

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Traductor de texto a español signado con imágenes

Modalidad: Proyecto Tecnológico

Alumno:

Cristian Arturo Plaza Cuadras
2112000578
plazac2092@gmail.com

Asesor:

Dr. Leonardo Daniel Sánchez Martínez
Departamento de Sistemas
Asociado
ldsm1985@gmail.com

Co-asesor:

Dr. José Alejandro Reyes Ortiz
Departamento de Sistemas
Asociado
jaro@correo.azc.uam.mx

Trimestre 2018 Primavera

Fecha: 30 de junio de 2018

Yo, LEONARDO DANIEL SÁNCHEZ MARTÍNEZ, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. Leonardo Daniel Sánchez Martínez

Yo, JOSÉ ALEJANDRO REYES ORTIZ, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. Alejandro Reyes Ortiz

Yo, CRISTIAN ARTURO PLAZA CUADRAS, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco para publicar el presente documento en la Biblioteca Digital, así como en el repositorio Institucional de UAM Azcapotzalco.



Cristian Arturo Plaza Cuadras

Agradecimientos

A mis padres, por plantarme en este mundo, por alimentarme y cuidarme hasta que maduré, por apoyarme y creer en mí.

A Lucero Sánchez Velasco, por apoyar la causa sin pedir nada a cambio, por su paciencia y por su ayuda brindada a todo lo largo de este proyecto.

A Leonardo, por tratarme como a una persona y no como a una obligación que cumplir. Por aconsejarme, motivarme y confiar en mi manera de hacer las cosas.

A mis asesores de proyecto, por aceptar trabajar conmigo, por sus observaciones, comentarios, por su ayuda y el excelente trato que me brindaron.

A la UAM como institución, porque tiene un espíritu, un carácter y una forma de ser propios. Algo que todos los miembros de la UAM percibimos; que nos atormenta a veces, pero que también nos encanta. Porque podemos decir: “la UAM” y sentir que estamos hablando de un hogar.

Resumen

En este documento se presenta el desarrollo de una aplicación de escritorio que pretende disminuir los problemas de comunicación entre personas con discapacidad auditiva y hablantes. Dicha aplicación permite traducir palabras del español escrito, al lenguaje de señas mexicanas a través de imágenes, las cuales se conformaron tomando fotografías de los movimientos referentes a las señas y posteriormente editándolas para colocar las direcciones y trayectorias que siguen dichos movimientos, obteniendo así un conjunto de 500 imágenes que alimentan la aplicación.

La aplicación procesa el texto para poder manejar símbolos y signos de puntuación, dando como resultado una herramienta capaz de traducir cualquier palabra en español al lenguaje de señas mexicano, sin importar la cantidad y tipo de símbolos utilizados en el texto.

La aplicación además permitirá que las personas que no tienen discapacidades auditivas y estén interesadas en aprender las señas, puedan hacerlo a través de esta herramienta.

Índice

Agradecimientos	3
Resumen	3
Índice	4
Índice de figuras	6
Índice de tablas	6
Introducción	7
Justificación	7
Objetivos	8
Objetivo General	8
Objetivos Específicos	8
Trabajos relacionados	8
Marco Teórico	8
Java	9
JavaFx	9
Interfaz Gráfica	9
Modelo vista controlador	9
Modelo	10
Vista	10
Controlador	11
SceneBuilder	11
FXML	11
Descripción del proyecto	12
<i>Hardware</i> utilizado	12
<i>Software</i> utilizado	12
Requerimientos	13
Desarrollo	13
Metodología empleada	13
Módulo de selección del vocabulario	14
Probabilidad de que aparezcan las palabras	15
Alfabeto, números y símbolos	17
Palabras orientadas a temas	18
Plurales y femeninos	19

Módulo de almacenamiento	20
Módulo de procesamiento de Texto.....	20
Espacios entre palabras.....	20
Signos de puntuación.....	21
Enter	23
Números:	23
Manejo de palabras.....	23
Módulo de traducción de texto a LSM.....	23
Módulo de Interfaz de usuario.....	23
Usabilidad.....	24
Interpretación de las imágenes	25
Estructura de la aplicación.....	29
Diagrama de casos de uso:	30
Caso de uso Subir archivo	31
Caso de uso Escribir en cuadro	32
Caso de uso Adelantar imagen	32
Caso de uso Retroceder Imagen	33
Caso de uso Traducir texto	33
Caso de uso Solicitar ayuda.....	34
Portabilidad.....	34
Resultados.....	35
Trabajo futuro	35
Conclusiones.....	36
Referencias	37
Entregables	42

Índice de figuras

Figura 1. MVC de la aplicación presentada.	10
Figura 2. Programa SceneBuilder.	12
Figura 3. “Araña” expresado en LSM.	13
Figura 4. Módulos que componen el proyecto.	14
Figura 5. Abogado, abogada y abogados en LSM.	20
Figura 6. Manejo de signos de puntuación	21
Figura 7. Gato en LSM.	22
Figura 8. Interfaz gráfica del traductor de texto a español signado.	24
Figura 9. Elementos comunes en las imágenes.	25
Figura 10. Palabra correr, expresada en LSM	26
Figura 11. Palabra nacional expresada en LSM	27
Figura 12. Palabra niño, expresada en LSM	28
Figura 13. Palabra cantar, expresada en LSM	28
Figura 14. Estructura de la aplicación.	29
Figura 15. Botones con eventos en la interfaz de usuario.	30
Figura 16. Diagrama de casos de uso.	31

Índice de tablas

Tabla 1. Frecuencias de repetición de algunas palabras del idioma español.	15
Tabla 2. Algunas de las palabras que más se repiten en el idioma español.	17
Tabla 3. Lista de letras números y símbolos.	18
Tabla 4. Grupos de palabras por tema.	19
Tabla 5. Grupos de palabras por tema.	19

Introducción

La OMS estima que 360 millones de personas sufren pérdida auditiva, 328 millones de adultos y 32 millones de niños, más del 5% de la población mundial. [1].

En México 694,451 habitantes del país padecen deficiencia auditiva [2]. Alrededor de un séptimo de esta cifra es capaz de comunicarse a través del lenguaje de señas [3].

Desafortunadamente, este tipo de personas son excluidas del aprendizaje formal impartido en las escuelas del país. Además, las que logran tener acceso a la educación se encuentran con el problema de que la estructura del lenguaje del idioma español es diferente a la del lenguaje de señas y se les dificulta aprender la sintaxis.

Existe un lenguaje intermedio entre el lenguaje de señas mexicanas (LSM) y el español hablado, este es el español signado, que, básicamente es la traducción de las palabras en el orden en que las usan los hablantes, a la seña equivalente de las mismas. Este lenguaje intermedio puede facilitar la comprensión de la estructura del español hablado a las personas sordas y puede hacer que ambos grupos de personas puedan establecer un canal de comunicación primitivo (como suelen comunicarse las personas que intentan aprender un nuevo idioma). Teniendo presentes las cualidades de este lenguaje intermedio, se pretende diseñar un programa con interfaz de usuario, que permita a las personas hablantes obtener la traducción de un texto ingresado, a su equivalente en español signado.

Justificación

Teniendo en cuenta que alrededor de 100 mil personas se comunican a través de la lengua de señas en México, cifra que es mayor al número de hablantes de muchas lenguas indígenas como el popoluca o el huichol y que existen casi 700 mil personas con potencial de comunicarse de esta manera, cuyo número se equipara a lenguas indígenas importantes como el mixteco, mazateco, zapoteco y otomí [4], parece importante proponer medidas para la inclusión de este sector de la población.

En México solo hay 40 intérpretes certificados de lengua de señas, según un estudio presentado en [5]. Lo anterior refleja la falta de inclusión e importancia que se le da a este sector de la población afectado. Es evidente que se necesitan medios que permitan a las personas con discapacidad auditiva acceder a la información de interés a la que toda la población tiene derecho y esto puede lograrse desarrollando las herramientas adecuadas para que las personas puedan aprender la lengua sin necesidad de depender de factores como la disponibilidad de cupo de estudiantes en las escuelas establecidas para este fin, la cercanía de las mismas o el recurso económico para pagarlas. Por esta razón se pretende diseñar una herramienta capaz de convertir una cadena de texto en una secuencia de imágenes equivalentes a la seña de cada palabra. Esta implementación pretende facilitar la comunicación entre los grupos humanos hablantes y no hablantes, sin la necesidad de tener que contar con un intérprete humano.

Aunque existen aplicaciones que intentan lograr que más personas se acerquen al LSM e incluso puedan aprender un poco, en México no se ha desarrollado un programa con las características que aquí se proponen, además, dicha herramienta puede expandirse a

tantas palabras como tiempo se disponga para dedicarle al módulo de imágenes en el futuro.

Objetivos

Objetivo General

Diseñar e implementar un programa con interfaz de usuario que, a partir de una cadena de texto ingresado, arroje como resultado una secuencia de imágenes equivalentes a la traducción en español signado del texto.

Objetivos Específicos

Diseñar e implementar un módulo para la selección del vocabulario de LSM.

Diseñar e implementar un módulo para almacenar vocabulario.

Diseñar e implementar un módulo para el procesamiento de texto.

Diseñar e implementar un módulo que permita traducir texto a señas.

Diseñar e implementar una interfaz de usuario.

Trabajos relacionados

Existen diversos trabajos que buscan apoyar al sector de la población con discapacidad para comunicarse por medio del habla. El propósito de estos trabajos es facilitar la inclusión de dicho sector mediante el desarrollo de herramientas que ayudan a establecer puentes de comunicación entre las personas discapacitadas y las que no lo son. Un ejemplo de este tipo de herramientas se presenta en [6]. Esta es una herramienta que trata de beneficiar a un sector de la población con problemas de lenguaje hablado, utilizando imágenes para la comunicación con los usuarios, específicamente se enfoca en el sector infantil, parecido al trabajo presentado en [7]. Otro ejemplo de este tipo de trabajos es el sistema para expresar las necesidades básicas en personas con trastornos del habla [8]. Este proyecto no utiliza lenguaje de señas para su fin, sino una serie de comandos que permiten comunicar necesidades básicas a través de una voz artificial y su implementación se compone en su mayoría de hardware.

Otros programas no se enfocan tanto en diseñar herramientas sino en fomentar la inclusión de los sectores afectados como en [9]. Incluso existen proyectos que pretenden establecer un canal inverso al que se propone en este documento, es decir, su objetivo es tomar las señas y convertirlas en texto. Podemos ver ejemplos de ello en [10] y [11].

Marco Teórico

Haciendo una síntesis que nos ayude a visualizar los elementos que debemos conocer para entender el presente proyecto, podemos decir que se trata de: “Una aplicación de escritorio basada en java que utiliza su complemento JavaFx para la interfaz gráfica. Dicha aplicación está basada en un patrón de diseño de software llamado: Modelo Vista Controlador”.

Teniendo lo anterior en mente pasaremos a explicar cada elemento de la descripción para comprenderla completamente y más adelante se explicará a detalle cómo está constituido el proyecto.

Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para ejecutarse en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones Web basadas en el modelo cliente-servidor, con unos diez millones de usuarios reportados. [12]

JavaFx

JavaFX es una familia de productos y tecnologías de Oracle (inicialmente *Sun Microsystems*), para la creación de aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son JavaFX Script y JavaFX *Mobile*, aunque hay más productos JavaFX. [13].

Interfaz Gráfica

Las GUI o interfaces gráficas son un programa informático que realiza la función de interfaz de usuario. Está formado por imágenes y objetos gráficos, que representan la información y acciones que se encuentran en la interfaz. Su objetivo es el de crear un entorno visual fácil de usar para que fluya la comunicación con el sistema operativo. [14]

Modelo vista controlador

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo, Vista y Controlador). El Patrón MVC se ve frecuentemente en aplicaciones Web, donde la Vista es la página HTML y el código que provee de datos dinámicos a la página; el Modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; el Controlador es el responsable de recibir los eventos de entrada desde la Vista [15]. En la Figura 1 se observa este tipo de modelo enfocado a la aplicación presentada en este proyecto.

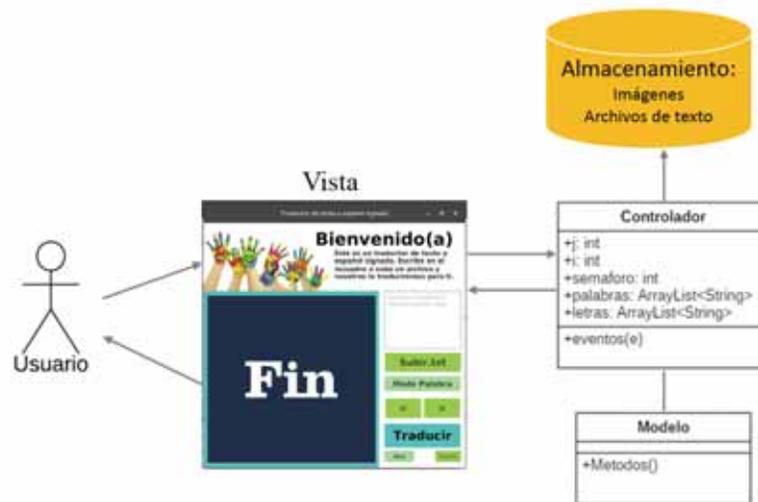


Figura 1. MVC de la aplicación presentada.

Modelo

Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, o calculando los totales e impuestos del carrito de compra. Esto quiere decir que aquí se operan los datos y las reglas de negocio asociadas al sistema, incluyendo el análisis sintáctico y el procesamiento de los datos de entrada y de los datos de salida.

El Modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo, por ejemplo, un fichero que actualiza los datos, un temporizador que desencadena una inserción, etc. Un ejemplo de MVC con un modelo pasivo (aquel que no notifica cambios en los datos) es la navegación web, que responde a las entradas del usuario, pero no detecta los cambios en datos del servidor.

Vista

Este presenta el Modelo, usualmente la interfaz de usuario. La vista es la capa de la aplicación que ve el usuario en un formato adecuado para interactuar, en otras palabras, es nuestra interface gráfica.

Las vistas son responsables de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de actualización para que sea invocado por el controlador o por el modelo cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes.

Controlador

El Controlador es la capa que controla todo lo que puede realizar la aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Está compuesto por acciones que se representan con funciones en una clase. Por ejemplo, se tiene el controlador llamado “Clientes”, y este controlador puede realizar las acciones “Crear”, “Editar”, “Listar” entre otras.

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contener reglas de gestión de eventos, del tipo “SI Evento Z, entonces, Acción W”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método *Actualizar()*. Una petición al modelo puede ser *Obtener_tiempo_de_entrega()*.

SceneBuilder

JavaFX SceneBuilder es una herramienta de diseño visual que permite a los usuarios diseñar rápidamente interfaces de usuario de aplicaciones JavaFX, sin codificación. Los usuarios pueden arrastrar y soltar los componentes de la interfaz de usuario en un área de trabajo, modificar sus propiedades, aplicar hojas de estilo y el código FXML para el diseño que están creando se genera automáticamente en el fondo. El resultado es un archivo FXML que luego se puede combinar con un proyecto Java vinculando la interfaz de usuario a la lógica de la aplicación [16].

En la Figura 2 podemos ver que el programa SceneBuilder cuenta con un menú de elementos en el lado izquierdo, tales como paneles, controles, documentación, etc., que pueden ser agregados fácilmente arrastrándolos al área de dibujo.

FXML

FXML es un formato jerárquico, se puede crear al instante de manera dinámica si es necesario, permitiendo agregar o crear elementos de UI dinámicos en puntos específicos durante el tiempo de ejecución de la aplicación. En muchos casos, al escribir FXML se obtienen definiciones de interfaz de usuario más cortas [17].

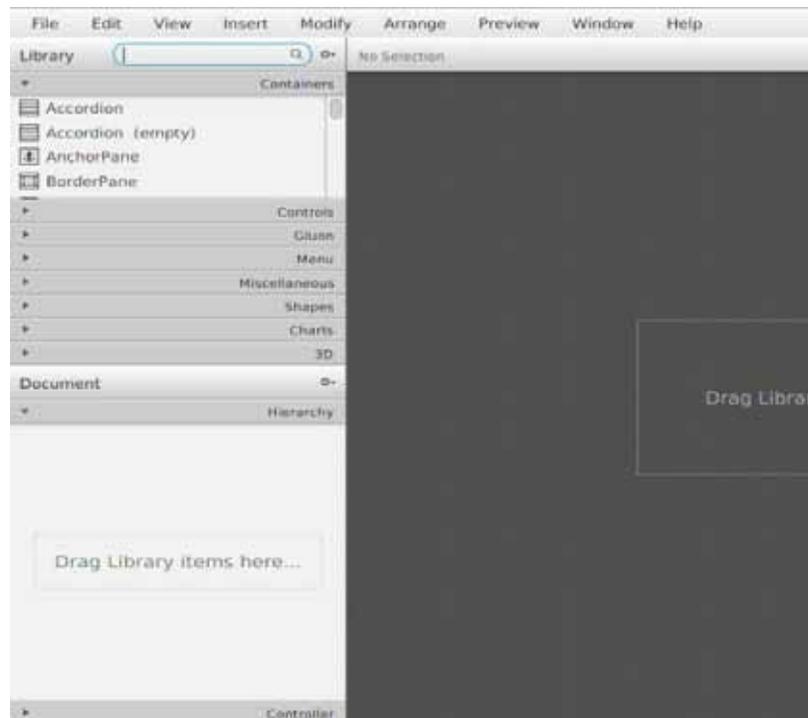


Figura 2. Programa SceneBuilder.

Descripción del proyecto

La aplicación convierte un texto ingresado a través de un archivo con extensión .txt en una secuencia de imágenes que corresponden cada una a la seña, en el lenguaje de señas mexicanas, de cada palabra encontrada en el archivo.

Dicho archivo puede subirse a través de la aplicación o generarse al escribir directamente en un área de texto de la misma.

Adicionalmente la aplicación puede mostrar no solo palabras, sino también símbolos y números, además de ser capaz de deletrear palabras que no se encuentren en el módulo de almacenamiento a través de imágenes de señas de cada letra del alfabeto.

Hardware utilizado

Para llevar a cabo este proyecto se utilizaron dispositivos de fácil acceso: teléfono celular con cámara para la toma de fotografías y una computadora común para el desarrollo de la aplicación.

Software utilizado

La aplicación se diseñó sobre una distribución del sistema Operativo Linux, específicamente un Ubuntu 16.04 LTS de 64 bits.

Para programar la funcionalidad de la aplicación se utilizó el lenguaje Java y JavaFX, el *IDE* eclipse en sus versiones Oxygen y Photon con la extensión e(fx)clipse y la herramienta SceneBuilder para el diseño de la interfaz gráfica.

Las fotografías que alimentan al sistema fueron editadas en su totalidad para colocarles uno o varios identificadores (flechas, nombres, puntos y/o números), además, la mayoría de las imágenes es un *collage* de dos (o a veces tres) fotografías que se unen para formar una seña, como la que se muestra en la Figura 3.

Para llevar a cabo la edición de imágenes se utilizó el software libre Gimp y el editor de imágenes de WhatsApp. Para la unión de fotografías se utilizó la aplicación en línea Fotoefectos [18].



Figura 3. “Araña” expresado en LSM.

Requerimientos

La aplicación puede ejecutarse en cualquier computadora que soporte Java 8 o superior. Ha sido probada en diferentes computadoras, siendo la de recursos más limitados una con procesador Intel Celeron trabajando a 1.8GHz con 2 gigas de *Ram*, en este caso el comportamiento ha sido adecuado.

Desarrollo

Metodología empleada

Para implementar el proyecto, se requiere llevar a cabo 5 módulos que se muestran en la Figura 4 y que se explicarán a continuación:

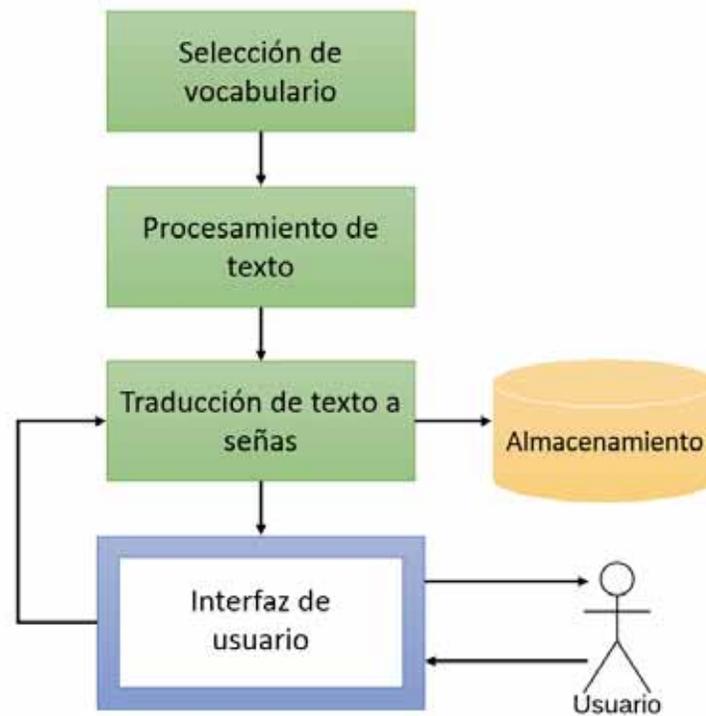


Figura 4. Módulos que componen el proyecto.

Módulo de selección del vocabulario

El Diccionario de la Real Academia Española (DRAE) contiene alrededor de 90 000 palabras. Sin embargo, un estudio de la BBC [19], señala que un niño es capaz de comunicarse solo con 100 palabras y que una persona puede comunicarse sin apuros sabiendo mil quinientas. Teniendo en cuenta que el lenguaje de señas tiene una seña equivalente para cada letra del alfabeto, las cuales se usarán para traducir palabras que no han sido encontradas en el sistema, se ha considerado suficiente almacenar la traducción, en imágenes de LSM, de 500 palabras.

Este módulo fue el encargado de establecer los criterios adecuados para seleccionar las 500 palabras del lenguaje español y con base en ellas componer un banco de 500 imágenes que alimentan la aplicación, entre ellas, algunas de las más utilizadas del idioma.

En un principio se había propuesto obtener la traducción de las 500 palabras que más se repiten en el español, las cuales podemos ver en una lista publicada por la RAE [20]. Sin embargo, después de analizar esa lista, se encontraron ciertos factores que no fueron tomados en cuenta, por ejemplo, que el español es un idioma hablado por muchos países y por lo tanto se encuentran palabras como: España, que no aplican a las probabilidades de uso lingüístico de México.

Otro factor interesante es que muchas de las palabras que componen la lista, son homónimas, ejemplos de ello se pueden ver en la Tabla 1. La palabra número 13, “por”, puede aparecer en diferentes contextos ya que puede utilizarse para explicar acciones, por ejemplo en la frase “Lo hice por error”. En ciencias exactas es el nombre del signo “x” y se refiere a la multiplicación. Otro uso se da al referenciar ubicaciones, por ejemplo, “Está por ahí”, etc. La palabra 18, “para” es una preposición y también es el presente en tercera persona del verbo parar.

Nos encontramos con otro factor: el hecho de que en lenguaje de señas hay palabras del español que aparecen en la lista y no tienen traducción al lenguaje de señas como “del”, “al”, entre otras.

La escasez de traductores en el país también ha representado un reto. Se ha conseguido la mayor parte de las palabras estudiando lecciones del LSM en internet, como las presentadas en [21-64] y en los libros electrónicos presentados en [65-67]

Orden	Frecuencia absoluta
13. por	1,561,904
18. para	1,062,152
22. como	773,465
23. más	661,696
32. sobre	289,704
33. este	285,461
35. entre	267,493
45. era	219,933
64. parte	148,750
104. estado	100,757
111. general	95,768

Tabla 1. Frecuencias de repetición de algunas palabras del idioma español.

Teniendo los factores anteriores en cuenta y considerando que se planea que la aplicación funcione también para que las personas que sí pueden hablar puedan aprender a signar, se han elegido nuevos criterios que se presentan a continuación:

- Probabilidad de que aparezcan las palabras.
- Alfabeto, números y símbolos.
- Palabras orientadas a temas.
- Plurales y femeninos.

Probabilidad de que aparezcan las palabras

Aunque hemos dicho que este método causaría conflictos, se cree que al combinarse con los demás que planteamos, se puede lograr un equilibrio. Se redujo el número de palabras elegidas por probabilidad de aparición en un texto, de 500 a 231, las cuales se muestran en la Tabla 2. A la izquierda de cada palabra, podemos observar su lugar en el ranking de probabilidad de aparición en un texto. Los números se muestran ascendentes pero no

consecutivos, esto se debe ya sea a que no hay seña para la palabra o bien no se pudo conseguir su traducción.

Palabras que más se repiten en el idioma español (puesto palabra)							
1	de	111	general	228	guerra	357	persona
5	en	113	poco	229	desarrollo	363	dios
13	por	115	presidente	233	lado	367	México
14	con	117	ante	235	tu	372	amor
15	no	120	algo	238	mano	373	policía
19	es	121	hacia	240	estar	374	puerta
23	más	122	casa	241	santo	380	tampoco
25	pero	123	ellos	242	número	384	campo
32	sobre	124	ayer	245	centro	387	saber
34	ya	127	mucho	246	padre	394	dinero
35	entre	128	mientras	254	último	398	hijo
36	cuándo	131	momento	255	madre	400	cien
37	todo	132	millones	259	cinco	402	hablar
38	esta	135	hombre	261	hombres	404	minutos
39	ser	139	lugar	262	información	406	caminar
41	dos	141	nacional	263	ojo	407	seis
42	también	142	trabajo	264	muerte	408	quién
46	muy	144	mejor	265	nombre	410	dirección
47	años	145	nuevo	268	mujeres	411	papel
48	hasta	146	decir	269	siglo	419	capital
49	desde	147	algunos	270	todavía	424	espacio
51	mío	148	entonces	272	mañana	430	estudio
53	qué	153	cómo	275	hora	436	cultura
54	sólo	158	pasado	277	pueblo	438	alto
56	yo	160	medio	279	dar	443	paz
58	vez	165	nunca	280	problema	452	julio
61	así	166	poder	283	tú	457	cara
62	nos	168	ver	284	derecho	465	bueno
63	ni	171	partido	285	verdad	467	niño
66	él	172	personas	287	unidos	468	difícil
67	uno	173	grupo	290	junto	469	joven
69	bien	174	contar	295	tierra	470	futuro
70	tiempo	179	cual	296	equipo	472	mes
71	mismo	181	mujer	297	segundo	473	pronto
73	ahora	182	frente	298	director	484	atención
74	cada	185	cosas	300	cierto	491	investigar
76	vivir	186	fin	303	nivel	508	universidad
77	otro	187	ciudad	305	familia	515	hijos
78	después	189	social	306	largo	519	jugar
81	aunque	191	tener	307	partir	538	pasar

83	eso	194	historia	309	llegar	542	gracias
86	gobierno	195	muchos	310	propio	545	defender
88	durante	198	cuatro	312	cosa	567	sol
89	siempre	199	dentro	313	primero	568	jóvenes
90	día	201	punto	314	seguridad	575	libre
92	ella	204	cualquier	316	mal	580	cine
93	tres	205	noche	317	algún	581	salir
94	sí	207	agua	317	tratar	598	domingo
98	país	209	haber	321	semana	599	favor
99	según	212	bajo	325	voz	631	blanco
100	menos	215	ejemplo	327	señor	636	ayuda
101	mundo	216	acuerdo	328	mil	642	dónde
102	año	218	usted	330	proyecto	660	línea
103	antes	219	estados	331	mercado		
104	estado	221	nadie	333	luz		
105	contra	222	países	334	claro		
107	forma	225	tarde	337	monedas		
109	nada	226	ley	341	querer		
110	hacer	227	importante	354	libro		

Tabla 2. Algunas de las palabras que más se repiten en el idioma español.

Alfabeto, números y símbolos

Para poder deletrear las palabras que no se encuentran en el banco de imágenes, se incluyeron las señas de cada letra del alfabeto, también se contemplan señas de números, ya que pueden aparecer en los textos, sin embargo, no incluimos la del 0 ya que la comunidad sordomuda no lo maneja, pero se incluye su grafía.

Un archivo de texto también puede incluir signos de puntuación o símbolos matemáticos, la lista de letras números y símbolos se muestra en la Tabla 3.

Alfabeto, números y símbolos			
0	(c	t
1	[d	u
2]	e	ú
3	{	é	ü
4	}	f	v
5	@	g	w
6	*	h	x
7	&	í	y
8	#	i	z
9	%	j	
)	+	k	

-	<	l
;	=	m
:	>	n
!		o
i	~	ó
?	\$	p
¿	á	q
'	a	r
"	b	s

Tabla 3. Lista de letras números y símbolos.

Palabras orientadas a temas

Debido a que se planea que la aplicación pueda servir a las personas que no conocen el lenguaje de señas, para que lo aprendan, se han incluido palabras específicas que pertenecen a temas de uso cotidiano, dichas palabras fueron obtenidas de lecciones publicadas en la plataforma YouTube, realizadas por traductores dedicados a la difusión del LSM, como los presentados en [21-64].

En las Tablas 4 y 5 se muestran, dichas palabras y los temas a los que pertenecen.

Palabras orientadas a temas					
Cuerpo	Profesiones	Animales	Emociones	Hogar	Escuela
barba	abogada	abeja	aburrido	baño	diagonal
barbilla	abogado	abejas	amistad	cama	ejemplos
bigote	abogados	araña	apenado	camas	entrada
boca	diosa	arañas	contento	cocina	internet
brazo	directores	caballo	enamorado	comedor	miles
cabello	doctor	caballos	enojado	computadora	millón
cabeza	doctores	chango	felicidad	computadoras	papeles
cadera	enfermera	changos	feliz	estufa	signo
ceja	maestra	coneja	preocupado	moneda	tercero
cintura	maestro	conejo	serio	pisos	universidades
codo	maestros	conejos	sorprendido	plato	cientos
corazón	policías	gato	triste	platos	cuántos
cuello	presidentes	gatos		sala	
dedo		hipopótamo		santos	
diente		hipopótamos		silla	
espalda		jirafa		sillas	
estómago		osa		teléfono	
hombro		oso		teléfonos	
labios		osos		ventana	
lengua		pájaro		ventanas	
muñeca		pájaros		suyo	

músculo		peces			
músculos		perro			
nariz		perros			
oreja		rinoceronte			
párpado		tigre			
pecho					
piernas					

Tabla 4. Grupos de palabras por tema.

Palabras orientadas a temas 2					
Verbos	Calendario	Familia	Colores	Alimentos	Lugares
abrir	abril	abuelo	amarillo	carne	campos
bailar	agosto	bebé	azul	frutas	casas
beber	diciembre	esposo	café	mango	cielo
besar	enero	hermano	negro	mangos	departamento
borrar	febrero	hermanos	rojo	manzana	departamentos
cantar	hoy	hija	rosa	naranja	edificio
cerrar	invierno	niña	rosas	naranjas	edificios
construir	jueves	niños	verde	rico	escuela
copiar	junio	novio			escuelas
correr	lunes	señores			habitación
dibujar	martes	bebés			habitaciones
dormir	marzo				pueblos
entrar	mayo				rancho
escribir	miércoles				aquí
escuchar	minuto				
explicar	noviembre				
gustar	octubre				
nadar	otoño				
oler	primavera				
pasear	sábado				
platicar	septiembre				
poner	siglos				
saltar	temprano				
tocar	verano				
	viernes				

Tabla 5. Grupos de palabras por tema.

Plurales y femeninos

En el idioma español muchas palabras diferencian su género por la terminación en las vocales “a” u “o”, en LSM se diferencia el femenino agregando una seña adicional después

de la palabra, esta seña es la referente a la palabra “mujer”. Del mismo modo, en español se suele agregar la “s” para los plurales y en LSM se agrega la seña referente a “muchos” al final de la palabra. En la Figura 5 podemos observar un ejemplo. La imagen a) nos muestra cómo se signa abogado, en la imagen b) podemos ver que se le agrega una seña adicional, esta es la seña para “mujer” y por último en c) en lugar de “mujer” se agrega la seña de “muchos” para convertir la palabra abogado en abogados.



Figura 5. Abogado, abogada y abogados en LSM.

Módulo de almacenamiento

Este módulo es el encargado de establecer los lineamientos para el almacenamiento de imágenes y el formato de las mismas.

Originalmente se propuso crear una base de datos de imágenes y se planeaba convertirlas al formato base64 para ser manejadas con Mysql, sin embargo, se hicieron pruebas sobre carpetas y se concluyó que el tiempo de respuesta para acceder directamente a una carpeta con todas las imágenes, era aceptable.

El tiempo ahorrado en este módulo permitió enfocarse en la robustez del módulo de procesamiento de texto, específicamente en la validación de palabras.

Módulo de procesamiento de Texto

El objetivo de este módulo es adaptar la información contenida en un archivo de texto para poder producir un arreglo de palabras manejables por el sistema, eliminando errores y elementos no deseados.

Existen traductores de idiomas que van de texto a texto y que conservan todo símbolo, signo de puntuación, saltos de línea y espacios al momento de traducir, sin embargo para fines de esta aplicación podemos descartar algunos de estos elementos como espacios consecutivos entre palabras o cadenas de símbolo que el usuario pudiera no conocer y que solo entorpecerían la comunicación.

Espacios entre palabras.

A nuestro usuario no va a interesarle cuantos espacios hay entre una palabra sino lo que estas quieren decir, además de que puede que los espacios adicionales hayan sido introducidos por error en el archivo, por ejemplo, tenemos el siguiente enunciado: “hay un

gato en la banqueta”, accidentalmente un descuido del escritor hizo que el enunciado quedará de la siguiente manera: “hay un gato en la banqueta”. Podemos ver que se colocaron espacios de más que al usuario no van a interesarle por lo cual se pueden eliminar, sin embargo, si se cambia entre transiciones de palabras que no se encuentran en el módulo de almacenamiento y por lo cual se tienen que deletrear, se agregará un espacio entre ellas para que el usuario pueda identificar donde termina una palabra y empieza otra.

Signos de puntuación

Otro aspecto a tener en cuenta, es que las personas sordomudas no tienen una traducción signada para los signos de puntuación tales como: comas, puntos, punto y coma etc. Y aquellos que no saben leer ni siquiera tienen noción de que existen, es por ello que en esta aplicación los signos y símbolos no se mostraran directamente en el panel de imágenes con el fin de evitar confusiones entre los usuarios, en su lugar y pensando en aquellos usuarios que saben leer, están aprendiendo o quieren aprender, tales símbolos se mostrarán en el botón *Aux* a menos que aparezcan independientes, en cuyo caso se mostrarán en el panel. En la Figura 6 podemos ver un ejemplo en el que se quiere traducir el texto: hombre. En esta se observa en a) que el punto que acompaña a hombre se muestra en el botón *Aux* y que en b) el punto que está solo, se muestra en el panel.



Figura 6. Manejo de signos de puntuación

En la Figura 6 vemos que al punto del botón *Aux*, le preceden 3 asteriscos (***) . Esta terminología se va a utilizar para denotar que hay una palabra. Es decir, que si vemos por ejemplo: !*** significa que antes de la palabra hay un signo de admiración. Un signo después de los asteriscos, p.e. ***! querrá decir que el símbolo se encuentra después de la

palabra y dos símbolos que los rodean p.e. ¡**!, significará que la palabra se encuentra en medio de estos símbolos.

Se ha decidido eliminar los signos consecutivos alrededor de palabras, por ejemplo: “#%&hola/() (la palabra hola, rodeada de símbolos) ya que si bien, mostrar un signo de puntuación servirá como guía, la aplicación está pensada para un sector cuyo gran número de personas no saben leer y se quiere evitar que se sientan abrumadas al encontrar demasiados elementos que no son de su entendimiento. Podemos ver un ejemplo de eliminación de signos en la Figura 7 donde la palabra “Gato” aparece rodeada de muchos símbolos que son descartados para mostrar en el panel solamente la palabra.



Figura 7. Gato en LSM

Los símbolos que aparezcan sin palabras serán mostrados en el panel de imágenes, con el fin de marcar una diferencia, y que el usuario sepa que es un símbolo solitario aunque no entienda qué es.

Aunque se han incluido imágenes de la mayoría de símbolos que se pueden introducir por teclado, podría darse el caso de encontrar algún símbolo desconocido, a lo que la aplicación responderá ignorándolo y pasando al siguiente elemento en el texto.

Enter

Los saltos de línea sí serán comunicados al usuario, para que pueda tener noción de cuantos párrafos ha leído o cuando está a punto de cambiar de tema el texto.

Números:

Los números serán mostrados del 0 al 9 en la aplicación, sin embargo, las personas sordomudas tienen una noción un poco diferente de ellos, por ejemplo, no utilizan el cero, por lo que se incluirá una imagen del número y no una seña en la carpeta de imágenes.

Manejo de palabras

Internamente las letras se pasan a minúsculas con la función *toLowerCase()* para evitar tener que almacenar varias imágenes con el mismo significado, por ejemplo la palabra podría aparecer al inicio del texto y empezar en mayúscula o estar toda en mayúsculas, tendríamos que tener 3 imágenes iguales guardadas con nombres diferentes para los diferentes casos. Convirtiendo las palabras a letras minúsculas se soluciona este problema.

Se han mantenido los acentos puesto que un acento puede cambiar el significado de una palabra, por ejemplo, esta y está.

Módulo de traducción de texto a LSM

El objetivo de este módulo es dar acceso al banco de imágenes para obtener una secuencia de estas a partir del texto que se desea traducir y que ya ha sido procesado anteriormente. Para lograr esto se implementó un sistema basado en rutas de imágenes. Estas rutas son generadas por las funciones *cambiarImagenPalabra()* y *cambiarImagenLetra()* que explicaremos a detalle más adelante y que permiten obtener acceso a imágenes específicas del banco de imágenes.

Módulo de Interfaz de usuario

Este módulo tiene como objetivo establecer una comunicación sencilla con el usuario que pretenda utilizar la aplicación. Se decidió utilizar una sola ventana, la cual se muestra en la Figura 8, ya que en esta se pueden incluir todos los elementos necesarios para comunicarse con la aplicación.



Figura 8. Interfaz gráfica del traductor de texto a español signado.

Usabilidad

En la Figura 8 podemos observar los elementos de la interfaz de usuario numerados, los cuales se explican a continuación:

- 1.-El cuadrado azul contiene un panel de imágenes que, como el nombre lo dice, nos permitirá mostrar las imágenes de cada seña.
- 2.-Este panel de imagen tiene la función de mostrar contenido relacionado con el tema, en este caso unas manos pintadas.
- 3.-Panel destinado a mostrar un texto de bienvenida para el usuario lector. En el elemento 1 se puede colocar una imagen de bienvenida para los usuarios que no saben leer.
- 4.-Cuadro de texto que tiene la función de permitir al usuario ingresar algún texto que desea traducir.
- 5.-Como el nombre lo indica, este botón permitirá al usuario subir un archivo de texto a través de la clase FileChooser.
- 6.-Este panel tiene la función de avisar al usuario si la imagen que está viendo corresponde a una palabra del texto o a una letra de una palabra.

7 y 8.- Son botones que nos permiten retroceder o adelantar las imágenes respectivamente.

9.-El botón de traducir prepara a la aplicación para poder navegar en el texto con los botones 7 y 8.

10.- Este botón se encarga de mostrar signos de puntuación que acompañen a las palabras, por ejemplo, si la palabra tenía una coma el botón *Aux* mostrará “***,” si la palabra está entre signos de interrogación mostrará ¿***?

11.-El botón de ayuda (?) abrirá un archivo en formato PDF que explique al usuario como se utiliza la aplicación.

Interpretación de las imágenes

Las imágenes son en su mayoría como la mostrada en la Figura 9, se componen de 2 fotografías unidas, flechas de dirección y texto.

Las imágenes unidas expresan que hay que pasar del movimiento 1 al 2, las flechas indican la dirección y trayectoria que debe seguir este movimiento. Si no hay flechas significa que debe pasarse de 1 a 2 en el movimiento más directo posible.

Además de lo anterior, las imágenes pueden contener los siguientes elementos:

Flechas bidireccionales

Quiere decir que debemos ir de una dirección a la otra indistintamente a la hora de realizar el movimiento. Un ejemplo de esto lo vemos en la Figura 10.

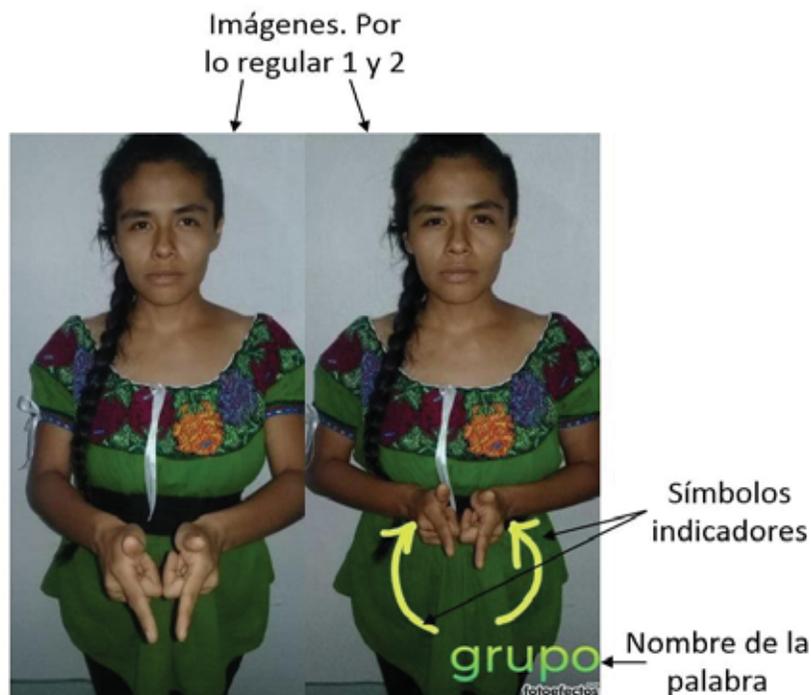


Figura 9. Elementos comunes en las imágenes.



Figura 10. Palabra correr, expresada en LSM

Números

En las imágenes formadas por 2 movimientos, estos deben ejecutarse de izquierda a derecha, sin embargo, algunas señas necesitarán más de 2 movimientos. Para no confundir al usuario, estos movimientos estarán numerados como se muestra en la Figura 11.



Figura 11. Palabra nacional expresada en LSM

Puntos junto a la seña

En la Figura 12 podemos observar que junto a la seña hay 3 puntos, esto significa que el usuario debe dar golpecitos en esa misma posición (al menos 3 sin separar más de 10 centímetros la mano que golpea).

Si los puntos se encuentran en la parte superior de la imagen, como en la Figura 13 y no junto a la seña, significa que el usuario debe repetir los 2 o tres movimientos mostrados en la imagen, 2 o tres veces.



Figura 12. Palabra niño, expresada en LSM



Figura 13. Palabra cantar, expresada en LSM

Estructura de la aplicación

En términos de Proyecto de Java, la aplicación consta de un archivo FXML que genera la vista y se comunica con el controlador (que absorbe la parte del modelo), el controlador y el archivo *Main*. Cuenta con una carpeta llamada “*ImágenesSennias*” que contiene todas las imágenes que alimentan al traductor y otra carpeta llamada *iconos* que alimenta a los elementos de la interfaz. La carpeta del proyecto cuenta también con un manual de usuario en PDF y un archivo de texto, que es el que procesa el traductor.

Cabe señalar que la mayoría de las imágenes que alimentan la aplicación fueron creadas especialmente para esta, sin embargo, algunas imágenes utilizadas en la interfaz gráfica fueron tomadas de bancos de imágenes libres.

El modelo y el controlador son manejados en un solo archivo para evitar la pérdida de información de las variables globales que, aunque podían pasarse por parámetros, su constante uso habría hecho que separar el modelo del controlador en términos de código lo tornara menos entendible, difícil de manejar y más susceptible a errores.

En la Figura 14 se observan los elementos de la aplicación y como se comunican entre ellos.

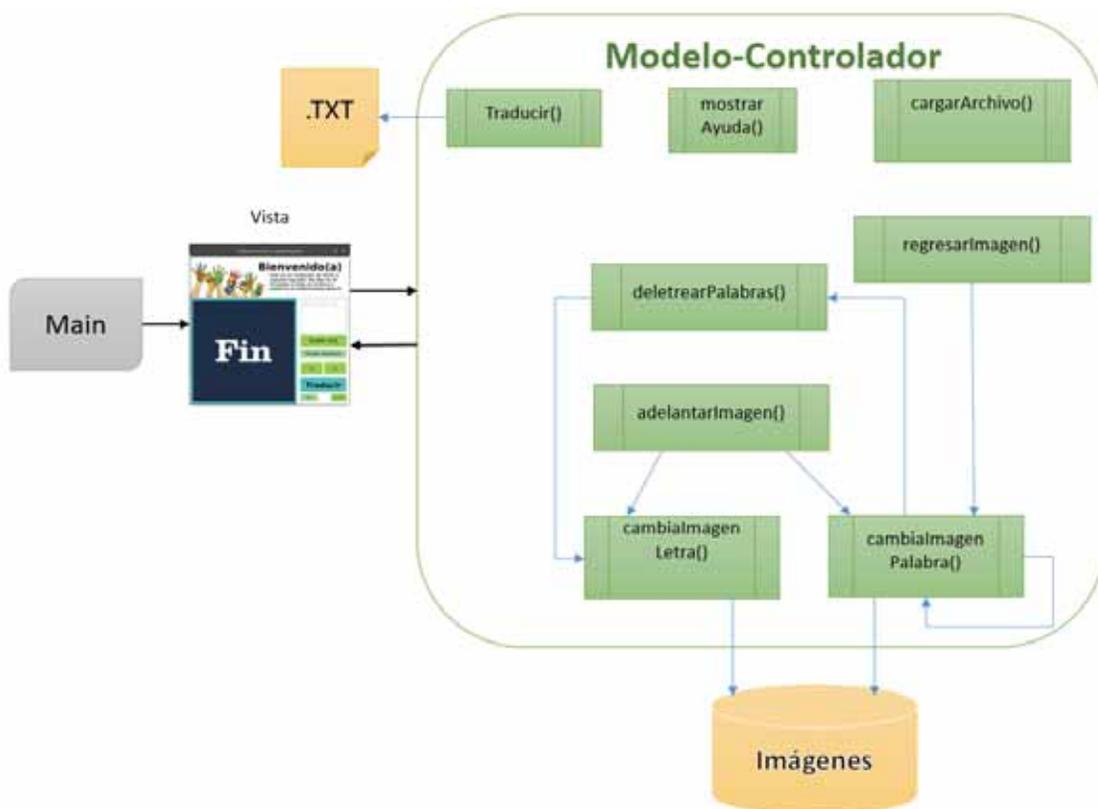


Figura 14. Estructura de la aplicación.

En los recuadros verdes se pueden ver todos los métodos, sin embargo, solo los mostrados en la Figura 15 son llamados al desencadenarse un evento. Dichos eventos ocurren al presionar los botones de la interfaz.

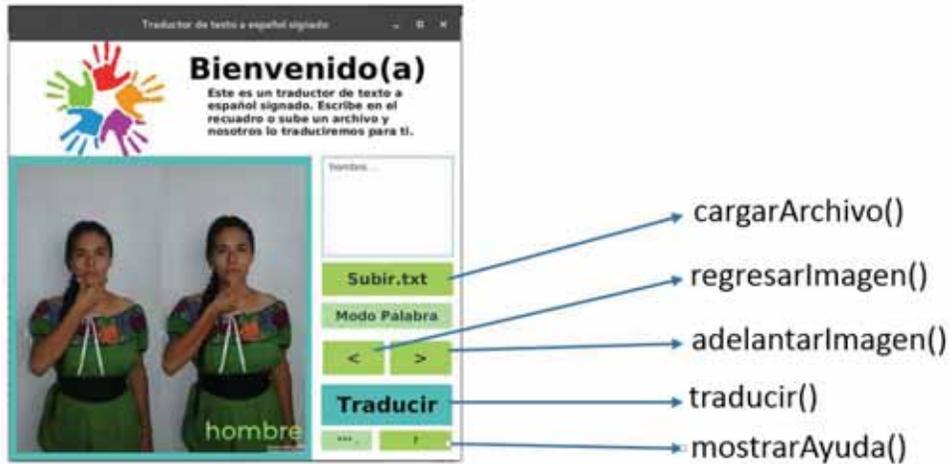


Figura 15. Botones con eventos en la interfaz de usuario.

Diagrama de casos de uso:

El traductor puede encontrarse en 2 modos: modo palabra o modo letra, los cuales muestran imágenes de letras o palabras en el panel de imágenes respectivamente. El modo por defecto es el de palabra. El modo letra se activa cuando una palabra solicitada no se encuentra entre las contenidas. Para decidir qué modo utilizar, el programa utiliza una bandera llamada “semaforo” y dos contadores “i” y “j”.

A continuación veremos cómo funciona la aplicación, dependiendo el evento que ocurra. En la Figura 16 vemos el diagrama de casos de uso (DCU) y después pasaremos a explicar cada uno de ellos.

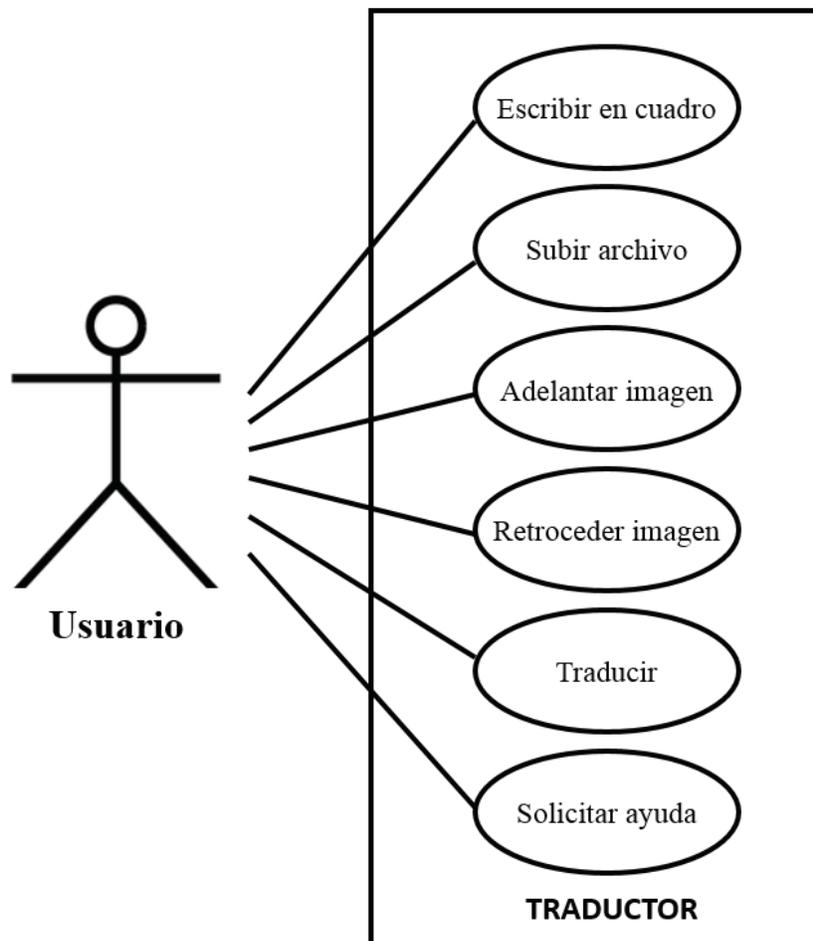


Figura 16. Diagrama de casos de uso.

Caso de uso Subir archivo

Botón: Subir.txt.

Descripción: Permite al usuario cargar un archivo al módulo de almacenamiento del programa para poder operarlo y traducirlo.

Evento: *cargarArchivo()*.

Métodos involucrados: Tiene asociado el método *cargarArchivo()*.

Flujo de actividades:

- 1.-Cambia el texto del botón btnTraducir a "Traducir" ya que si se ha presionado dicho botón previamente su texto cambia para indicar que la traducción está lista y hay que volverlo a su estado original.
- 2.-Permite al usuario seleccionar un archivo de su ordenador a través de la clase *FileChooser()*. Para esta aplicación, con el fin de facilitar las cosas, se ha configurado para mostrar solo archivos que terminan en.txt.
- 3.-Se obtiene la ruta absoluta del archivo seleccionado y se copia en la carpeta del proyecto para procesarlo, esto se logra haciendo una llamada al sistema operativo con el método *exec()* de la clase *Runtime*.

4.-Se cambia el texto del botón Subir.txt a “Subido” para indicarle al usuario que el archivo está listo para traducirse, además se coloca una imagen en el panel 1 (ver Figura 8) para que los usuarios que no saben leer entiendan lo que pasa.

Alternativo:

El usuario puede cancelar la subida del archivo y ya no se efectúan los pasos 1, 3 y 4.

Sí algo sale mal al subir el archivo se envía un mensaje de error en botón Subir.txt.

Condiciones previas: Tener el comando correcto para hacer la copia del archivo. En este caso es un comando para Linux.

Caso de uso Escribir en cuadro

Descripción: El usuario escribe texto en un *TextArea* de la interfaz.

Flujo de actividades: El área de texto almacena el texto ingresado por el usuario en un buffer en espera de ser utilizado.

Alternativo: No escribir en el área de texto.

Métodos involucrados: Método del *TextArea* *getText()*;

Condiciones previas: Ninguna

Caso de uso Adelantar imagen

Botón: > (El elemento 8 de la Figura 8).

Descripción: El usuario presiona el botón “>” y se muestra la imagen correspondiente a la seña de la siguiente palabra en el texto que se está traduciendo.

Evento: *adelantarImagen()*. Se encarga de validar el fragmento de texto que se esté analizando, convertirlo en algo manejable para los métodos con los que se comunica y de actualizar banderas.

Métodos involucrados:

cambialmagenPalabra().- Verifica que le han pasado un nombre de imagen válido y de eliminar símbolos no deseados, actualiza el botón *Aux* si hay símbolos junto a la palabra y llama a la función *deletrearPalabras()* si la palabra no se encuentra en el banco de imágenes. Además actualiza la imagen que se está mostrando al usuario.

El método *cambialmagenPalabra()* se llama a sí mismo de manera recursiva y se encarga de validar también los siguientes aspectos:

Palabras rodeadas de símbolos: Como se sabe, en el idioma escrito se encuentran constantemente signos de puntuación que rodean las palabras, por ejemplo:

palabra, palabra.

Palabra: palabra;

¿palabra palabra?

Palabra! ¡palabra

(palabra palabra)

(palabra)

Dichos símbolos no se utilizan en el habla y no forman parte de la palabra en sí, tampoco de una conversación en LSM.

El método valida estos casos y retira los símbolos para poder hacer la búsqueda de la palabra. Los símbolos consecutivos que se encuentran junto a palabras son descartados para

no confundir al usuario, sin embargo se colocan en el botón Aux para servir de guía al usuario que sepa interpretarlos (un usuario que sabe leer).

El método también valida si la secuencia de caracteres no es una palabra sino números y los muestra por separado.

deletrearPalabras().-Descompone una palabra en letras para que sean procesadas por la función *cambialmagenLetra()* y actualiza banderas.

cambialmagenLetra().- Actualiza banderas y muestra imágenes correspondientes a letras de palabras.

Flujo de actividades:

1.-El botón Traducir se cambia de texto a “Traducir” con el propósito de recordar al usuario (lector) que puede traducir cualquier otro archivo o texto ingresado en el área de texto cuando lo desee o también reiniciar el flujo del archivo que se está traduciendo.

2.-El programa llama al método *cambialmagenPalabra()* o al método *cambialmagenLetra()* dependiendo de si una bandera “semaforo” se encuentra en el valor 0 o 1 respectivamente. El evento *adelantarImagen()* también es encargado de detectar si se ha llegado al final del archivo o de un arreglo de letras de alguna palabra.

Condiciones previas: El archivo de texto debe contener al menos un carácter.

Alternativo: Puede ocurrir que un símbolo no se encuentre guardado en el módulo de almacenamiento, en este caso dicho símbolo será ignorado.

Caso de uso Retroceder Imagen

Botón: <

Descripción: El usuario presiona el botón “<” y se muestra la imagen correspondiente a la seña de la palabra anterior en el texto que se está traduciendo.

Evento: *regresarImagen()*;

Métodos involucrados:

cambialmagenPalabra();

Flujo de actividades:

1.-El botón Traducir se cambia de texto a “Traducir”.

2.-Se muestra la palabra anterior en el archivo de texto que se está procesando.

Condiciones previas: El archivo de texto debe contener al menos un carácter.

Caso de uso Traducir texto

Botón: Traducir.

Descripción: El usuario presiona el botón Traducir y el programa se encarga de alistar los componentes necesarios para mostrar las imágenes de las señas equivalentes a las palabras escritas que el usuario pretende traducir.

Métodos con los que se comunica:

Métodos de lectura y escritura de archivos y métodos de manipulación de listas.

Flujo de actividades:

1.-El usuario presiona el botón traducir después de subir un archivo o ingresar texto en el área de texto.

2.-Se crea un canal de lectura a un archivo “texto.txt” que de no existir se crea.

3.-Se toman las palabras que se encuentren en el archivo texto.txt y se procesan, es decir, se separan cadenas de caracteres cada que se encuentra un " " (espacio en blanco) y se colocan en una lista.

4.-Se cambia el texto del botón Traducir a "Hecho" para indicar al usuario que se realizó la acción. En el panel de imágenes se coloca una palomita con el mismo fin hacia personas que no saben leer.

Alternativo: El usuario ingresa un texto a través del teclado usando el área de texto. En este caso el texto se obtiene del área de texto al presionar el botón Traducir y se manda a un archivo desde el cual será procesado, luego se siguen los pasos mencionados anteriormente.

El usuario no ingresa texto ni sube un archivo. En este caso se traduce lo último que quedó en el archivo texto.txt.

Condiciones previas: ninguna.

Caso de uso Solicitar ayuda

Botón: ?

Descripción: El usuario presiona el botón "?" y se abre un archivo PDF con un manual de usuario.

Métodos con los que se comunica: Métodos de la clase File para procesar el archivo PDF.

Flujo de actividades:

1.-Se envía un comando para el sistema operativo a través del método *exec()*. Dicho comando le dice al S.O. que habrá el archivo PDF llamado "ayuda.pdf".

Condiciones previas: Tener el archivo PDF en la carpeta del proyecto.

Los métodos explicados anteriormente se encargan además de manipular los índices de las palabras o letras, según se requiera para poder mostrar la palabra o letra, anterior o siguiente, manejando dos iteradores distintos "i" y "j" que se coordinan con la variable de bandera para saber cuándo incrementarse o decrementarse.

Se ha decidido no retroceder entre letras pues no es algo natural, sería como intentar leer al revés y el usuario tendría que estar regresando tantos de letras para poder verlas de nuevo en orden, por ello al retroceder a una imagen que no se encuentra, el deletreo empieza desde la letra inicial de la palabra y continua hacia adelante.

Los símbolos y signos de puntuación se muestran en imágenes de sí mismos (no en seña) ya que no tienen equivalente en señas.

Portabilidad

La aplicación depende de comandos que no pueden ser sujetos simplemente a una elección entre Linux o Windows ya que también dependen de en qué lugar se ha almacenado el proyecto para establecer las rutas de los directorios, es por ello que no se ha implementado una función de reconocimiento del sistema operativo. Quien quiera utilizar la aplicación deberá verificar que la ruta del programa coincida con la ubicación del proyecto. Si lo anterior es correcto la aplicación se ejecutará en un entorno de desarrollo eclipse con el complemento *e(fx)clipse* instalado.

Resultados

Inicialmente en la propuesta se plantearon los objetivos:

- Diseñar e implementar un módulo para la selección del vocabulario de LSM.
- Diseñar e implementar un módulo de almacenamiento del vocabulario.
- Diseñar e implementar un módulo para la administración del vocabulario.
- Diseñar e implementar un módulo que permita traducir texto a LSM.
- Diseñar e implementar una interfaz de usuario.

Que más adelante en este proyecto se modificaron muy levemente para dejar el objetivo 3 como:

- Diseñar e implementar un módulo para el procesamiento de texto.

Pues se decidió que tiene un nombre más adecuado para sus funciones.

Todos los objetivos planteados en la propuesta fueron alcanzados con satisfacción.

La aplicación cumple con lo prometido e incluso se le ha encontrado la cualidad de que no solo servirá para comunicarse con la comunidad sordo muda sino también de aprender el lenguaje de señas.

A pesar de que todos los objetivos fueron alcanzados se ha detectado potencial en la aplicación. Los posibles cambios, mejoras y adiciones se plantean en la sección: Trabajo Futuro.

Un cambio importante que se realizó en comparación con lo propuesto fue que las 500 imágenes no se limitaron únicamente a las más repetidas, sino que se agregaron palabras orientadas a temas. La lista completa de palabras, letras, símbolos y números que maneja la aplicación se anexa en un archivo de texto en el disco de este proyecto del mismo modo el código fuente de la aplicación.

Trabajo futuro

Durante el desarrollo del proyecto y su culminación se fueron encontrando posibles mejoras y aditamentos que podrían realizarse en la aplicación para mejorar los beneficios que puede brindarle al usuario, por ejemplo:

1.-Se pueden cambiar los botones de texto por iconos diseñados especialmente para que personas sordo-mudas puedan comprenderlos con facilidad. Lo anterior no se realizó porque se cree que es un trabajo para un diseñador, en colaboración del sector que va a beneficiarse. Intentar diseñarlo no sería imposible, pero hubiera consumido un tiempo significativo del desarrollo del proyecto.

2.-Hacerlo web. Una aplicación de este tipo merece estar disponible para el usuario común, es por ello que se recomienda hacerla web cuanto antes. No se planteó de esta manera por la poca experiencia del programador en cuanto a este tipo de aplicaciones.

3.-En la parte de ayuda, se incluye un manual dirigido a lectores ya que no se pudo conseguir un intérprete. Lo que se plantea es que el botón de ayuda abra un vídeo explicativo en lenguaje de señas para que de esta manera los usuarios con problemas de audición puedan comprenderlo mejor.

4.-Reproducción automática. La presente aplicación requiere que se presionen las flechas de dirección para ir navegando entre las palabras, sin embargo se puede implementar una opción de reproducción automática para las personas que ya estén familiarizadas y que no quieren estar presionando un botón por cada palabra.

5.-Pensando en traductores profesionales como el de Google, no está mal proponer un módulo que convierta voz a texto para que una persona le dicte a la aplicación lo que quiere traducir.

Conclusiones

Se puede notar que no hubo grandes cambios en cuanto al desarrollo del proyecto presentado en este documento y lo planteado en la propuesta de dicho proyecto, solo cambios menores que surgieron al profundizar en el desarrollo del mismo.

Se debe tener claro que el LSM no es lo mismo que el español signado, a las personas que no saben leer les costará un poco adaptarse a la estructura del lenguaje escrito, sin embargo, ver las señas y no palabras en forma de texto, también les ayudará a familiarizarse con la escritura del lenguaje.

Para aquellos que no estén interesados en aprender sobre el lenguaje escrito, sería bueno implementar una opción de traducción natural. Evidentemente este es un tema muy complejo que requiere de mucho tiempo, es por ello que se plantea como trabajo futuro.

Referencias

- [1] BBC, “Las 5 cosas menos conocidas sobre la sordera”, 2014. [En línea]. Disponible en: http://www.bbc.com/mundo/noticias/2014/08/140808_salud_cinco_cosas_que_no_sabe_sordera_lv. [Accedido: 8-nov-2017]
- [2] Montserrat Sánchez Maldonado, “César: el maestro de lenguaje de señas al que vivir en silencio no lo detuvo”, 2016. [En línea]. Disponible en: <http://www.animalpolitico.com/2016/09/lenguaje-personas-sordas-mexico/>. [Accedido: 8-nov-2017].
- [3] Smith-Stark, Thomas C, La lengua manual mexicana. Unpublished paper: Colegio de México, 1986.
- [4] INEGI, “Lenguas indígenas en México”, 2015. [En línea]. Disponible en: http://cuentame.inegi.org.mx/hipertexto/todas_lenguas.htm. [Accedido: 8-nov-2017]
- [5] SIPSE, “En México se hacen ciegos ante los sordos”, 2016. [En línea]. Disponible en: <http://sipse.com/mexico/sordos-discapacidad-gobierno-mexico-224324.html>. [Accedido: 8-nov-2017].
- [6] J. A. López Castañeda, “Sistema de Enseñanza para la Articulación del Lenguaje por medio de Fonemas para Niños con Problemas del Habla”, proyecto terminal, Universidad Autónoma Metropolitana, 2012.
- [7] E. Solano Duarte, “Prototipo de una aplicación para PDA que facilite la comunicación de las necesidades de un niño con problemas del habla”, proyecto terminal, Universidad Autónoma Metropolitana, 2011.
- [8] A. Rebollar Gómez, “Sistema para expresar las necesidades básicas en personas con trastornos del habla”, proyecto terminal, Universidad Autónoma Metropolitana, 2017.
- [9] L. A. Blanco Monterrosa, “¿Cómo fomentar el lenguaje de señas colombianas en los estudiantes de séptimo grado (7) de la institución educativa de ternera?”, tesis de licenciatura”, Univ. De Cartagena, 2016. [En línea]. Disponible en: <http://190.242.62.234:8080/jspui/bitstream/11227/5297/1/Tesis%20Final.pdf>. [Accedido: 8-nov-2017].

- [10] F. P. Priego Pérez, “Reconocimiento de imágenes del lenguaje de señas mexicano”, IPN, 2012. [En línea]. Disponible en: <http://tesis.ipn.mx/bitstream/handle/123456789/15991/XM%2013.05.pdf?sequence=1&isAllowed=y>. [Accedido: 8-nov-2017].
- [11] P. A. Espinosa Aguilar, H. A. Pogo León, “Diseño y construcción de un guante prototipo electrónico capaz de traducir el lenguaje de señas de una persona sordomuda al lenguaje de letras”, tesis terminal, Univ. Politécnica Salesiana, 2013 [En línea]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/4211/1/UPS-CT002598.pdf>. [Accedido: 8-nov-2017].
- [12] Wikipedia, “Java (lenguaje de programación)” [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). [Accedido: 20-jun-2018].
- [13] Wikipedia, “JavaFX” [En línea]. Disponible en: <https://es.wikipedia.org/wiki/JavaFX>. [Accedido: 20-jun-2018].
- [14] EDU, “Qué son las interfaces gráficas o GUI” [En línea]. Disponible en: <http://fp.uoc.edu/blog/que-son-las-interfaces-graficas-o-gui/>. [Accedido: 20-jun-2018].
- [15] WORDPRESS, “El Patrón MVC (Modelo Vista Controlador)” [En línea]. Disponible en: <https://prestashop5estrellas.wordpress.com/2010/03/29/el-patron-mvc-modelo-vista-controlador/>. [Accedido: 20-jun-2018].
- [16] Oracle, “JavaFX SceneBuilder” [En línea]. Disponible en: <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>. [Accedido: 20-jun-2018].
- [17] Oracle, Utilizando FMXMLoader [En línea]. Disponible en: <http://www.oracle.com/technetwork/es/articles/java/fxmlloader-parte1-4414430-esa.html>. [Accedido: 20-jun-2018].
- [18] fotoefectos, “Hacer collage de dos fotos verticalmente” [En línea]. Disponible en: <https://www.fotoefectos.com/cat-collages/collageMix2>. [Accedido: 20-jun-2018].
- [19] BBC, “¿Cuántas palabras se necesitan para comunicarse?” [En línea]. Disponible en: http://www.bbc.com/mundo/noticias/2011/04/110330_palabras_ingles_lp. [Accedido: 8-nov-2017].
- [20] RAE, “1000 formas” [En línea]. Disponible en: http://corpus.rae.es/frec/1000_formas.TXT. [Accedido: 20-jun-2018].

- [21] Aprendiendo Lengua de Señas Mexicana, “Verbos 6”, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=PWpL9Ji9qXY> [Accedido: 20-julio-2018]
- [22] Daniela Pardo, PALABRAS ADICIONALES LSM, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=e-HKeV1qgG4&t=> [Accedido 20-julio-2018]
- [23] Digital Brain México, “Curso de Lengua de Seña Mexicanas - Preposiciones”, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=bELI5FKCjuY>. [Accedido: 20-jun-2018].
- [24] El Rincón de Molteni, “Vocabulario Común en la LSM”, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=d74Y5sE0868>. [Accedido: 20-jun-2018].
- [25] El Rincón de Molteni, “Vocabulario en LSM Cine”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=mqvu-fvdwac&t=>. [Accedido: 20-jun-2018].
- [26] Digital Brain México, “Curso de Lengua de Seña Mexicanas - Adverbios”, 2016. [En línea]. Disponible en: https://www.youtube.com/watch?v=1gQpBs8_PrU. [Accedido: 20-jun-2018].
- [27] El Rincón de Molteni, “Vocabulario Legal en LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=P7FGpKtlztl>. [Accedido: 20-jun-2018].
- [28] El Rincón de Molteni, “Oficios y Profesiones en LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=zjNNVRsMhVI>. [Accedido: 20-jun-2018].
- [29] El Rincón de Molteni, “Vocabulario Médico 2da Parte LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=DSpelpOVrlg>. [Accedido: 20-jun-2018].
- [30] El Rincón de Molteni, “Vocabulario Escolar en LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=Md9EaeQ9KIY>. [Accedido: 20-jun-2018].
- [31] El Rincón de Molteni, “Adverbios de Tiempo LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=CwHSLLMMe2o>. [Accedido: 20-jun-2018].
- [32] José Manuel Ochoa Alcantar, “Adjetivos en Lengua de Señas Mexicana”, 2015. [En línea]. Disponible en: <https://www.youtube.com/watch?v=-wfgCbslmlA>. [Accedido: 20-jun-2018].
- [33] El Rincón de Molteni, “Vocabulario Común Médico en LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=wEtVaJd1SHk>. [Accedido: 20-jun-2018].

- [34] Aprendiendo Lengua de Señas Mexicana, “Verbos 1”, 2015. [En línea]. Disponible en: <https://www.youtube.com/watch?v=vYqeK0o8Oso>. [Accedido: 20-jun-2018].
- [35] El Rincón de Molteni, “Sustantivos, adjetivos, sentimientos y estados de ánimo en LSM”, 2016 . [En línea]. Disponible en: <https://www.youtube.com/watch?v=nKGH3mJxRcs>. [Accedido: 20-jun-2018].
- [36] Terapia Física / Lengua de Señas, “Verbos - LSM”, 2015. [En línea]. Disponible en: <https://www.youtube.com/watch?v=Ql49yQh6ku8>. [Accedido: 20-jun-2018].
- [37]]El Rincón de Molteni, “Verbos Comunes en LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=USQOKJSForE>. [Accedido: 20-jun-2018].
- [38] Beatriz Cota Ponce, “LSM Algunos verbos narrativos”, 2014 . [En línea]. Disponible en: <https://www.youtube.com/watch?v=AWYEpjX2d8Q>. [Accedido: 20-jun-2018].
- [39] CREE DIF NL, “LENGUA DE SEÑAS MEXICANA TEMA 8: VERBOS COMUNES Diccionario Español-LSM”, 2013. [En línea]. Disponible en: <https://www.youtube.com/watch?v=l3NybQ82gCo>. [Accedido: 20-jun-2018].
- [40] José Manuel Ochoa Alcantar, “Verbos 5 septiembre”, 2015. [En línea]. Disponible en: <https://www.youtube.com/watch?v=snmGraMm7e8>. [Accedido: 20-jun-2018].
- [41] UT Santa Catarina, “UTSC - Vocabulario común y palabras relacionadas con discapacidad, LSM (10/10)”, 2013. [En línea]. Disponible en: <https://www.youtube.com/watch?v=FndMn6XQRUo>. [Accedido: 20-jun-2018].
- [42] Vanessa Hernández, “ADJETIVOS LSM”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=d5qk01EcSPE>. [Accedido: 20-jun-2018].
- [43] Daniela Pardo, “CONCEPTOS RELACIONADOS CON MEDIOS DE TRANSPORTE LSM”. 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=LxfVYd36EpQ>. [Accedido: 20-jun-2018].
- [44] Daniela Pardo, “PROFESIONES Y CONCEPTOS RELACIONADOS LSM”, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=Kz9SJDVW-zc>. [Accedido: 20-jun-2018].
- [45] Dilo en señas, “Guía Dilo en señas - Liderazgo”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=4HHBdnzLBNI>. [Accedido: 20-jun-2018].
- [46] Dilo en señas, “Guía Taxi Cobra”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=Tx2ny65eP1l>. [Accedido: 20-jun-2018].

- [47] Dilo en señas, “Guía Dilo en señas - Billiards Pool - Bar Deportivo”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=tFg4K038Vfs>. [Accedido: 20-jun-2018].
- [48] Dilo en señas, “Guía Dilo en señas - Windsor Eventos - Eventos Sociales”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=soNPgTLiwNE>. [Accedido: 20-jun-2018].
- [49] Dilo en señas, “Guía Dilo en señas - LSM - CIRI - Innovación & Inclusión”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=rgRvbVRYyWM>. [Accedido: 20-jun-2018].
- [50] Dilo en señas, “Guía Dilo en señas - LSM - Turista”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=YN79yDqVooc>. [Accedido: 20-jun-2018].
- [51] Dilo en señas, “Guía Dilo en señas - LSM - Playa”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=waVau46MbXA>. [Accedido: 20-jun-2018].
- [52] Dilo en señas, “Guía Dilo en señas - LSM - CIRI - Innovación & Inclusión”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=rgRvbVRYyWM>. [Accedido: 20-jun-2018].
- [53] Dilo en señas, “Tutorial Dilo en señas - LSM - Abecedario”, 2018. [En línea]. Disponible en: <https://www.youtube.com/watch?v=7B1C5fFejN8>. [Accedido: 20-jun-2018].
- [54] Dilo en señas, “Tutorial Dilo en señas - LSM - Mujeres y profesiones”, 2018. [En línea]. Disponible en: <https://www.youtube.com/watch?v=uHFPgpZq0T0>. [Accedido: 20-jun-2018].
- [55] Dilo en señas, “Tutorial Dilo en señas - LSM - Frutas”, 2018. [En línea]. Disponible en: https://www.youtube.com/watch?v=p_nOgCM0-IU. [Accedido: 20-jun-2018].
- [56] Dilo en señas, “Tutorial Dilo en señas - LSM - Animales (2)”, 2018. [En línea]. Disponible en: <https://www.youtube.com/watch?v=U4TdXRuY9nw>. [Accedido: 20-jun-2018].
- [57] Dilo en señas, “Tutorial Dilo en señas - LSM - Emociones”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=COnbIBGoAQc>. [Accedido: 20-jun-2018].
- [58] Dilo en señas, “Tutorial Dilo en señas - LSM - Lugares”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=1lixb-pHpWQ>. [Accedido: 20-jun-2018].

- [59] Dilo en señas, “Tutorial Dilo en señas - LSM - Verbos”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=KssVlw6Ha1E>. [Accedido: 20-jun-2018].
- [60] Dilo en señas, “Tutorial Dilo en señas - LSM - Animales”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=KNfg9iaB5MQ>. [Accedido: 20-jun-2018].
- [61] Dilo en señas, “Tutorial Dilo en señas - LSM - Escuela”, 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=jf-hzeePRA>. [Accedido: 20-jun-2018].
- [62] Dilo en señas, “Tutorial Dilo en señas - LSM - Colores”, 2016. [En línea]. Disponible en: https://www.youtube.com/watch?v=ayXvMdd_InQ. [Accedido: 20-jun-2018].
- [63] Dilo en señas, “Tutorial Dilo en señas - LSM - Hogar”, 2016. [En línea]. Disponible en: <https://www.youtube.com/watch?v=296AZixuwPs>. [Accedido: 20-jun-2018].
- [64] Dilo en señas, “Tutorial Dilo en señas - LSM - Familia”, 2016. [En línea]. Disponible en: https://www.youtube.com/watch?v=wNbf9yt_18w. [Accedido: 20-jun-2018].
- [65] César Ernesto Escobedo Delgado (coordinador), *“Diccionario de Lengua de Señas Mexicana de la Ciudad de México”*. Primera edición. CDMX: México, 2017.
- [66] Mariela Gómez Roquero (coordinadora), *“Manos con voz - Diccionario de Lengua de Señas Mexicanas”*. Primera edición. México: Libre Acceso A. C. y Consejo Nacional para Prevenir la Discriminación, 2011.
- [67] Sistema Municipal DIF, *“Manual de Lengua de Señas Mexicana”*. Primera edición. México: Dirección de Atención a Personas con Discapacidad, 2014.

Entregables

El proyecto completo se adjuntará en un disco. Incluye el proyecto en eclipse, el documento del proyecto y un manual de usuario en formato PDF.