas se van almacenando en la ruta /play_animation en una cola llamada animations, en la cual se hace la operación pop al hacerse una petición en la ruta /serve_gloss. Código del Servidor se puede consultar en el Anexo B.

5.7. Cliente Unity y Cliente Terminal

Al usuario ingresar la glosa que desea animar en el terminal, se crea una petición HTTP la cual se comunica con el servidor en Flask. A la vez, el programa Unity está constantemente realizando polling a la ruta /serve_gloss en la que espera recibir la información de la sub-animación para reproducirla. El servidor consulta la información de la glosa ingresada por el usuario en el banco de animaciones de donde la obtiene, la interpreta y la envía al programa Unity como se puede ver en el diagrama de la Figura 18. Código del Cliente Terminal se puede consultar en el Anexo A.

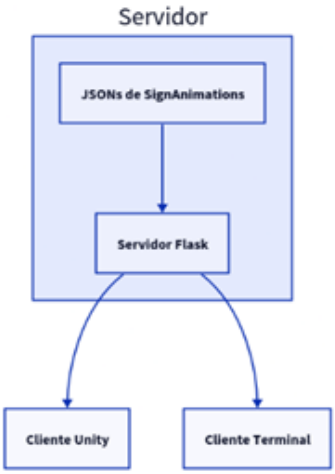


Figura 18: Diagrama del sistema.

5.8. Programa en Unity

La conexión del programa con el servidor se realiza a través de una petición HTTP en la que se recibe un JSON con la información de un gesto ordenada como se muestra en la Figura 19 a la ruta /serve_gloss.

Luego se inicializa el AnimationControllerLayer encontrando las capas rigthArm y leftArm correspondientes a cada brazo de la animación. Esto nos permite modificar las máquinas de estado de cada layer en tiempo de ejecución.

Como se mencionó anteriormente, el cliente realiza polling para recibir nuevos gestos. Al ser recibido, se crea una máquina de estados utilizando el método addAnimation, donde son ordenados según el tiempo indicado en el atributo startTime. De esta forma, se van formando los subAnimations de manera independiente en las máquinas de estado de cada brazo, dando origen a una instancia de SignAnimations. Código del Programa en Unity se puede consultar en el Anexo C.

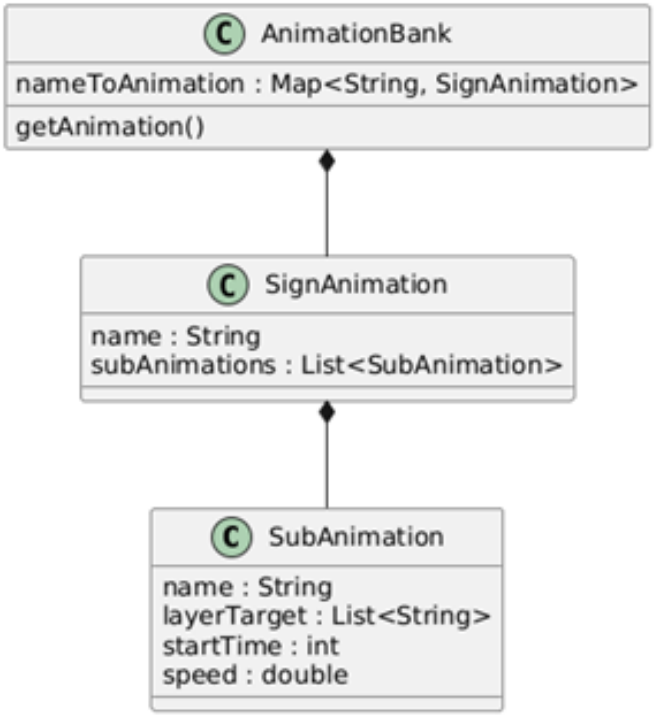


Figura 19: Diagrama de clases.

La relación entre las clases AnimationBank, SignAnimation y SubAnimation involucradas en el cliente y el servidor está reflejada en la Figura 19.

6. Resultados

Para probar la implementación de este trabajo crearon 14 subgestos y se realizaron dos tipos de frases, la frase simple que implica utilizar una glosa y la frase compleja que implica utilizar dos o más glosas. Cabe señalar que las frases utilizadas no corresponden a una traducción del español a la lengua de señas chilena, sino más bien de glosa a la representación gestual en LSCh y no necesariamente son correctas gramaticalmente.

La puesta en marcha de los programas se realiza a través de dos terminales del computador, en la Figura 20 se puede ver la ejecución del servidor antes de iniciar la comunicación. El programa en funcionamiento se puede ver en [44].

Figura 20: Terminal ejecutando el servidor.

La otra terminal se utiliza como una interfaz que, al ejecutarse, permite al usuario ingresar el nombre de las glosas, ya sea escribiendo el nombre de cada una de manera individual o construyendo una oración con ellas, como se puede ver en la Figura 20.

Figura 21: Terminal ejecutando la interfaz de línea de comando.

Finalmente, se ejecuta el proyecto en Unity como se puede ver en la Figura 21, en donde también podemos observar que en la consola del programa aparecen mensajes correspondientes a las peticiones que realiza, esperando recibir una glosa.

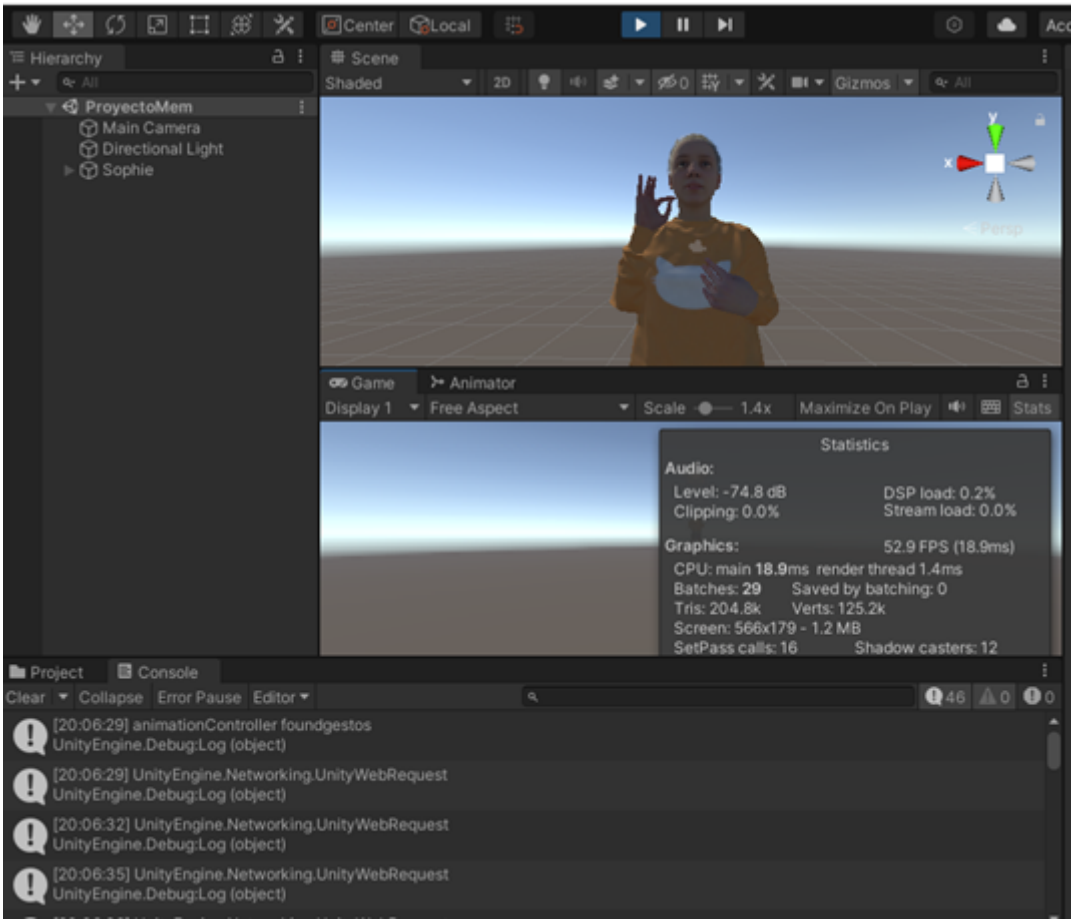


Figura 22: Captura del proyecto funcionando en el programa Unity.

6.1. Frase simple

Para este ejemplo se utilizaron dos frases simples ya vistas anteriormente en el trabajo, correspondientes a: “BUENOS-DIAS” y “BUENAS-TARDES”.

Ambas están compuestas por el gesto “BUENO”, pero también comparten en común la posición del brazo izquierdo, ya que éste se mantiene durante la realización de la seña, siendo el brazo derecho el que representa la diferencia entre ambas.

Para representar estas frases se utilizaron las sub-animaciones correspondientes a “bueno”, “día_der”, “tarde_der” y “día_tarde_Izq” de esta forma nos ahorramos el tener que repetir el movimiento realizado por el brazo izquierdo en ambas frases y el “bueno” que también está

incluido en ambas frases.

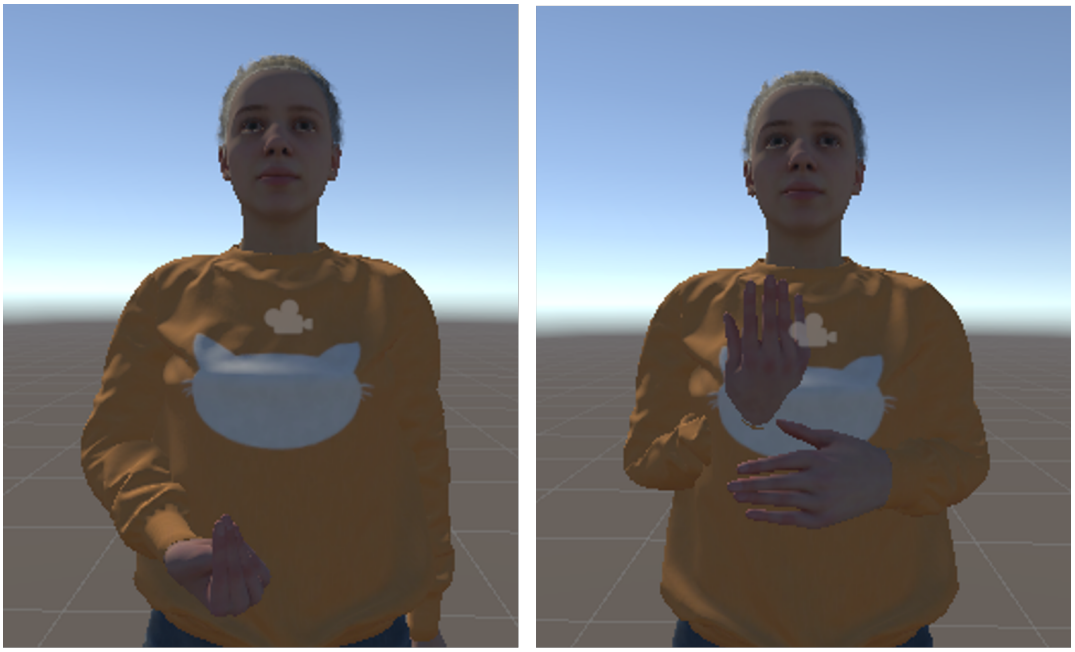


Figura 23: Seña BUENOS-DIAS realizada en Unity.

En la Figura 23 se muestran capturas del programa Unity donde Sophie realiza las señas BUENO (a) y DIA (b) que componen la seña BUENOS-DIAS.

6.2. Frase compleja

En este ejemplo se formaron dos oraciones compuestas de cuatro glosas cada una correspondientes a: “YO BAILAR CUECA FONDA” e “HIJA JUGAR ABUELA PATIO”.

Para representar la primera oración se utilizaron las sub-animaciones: “yo”, “bailar”, “cueca”, “pieza”. En este caso, las glosas CUECA y FONDA comparten la sub-animación cueca, por lo que ahorramos representar este gesto nuevamente.

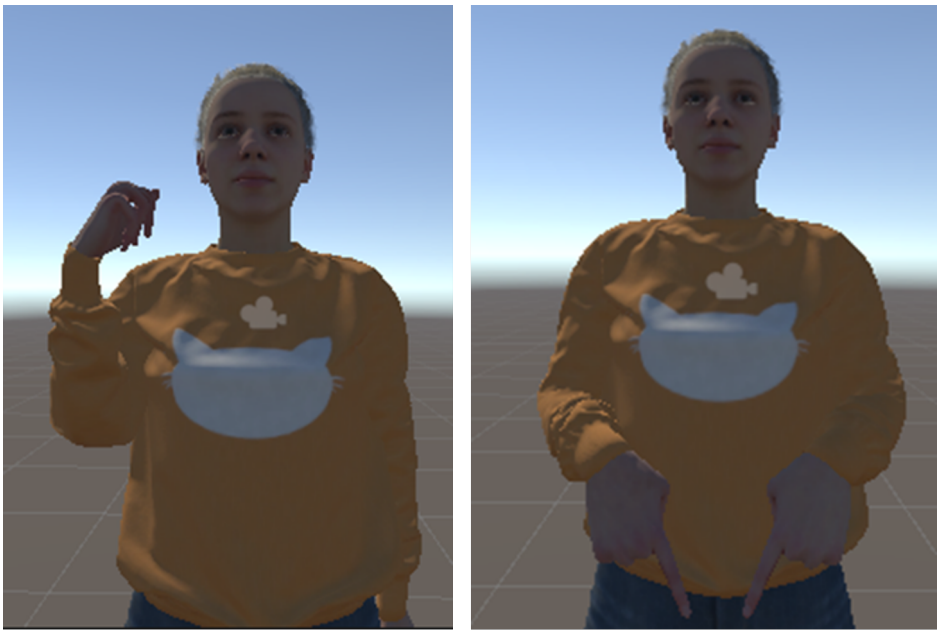
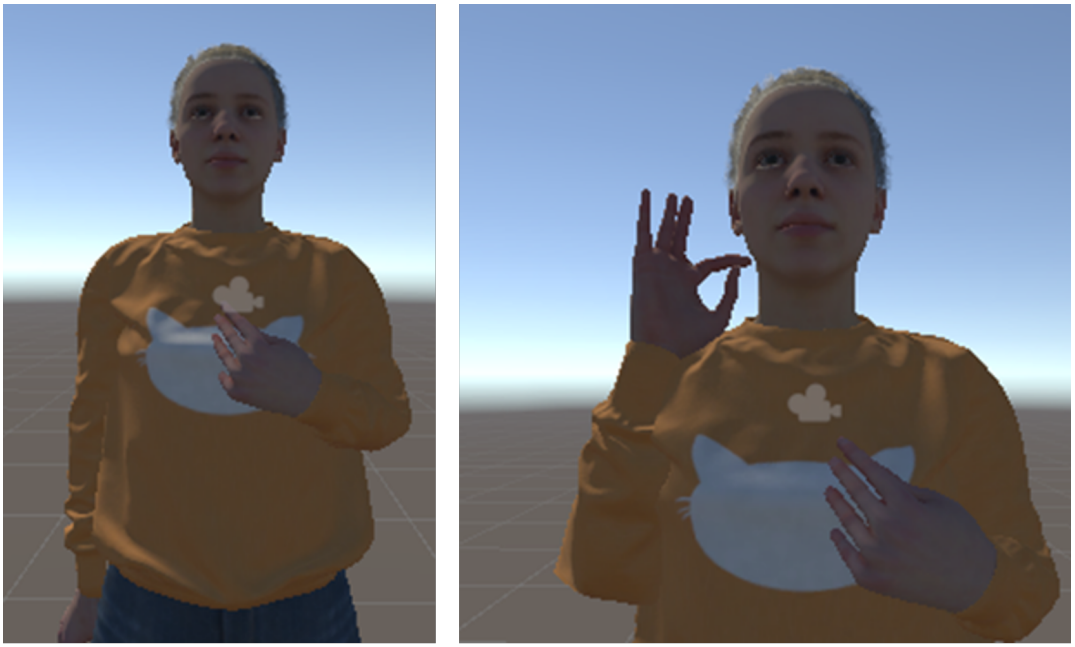


Figura 24: Seña FONDA realizada en Unity.

En la Figura 24 se muestran capturas de Sophie realizando las señas CUECA (a) y PIEZA (b) que componen la seña FONDA.

Mientras que en la segunda oración se utilizaron las sub-animaciones correspondientes a: “hij”, “mujer”, “jugar”, “abue”, “pieza”. En este caso HIJA y ABUELA son señas compuestas que tienen en común la seña MUJER, pero también ocurre esto con la seña PATIO, ya que PATIO se compone de la seña JUGAR y PIEZA.



a) b)

Figura 25: Seña HIJA realizada en Unity.

En la Figura 25 se muestran las capturas de Sophie realizando las señas correspondientes a HIJO/A (a) y MUJER (b) que componen la seña HIJA.

A continuación, se muestran en las Figuras 26 y 27 el comportamiento de los programas al recibir frases correspondientes a oraciones compuestas de una o más señas.

Figura 26: Terminal usuario con frases ejemplo.

Es necesario recalcar que en el terminal deben ingresarse las frases en estilo glosa, aunque por simplicidad se permite el uso de minúsculas para representarlas. Es necesario incluir el

guión que las une, como es en el caso de “buenos-dias” y “buenas-tardesçomo se puede ver en la Figura 26

En el caso del servidor, este se encuentra realizando peticiones constantemente. En la Figura 20 se puede notar que en el mensaje aparece la numeración 404 y 200, esto corresponde a los códigos de respuesta HTTP “Not Found” y “OK” respectivamente [45]. Donde, en caso de obtener 404, implicaría que no se ha recibido la animación y 200 si ésta se recibió correctamente.

También se ve cómo el número 200 se repite seguidamente, porque en el caso de las oraciones compuestas por más glosas, éstas se dividen y cada resultado de animación se envía al servidor individualmente.

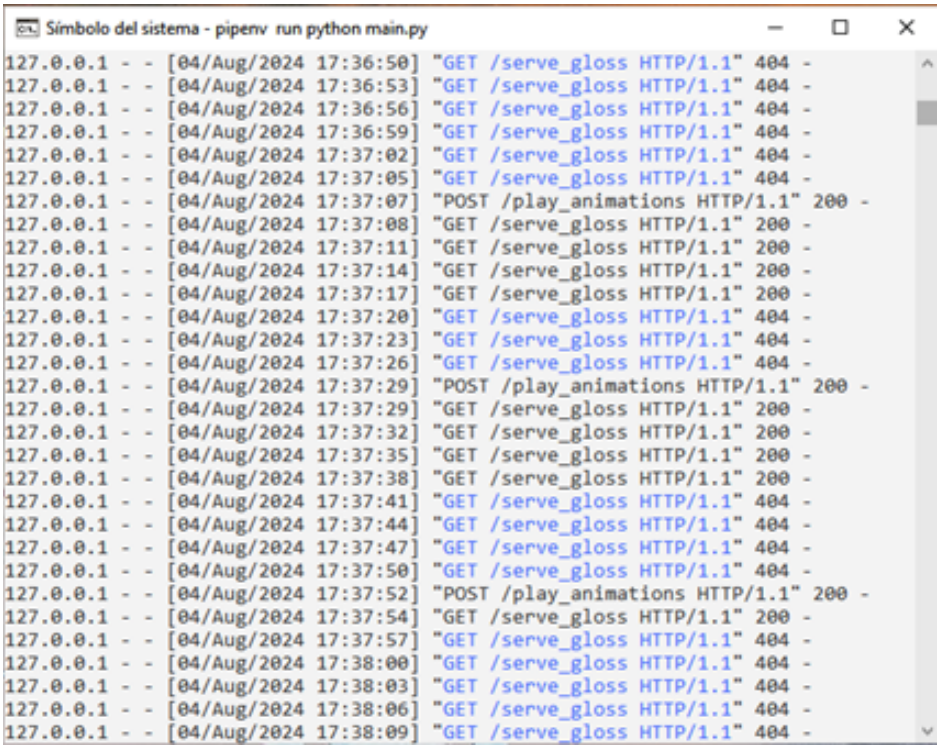


Figura 27: Captura del servidor.

7. Revisión de Resultados

Por medio de lo expuesto en este trabajo podemos considerar que la alternativa propuesta de construir gestos a partir de sub-gestos es realizable y presenta una ventaja al reutilizar sub-animaciones en la composición de los mismos, todo esto debido a la estructura que posee la lengua de señas que permite formar "palabras"mediante la unión de .ºtras palabras"de manera similar a los lexemas que se utilizan en el español. También, esta forma es beneficiosa cuando se trata de señas que requieren de la repetición de una misma acción, es decir, aquí podemos reutilizar una sub-animación el número de veces que se requiera.

La versatilidad de la estructura de almacenamiento mostrada en este trabajo hace relativamente simple agregar nuevos gestos, así como establecer otros significados.

La calidad de la animación es un punto que necesita bastante mejora ya que al realizar algunas señas la animación se "solapa.º la altura del gesto es incorrecta, como se puede ver en la Figura 28. Por ejemplo, al realizarse la seña "MUJER", las manos deben tocar la oreja, imitando el acto de tocar un aro que se encuentre, por lo que podemos observar que la mano no alcanza a llegar a la oreja y se realiza el gesto más abajo.

Figura 28: Seña MUJER con errores de animación

A continuación, se muestra en la tabla 1 donde se realiza un recuento de cuántas frases se pueden formar uniendo 10 de los sub-gestos creados dispuestos a la izquierda; las veces que es reutilizado un subgesto dentro de la misma frase es indicado con un número a la izquierda del *.

subgesto	1	2	3	4	5	6	7	8	9	10	11	12	...
YO	*	*	*	*	*	*	*				*		...
JUGAR	2*	2*	2*	2*			*	*	*	*			...
BAILAR					*	*					*	*	...
HIJ			*	*	*	*	*	*	*	*		*	..
ABUEL	*	*							*	*		*	...
MUJER	*		*		*		*		2*	2*		*	...
HOMBRE		*		*		*		*				*	...
PIEZA	*	*		*	*	*	*	*			*	*	...
CUECA					2*						2*	2*	...
TECLEAR							*	*					

Cuadro 1: Tabla de cantidad de frases con que pueden formar los subgestos

Por ejemplo:

- **Frase 1**: YO JUGAR ABUELA PATIO, donde ABUELA se compone de ABUEL + MUJER y PATIO se compone de PIEZA y JUGAR. Cómo JUGAR se repite dos veces, aparece marcado en la tabla como 2*.
- **Frase 4**: YO JUGAR HIJA PATIO, donde HIJO se compone de HIJ + HOMBRE y PATIO se compone de PIEZA y JUGAR. Cómo JUGAR se repite dos veces, aparece marcado en la tabla como 2*.
- **Frase 5**: YO BAILAR CUECA HIJA FONDA , donde HIJA se compone de HIJ + MUJER y FONDA se compone de PIEZA y CUECA. Cómo CUECA se repite dos veces, aparece marcado en la tabla como 2*.
- **Frase 7**: YO JUGAR HIJA OFICINA, donde HIJA se compone de HIJ + MUJER y PATIO de JUGAR y PIEZA.
- **Frase 9**: ABUELA JUGAR HIJA, donde HIJA se compone de HIJ + MUJER y ABUELA de ABUEL + MUJER. Cómo MUJER se repite dos veces, aparece marcado en la tabla como 2*.

De esta manera podemos seguir armando frases.Notar que el orden de los subgestos que forman un gesto es importante, como por ejemplo, para formar el gesto ABUELA, el subgesto ABUEL debe ir primero que MUJER, y en OFICINA el subgesto PIEZA va primero y luego TECLEAR.

Sería interesante además, abordar la posibilidad de construir un "traductor completo". Realicé este esquema en mis trabajos anteriores que se puede ver en la figura 29 como un resumen de lo que implicaría llevar a cabo esta idea. La sección que abarca este trabajo se concentra sólo en una parte del cuadro amarillo, que puede ser mejorada en acompañamiento de artistas especialistas en animación, lingüistas y personas sordas.

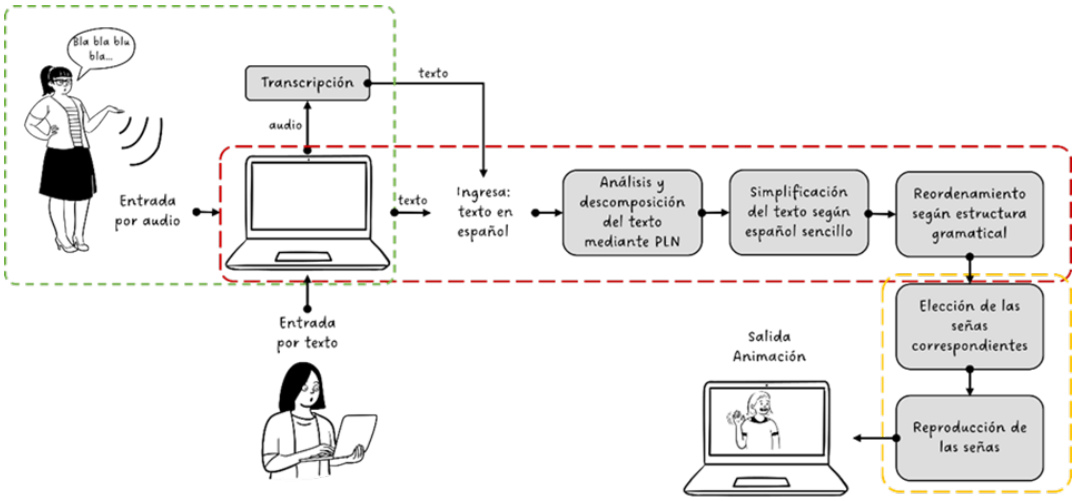


Figura 29: Diagrama de solución propuesta completa con diferentes etapas señalizadas en verde, rojo y amarillo.

8. Conclusión y trabajo futuro

Este trabajo es una primera aproximación al problema, por lo que queda realizar la construcción de un banco de sub-animaciones más completo, que permita aumentar el número de señas que se puedan representar mediante sub-señas. También sería importante incluir gestos realizados por el rostro o sub-gestos que incluyan otras partes del cuerpo, como las piernas o la postura corporal completa, cómo sería el caso para la seña correspondiente a "pingüino", que se vio en la figura 7.

Se podría mejorar la calidad de animación, eligiendo parámetros más adecuados al momento de utilizar el IK y las formas de interpolación que incluye el programa o aumentando el número de frames entre cada *keyframes*.

También sería interesante abordar si es factible aplicar esta metodología a otras lenguas de señas aprovechando la similitud existente entre las lenguas viso-gestuales, recordar que nuestra lengua comparte orígenes con las lenguas de la familia hispano-francesa, por lo que comparten gestos, mas no siempre el mismo significado, pero esto permitiría de igual manera la reutilización de gestos para ampliar el banco a uno más internacional.

Se considera muy importante generar mayor concientización respecto al estudio de la lengua de señas chilena que permita crear herramientas tanto lingüísticas como tecnológicas, para poder integrar en su totalidad a las personas sordas y que su condición no sea un impedimento para desarrollarse como cualquier otro individuo dentro de esta sociedad, facilitando su acceso a derechos tan importantes como lo son la salud y la educación.

Anexo A. Cliente Terminal

```
import requests  # Importa la librería requests para realizar solicitudes HTTP
while True:
    play_animations = input()  # Solicita al usuario que ingrese animaciones
    #Envía una solicitud POST a la URL especificada con un JSON que contiene las
    ↪  animaciones ingresadas
    requests.post("http://127.0.0.1:5000/play_animations", json={"animations":
    ↪  play_animations.split()})
```

Anexo B. Servidor

```python
# Importación de bibliotecas necesarias
from flask import Flask, jsonify, request, abort  # Importa Flask y funciones para
↪  manejar respuestas JSON
from copy import copy # Importa copy para crear copias de objetos si es necesario
from dataclasses import dataclass, asdict  # Importa dataclasses para facilitar la
↪  creación de clases
import os  # Importa os para interactuar con el sistema operativo
import json  # Importa json para manejar datos en formato JSON
from collections import deque  # Importa deque para utilizar colas

# Inicialización de variables globales

animationBank = None  # Variable que almacenará un banco de animaciones
animation = deque()  # Cola para gestionar las animaciones a reproducir
app = Flask(__name__, static_url_path='')  # Inicializa la aplicación Flask

@app.route("/serve_gloss")  # Define la ruta para servir glosas
def json_gloss():
    global animation  # Accede a la variable global animation
    try:          # Intenta devolver el primer elemento de la cola como JSON
        return jsonify(asdict(animation.popleft()))  # Convierte el objeto a
        ↪  diccionario y lo devuelve
except IndexError:  # Maneja el caso en el que la cola esté vacía
    abort(404)  # Devuelve un error 404 si no hay elementos

@app.route("/play_animation", methods=["POST"])  # Define la ruta para reproducir
↪  una animación
def play_animation():
    global animation  # Accede a la variable global animation
    if request.method == "POST":  # Verifica si la solicitud es un POST
        receivedAnimation = request.json["animation"]  # Obtiene la animación del
        ↪  JSON recibido
        signAnimation = animationBank.getAnimation(receivedAnimation)  # Obtiene la
        ↪  animación del banco
        animation.append(signAnimation)  # Agrega la animación a la cola
return "ok", 200  # Devuelve un mensaje de éxito
```

```python
def play_animations():
    global animation  # Accede a la variable global animation
    if request.method == "POST":  # Verifica si la solicitud es un POST
        receivedAnimations = request.json["animations"]  # Obtiene la lista de
        ↪  animaciones del JSON recibido
        for receivedAnimation in receivedAnimations:  # Itera sobre cada animación
        ↪  recibida
            signAnimation = animationBank.getAnimation(receivedAnimation)  # Obtiene
            ↪  la animación del banco
            animation.append(signAnimation)  # Agrega la animación a la cola
    return "ok", 200  # Devuelve un mensaje de éxito


@dataclass
class SubAnimation:
    name: str  # Nombre de la subanimación
    layerTarget: list[str]  # Lista de capas a las que se aplica la subanimación
    startTime: int  # Tiempo de inicio de la subanimación
    speed: float  # Velocidad de la subanimación

    @classmethod
    def fromJson(cls, jsonAnimation):
        # Crea una instancia de SubAnimation a partir de un diccionario JSON
        return cls(
            jsonAnimation["name"],
            jsonAnimation["layerTarget"],
            jsonAnimation["startTime"],
            jsonAnimation["speed"]
        )

@dataclass
class SignAnimation:
    name: str  # Nombre de la animación de señal
    subAnimations: list[SubAnimation]  # Lista de subanimaciones asociadas

    @classmethod
    def fromJson(cls, jsonAnimation):
        # Crea una instancia de SignAnimation a partir de un diccionario JSON
        subAnimationsList = []
        # Itera sobre cada subanimación en el JSON
        for subAnimationjson in jsonAnimation["subAnimations"]:
            # Convierte el JSON de cada subanimación a un objeto SubAnimation
            subAnimationsList.append(SubAnimation.fromJson(subAnimationjson))

        return cls(jsonAnimation["name"], subAnimationsList)
```

```python
@dataclass
class AnimationBank:
    animationNameToSignAnimation: dict[str, SignAnimation]  # Mapa de nombres de
    ↪  animaciones a sus objetos SignAnimation

    def getAnimation(self, name: str):
        # Devuelve la animación de señal asociada al nombre dado
        return self.animationNameToSignAnimation[name]

def main():
    global animationBank  # Declara animationBank como global
    animations = {}  # Diccionario para almacenar animaciones

    # Recorre el directorio "animations" para cargar las animaciones
    for root, dirs, files in os.walk("animations"):
        for file in files:
            # Abre cada archivo JSON y lo carga como una animación de señal
            with open(f'animations/{file}', 'r') as f:
                signAnimation = SignAnimation.fromJson(json.loads(f.read()))
                animations[signAnimation.name] = signAnimation  # Agrega la
                ↪  animación al diccionario

    animationBank = AnimationBank(animations)  # Crea el AnimationBank con las
    ↪  animaciones cargadas
    app.run(debug=True)  # Inicia la aplicación en modo de depuración

if __name__ == "__main__":
    main()  # Llama a la función main
```

Anexo C. Programa en Unity

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Networking;
using UnityEditor; // Importa el espacio de nombres UnityEditor, necesario para
↪  AnimatorController.
using UnityEditor.Animations; // Importa el espacio de nombres
↪  UnityEditor.Animations para trabajar con AnimatorControllerLayer y
↪  AnimatorStateMachine.
using System.Linq; // Importa el espacio de nombres System.Linq para usar métodos
↪  como Count() y ToList().
using System; // Importa el espacio de nombres System, aunque en este caso no se usa
↪  explícitamente.

public class AnimationScript : MonoBehaviour
{
    // Variables privadas para almacenar el componente Animator y las capas del
    ↪  controlador de animación para los brazos.
    private Animator interprete;
    private AnimatorControllerLayer leftArm;
    private AnimatorControllerLayer rightArm;

    // Método llamado al inicio del script.
    void Start()
    {
        // Obtiene el componente Animator adjunto a este GameObject.
        interprete = GetComponent<Animator>();

        // Intenta obtener el AnimatorController del Animator. Necesita un cast a
        ↪  AnimatorController.
        var animatorController = interprete.runtimeAnimatorController as
        ↪  AnimatorController;

        // Comprueba si se encontró el AnimatorController.
        if (animatorController)
        {
            Debug.Log("animationController found" + animatorController.name);
        }
        else
        {
            Debug.Log("animationController not found");
        }
```

```
    // Itera a través de las capas del AnimatorController para encontrar las
    ↪  capas llamadas "leftArm" y "rightArm".
    for (int i = 0; i < animatorController.layers.Count(); i++)
    {
        if (animatorController.layers[i].name == "leftArm")
        {
            leftArm = animatorController.layers[i];
        }
        else if (animatorController.layers[i].name == "rightArm")
        {
            rightArm = animatorController.layers[i];
        }
    }

    // Inicia una corrutina para realizar una petición GET a la URL
    ↪  especificada.
    StartCoroutine(getRequest("http://127.0.0.1:5000/serve_gloss"));
}

// Método para agregar una animación basada en la información recibida.
void addAnimation(SignAnimation animation)
{
    // Inicializa listas para almacenar las sub-animaciones para cada brazo.
    List<SubAnimation> leftArmSubAnimation = new List<SubAnimation>();
    List<SubAnimation> rightArmSubAnimation = new List<SubAnimation>();

    // Itera a través de las sub-animaciones de la animación principal.
    for (int i = 0; i < animation.subAnimations.Count(); i++)
    {
        var subAnimation = animation.subAnimations[i];
        // Comprueba a qué capa debe aplicarse la sub-animación.
        if (subAnimation.layerTarget.Contains("leftArm"))
        {
            leftArmSubAnimation.Add(subAnimation);
        }
        if (subAnimation.layerTarget.Contains("rightArm"))
        {
            rightArmSubAnimation.Add(subAnimation);
        }
    }
    // Ordena las sub-animaciones por su tiempo de inicio.
    leftArmSubAnimation = leftArmSubAnimation.OrderBy(o =>
    ↪  o.startTime).ToList();
    rightArmSubAnimation = rightArmSubAnimation.OrderBy(o =>
    ↪  o.startTime).ToList();
```

```
    // Limpia los estados existentes en las máquinas de estado de las capas de
    ↪ los brazos.
    leftArm.stateMachine.states = Array.Empty<ChildAnimatorState>();
    rightArm.stateMachine.states = Array.Empty<ChildAnimatorState>();

    // Llama a la función para agregar las sub-animaciones a las máquinas de
    ↪ estado correspondientes.
    addSubAnimations(leftArmSubAnimation, leftArm.stateMachine);
    addSubAnimations(rightArmSubAnimation, rightArm.stateMachine);
}

// Método para agregar una lista de sub-animaciones a una máquina de estado del
↪ Animator.
void addSubAnimations(List<SubAnimation> subAnimations, AnimatorStateMachine
↪ stateMachine)
{
    // Si no hay sub-animaciones, retorna.
    if (subAnimations.Count() == 0) return;

    // Obtiene la primera sub-animación de la lista.
    var subAnimation = subAnimations[0];
    // Carga el AnimationClip desde los recursos usando el nombre de la
    ↪ sub-animación.
    var currentAnimationClip = Resources.Load<AnimationClip>(subAnimation.name);
    // Crea un nuevo estado en la máquina de estado con el nombre de la
    ↪ sub-animación.
    var currentState = stateMachine.AddState(subAnimation.name);
    // Asigna el AnimationClip al estado.
    currentState.motion = currentAnimationClip;
    // Establece este estado como el estado por defecto de la máquina de
    ↪ estado.
    stateMachine.defaultState = currentState;
    // Guarda el estado actual para crear transiciones desde él.
    var lastState = currentState;
```

```
    // Itera a través del resto de las sub-animaciones para crear estados y
    ↪  transiciones.
    for (int i = 1; i < subAnimations.Count(); i++)
    {
        subAnimation = subAnimations[i];
        currentAnimationClip = Resources.Load<AnimationClip>(subAnimation.name);
        currentState = stateMachine.AddState(subAnimation.name);
        currentState.motion = currentAnimationClip;
        // Crea una transición desde el estado anterior al estado actual.
        var transition = lastState.AddTransition(currentState);
        // Indica que la transición debe ocurrir cuando el estado anterior
        ↪  termine.
        transition.hasExitTime = true;
        lastState = currentState;
    }
}


// Corrutina para realizar una petición GET a una URL de forma periódica.
IEnumerator getRequest(string uri)
{
    // Bucle infinito para realizar peticiones continuamente.
    while (true)
    {
        // Crea una nueva petición GET a la URI especificada.
        UnityWebRequest uwr = UnityWebRequest.Get(uri);
        Debug.Log(uwr);
        // Envía la petición y espera a que se complete.
        yield return uwr.SendWebRequest();

        // Comprueba si hubo un error de red durante la petición.
        if (uwr.isNetworkError)
        {
            Debug.Log("Error While Sending: " + uwr.error);
        }
        // Comprueba si la petición fue exitosa (código de respuesta 200).
        else if (uwr.responseCode == 200)
        {
            Debug.Log("Recibido: " + uwr.downloadHandler.text);
            // Deserializa el JSON recibido en un objeto SignAnimation.
            SignAnimation animation_recv =
            ↪  SignAnimation.CreateFromJSON(uwr.downloadHandler.text);
            // Llama a la función para agregar la animación recibida.
            addAnimation(animation_recv);
        }
        // Espera 3 segundos antes de la siguiente petición.
        yield return new WaitForSeconds(3);
    }
}
```

```
    // Método llamado en cada frame.
    void Update()
    {

    }

    // Método privado para resetear todos los triggers del Animator.
    private void ResetAllTriggers()
    {
        // Itera a través de todos los parámetros del Animator.
        foreach (var p in interprete.parameters)
        {
            // Comprueba si el parámetro es de tipo Trigger.
            if (p.type == AnimatorControllerParameterType.Trigger)
            {
                // Resetea el trigger.
                interprete.ResetTrigger(p.name);
            }
        }
    }
}

// Clase serializable para representar una SubAnimation.
[System.Serializable]
public class SubAnimation
{
    public string name;
    public List<string> layerTarget;
    public double startTime;
    public double speed;

    // Método estático para crear un objeto SubAnimation desde un string JSON.
    public static SubAnimation CreateFromJSON(string jsonString)
    {
        return JsonUtility.FromJson<SubAnimation>(jsonString);
    }
}
```

```
// Clase serializable para representar una SignAnimation.
[System.Serializable]
public class SignAnimation
{
    public string name;
    public List<SubAnimation> subAnimations;

    // Método estático para crear un objeto SignAnimation desde un string JSON.
    public static SignAnimation CreateFromJSON(string jsonString)
    {
        return JsonUtility.FromJson<SignAnimation>(jsonString);
    }
}
```

Referencias

[1] FAXPG, "Historia de la LSE." [Online]. Available: https://www.faxpg.es/historia-lse-es.html

[2] G. FERRERI, "The deaf in antiquity," *American Annals of the Deaf*, vol. 51, pp. 460–473, 1906.

[3] E. A. F., "What did st. augustine say?" *American Annals of the Deaf*, vol. 57, pp. 108–120, 1912.

[4] A.G.Ricao, "Historia del alfabeto dactilológico español," 2006. [Online]. Available: https://cultura-sorda.org/historia-del-alfabeto-dactilologico-espanol/#

[5] E.Z.Pascual, "Pedro ponce de león." [Online]. Available: https://historia-hispanica.rah.es/biografias/36520-pedro-ponce-de-leon

[6] "Sahagún rinde homenaje al padre de la educación inclusiva," 2021. [Online]. Available: https://sahagundigital.com/art/10778/sahagun-rinde-homenaje-al-padre-de-la-educacion-inclusiva.

[7] british sign.co.uk. [Online]. Available: https://www.british-sign.co.uk/fingerspelling-alphabet-charts

[8] J.P.Bonet, *Reducción de las letras y arte para enseñar a hablar a los mudos*, 1620.

[9] A. Oviedo, "El libro «refugium infirmorum» (madrid, 1593), del monje franciscano fray melchor de yebra," 2007. [Online]. Available: https://cultura-sorda.org/refugium-infirmorum/

[10] A.Oviedo, "La vida y la obra del abad charles michel de l´epée," 2007. [Online]. Available: https://cultura-sorda.org/abad-de-lepee/

[11] Davius. [Online]. Available: https://es.wikipedia.org/wiki/Lengua_de_se%C3%B1as#/media/Archivo:Main_Sign_Language_Families.png

[12] A. Oviedo, "La vida y la obra de samuel heinicke (1727 - 1790)," 2007. [Online]. Available: https://cultura-sorda.org/samuel-heinicke/

[13] A.Oviedo, "El 2do. congreso internacional de maestros de sordomudos, celebrado en milán, italia, del 6 al 11 de septiembre de 1880," 2006. [Online]. Available: https://cultura-sorda.org/el-2do-congreso-internacional-de-maestros-de-sordomudos-milan-1880/

[14] M.Jay, "History of american sign language," 2023. [Online]. Available: https://www.startasl.com/history-of-american-sign-language/

[15] B.Pare, "William stokoe," 2021. [Online]. Available: https://www.startasl.com/william-stokoe

[16] C. Jullian, "Un héroe francés en el silencio: Eduardo huet y la conformación de la identidad sorda en méxico," *Centro de estudios mexicanos y centroamericanos*, pp. 385–410, 2008.

[17] M.C.Aldrete, "La educación del sordo en méxico siglos xix y xx: La escuela nacional de sordomudos," 2009. [Online]. Available: https://cultura-sorda.org/la-educacion-del-sordo-en-mexico-siglos-xix-y-xx-la-escuela-nacional-de-sordomudos/

[18] LADR, "About nicaraguan sign language." [Online]. Available: https://www.columbia.edu/~as1038/about-nsl.html

[19] A. A. Latina, "Aprendiendo lenguaje de señas nicaragüense," 2017. [Online]. Available: https://asb-latam.org/aprendiendo-lenguaje-senas-nicaraguense/

[20] S. Veinberg, "Argentinien: Anfänge und entwicklung der erziehung gehörloser (argentina: Inicios y desarrollo de la educación del sordo)," 1996.

[21] M. I. Massone and S. M. Menéndez, "Una aproximación interaccional para el análisis de la lengua de señas argentina," 1996. [Online]. Available: https://cultura-sorda.org/una-aproximacion-interaccional-para-el-analisis-lsa/

[22] V. H. Fernández, "Estudio de la poblaciÓn sorda en chile: EvoluciÓn histÓrica y perspectivas lingÜÍsticas, educativas y sociales," 2010. [Online]. Available: https://valeria-herrera.blogspot.com/2010/04/estudio-de-la-poblacion-sorda-en-chile.html#more

[23] Millaray. [Online]. Available: https://es.pinterest.com/pin/547820742188920598/

[24] X. A. Robertson, D. A. Quintela, and I. C. Ramírez, *Diccionario Bilingüe Lengua de Señas Chilena-Español Tomo I.* Universidad Metropolitana de Ciencias de la Educación, 2009.

[25] X. Robertson, D. A. Quintela, and I. C. Ramírez, *Diccionario Bilingüe Lengua de Señas Chilena-Español Tomo II.* Universidad Metropolitana de Ciencias de la Educación, 2009.

[26] P. A. S. Uribe, "Análisis descriptivo de la categoría gramatical de aspecto en la lengua de señas chilena," 2015.

[27] A. Moryossef and Y. Goldberg, "Sign Language Processing," https://sign-language-processing.github.io/, 2021.

[28] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[29] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306.

[30] "Singer link." [Online]. Available: https://ohiostate.pressbooks.pub/graphicshistory/chapter/13-2-singer-link/

[31] "Programming and artistry." [Online]. Available: https://ohiostate.pressbooks.pub/graphicshistory/chapter/2-2-programming-and-artistry/

[32] E. UNIVERSITY, "¿cuáles son los mejores programas de animación 3d?" 2021. [Online]. Available: https://www.esic.edu/rethink/tecnologia/mejores-programas-de-animacion-3d

[33] Blender, *Blender Documentation*. [Online]. Available: https://docs.blender.org/

[34] CORDIS, "D2.1 sign language animation preliminary development and production," 2018.

[35] E. L. R. A. (ELRA), "Cuny american sign language motion-capture corpus: First release," 2018.

[36] R. Parent, *Computer Animation: Algorithms and Techniques*. Elsevier, 2012.

[37] S. Tomás, "Talleres dae — taller de lengua de señas," 2021. [Online]. Available: https://www.youtube.com/watch?v=cy9eABjXxjg&list=LL&index=19.

[38] R. M. Muñoz, "Saludos y preguntas lengua de señas chilena," 2016. [Online]. Available: https://www.youtube.com/watch?v=95Qu2xavAsA

[39] C. de Innovación Mineduc, "Vocabulario lengua de señas chilena (lsch)," 2014. [Online]. Available: https://www.youtube.com/watch?v=J-3jA42TdMw

[40] L. B. Chile, "Capítulo 1, el alfabeto manual en lengua de señas chilena," 2022. [Online]. Available: https://www.youtube.com/watch?v=LfLP7vs4PEs&list=PLcvDLY-MxzIoJbCPM3aQk4dwP9K5aDA-M

[41] P. Monster, "Sketchfab," 2023. [Online]. Available: https://sketchfab.com/3d-models/indian-teenager-b2804b40f671431ca7b5cc53b239175e

[42] A. S. Incorporated, "Mixamo characters." [Online]. Available: https://www.mixamo.com/#/?page=1&query=Sophie&type=Character

[43] Unity, *Unity Documentation*. [Online]. Available: https://docs.unity3d.com/Manual/index.html

[44] P. Castillo, 2025. [Online]. Available: https://youtu.be/EVy_x62qn2M

[45] M. Corporation, *HTTP response status code*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA
RESUMEN DE MEMORIA DE TITULO

Departamento	**:** Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería civil electrónica
Nombre del memorista	: Paula Geraldine Castillo Osses
Título de la memoria	**:** Construcción de gestos en Lengua de Señas Chilena a partir de sub-gestos
Fecha de la presentación oral	: miércoles 2 de abril de 2025
Profesor(es) guía	: Mario Medina
Profesor(es) revisor(es)	: Pamela Guevara – Sebastián Godoy
Concepto	**:**
Calificación	**:**

Resumen (máximo 200 palabras)

En este trabajo se presenta la implementación de un programa que permite realizar la conversión de glosas en lengua de señas chilena a animaciones 3D.

El objetivo de este trabajo es presentar una alternativa en la representación de la construcción de señas de la lengua de señas chilena mediante sub-gestos, ya sea que posean un significado o no, pero que compuestos puedan formar diferentes gestos aprovechando la estructura de esta lengua.

Esto se realiza a través de programas de animación 3D, como Blender donde se crea cada sub-animación, posteriormente estas son estructuradas de tal forma que exista una reutilización de cada sub-gesto en la formación de un gesto, siendo finalmente reproducidas en un motor de gráfico como Unity.