

---

# Text2LSE: Traductor de Texto a Lengua de Signos Española (LSE)

---

Text2LSE: Text to Spanish Sign Language Translator



**TRABAJO DE FIN DE GRADO 2019/2020**

**AUTORES**

Sara Vegas Cañas  
Miguel Rodríguez Cuesta  
Alejandro Torralbo Fuentes

**DIRECTORES**

Virginia Francisco Gilmartín  
Antonio Fernando García Sevilla

*Grado de Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid*

*Convocatoria 2019/2020*

Este documento está preparado para ser imprimido a doble cara.

# Text2LSE: Traductor de Texto a Lengua de Signos Española (LSE)

**Text2LSE: Text to Spanish Sign Language  
Translator**

**Trabajo de Fin de Grado en Ingeniería Informática**

## **AUTORES**

**Sara Vegas Cañas  
Miguel Rodríguez Cuesta  
Alejandro Torralbo Fuentes**

## **DIRECTORES**

**Virginia Francisco Gilmartín  
Antonio Fernando García Sevilla**

*Grado de Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid*

*Convocatoria 2019/2020*



# Resumen

En la sociedad actual en la que vivimos es imprescindible la comunicación, ya que esta permite la integración e interacción social. Actualmente, muchas personas que sufren discapacidad auditiva se enfrentan a numerosas barreras comunicativas y necesitan el apoyo de métodos alternativos de comunicación, como la Lengua de Signos (LS), para integrarse completamente en la sociedad. Hoy en día, casi ningún material audiovisual cuenta con la LS integrada, lo que supone que muchas personas con discapacidad auditiva no tengan acceso a ciertos servicios fundamentales, como por ejemplo avisos por megafonía en una estación de tren o un hospital. En este TFG hemos desarrollado una herramienta gratuita que permite traducir un texto en castellano a Lengua de Signos Española (LSE) en formato vídeo o imagen y de forma instantánea. Text2LSE permitirá así incorporar la LS a cualquier material audiovisual, lo que contribuye favorablemente a la adaptación de este colectivo y su inclusión en la sociedad. A día de hoy no existe ninguna herramienta que sea capaz de traducir castellano a LSE en tiempo real, por lo que nuestro TFG puede ayudar a cubrir esa necesidad y mejorar la calidad de vida de las personas con discapacidad auditiva. La herramienta se ha desarrollado siguiendo una Arquitectura Orientada a Servicios (SOA), por lo que otros desarrolladores podrán integrar los servicios que hemos desarrollado para implementar cada una de las funcionalidades de nuestra aplicación en sus propias aplicaciones. Además, se ha realizado una evaluación de las traducciones proporcionadas por nuestra herramienta y, aunque hay mucho margen de mejora, los resultados han sido muy satisfactorios. Con el desarrollo de este TFG hemos establecido una buena base para la traducción a LS, ya que la aplicación es capaz de traducir un número considerable de oraciones, fácilmente ampliable si se continua con el trabajo realizado.

## Palabras clave

- Discapacidad auditiva
- Accesibilidad
- Lengua de Signos
- Lengua de Signos Española
- Procesamiento del Lenguaje Natural
- Servicios web
- Aplicación web

# Abstract

In today's society, communication is essential, since it allows social integration and interaction. Nowadays, many people with hearing disabilities face numerous communication barriers and need the support of alternative communication methods, such as Sign Language (SL), to fully integrate into society. Today, most audiovisual material have not integrated SL, which means that many people with hearing disabilities do not have access to certain essential services, such as public address announcements at train stations or hospitals. In this project, we have developed a free tool that allows to instantly translate a Spanish text into Spanish Sign Language (SSL) in video or image format. Thus, Text2LSE will allow SL to be incorporated into any audiovisual material, which contributes favorably to the adaptation of this group and its inclusion into society. Nowadays, there is no tool that is capable of translating Spanish into SSL in real time, so our project can help cover that need and improve quality of life for people with hearing disabilities. The tool has been developed following a Service Oriented Architecture (SOA), so other developers will be able to integrate the functionalities of our application into their own applications. Additionally, an evaluation of the translations provided by our tool has been carried out. Although there is room for improvement, results have been very satisfactory. The development of this project allowed us to establish a good basis for the translation to SL, since the application is capable of translating a considerable number of sentences. Nevertheless, this work is easily expandable should it be continued.

## Keywords

- Hearing disabilities
- Accessibility
- Sign Language
- Spanish Sign language
- Natural Language Processing
- Web services
- Web Application

# Índice

<b>Resumen</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Estructura del documento . . . . .	3
<b>1. Introduction</b>	<b>5</b>
1.1. Motivation . . . . .	5
1.2. Goals . . . . .	6
1.3. Document structure . . . . .	7
<b>2. Estado del Arte</b>	<b>9</b>
2.1. La discapacidad auditiva . . . . .	9
2.2. Sistemas Aumentativos y Alternativos de Comunicación (SAAC)	10
2.3. La Lengua de Signos . . . . .	12
2.3.1. Reglas de la Lengua de Signos Española . . . . .	14
2.3.2. Bancos de videos e imágenes LSE . . . . .	24
2.4. Aplicaciones de Traducción de Texto a LSE . . . . .	28
2.4.1. Text2Sign . . . . .	28
2.4.2. TextoSign . . . . .	29
2.4.3. StorySign . . . . .	29
2.4.4. TeCuento . . . . .	29
2.4.5. Conclusiones . . . . .	31
2.5. Procesamiento del Lenguaje Natural . . . . .	31
2.5.1. Niveles del Procesamiento del Lenguaje Natural . . . . .	32
2.5.2. Herramientas PLN . . . . .	33
<b>3. Tecnologías utilizadas</b>	<b>35</b>
3.1. Servicios Web . . . . .	35

3.1.1.	Servicios Web SOAP . . . . .	35
3.1.2.	Servicios Web REST . . . . .	36
3.1.3.	Ventajas de Servicios Web . . . . .	37
3.1.4.	Desventajas de Servicios Web . . . . .	37
3.2.	Flask . . . . .	37
3.3.	Nginx . . . . .	38
3.4.	Google Colaboratory . . . . .	39
<b>4.</b>	<b>Metodologías utilizadas</b>	<b>41</b>
4.1.	Metodologías de gestión del proyecto . . . . .	41
4.2.	Trello . . . . .	43
4.3.	Reuniones . . . . .	44
4.4.	Control de versiones . . . . .	44
4.4.1.	Estructura de Trabajo . . . . .	45
<b>5.</b>	<b>Text2LSE</b>	<b>47</b>
5.1.	Arquitectura . . . . .	47
5.2.	Back-End . . . . .	50
5.2.1.	Recursos LSE . . . . .	50
5.2.2.	Servicio web de traducción de palabra a vídeo LSE . . . . .	50
5.2.3.	Servicio web de traducción de palabra a imagen LSE . . . . .	51
5.2.4.	Servicio web para Traducción de texto en castellano a texto en LSE . . . . .	53
5.2.5.	Servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de vídeos de ARASAAC . . . . .	60
5.2.6.	Servicio web de traducción de texto en LN a texto LSE adaptado a las imágenes del catálogo de ARASAAC . . . . .	61
5.2.7.	Servicio web para traducción de texto a LSE (vídeo) . . . . .	63
5.2.8.	Servicio web para traducción de texto a LSE (imagen) . . . . .	65
5.3.	Front-End . . . . .	67
<b>6.</b>	<b>Evaluación</b>	<b>75</b>
6.1.	Diseño de la evaluación . . . . .	75
6.2.	Resultados de la Evaluación . . . . .	77
6.3.	Análisis de los Resultados . . . . .	79
<b>7.</b>	<b>Trabajo individual</b>	<b>85</b>
7.1.	Sara Vegas Cañas . . . . .	85
7.2.	Alejandro Torralbo Fuentes . . . . .	86
7.3.	Miguel Rodríguez Cuesta . . . . .	88
<b>8.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>91</b>

---

8.1. Conclusiones . . . . .	91
8.2. Trabajo Futuro . . . . .	95
<b>8. Conclusions and Future Work</b>	<b>101</b>
8.1. Conclusions . . . . .	101
8.2. Future Work . . . . .	105
<b>A. Corpus de oraciones de instrucciones de seguridad de un vuelo</b>	<b>111</b>
<b>B. Corpus de oraciones de un capítulo de una serie infantil</b>	<b>115</b>
<b>C. Corpus de subtítulos de un trailer de una película</b>	<b>117</b>
<b>D. Corpus de subtítulos de un anuncio</b>	<b>119</b>
<b>E. Corpus de frases simples</b>	<b>121</b>
<b>F. Traducción de los corpus mediante Text2LSE</b>	<b>123</b>
F.1. Corpus de oraciones de instrucciones de seguridad de un vuelo	123
F.2. Corpus de oraciones de un capítulo de una serie infantil . . .	152
F.3. Corpus de subtítulos de un trailer de una película . . . . .	169
F.4. Corpus de subtítulos de un anuncio . . . . .	175
F.5. Corpus de frases simples . . . . .	180
<b>Bibliografía</b>	<b>193</b>



# Índice de figuras

2.1. Representación del abecedario en el Sistema Braille . . . . .	12
2.2. Representación de distintas palabras con pictogramas . . . . .	13
2.3. Representación de la configuración de la mano en Lengua de Signos . . . . .	15
2.4. Signo “ <i>AYUDAR</i> ” en LSE del banco de imágenes ARASAAC	16
2.5. Representación de la orientación de las manos en Lengua de Signos . . . . .	16
2.6. Signo “ <i>VISERA</i> ” en LSE del banco de imágenes ARASAAC .	17
2.7. Signo “ <i>OIR</i> ” en LSE del banco de imágenes ARASAAC . . .	17
2.8. Ejemplos de tipos de movimientos realizados al signar en LSE	18
2.9. Signo “ <i>MUCHO</i> ” en LSE del banco de imágenes ARASAAC	18
2.10. Uso de texto para la enseñar la estructura de la LSE . . . . .	19
2.11. Signo “ <i>HOMBRE</i> ” del banco de imágenes ARASAAC . . . . .	20
2.12. Signo “ <i>MUJER</i> ” del banco de imágenes ARASAAC . . . . .	20
2.13. Signo “ <i>ABUELO</i> ” del banco de imágenes ARASAAC . . . . .	21
2.14. Signo “ <i>ABUELA</i> ” del banco de imágenes ARASAAC . . . . .	21
2.15. Signo “ <i>NIÑO</i> ” del banco de imágenes ARASAAC . . . . .	21
2.16. Signo “ <i>NIÑA</i> ” del banco de imágenes ARASAAC . . . . .	22
2.17. Signo “ <i>UNO</i> ” del banco de imágenes ARASAAC . . . . .	23
2.18. Signo “ <i>MUCHOS</i> ” del banco de imágenes ARASAAC . . . . .	23
2.19. Signo “ <i>QUERER</i> ” en LSE del banco de imágenes ARASAAC	25
2.20. Signo “ <i>SEÑALAR</i> ” en LSE del banco de imágenes ARASAAC	25
2.21. Signo “ <i>HOLA</i> ” en LSE del banco de imágenes ARASAAC . .	26
2.22. Recursos LSE de la Fundación CNSE . . . . .	27
2.23. Imagen de la página web SpreadTheSign . . . . .	28
2.24. Maya, avatar usado por TextoSign. . . . .	29
2.25. Imagen de la aplicación StorySign. . . . .	30
2.26. Imagen de la aplicación TeCuento. . . . .	30
2.27. Análisis sintáctico frase “Mi padre es abogado” . . . . .	32
2.28. Benchmark de diferentes procesadores de lenguaje . . . . .	33

3.1. Vista de un cuaderno de Google Colaboratory . . . . .	40
4.1. Estructura “ <i>Git</i> ” utilizada en el proyecto. . . . .	46
5.1. Aplicación web Text2LSE . . . . .	48
5.2. Esquema Proxy Inverso . . . . .	49
5.3. Flujo del servicio de Traducción de una palabra a vídeo en LSE	52
5.4. Imagen de error devuelta en la traducción de una palabra a LSE . . . . .	53
5.5. Flujo del servicio de Traducción de una palabra a imagen en LSE . . . . .	54
5.6. Imagen devuelto por el servicio de traducción de la palabra “ <i>coche</i> ” a LSE . . . . .	55
5.7. Flujo servicio de traducción de texto en castellano a texto en LSE . . . . .	56
5.8. Dependencias de una frase . . . . .	57
5.9. Tokens de la frase “ <i>Los niños meredaron chocolate</i> ” . . . . .	58
5.10. Displacy grafo de dependencias . . . . .	58
5.11. Árbol de ls oración “ <i>Los niños merendaron chocolate</i> ” . . . . .	59
5.12. Flujo servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de vídeos de ARASAAC . . . . .	62
5.13. Flujo servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de imágenes de ARASAAC . . . . .	64
5.14. Flujo del servicio web para traducción de texto a LSE (vídeo)	66
5.15. Flujo del servicio web para traducción de texto a LSE (ima- gen) . . . . .	68
5.16. JSON devuelto por el servicio web para traducción de texto a LSE (imagen) . . . . .	69
5.17. Flujo de la aplicación web al seleccionar la traducción en for- mato vídeo. . . . .	70
5.18. Flujo de la aplicación web al seleccionar la traducción en for- mato imagen. . . . .	71
5.19. Flujo de la aplicación web al seleccionar la traducción en for- mato texto. . . . .	72
5.20. Aplicación Web vista desde un dispositivo móvil . . . . .	73
6.1. Representación del signo “por favor” en LSE . . . . .	80
8.1. Oración con tiempo verbal y posesivos en LSE traducida por Text2LSE . . . . .	92
8.2. Oración con tiempo verbal y posesivos en LSE traducida por Text2LSE . . . . .	93
8.3. Oración con género y número en LSE traducida por Text2LSE	93

---

8.4. Oración con adjetivos en LSE traducida por Text2LSE . . . .	94
8.5. Ejemplo de mala traducción de un nombre propio realizada por Text2LSE . . . . .	96
8.6. Ejemplo de buena traducción de un nombre propio realizada por Text2LSE . . . . .	96
8.7. Traducción correcta de la palabra rampas realizada por Text2LSE	96
8.8. Traducción incorrecta de la palabra rampas realizada por Text2LSE	97
8.9. Ejemplo de mala traducción de tiempos compuestos realizada por Text2LSE . . . . .	97
8.10. Ejemplo de buena traducción de tiempos compuestos realizada por Text2LSE . . . . .	98
8.11. Signo en LSE para la frase hecha “Secar el pelo” . . . . .	98
8.12. Signo en LSE para la frase hecha “Secar el pelo” realizado por Text2LSE . . . . .	98
8.1. Verbal tense and possessive sentence in SSL translated by Text2LSE . . . . .	102
8.2. Verbal tense and possessive sentence in SSL translated by Text2LSE . . . . .	103
8.3. Sentence with gender and number in SSL translated by Text2LSE	103
8.4. Sentence with adjectives in SSL translated by Text2LSE . . . .	104
8.5. Example of bad translation of a proper name made by Text2LSE . . . . .	105
8.6. Example of a good translation of a proper name made by Text2LSE . . . . .	106
8.7. Correct translation of the word “rampas” by Text2LSE . . . .	106
8.8. Incorrect translation of the word “rampas” by Text2LSE . . . .	107
8.9. Example of bad translation of compound times made by Text2LSE	107
8.10. Example of good translation of compound times made by Text2LSE . . . . .	107
8.11. Sign in SSL for the phrase “Secar el pelo” . . . . .	108
8.12. Sign in SSL for the phrase “Secar el pelo” made by Text2LSE	108



# Índice de Tablas

6.1. Resultados de la evaluación: traducciones correctas . . . . .	78
6.2. Resultados de la evaluación: traducciones no exactas . . . . .	78
6.3. Resultados de la evaluación: traducciones entendibles . . . . .	78
6.4. Oraciones traducidas incorrectamente por cada tipo de error .	82
6.5. Resultados de la evaluación: traducciones correctas en texto LSE . . . . .	82
6.6. Resultados de la evaluación: traducciones no exactas en texto LSE . . . . .	83
6.7. Resultados de la evaluación: traducciones entendibles en texto LSE . . . . .	83
6.8. Resultados de la evaluación: signos buscados y signos no en- contrados . . . . .	84



# Capítulo 1

## Introducción

En la sección 1.1 de este capítulo se explicará la motivación que hay detrás de la realización de este TFG, y en la sección 1.2 los objetivos que nos marcamos al comienzo del proyecto. Por otro lado, en la sección 1.3 se detallará la estructura que sigue el presente documento.

### 1.1. Motivación

En la sociedad actual en la que vivimos hay una gran cantidad de personas con distintas discapacidades: cognitivas, físicas, visuales, auditivas... En algunos casos, dichas discapacidades hacen que estas personas se sientan apartadas del resto de la sociedad por no tener acceso completo a ciertos servicios fundamentales, como por ejemplo avisos por megafonía en lugares públicos como aeropuertos u hospitales, lo que dificulta su día a día. Gracias a los avances tecnológicos, se están desarrollando soluciones para ayudar a estos colectivos a acceder a distintos servicios y así integrarlos en la sociedad. Un ejemplo son las aplicaciones basadas en pictogramas (imágenes simples que representan un concepto), cuya finalidad es ayudar a comunicarse a personas con algún tipo de discapacidad cognitiva que les impide comunicarse usando el lenguaje natural.

En España, un 2,3 % de la población total (alrededor de un millón de personas) sufre algún tipo de discapacidad auditiva. Este colectivo usa distintos métodos de comunicación, que dependen tanto de la edad con la que empezaron a padecer sordera, como de su grado de pérdida auditiva. Las personas con discapacidad auditiva tienen las mismas necesidades de obtener información del entorno que el resto de la población, pero a pesar de disponer de las mismas capacidades cognitivas, se enfrentan a numerosas barreras comunicativas, que dificultan su proceso de aprendizaje y la capacidad de relacionarse con su entorno mediante la audición y la lengua oral. Esto se debe a que en muchas ocasiones el audio es el único canal para comunicar

información, ya sea en películas, informativos, conversaciones cotidianas, actividades educativas, megafonía o vídeos.

El método alternativo al lenguaje oral más extendido entre las personas con discapacidad auditiva es la Lengua de Signos (LS). La LS no es universal, sino que varía en función de la región, es decir, no existe una lengua de signos común en todo el mundo, disponiendo cada país de una o varias. En España desde el año 2007 hay dos lenguas reconocidas oficialmente: la Lengua de Signos Española (LSE) y la Lengua de Signos Catalana (LSC).

Hoy en día no todos los audios están complementados con la LS. Esto hace que las personas con discapacidad auditiva no terminen de estar integradas en la sociedad. Las personas sordas no se encuentran en igualdad de condiciones a la hora de recibir información, ya que en la mayoría de los casos el audio es el único canal para comunicar información en películas, informativos, programas, avisos en lugares públicos, etc. En muchos de los casos se cuenta con la posibilidad de añadir subtítulos que acompañen el audio, pero no es suficiente, ya que los subtítulos no son tan efectivos como la LS. La LS es ideográfica (permite representar conceptos comunes con un solo gesto), lo que hace que se tarde menos en recibir la información con la LS que con los subtítulos. Por todo esto es importante fomentar y facilitar el uso de la LS como lengua alternativa a la lengua oral.

Contar con una herramienta capaz de traducir un texto a Lengua de Signos resultaría de gran ayuda para personas con discapacidad auditiva, ya que ofrecería la posibilidad de introducir la LS a cualquier audio que disponga de subtítulos, e incluso de traducir directamente cualquier audio mediante el apoyo de herramientas de traducción de voz a texto.

## 1.2. Objetivos

El objetivo principal del proyecto es crear una herramienta gratuita que sea capaz de traducir un texto en lenguaje natural escrito en castellano a la Lengua de Signos Española (LSE) en tiempo real. La aplicación creada permitirá incorporar la LSE a cualquier material audiovisual de manera sencilla, y además servirá de apoyo a las personas en proceso de aprendizaje de la LSE.

La aplicación estará basada en una Arquitectura Orientada a Servicios (SOA), es decir, tendrá implementadas pequeñas funcionalidades como servicios web, fácilmente reutilizables por otros programadores que deseen integrar en sus aplicaciones las funcionalidades que vamos a desarrollar. Estos microservicios recibirán un texto en castellano y devolverán la traducción a LSE en varios formatos (video e imagen), haciendo uso de herramientas de

Procesamiento de Lenguaje Natural (PLN) para ello.

En cuanto a los objetivos académicos, con este proyecto pretendemos poner en práctica los conocimientos adquiridos a lo largo del Grado en Ingeniería Informática, así como ampliar nuestros conocimientos y competencias.

### **1.3. Estructura del documento**

En el Capítulo 2 se habla de la discapacidad auditiva y de los medios para comunicarse de los que disponen las personas con dicha discapacidad. También se da una visión más profunda de la Lengua de Signos, en concreto de la Lengua de Signos Española. Esto se complementa con un análisis de las herramientas que existen actualmente que ayudan a comunicarse y a aprender la LSE. En el Capítulo 3 se presentan las herramientas utilizadas para el desarrollo de la aplicación. En el Capítulo 4 se explica con detalle la metodología de trabajo seguida por los integrantes del grupo durante el desarrollo de este TFG. En el Capítulo 5 se explica en detalle el trabajo realizado. En el Capítulo 6 se explica la evaluación y pruebas realizadas a la aplicación desarrollada y se detallan los resultados. En el Capítulo 7 cada integrante del grupo explica el trabajo individual realizado a lo largo de todo el proyecto. Finalmente, en el Capítulo 8 se presentan las conclusiones y se detallan algunos trabajos futuros que se podrían llevar a cabo para mejorar la aplicación.



# Chapter 1

## Introduction

The motivation behind the implementation of this project will be explained in section 1.1. The objectives that we set ourselves at the beginning of the project will be presented in Section 1.2. Finally, the structure of this document will be detailed in Section 1.3.

### 1.1. Motivation

In current society in which we live, there is a large number of people with different disabilities: cognitive, physical, visual, hearing... In some cases, these disabilities make these people feel separated from the rest of society because they do not have complete access to certain fundamental services, like public address announcements in public places, such as airports or hospitals, which makes difficult their daily life. Current technological advances have allowed the development of solutions that allow these groups to access different services and, therefore, integrate them into society. As an example, applications based on pictograms (simple images that represent a concept), help people with some level of cognitive disability (which does not allow them to communicate using natural language) to communicate.

In Spain, 2.3% of the total population (around one million people) suffers from some type of hearing disability. This group uses different communication methods, which depend both on the age at which they become deaf, and on their degree of hearing loss. People with hearing disabilities have the same needs to obtain information as the rest of the population. But despite having the same cognitive abilities, they face numerous communication barriers, which hinder their learning process and the ability to relate to their environment through listening and speaking. This is because on many occasions audio is the only channel to communicating information, whether in films, news, everyday conversations, educational activities, public address systems or videos.

The alternative method to oral language most used among hearing impaired people is Sign Language (SL). The SL is not universal, it varies depending on the region, for example, there is no common worldwide sign language, but each country has one or more than one. In Spain, there are two officially recognized languages since 2007: Spanish Sign Language (SSL) and Catalan Sign Language (CSL).

Nowadays not all audios are complemented by the SL. This means that hearing impaired people are not fully integrated into society. Deaf people are not on equal terms when it comes to receiving information, since audio is in most cases the main channel to communicate information in movies, news, programs, notices in public places, etc. In many cases subtitles can be added to accompany the audio, but that is not enough, since subtitles are not as effective as the SL. The SL is ideographic (it allows the representation of common concepts with a single gesture), which makes information be received faster through sign language. It is therefore important to promote and facilitate the use of SL as an alternative language to the oral language.

A tool capable of translating text into Sign Language would be of great help to hearing impaired people, as it would offer the possibility of introducing the SL to any subtitled audio, and even of directly translating any audio through the support of voice to text translation tools.

## 1.2. Goals

The main goal of this project is to create a free tool capable of translating a natural language text written in Spanish into Spanish Sign Language (SSL) in real time. This application will allow the SSL to be embedded on any audiovisual material in a simple way. It will also support people in the SSL learning process.

The application will be based on a Service Oriented Architecture (SOA): small functionalities will be implemented as web services, easily reusable by other programmers who wish to integrate the functionalities that we will develop into their applications. These microservices will receive a text in Spanish and return the SSL translation in different formats (video and image), using Natural Language Processing (NLP).

Regarding the academic objectives, our intention is to put into practice everything learned during the Degree in Computer Engineering, as well as to expand our knowledge and skills.

### **1.3. Document structure**

In Chapter 2 hearing impairment and the different ways of communication available for hearing impaired people are discussed. Also, a deeper insight into Sign Language, specifically Spanish Sign Language is provided. This is complemented by an analysis of the tools currently available to help communicate and learn about SSL. The tools used to develop the application are presented in Chapter 3. In Chapter 4, the working methodology followed by the members of the group during the development of this project is explained in detail. A detailed description of the undertaken work is provided in Chapter 5. The results of the evaluation and testing of the application are shown in Chapter 6. The individual work done throughout the project is explained in Chapter 7. Finally, in Chapter 8, conclusions are presented and some future work are suggested.



## Capítulo 2

# Estado del Arte

En este capítulo se introduce la discapacidad auditiva y la Lengua de Signos Española (LSE), junto con algunas aplicaciones de traducción a LSE existentes y los conceptos básicos del Procesamiento del Lenguaje Natural (PLN). En la sección 2.1 de este capítulo se introducen los tipos y características más importantes de la discapacidad auditiva. En la sección 2.2 se presentan las distintas vías de comunicación alternativa que utilizan las personas con discapacidad auditiva para relacionarse. A continuación, en la sección 2.3 se da una explicación detallada de qué es la Lengua de Signos y por qué es la vía de comunicación más usada por las personas con discapacidad auditiva. Posteriormente, en la sección 2.4 se muestran las soluciones de accesibilidad digital que existen para personas con discapacidad auditiva, así como tecnologías ya desarrolladas similares a la aplicación que se va a desarrollar en este TFG. Finalmente, en la sección 2.5 se explican los conceptos más importantes del Procesamiento del Lenguaje Natural (PLN).

### 2.1. La discapacidad auditiva

La discapacidad auditiva es la dificultad que sufren algunas personas para percibir el sonido a través del oído. El colectivo de personas con déficit auditivo es muy heterogéneo, influyendo factores como la edad en la que aparece la sordera, el grado de ésta, así como factores de su entorno educativo y social. Existen distintas clasificaciones (Discapnet, 2019):

1. Según el grado de pérdida de audición:

- **Audición normal:** se perciben sonidos por encima de 20 decibelios.
- **Hipoacusia:** pérdida parcial de la audición.

- **Leve:** no se perciben sonidos inferiores a 40 decibelios.
  - **Moderada:** se presentan pérdidas entre 40 y 70 decibelios
  - **Severa:** pérdida de entre 70 y 90 decibelios. En este grado se requiere de ayudas auditivas, como prótesis o implantes. A partir de pérdidas de 75 decibelios la Seguridad Social considera al individuo persona sorda.
- **Sordera (Cofosis):** pérdida total de la audición. Se precisa de la ayuda de códigos de comunicación alternativa.
2. Según su etiología:
- **Genéticas:** factores hereditarios influyen en la pérdida de audición del individuo.
  - **Adquiridas:** influyen factores externos como golpes o exposición a ruidos fuertes.
  - **Congénitas:** la pérdida de audición está presente desde el nacimiento del individuo.
3. Según el momento de la aparición se distinguen:
- **Prelocutivas:** la sordera aparece antes de que el individuo haya aprendido el lenguaje oral. La gran mayoría de este colectivo no sabe ni leer ni escribir.
  - **Postlocutivas:** la persona es capaz de comunicarse oralmente antes de la aparición de la discapacidad. En condiciones normales conoce el lenguaje escrito.

Muchas personas con discapacidad auditiva necesitan apoyar la comunicación oral con Sistemas Aumentativos y Alternativos de Comunicación (SAAC). En la siguiente sección se profundizará en estos métodos de comunicación y se explicarán con detalle los tipos que más se utilizan.

## 2.2. Sistemas Aumentativos y Alternativos de Comunicación (SAAC)

Los Sistemas Aumentativos y Alternativos de Comunicación (SAAC) (ARASAAC, 2019) son un conjunto de recursos y formas de expresión distintas al lenguaje hablado, cuyo objetivo es facilitar la comprensión y la expresión del lenguaje a personas que tienen dificultades en la adquisición del habla y/o en la escritura. En los casos graves en los que la capacidad de expresión verbal es nula, estos sistemas se denominan sistemas alternativos.

De esta forma, las personas con este tipo de dificultades pueden expresar sus deseos, intercambiar conocimientos, opiniones e incluso expresar su propia personalidad de manera mucho más eficiente e inteligible para los demás, enriqueciendo así su campo de experiencia.

Las personas que más usan este tipo de sistemas son aquellas que presentan discapacidades motoras, las cuales carecen de un habla comprensible por los demás. También existen otros colectivos que requieren de estos sistemas, como pueden ser personas con discapacidad intelectual, cognitiva y física, así como discapacidades sensoriales, como la sordera.

Dependiendo de las necesidades de cada persona se pueden distinguir dos tipos de SAAC:

- **SAACs gestuales** (Torres, 2001): No se apoyan en ningún soporte físico (libros, dispositivos tecnológicos, etc), sino que se usa el propio cuerpo. Algunos ejemplos son los siguientes:
  - **Bimodal:** es el uso simultáneo de palabras articuladas y signos gestuales manteniendo la estructura sintáctica de la lengua oral.
  - **Oral signado:** es el uso simultáneo de palabras y signos manteniendo la estructura sintáctica de la lengua oral. La diferencia con el bimodal radica en que el bimodal signa preferentemente las palabras con contenidos semánticos, mientras que el oral signado signa de manera restrictiva todas y cada una de las palabras de la frase oral, hay paralelismo exacto entre palabras y signos.
  - **Lectura de labios:** es una técnica con la que una persona comprende lo que se le habla observando los movimientos de los labios de su interlocutor e interpretando los fonemas que éste produce.
  - **Palabra complementada:** es un sistema que posibilita la comunicación con personas sordas o con discapacidad auditiva mediante el uso simultáneo de la lectura de labios y una serie de gestos manuales que lo complementan.
  - **Dactilología:** Consiste en representar letras del alfabeto mediante formas manuales. Todas las lenguas de signos hacen uso de la dactilología en mayor o menor medida.
- **SAACs gráficos:** Se apoyan en un soporte físico que varía según las necesidades del individuo. Algunos ejemplos son:

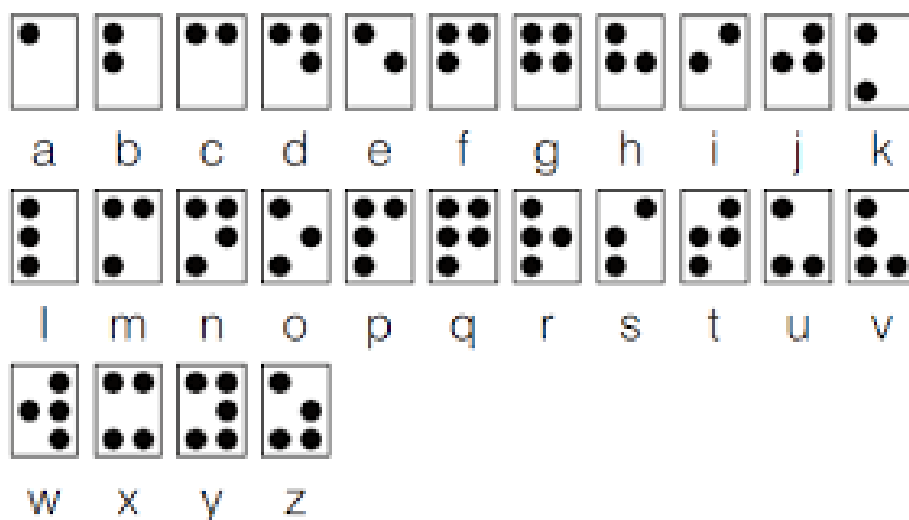


Figura 2.1: Representación del abecedario en el Sistema Braille

- **Braille:** es un sistema de lectura y escritura táctil, ya que los puntos que forman el sistema se encuentran elevados en la superficie. Es utilizado por las personas con discapacidad visual o ciegos para poder escribir y leer *textos*, libros y documentos. En la Figura 2.1 se muestra el abecedario en Braille.
- **Pictogramas:** sistema que utiliza dibujos simples o símbolos para comunicarse de forma sencilla. Los símbolos son diseñados con el fin de representar las palabras y conceptos de uso más común. Existen diversos sistemas pictográficos, como por ejemplo MIC o ARASAAC, cuyos pictogramas son los más utilizados en España. En la Figura 2.2 se muestra la representación de varios elementos comunes mediante pictogramas.

### 2.3. La Lengua de Signos

La Lengua de Signos (LS) (Muñoz y et.al., 2002) es una lengua natural gestual que sirve para ayudar a integrarse a las personas con discapacidad auditiva o dificultad en el habla, y a personas que se quieran comunicar con ellas. La Lengua de Signos es considerada como SAAC gestual sin ayuda, es decir, no necesita apoyo de ningún sistema de información (imágenes, pictogramas).

La LS es una lengua rica y compleja gramaticalmente, es decir, no es una simple representación literal de la lengua oral. Además, es muy expresiva,














 BATA	 BAÑO	 BALÓN	 BASURA	 BAILE
 BESO	 BEBÉ	 BEBER	 BICI	 BOTELLA
 BOTA	 BOLA	 BOCA	 BODA	 BOTÓN
 BOMBÓN	 BOMBA	 BORRA	 BUENO	 BURRO

Figura 2.2: Representación de distintas palabras con pictogramas

ya que permite expresar sentimientos y emociones a la hora de comunicarse mediante el énfasis en los gestos y expresiones faciales.

El desarrollo de esta lengua necesita de unas capacidades tanto cognitivas como motrices, así como de un entrenamiento específico. Este sistema puede tener una doble finalidad:

- **Como elemento de comunicación:** Para hacer posible la comunicación de manera alternativa al habla o paliar las limitaciones que provoca la pérdida auditiva y así mejorar la integración social de las personas que sufren esta discapacidad.
- **Como elemento de desarrollo intelectual:** En personas sin ningún tipo de discapacidad auditiva, la capacidad lingüística es principalmente auditiva, siendo ésta la que más influye en la educación debido a que es la vía de adquisición del lenguaje. Para las personas que sí sufren de dicha discapacidad, el uso de estos sistemas es determinante en su desarrollo intelectual, sobre todo cuando el lenguaje verbal no esté adquirido.

La LS no es una lengua universal, es decir, no existe una Lengua de Signos común para todo el mundo ni para todos los idiomas. Cada país puede contar con una o varias LS oficiales y no son únicas para cada lengua. Esto quiere decir que dos países que comparten lengua oral oficial (como España

y Argentina), tienen Lenguas de Signos totalmente diferentes. Incluso puede ocurrir que una misma LS presente diferencias dependiendo de en qué región del país se utilice.

En España hay alrededor de 1.064.000 personas mayores de seis años con algún grado de discapacidad auditiva, de las cuales el 1,25 % (13.300 personas) utilizan la Lengua de Signos para comunicarse, según los últimos datos aportados por el Instituto Nacional de Estadística (INE)<sup>1</sup>. Esto se debe a que la mayoría de las personas con esta discapacidad son capaces de comunicarse a través del lenguaje oral, ya que su grado de discapacidad se lo permite y les ayuda a integrarse con el entorno debido a que no es común saber Lengua de Signos si no sufres de dicha discapacidad, siendo únicamente alrededor de 400.000 las personas que saben comunicarse mediante dicha lengua en nuestro país. Desde el año 2007 España cuenta con dos LS reconocidas oficialmente: la Lengua de Signos Española (LSE) y la Lengua de Signos Catalana (LSC), siendo la LSE la más utilizada en nuestro país.

La LSE es una lengua normativizada, es decir, consta de unas reglas que marcan el correcto uso de la misma, tal y como se muestra en la siguiente subsección.

### 2.3.1. Reglas de la Lengua de Signos Española

A continuación, se explica el conocimiento básico de las normas de la LSE en lo referente a la gramática, fonología, sintaxis y morfología.

#### 2.3.1.1. Fonología de la Lengua de Signos Española

La fonología (Muñoz y et.al., 2002), en lo relativo a la LSE, se refiere al estudio de la estructura y organización interna de los signos.

Los signos de la LSE están formados por siete elementos esenciales:

- **Forma o configuración de la mano o manos** que intervienen en el signo. Es importante remarcar que no se utilizan los términos “mano izquierda” y “mano derecha”, ya que dependiendo de la persona, la mano dominante puede ser una u otra. Por eso se utilizan los términos “mano dominante” y “mano no dominante” para señalar con qué mano o manos signar. Para indicar qué mano es la dominante basta con levantarla y rotarla (tal y como se muestra en la Figura 2.3)
- **Orientación** de la mano o manos al signar respecto al cuerpo del individuo. Por ejemplo, signos como “*AYUDAR*”, el cual podemos observar

---

<sup>1</sup><https://www.ine.es/>



Figura 2.3: Representación de la configuración de la mano en Lengua de Signos

en la Figura 2.4, van acompañados de dirección para indicar a qué persona se refiere el emisor. En la Figura 2.5 se muestran las distintas direcciones que pueden acompañar a los gestos.

- **Lugar de articulación** del signo, es decir, el espacio en el que se realiza el movimiento. Puede ser ante el pecho, el hombro, la frente o los labios, entre otros. Por ejemplo, en la Figura 2.6, el signo referente a la palabra “visera” se realiza ante la frente.
- **Plano** en el que se realiza el signo, es decir, la distancia de realización del signo con respecto al cuerpo del individuo. Sirve para indicar el tiempo verbal de los gestos. Los gestos signados más cerca del cuerpo indican pasado y lo más alejados futuro.
- **Punto de contacto** de la mano dominante con el cuerpo del individuo al realizar el signo. Por ejemplo, al signar “oir” el punto de contacto sería la oreja, tal y como se muestra en la Figura 2.7.
- **Movimiento** de las manos al realizar un signo. Puede ser giratorio, de vaivén o quebrado, entre otros. En la Figura 2.8 podemos observar la dirección y sentido con los que se realizan algunos de estos movimientos.
- **Componente no manual** del signo, que puede ser desde expresiones faciales hasta movimientos corporales del individuo. Por ejemplo, en la Figura 2.9, la intérprete quiere expresar una cantidad abundante, acompañando al signo de una expresión facial que enfatiza una gran cantidad.



Figura 2.4: Signo “AYUDAR” en LSE del banco de imágenes ARASAAC

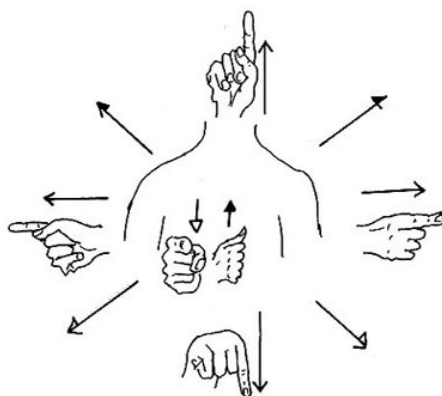


Figura 2.5: Representación de la orientación de las manos en Lengua de Signos



Figura 2.6: Signo “*VISERA*” en LSE del banco de imágenes ARASAAC



Figura 2.7: Signo “*OIR*” en LSE del banco de imágenes ARASAAC

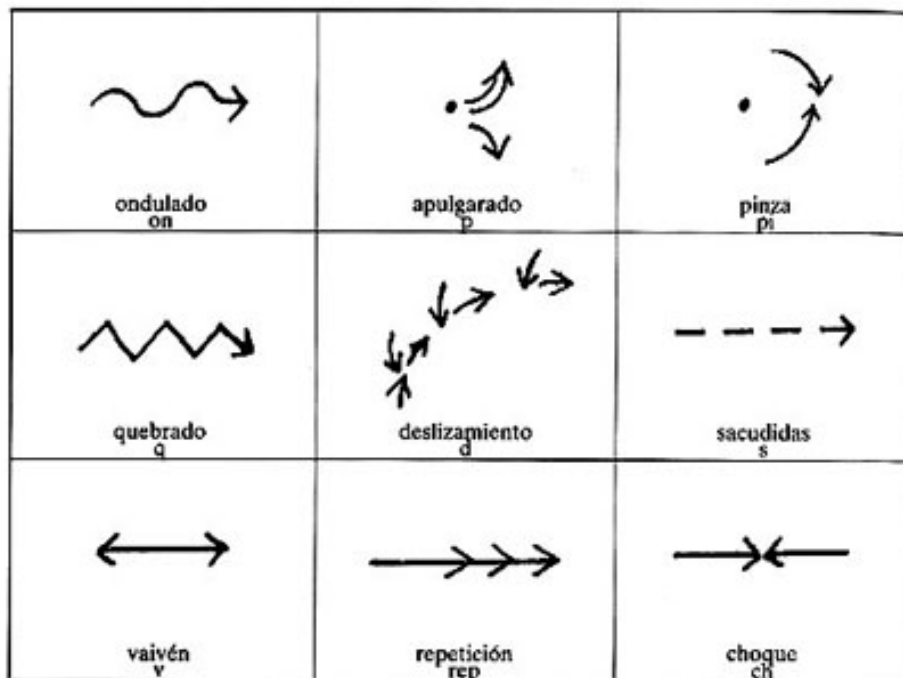


Figura 2.8: Ejemplos de tipos de movimientos realizados al signar en LSE



Figura 2.9: Signo “MUCHO” en LSE del banco de imágenes ARASAAC

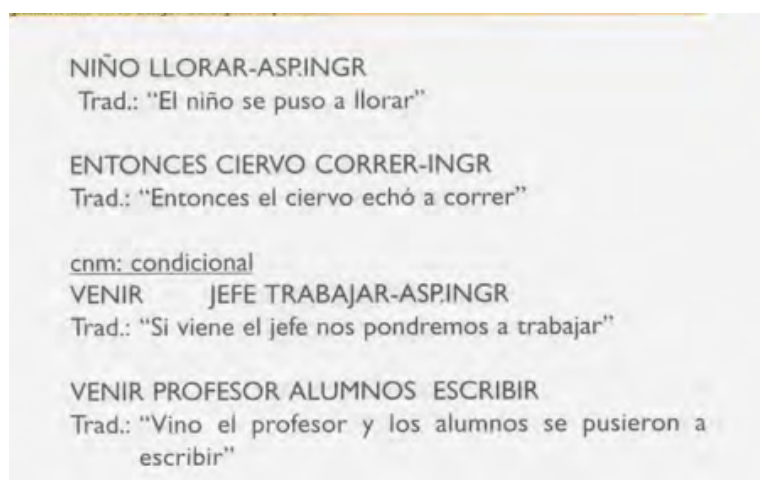


Figura 2.10: Uso de texto para la enseñanza de la estructura de la LSE

### 2.3.1.2. Sintaxis de la Lengua de Signos Española

Las oraciones en la LSE no se estructuran igual que en castellano. La estructura básica de las oraciones se compone de sujeto-objeto-verbo, aunque existen excepciones. Por ejemplo, la oración "Él come patatas" en LSE se traduciría como "ÉL PATATAS COMER". Según se van incorporando elementos a las oraciones, la estructura se va volviendo más compleja siguiendo un orden temporal, es decir, las acciones se nombran en el orden en el que suceden. Por ejemplo, la oración "Después de comer me fui a dormir", se traduciría a LSE como "YO COMER FIN DORMIR IR", indicando que primero se realizó la acción de comer y después la acción de irse a dormir.

Para aprender esta estructura en muchos libros de aprendizaje de la Lengua de Signos (Muñoz y et.al., 2002) utilizan el texto en LSE. Este texto en LSE sirve para comparar las diferencias a la hora de estructurar una misma frase en castellano y en LSE como se puede ver en la Figura 2.10. A lo largo de la memoria cuando nos referimos a texto LSE nos referimos a este tipo de estructura.

### 2.3.1.3. Morfología de la Lengua de Signos Española

En la lengua oral, los morfemas de género sirven principalmente para marcar la concordancia entre el sustantivo y el adjetivo. En la LSE esto no existe, ya que tanto los adjetivos como los sustantivos son palabras invariables y además, no existen los artículos. Los elementos inanimados no tienen género en la LSE, por lo que no es necesario señalarlos, mientras que los elementos animados sí que precisan de distinción. En caso de que se quiera hacer énfasis en el género de alguna palabra, al terminar de señalar la palabra



Figura 2.11: Signo “*HOMBRE*” del banco de imágenes ARASAAC



Figura 2.12: Signo “*MUJER*” del banco de imágenes ARASAAC

en cuestión se añaden los signos de “*HOMBRE*” (Figura 2.11) o de “*MUJER*” (Figura 2.12).

En las Figuras 2.13 y 2.14 podemos observar cómo se añaden los signos “*HOMBRE*” y “*MUJER*” para marcar el género de las palabras “*abuela*” y “*abuelo*”.

Existen algunas excepciones, como el caso de los signos para “*NIÑO*” y “*NIÑA*”, los cuales tienen su propio signo para definir el género de la palabra. Podemos verlos en la Figura 2.15 y en la Figura 2.16 respectivamente.

Por otra parte, los morfemas de número en la LSE sirven únicamente para expresar cantidad. Existen varias formas de expresar cantidad según la categoría gramatical de la palabra a la que acompañen:



Figura 2.13: Signo “ABUELO” del banco de imágenes ARASAAC



Figura 2.14: Signo “ABUELA” del banco de imágenes ARASAAC



Figura 2.15: Signo “NIÑO” del banco de imágenes ARASAAC



Figura 2.16: Signo “NIÑA” del banco de imágenes ARASAAC

■ **Sustantivos:**

- Si no queremos enfatizar el número o queremos expresar solo una unidad, se signa la palabra sin añadir nada a ella. Por ejemplo, la frase *“La puerta de la universidad es de color gris”* en LSE sería *“UNIVERSIDAD PUERTA COLOR GRIS”*.
- Se pueden añadir palabras que expresen cantidad, como cuantificadores definidos, por ejemplo *“uno”*, cuyo signo podemos observar en la Figura 2.17, o cuantificadores indefinidos, por ejemplo *“muchos”*, cuyo signo podemos observar en la Figura 2.18.
- Se puede enfatizar la cantidad de un sustantivo mediante la expresión facial de la cual se puede distinguir si es mucha o poca la cantidad a la que se quiere referir el emisor. Por ejemplo, en la Figura 2.18 aparece una intérprete realizando el signo *“MUCHOS”*. En ella podemos observar el gesto facial indicando una gran cantidad. Esta gesticulación puede sustituir al propio signo *“MUCHOS”* en una oración si se realiza mientras se signa la oración.

■ **Adjetivos:** no varían en cuanto a número.

- **Verbos:** el plural se realiza principalmente mediante la repetición del verbo y recae sobre el objeto. Por ejemplo, en la oración *“Yo viajo a muchos sitios”* se repetiría el verbo viajar: *“YO VIAJAR VIAJAR SITIO”*.



Figura 2.17: Signo “*UNO*” del banco de imágenes ARASAAC



Figura 2.18: Signo “*MUCHOS*” del banco de imágenes ARASAAC

#### 2.3.1.4. Verbos de la Lengua de Signos Española

El verbo en la LSE (Fernández, 2008) presenta unos rasgos gramaticales propios que es necesario tener en cuenta. Se distinguen tres clases básicas de verbos en LSE:

- **Verbos planos:** Los verbos planos son aquellos cuyo signo no se modifica para marcar información complementaria, como por ejemplo el género, el número o a quién va dirigida la acción. Para añadir este tipo de información se añaden palabras independientes, como cuantificadores (uno, dos, etc), pronombres personales (yo, él, etc). Por ejemplo, *Querer, Comer, Pensar, Conducir, etc.* Podemos observar en la Figura 2.19 como el signo de la palabra “*Querer*” no aporta ningún tipo de información adicional por sí mismo.
- **Verbos de concordancia:** Los verbos de concordancia son aquellos que en la realización del propio signo se añade información adicional, como por ejemplo, a quién va dirigida la acción mediante la articulación del signo. Esto podemos observarlo en la Figura 2.4, que muestra el signo de la palabra “*ayudar*”, el cual termina señalando en el espacio a la persona a la que se está ayudando. Algunos ejemplos de verbos de concordancia son: *Avisar, Ayudar, Contar, Cuidar, Responder, etc.*
- **Verbos espaciales:** Los verbos espaciales son aquellos que no necesitan un signo específico para expresar su significado, ya que indican la situación en el espacio de algo o alguien a través de la dirección y la velocidad con la que se realiza el signo. Al igual que en los verbos planos, se puede añadir información adicional como género o número a través de palabras independientes, como cuantificadores. Por ejemplo, “*Hay un ratón ahí*” se traduce como “*RATÓN (Señalando el sitio)*”. Esto podemos observarlo en la Figura 2.20, en la que se muestra el signo de la palabra “*señalar*”.

#### 2.3.2. Bancos de videos e imágenes LSE

A continuación, se muestran diferentes bancos de vídeos e imágenes existentes de la Lengua de Signos Española.

##### 2.3.2.1. ARASAAC

El Portal Aragonés de la Comunicación Aumentativa y Alternativa (ARASAAC)<sup>2</sup>, es un proyecto desarrollado en el año 2007 por el gobierno de Aragón. Tiene como objetivo la creación de un sistema pictográfico de comunicación y un conjunto de herramientas de libre distribución. Estos recursos

---

<sup>2</sup><http://www.arasaac.org/>



Figura 2.19: Signo “*QUERER*” en LSE del banco de imágenes ARASAAC



Figura 2.20: Signo “*SEÑALAR*” en LSE del banco de imágenes ARASAAC



Figura 2.21: Signo “HOLA” en LSE del banco de imágenes ARASAAC

facilitan la accesibilidad de carácter comunicativo y cognitivo en diversos ámbitos de la vida a todas las personas que lo puedan requerir. ARASAAC proporciona un catálogo con más de 8.000 pictogramas en color, en blanco y negro, fotografías, así como un banco de vídeos e imágenes a color de signos de la Lengua de Signos Española. En la Figura 2.21 se muestra el signo “HOLA” en LSE del banco de imágenes de LSE de ARASAAC. Este contenido está disponible como contenido descargable con licencia Creative Commons, es decir, se puede usar libremente sin ánimo de lucro.

ARASAAC ofrece una API<sup>3</sup> a través de la cual se pueden obtener sus pictogramas y los materiales generados con sus pictogramas a través de llamadas a los métodos de dicha API. Para obtener los recursos correspondientes a la LSE es necesario descargarlos a través del apartado de descargas de su página web<sup>4</sup>.

### 2.3.2.2. Banco de imágenes Fundacion CNSE

La Confederación Estatal de Personas Sordas (CNSE)<sup>5</sup> es una organización que atiende los intereses de las personas sordas y sus familias en España. Es la primera entidad asociativa de la discapacidad de nuestro país, y desde su creación se ha ocupado de incentivar el desarrollo y la participación social de dicho colectivo. En ella se integran 17 federaciones de personas sordas, una por cada comunidad autónoma, que, a su vez, mantienen afiliadas a más de 120 asociaciones provinciales y locales de todo el Estado. No obstante, la CNSE atiende cualquier necesidad relacionada con el colectivo de personas sordas, estén o no afiliadas a su movimiento asociativo.

<sup>3</sup><https://beta.arasaac.org/developers/api>

<sup>4</sup><http://www.arasaac.org/descargas.php>

<sup>5</sup><http://www.fundacioncnse.org/>



Figura 2.22: Recursos LSE de la Fundación CNSE

En 1998, la CNSE constituye la Fundación CNSE para la Supresión de las Barreras de Comunicación. Se trata de una organización estatal sin ánimo de lucro, desde la que se impulsa la investigación y el estudio de la Lengua de Signos Española y se trabaja por mejorar la accesibilidad de las personas sordas en todos los ámbitos y se promueve el desarrollo de proyectos que mejoren la calidad de vida de las personas sordas y de sus familias.

Esta fundación cuenta con un banco de imágenes<sup>6</sup> (ver Figura 2.22) y signos de la LSE, accesible a través de su página web. En esta página no existe la opción de descargar todas las imágenes de una vez, sino que es necesario hacerlo manualmente una a una, y no cuenta con un banco de videos de signos.

### 2.3.2.3. SpreadTheSign

SpreadTheSign<sup>7</sup> es un diccionario online administrado por el Centro Europeo de Lenguas de Signos, que cuenta con más de 400.000 signos en vídeo (ver Figura 2.23) de diferentes Lenguas de Signos, como la española, estadounidense o alemana, entre muchas otras, así como de un alfabeto dactilológico para cada una de ellas y frases previamente almacenadas. Es una herramienta de autoaprendizaje gratuita, accesible a través de su página web y desde

<sup>6</sup><http://www.fundacioncnse.org/educa/bancolse/>

<sup>7</sup><https://www.spreadthesign.com/es.es/search/>



Figura 2.23: Imagen de la página web SpreadTheSign

su aplicación para Android<sup>8</sup> e IOS<sup>9</sup>, aunque ninguna de estas aplicaciones permite la descarga de contenido.

## 2.4. Aplicaciones de Traducción de Texto a LSE

La variedad de recursos y aplicaciones orientadas a la integración social de las personas con discapacidad auditiva ha ido aumentando con el paso del tiempo. En esta sección se presentan aplicaciones con un objetivo similar al de nuestro TFG, cuyo propósito es ayudar en la integración de las personas que sufren de discapacidad auditiva.

### 2.4.1. Text2Sign

Text2Sign<sup>10</sup> es una aplicación tanto web como móvil gratuita que permite enviar un texto en formato PDF, imagen o Word a un equipo de intérpretes que se encarga de traducirlo a la LSE. Una vez realizada la traducción se puede descargar a través de la plataforma entre un periodo de 24 y 72 horas, dependiendo de la longitud del texto, ya que no es un servicio instantáneo y está limitado a una o dos páginas.

<sup>8</sup>[https://play.google.com/store/apps/details?id=com.spreadthesign.androidapp\\_paid&hl=es](https://play.google.com/store/apps/details?id=com.spreadthesign.androidapp_paid&hl=es)

<sup>9</sup><https://apps.apple.com/ni/app/spreadthesign/id438811366>

<sup>10</sup><http://text2sign.es/>

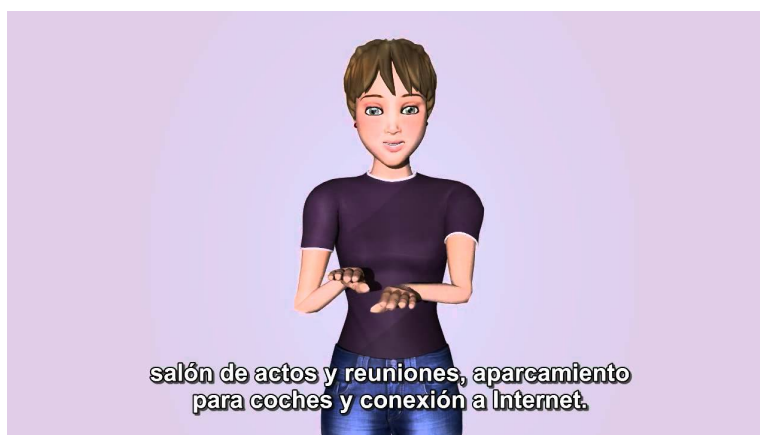


Figura 2.24: Maya, avatar usado por TextoSign.

### 2.4.2. TextoSign

TextoSign<sup>11</sup> es una herramienta software que funciona en páginas web, asistentes virtuales y dispositivos móviles que traduce en tiempo real un texto a la Lengua de Signos Española mediante un intérprete visual llamado Maya (Figura 2.24). Se trata de herramienta de pago y es utilizada hoy en día por La Caixa, Adif, Inturjoven y otras entidades.

### 2.4.3. StorySign

StorySign<sup>12</sup> es una aplicación para Android gratuita desarrollada por Huawei, que traduce en tiempo real una selección de cuentos infantiles a diferentes Lenguas de Signos como la española, italiana, francesa o inglesa. Para utilizarla es necesario abrir la aplicación y sostener el móvil encima del libro físico (especialmente diseñado para esta aplicación). Un avatar llamado Star interpreta el cuento en Lengua de Signos mientras la aplicación resalta cada palabra que Star interpreta, como podemos ver en la Figura 2.25.

### 2.4.4. TeCuento

TeCuento<sup>13</sup> es una aplicación Android gratuita que contiene diversos cuentos que se reproducen en vídeo con subtítulos y LSE, como podemos apreciar en la Figura 2.26. No se traduce en el momento, son vídeos previamente grabados por personas físicas, dando también la opción de poder escribir tu propio cuento y que lo traduzcan a LSE.

---

<sup>11</sup><http://textosign.es/>

<sup>12</sup><https://play.google.com/>

<sup>13</sup><https://play.google.com/>

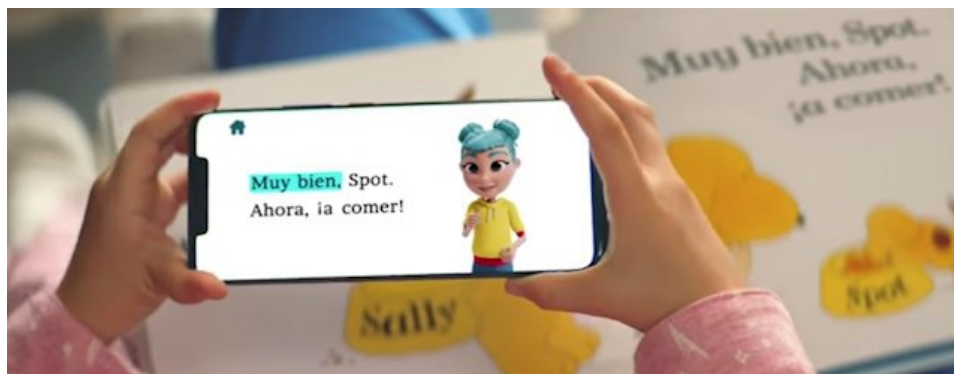


Figura 2.25: Imagen de la aplicación StorySign.



Figura 2.26: Imagen de la aplicación TeCuento.

### 2.4.5. Conclusiones

Como se ha podido comprobar, existe un abanico amplio de aplicaciones que son capaces de traducir texto a LSE. Estas herramientas, en su mayoría, cumplen el propósito de facilitar el uso de la Lengua de Signos a las personas que quieran comunicarse con ella. No obstante, presentan algunas limitaciones como licencias de pago, tiempo de respuesta lento o disponibilidad en un solo tipo de dispositivo.

Nuestro proyecto Text2LSE pretende solventar muchas de estas limitaciones proporcionando una aplicación gratuita que sea accesible desde cualquier dispositivo, ya sea web o móvil. El objetivo de Text2LSE será ofrecer una traducción al instante de cualquier frase o palabra en castellano a LSE y la posibilidad de integrar nuestra herramienta mediante servicios web en aplicaciones que requieran su uso.

## 2.5. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) (Juan Ignacio Bagnato, 2018) es un campo dentro de la inteligencia artificial y la lingüística aplicada que analiza el lenguaje humano, lo interpreta y le dan significado para que pueda ser utilizado por una máquina de manera práctica.

La complejidad del lenguaje hace que su tratamiento automático no sea nada fácil, por ello el PLN es un campo en pleno desarrollo, que necesita seguir evolucionando y superando los retos planteados por el propio lenguaje natural. Durante los últimos años ha ido creciendo de forma exponencial, debido a la capacidad computacional de la que disponemos hoy en día y a los avances realizados en la algoritmia, donde algunas de sus aplicaciones actuales son: la traducción automática, sistemas de recuperación de información, elaboración automática de resúmenes, entre otros.

Gramaticalmente, la Lengua de Signos Española es muy diferente al castellano (ver sección 2.3). Por ejemplo la frase “La niña es alta” en LSE se traduce mediante los signos “NIÑA ALTA”. Ambas frases tienen el mismo significado pero distinta estructura gramatical, por lo que para hacer la traducción hay que obtener información sobre la frase a traducir, como lemas, categorías gramaticales, morfemas, etc. De ahí la importancia del PLN en nuestro proyecto, ya que es esencial para poder obtener todo este tipo de información del Lenguaje.

### 2.5.1. Niveles del Procesamiento del Lenguaje Natural

Para poder obtener el significado de una frase será necesario hacer un análisis profundo de ella. El PLN suele hacer un análisis en 4 niveles (Eduardo Sosa, 1997):

- **Análisis Morfológico:** Consiste en identificar unidades léxicas o tokens (unidades del lenguaje con significación propia como son las palabras o signos de puntuación) y tipos de palabras (sustantivos, verbos, etc.). En el análisis morfológico se determina la forma, clase o categoría gramatical de cada palabra (etiquetado gramatical o POS (Part-of-speech) tagging).
- **Análisis Sintáctico:** Determina las relaciones de dependencia estructural entre las palabras de la oración. En este nivel se lleva a cabo la desambiguación, que es el proceso de identificar con qué sentido una palabra está usada en los términos de una oración, cuando la palabra en cuestión tiene polisemia, es decir, pluralidad de significados. Por ejemplo en la frase “*Deja el sobre que sobre sobre la mesa de la oficina*” la palabra “*sobre*” es polisémica, ya que actúa como sustantivo, verbo y preposición.

En la Figura 2.27 se muestra el análisis sintáctico de la frase “Mi padre es abogado”.

<b>Mi</b>	<b>padre</b>	<b>es</b>	<b>abogado</b>
<i>Det.</i>	<i>Núcleo</i>	<i>Verbo</i>	<i>S. Nominal</i>
<i>S. Nominal</i>		<i>Núcleo</i>	<i>Atributo</i>
		<i>S. Verbal</i>	
<i>Sujeto</i>		<i>Predicado</i>	

Figura 2.27: Análisis sintáctico frase “Mi padre es abogado”

- **Análisis Semántico:** El análisis semántico trata del significado de las palabras y de cómo los significados se unen para dar sentido a una oración. Es posible distinguir entre un análisis de significado independiente de contexto y un análisis de significado dependiente de contexto. El primero tratado por la semántica, se refiere al estudio del significado de las palabras de forma individual, mientras que el segundo estudia cómo estas palabras se conjugan para formar significados complejos y, para ello, se realiza el análisis pragmático.
- **Análisis Pragmático:** Es el encargado de analizar el significado de la frase en función del contexto en el que se desarrolla. En muchas

ocasiones las oraciones y las palabras no se pueden tomar su significado de forma literal, ya que dependen en gran medida del contexto en el que se desarrolla la acción.

En las aplicaciones que requieren de PLN no siempre se aplican todos los niveles de análisis, depende del objetivo de la aplicación. En nuestro caso solo serán necesarios los dos primeros niveles de análisis (morfológico y sintáctico).

### 2.5.2. Herramientas PLN

Existe una amplia variedad de herramientas informáticas para el PLN en diversos idiomas. A continuación presentamos algunas de ellas:

- **Spacy**<sup>14</sup>: es una librería de código abierto en Python. Actualmente es considerada una de las mejores herramientas para el PLN que destaca por su rapidez y precisión a la hora de realizar análisis sintácticos comparado con otras librerías, como se puede comprobar en la imagen 2.28

SYSTEM	YEAR	LANGUAGE	ACCURACY	SPEED (WPS)
spaCy v2.x	2017	Python / Cython	92.6	n/a <sup>Ⓢ</sup>
spaCy v1.x	2015	Python / Cython	91.8	13,963
ClearNLP	2015	Java	91.7	10,271
CoreNLP	2015	Java	89.6	8,602
MATE	2015	Java	92.5	550
Turbo	2015	C++	92.4	349

Figura 2.28: Benchmark de diferentes procesadores de lenguaje

Spacy actualmente permite hacer un procesamiento completo del lenguaje, permitiendo realizar los distintos análisis del PLN explicados en 2.5.1. Algunas funciones que dispone Spacy para facilitar el desarrollo del PLN son:

- **Tokenización**: Divide una oración en *tokens*, donde cada token representa cada una de las palabras que componen dicha oración. Además permite eliminar caracteres especiales como puntos, comas y signos de interrogación.
- **Part of Speech**: Es el etiquetado gramatical, a cada palabra le asigna su categoría gramatical.

<sup>14</sup><https://spacy.io/>

- **Lematización:** permite obtener el lema correspondiente de una palabra. Por ejemplo, el lema *gat-* representaría a las palabras *gatas* o *gatita*.
- **Natural Language ToolKit (NLTK)**<sup>15</sup>: es una librería dedicada al análisis del lenguaje natural en Python. En la actualidad incluye utilidades para más de 10 idiomas y fue creada por académicos e investigadores como una herramienta para crear funciones complejas de PLN. Esta biblioteca es muy útil cuando se quiere construir un modelo desde cero, ya que proporciona acceso a muchos algoritmos y facilita el desarrollo de algoritmos propios permitiendo realizar un procesamiento del lenguaje más adaptado a un proyecto en concreto. Con ella se realiza tanto análisis morfológico, como sintáctico, semántico y pragmático.
- **FreeLing**<sup>16</sup>: es una herramienta PLN de código abierto bajo el lenguaje de programación C++ que ha sido desarrollada por la Universidad Politécnica de Cataluña. La biblioteca permite realizar análisis morfológico, sintáctico, semántico y pragmático para una gran variedad de idiomas. Los desarrolladores pueden usar los recursos por defecto de FreeLing, ampliarlos, adaptarlos a dominios particulares e incluso desarrollar otros nuevos para idiomas específicos o necesidades particulares.

Una vez analizadas en detalle cada una de las herramientas presentadas en esta sección, se llegó a la conclusión de que Spacy es la herramienta más adecuada para nuestro TFG, ya que proporciona un análisis rápido y bastante preciso en español. Esto es de gran importancia en este proyecto, ya que necesitamos traducir textos en castellano en tiempo real, teniendo que hacer este procesamiento del lenguaje lo más rápido y preciso posible.

Spacy es muy fácil de implementar, ya que utiliza un lenguaje de programación simple, como es Python. Además también es fácil de aprender gracias a su manual de uso, ya que este cuenta con ejemplos claros y útiles para hacer un análisis, sin necesidad de ser expertos en el uso la herramienta.

---

<sup>15</sup><https://www.nltk.org/>

<sup>16</sup><http://nlp.lsi.upc.edu/freeling/index.php/node/1>

## Capítulo 3

# Tecnologías utilizadas

En este capítulo se habla de las tecnologías utilizadas a lo largo del desarrollo de nuestra aplicación, dando una breve descripción de cada una y comentando los motivos de su utilización con respecto a otras tecnologías.

### 3.1. Servicios Web

Según el W3C (World Wide Consortium)<sup>1</sup> un servicio web es un sistema software diseñado para soportar interacciones entre sistemas a través de la red. Proporcionan una forma estándar de interoperar entre aplicaciones software que se pueden ejecutar en diferentes plataformas, soportando aplicaciones desarrolladas en distintos lenguajes de programación.

Muchos de los servicios web están basados en la arquitectura SOA (Service Oriented Architecture). Es un estilo arquitectónico para la construcción de aplicaciones software en base a servicios disponibles, los cuales interactúan entre sí y se pueden combinar obteniendo nuevos servicios con más funcionalidades.

Los dos tipos de servicios web más utilizados en la actualidad son: Servicios Web SOAP y Servicios Web REST.

#### 3.1.1. Servicios Web SOAP

Los Servicios Web SOAP (IBM Corporation, 2014) son un tipo de servicios web cuyo propósito es que las aplicaciones implementadas en diferentes plataformas y lenguajes de programación sean capaces de intercambiar datos de una forma más sencilla.

---

<sup>1</sup><https://www.w3.org/>

Está basado en el protocolo SOAP (Simple Object Access Protocol) basado en el lenguaje XML, el cual permite el intercambio de información entre aplicaciones. El mensaje se codifica como un documento XML, que consta de un elemento *<Envelope>* que hace referencia a la raíz de cada mensaje, y este a su vez incorpora una cabecera opcional *<Header>* y un cuerpo *<Body>* obligatorio. La primera es utilizada para pasar información de la aplicación y la segunda contiene la información dirigida al destinatario final del mensaje. Dentro del *<Body>* está el subelemento *<Fault>*, que sirve para la notificación de errores.

Funciona por lo general con el protocolo HTTP, sin embargo, no está limitado solo a este protocolo, si no que puede ser enviado por otros tipos de protocolos diferentes, como TCP o POP3.

Para poder acceder al servicio web se necesita saber donde está ubicado y cuales son las funcionalidades del servicio. Para ello, se utiliza el lenguaje de descripción WSDL (Web Services Description Language), el cual está basado en XML y permite especificar a los clientes qué llamadas se pueden hacer a un servidor SOAP y qué tipo de respuesta esperar. A pesar de tener el archivo WSDL, es necesario saber dónde está toda esta información. Para ello se dispone del UDDI (Universal Description, Discovery and Integration Directory), que es un estándar basado en XML que actúa como repositorio donde se puede acudir a realizar búsquedas de Servicios Web específicos.

### 3.1.2. Servicios Web REST

Un Servicio Web REST (Manuel Rosa Moncayo, 2018) es una interfaz para conectar varios servicios web basados en el protocolo HTTP que define una gran cantidad de métodos, como son GET, POST, PUT y DELETE entre otros. Estos métodos pueden ser usados en diferentes circunstancias devolviendo los datos en distintos formatos como XML y JSON.

Rest tiene como peculiaridad que es un servicio sin estado, es decir, que por cada petición que recibe el servidor, los datos se pierden, ya que no es necesario mantener sesiones. Esto permite un uso más eficiente de la memoria y una mayor escalabilidad, permitiendo separar el cliente del servidor, facilitando así que las distintas partes de un proyecto se puedan realizar de manera más independiente.

Los objetos son manipulados a través de una URI (Uniform Resource Identifier), haciendo de identificador único de cada recurso del sistema REST. Esto junto con los métodos del protocolo HTTP hace posible la transferen-

cia de datos en el sistema aplicando operaciones concretas sobre un recurso determinado.

Para el desarrollo de este proyecto, se ha utilizado un servicio web REST, debido a su facilidad de uso y configuración.

### 3.1.3. Ventajas de Servicios Web

Las principales ventajas de los servicios web son<sup>2</sup>:

- Permiten una mayor operabilidad entre las aplicaciones software sin importar sus propiedades ni la plataforma en la que estén instalados.
- Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten y facilitan que software y servicios de diferentes compañías se puedan combinar formando un servicio funcional.

### 3.1.4. Desventajas de Servicios Web

Las principales desventajas de los servicios web son<sup>3</sup>:

- Las transacciones en este tipo de servicios están mucho menos desarrolladas que otros estándares abiertos.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida.
- Al estar muy ligados con HTTP, las medidas de seguridad que tratan de bloquear la comunicación entre programas se pueden burlar fácilmente.
- Existe poca información de servicios web para algunos lenguajes de programación.

## 3.2. Flask

Para el desarrollo del proyecto se ha decidido utilizar un servidor Flask para implementar los microservicios. Flask (Openwebinars, 2019) es un “micro” framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web y APIs. La palabra micro hace referencia a que Flask únicamente trae por defecto las herramientas necesarias para crear una aplicación web básica, aunque si se necesitan añadir nuevas funcionalidades hay

<sup>2</sup><https://sites.google.com/site/preyctodetics/home/servicios-web>

<sup>3</sup><https://sistemas3.wordpress.com/2007/06/14/web-services/>

un conjunto muy grande de extensiones (plugins) que se pueden instalar fácilmente. Por ello, Flask es muy recomendable para el desarrollo de aplicaciones que no requieran muchas extensiones o que se necesiten implementar de una forma ágil y rápida. También es muy recomendable para implementar microservicios.

Algunas de las características de Flask por las que decidimos desarrollar nuestros microservicios con este framework son las siguientes:

- Rapidez y facilidad en la instalación y configuración, a diferencia de otros frameworks como Django, que son más complejos y difíciles de utilizar.
- Es compatible con Python. Nuestros microservicios están implementados en dicho lenguaje.
- Incluye un servidor web de desarrollo. No se necesita una infraestructura con un servidor web para probar las aplicaciones, sino que de una manera sencilla se puede levantar un servidor web para ir viendo los resultados que se van obteniendo.
- Cuenta con depurador. Si tenemos algún error en el código que se está desarrollando, se puede depurar ese error y ver los valores de las variables.
- Flask es Open Source y está amparado bajo una licencia BSD (Wikis FDI, 2017), que es la licencia utilizada para los sistemas operativos BSD (Berkeley Software Distribution), y tiene menos restricciones en comparación con otras licencias como la GPL, estando muy cercana al dominio público.

### 3.3. Nginx

Nginx <sup>4</sup> es un servidor web de alto rendimiento, capaz de trabajar junto con diversas tecnologías de desarrollo y lenguajes. La asincronía es una de sus características fundamentales, junto con su rapidez, debido a que es un servidor web ligero. A día de hoy, además de sus capacidades de servidor HTTP, NGINX también puede funcionar como un servidor proxy para correo electrónico (IMAP, POP3 y SMTP) y un proxy inverso y equilibrador de carga para servidores (HTTP, TCP y UDP).

Las ventajas de Nginx y el porqué de su uso en este TFG son:

---

<sup>4</sup>Web oficial de Nginx <https://www.nginx.com/>

- Nginx es un software multiplataforma que se puede usar en la gran mayoría de los sistemas operativos, tanto en sistemas basados en Unix como en Windows.
- Consume menos recursos que la mayoría de servicios que hacen su misma función ya que hace un uso de memoria de forma estática y se mantiene bastante estable independientemente del volumen de tráfico.
- Proporciona un alto rendimiento sobre todo en casos de mucho tráfico.
- Puede ser usado como Proxy inverso cacheando el contenido de nuestros sitios web.

Se ha utilizado un servidor Nginx para alojar una aplicación web cuya finalidad es mostrar la funcionalidad de los diferentes microservicios implementados en el proyecto. Además, se ha configurado Nginx como servidor proxy inverso, redireccionando las peticiones según correspondan a la página web alojada en el propio Nginx o a los microservicios desarrollados alojados en el servidor Flask.

### 3.4. Google Colaboratory

Google Colaboratory<sup>5</sup> es una herramienta gratuita de Google que permite programar en Python y ejecutar el código en tiempo real. El usuario final no necesita instalar nada en su propio sistema, ya que se pueden importar las librerías necesarias y toda la carga del procesamiento se localiza en los servidores de Google.

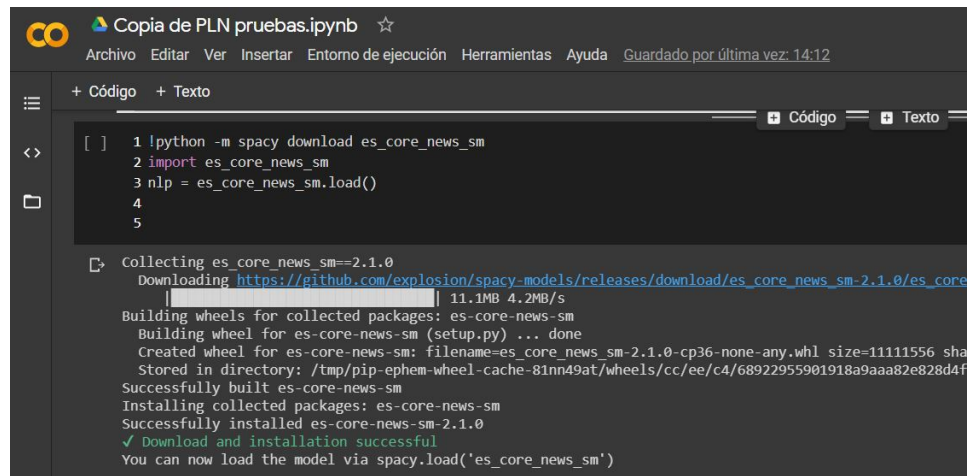
Con esta herramienta se crean documentos llamados “cuadernos de Colaboratory”, los cuales contienen celdas independientes en las que se puede introducir código ejecutable, texto, imágenes, comandos de la shell de Linux, código HTML, LaTeX, etc. Todos estos cuadernos son almacenados en la cuenta de Google Drive del usuario, haciendo que compartirlo con otras personas sea muy rápido y sencillo. Podemos ver un ejemplo de un cuaderno en la Figura 3.1.

Una de las opciones más llamativas de Colaboratory es la opción de poder procesar el código que necesite gran potencia a través de un acelerador, en este caso consiste en una GPU proporcionada por Google.

Para este proyecto se ha utilizado esta herramienta para realizar pruebas con PLN utilizando Spacy y NLTK de manera más rápida y sencilla, teniendo acceso los tres miembros del equipo en tiempo real a dichas pruebas. En este

---

<sup>5</sup><https://colab.research.google.com/notebooks/intro.ipynb>



```
Copia de PLN pruebas.ipynb ☆
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Guardado por última vez: 14:12

+ Código + Texto

[ ] 1 !python -m spacy download es_core_news_sm
    2 import es_core_news_sm
    3 nlp = es_core_news_sm.load()
    4
    5

Collecting es_core_news_sm==2.1.0
  Downloading https://pithub.com/explosion/spacy-models/releases/download/es_core_news_sm-2.1.0/es_core_news_sm-2.1.0-py3-none-any.whl (11.1MB) 4.2MB/s
Building wheels for collected packages: es-core-news-sm
  Building wheel for es-core-news-sm (setup.py) ... done
  Created wheel for es-core-news-sm: filename=es_core_news_sm-2.1.0-cp36-none-any.whl size=11111556 sha256=...
  Stored in directory: /tmp/pip-ephem-wheel-cache-81nn49at/wheels/cc/ee/c4/68922955901918a9aaa82e828d4f...
Successfully built es-core-news-sm
Installing collected packages: es-core-news-sm
Successfully installed es-core-news-sm-2.1.0
✓ Download and installation successful
You can now load the model via spacy.load('es_core_news_sm')
```

Figura 3.1: Vista de un cuaderno de Google Colaboratory

caso no ha sido necesario la utilización de la GPU ya que la ejecución del código generado no ha sido tan potente como para necesitarla. Una vez que las pruebas eran satisfactorias, se comenzaba el traspaso de código a la API del proyecto.

## Capítulo 4

# Metodologías utilizadas

En este capítulo se habla de la metodología de trabajo seguida por el equipo de este TFG durante el desarrollo del mismo, junto con algunas de las herramientas utilizadas para organizar y estructurar las tareas a realizar y mantener el código de la aplicación accesible y con sus correspondientes copias de seguridad.

### 4.1. Metodologías de gestión del proyecto

Al comienzo de este TFG, los integrantes del proyecto desconocíamos tanto su alcance como su viabilidad, por lo que necesitábamos implantar una metodología que nos permitiera adaptar nuestro trabajo a necesidades que fueran surgiendo a medida que íbamos avanzando. Debido a esto, decidimos seguir una metodología ágil.

Las metodologías ágiles (Iebschool, 2019) son aquellas que permiten adaptar la forma de trabajo a las condiciones particulares de cada proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Nuestros tutores han tomado el papel de Product Owner y han estado involucrados de manera muy activa a lo largo de todo el proyecto. Se les iba informando de los logros y progresos del mismo para optimizar el resultado final, obteniendo en todo momento una visión completa de su estado. La continua interacción entre los alumnos y los tutores tenía como objetivo asegurar que el resultado final era exactamente lo que se buscaba y necesitaba. Además, de este modo se detectaban de forma rápida tanto errores como problemas que pudieran aparecer a lo largo del proyecto, por lo que es posible dar respuesta a todos aquellos contratiempos que puedan darse desde el inicio. Todo esto fue posible gracias a que estábamos en permanente contacto a través del correo electrónico para poder resolver dudas puntuales,

y a las constantes reuniones que se realizaron tanto entre los integrantes del grupo como entre los alumnos y los tutores. Estas reuniones se explican con más en detalle en la sección 4.3.

A lo largo de todo el TFG se fueron realizando entregas parciales del proyecto. De este modo, se entregaba en el menor intervalo de tiempo posible una versión funcional de la aplicación y de la memoria, para que los tutores pudieran revisar el trabajo realizado y comprobar que el trabajo iba por buen camino.

La herramienta elegida para gestionar el trabajo realizado fue el conocido controlador de versiones Git, del que se habla con detalle en la sección 4.4

En nuestro proyecto seguimos algunas de las reglas de la metodología Kanban. Kanban (Iebschool, 2013) es una metodología ágil cuyo objetivo es gestionar de manera visual cómo se van completando las tareas. Kanban es una palabra japonesa que significa “tarjetas visuales”, donde Kan es “visual”, y Ban corresponde a “tarjeta”. En Kanban existen tres reglas que ayudan a obtener el máximo rendimiento del flujo de trabajo (Kanbantool, 2019):

- **Visualizar el flujo de trabajo:** Kanban destaca por ser una técnica de gestión de tareas muy visual, que permite ver a golpe de vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva. Esta metodología se basa en la implementación de un tablero que hará uso de una serie de tarjetas, que describirán las tareas, y que constará de varias columnas, una por cada estado de dichas tareas. La elección de los estados depende de las necesidades del proyecto, aunque los tres estados más comunes son “To Do”, “Doing” y “Done”. Según vayan surgiendo nuevas tareas, los miembros del equipo colocarán las tarjetas en la columna que corresponda, y las irán moviendo de columna a medida que el estado de la tarea avance. La visualización de todas las tareas contribuye a que todos los miembros del equipo se mantengan al corriente del trabajo. Todos los miembros del equipo trabajan en el mismo tablero y colaboran en tiempo real. Esta regla permite a todos los miembros conocer el estado del proyecto en cualquier momento, y facilita tanto el reparto de tareas como que los compromisos sean acordados y aceptados por todos.
- **Limitar la cantidad de Trabajo en Proceso:** WIP (Kanbanize, 2020), o “Work In Progress”, hace referencia al número de tareas máximo permitido en cada columna del tablero. Este límite es importante para prevenir “cuellos de botella” o incluso bloqueos.

- **Medir el flujo de tareas:** para medir el flujo de tareas, en Kanban se utilizan los términos “tiempo de entrega”, o “leadtime”, y “tiempo de ciclo”, o “cycletime”.
  - **Leadtime:** es el periodo de tiempo que transcurre entre la aparición de una nueva tarea y su finalización.
  - **Cycletime:** es el periodo de tiempo que transcurre entre que alguien del equipo comienza a trabajar en una tarea hasta que se completa.

La herramienta utilizada para implementar el tablero Kanban ha sido Trello, explicada con detalle en la sección 4.2. En la sección 4.2 también se explica en profundidad la forma en la que el equipo ha aplicado las tres reglas de Kanban.

## 4.2. Trello

A lo largo de todo el proyecto se ha utilizado la herramienta Trello<sup>1</sup> para implementar el tablero Kanban, y así visualizar las tareas y su estado de manera clara y en tiempo real. Trello permite crear tableros con distintos estados, a los cuales se les pueden añadir tareas. Esta herramienta permite ver de manera sencilla el estado del proyecto, facilitando así la organización de las tareas a realizar.

Durante nuestro TFG se ha utilizado un único tablero, al que se le ha llamado “Text2LSE”, en el que se han englobado todas las tareas referentes a la investigación, desarrollo y pruebas del proyecto. Se ha diferenciado el tipo de tarea a través de los colores de cada tarjeta:

- **Azul:** Tareas de desarrollo del código de la aplicación web.
- **Rojo:** Tareas de desarrollo del código de los servicios web.
- **Verde:** Tareas relacionadas con la memoria.

Para una mayor organización, el tablero se ha estructurado en cuatro estados:

- **Lista de Tareas:** Lista de tareas a realizar.
- **En proceso:** Tareas en proceso de desarrollo. Estas tareas están asignadas a un miembro del equipo. Se ha establecido para esta columna un WIP de una tarea por persona, con el fin de evitar bloqueos y “cuellos de botella”.

---

<sup>1</sup><https://trello.com/>

- **En revisión:** Tareas desarrolladas, pero pendientes de realizar pruebas para comprobar su correcto funcionamiento. Respecto a las tareas correspondientes al código, todos los miembros del equipo realizan pruebas unitarias para comprobar que el nuevo desarrollo funcione correctamente. Respecto a las tareas de memoria, cada nueva aportación de un integrante del grupo es revisada por el resto de los compañeros.
- **Hecho:** Tareas completadas y comprobadas.

### 4.3. Reuniones

Para una buena organización, desde el principio el equipo de trabajo se ha reunido una vez a la semana para asignar las diferentes tareas, comentar el trabajo realizado y poner en común todos los conocimientos adquiridos por cada uno. A lo largo de la semana también había reuniones a través de Skype para comentar y resolver las posibles dudas que podrían surgir durante el proceso.

El equipo también se ha mantenido en contacto vía email con los tutores comentando el estado del proyecto. A esto se añadieron reuniones mensuales con ellos para realizar correcciones de la memoria, revisar el progreso, resolver dudas y comentar las tareas necesarias para las siguientes fases.

De esta forma, los integrantes del equipo y los tutores han estado informados constantemente del progreso del trabajo, se ha podido hacer un reparto claro de tareas y se han podido establecer tiempos de entrega realistas y factibles.

A partir de marzo, debido a la situación causada por la Covid-19, se tuvieron que suprimir las reuniones presenciales. Se han mantenido las reuniones semanales de los miembros del equipo, pero a distancia a través de Skype. También se ha mantenido el contacto vía email con los tutores, y las reuniones con ellos se han realizado a través de Google Meet.

### 4.4. Control de versiones

Para realizar la gestión del proyecto se ha decidido usar la herramienta GitHub, que es una plataforma basada en el sistema de control de versiones Git, que permite ver y llevar el registro de todos los cambios realizados en el proyecto, organizarlos y resolver cualquier conflicto que pueda surgir.

Se ha utilizado la cuenta del grupo NIL<sup>2</sup>, que nos permite gestionar todo

---

<sup>2</sup><https://github.com/NILGroup/TFG-1920-Text2LSE.git>

el código del proyecto de manera privada, ya que únicamente los miembros de dicho grupo tienen acceso a él. Al finalizar este TFG, dicha cuenta se hará pública con la finalidad de que cualquier persona pueda consultar el código.

#### 4.4.1. Estructura de Trabajo

Debido a la necesidad de trabajar los tres miembros del equipo de forma independiente, se ha decidido estructurar las ramas de trabajo de forma que se impida interferir en el trabajo de los demás compañeros. Por este motivo, se han creado tres ramas principales, como se puede ver en la Figura 4.1: Memoria, API, y Web. Desde cada una de estas ramas se crearán ramas hijas según las funcionalidades a desarrollar, lo que permite que un miembro del equipo pueda trabajar sobre ella sin necesidad de modificar el resto del proyecto. Una vez finalizado el trabajo sobre esa rama hija, esta podrá eliminarse una vez realizada la correspondiente actualización sobre la rama padre. A continuación se detalla la estructura de las ramas de trabajo:

- **MASTER:** Rama principal del proyecto donde todo ha sido revisado. De ella dependen:
  - **Memoria:** Rama principal de la memoria que contiene la versión revisada más reciente. De ella se crean nuevas ramas hijas por capítulo y funcionalidad para poder trabajar cada uno en su parte de la memoria dentro de un mismo capítulo:
    - Memoria-Capítulo-N: *Memoria-Capítulo3*
      - ◇ Memoria-Capítulo-N-funcionalidad: *Memoria-Capítulo3-Flask*
  - **API:** Rama principal de la API que contiene el código más reciente y en funcionamiento. Contiene dos ramas hijas:
    - API-Video: Rama donde se desarrolla toda la parte de procesamiento de vídeo. A partir de ella se crean ramas hijas por funcionalidad a desarrollar.
      - ◇ API-Video-funcionalidad: *Api-Video-unaPalabra*
    - API-PLN: Rama donde se desarrolla el procesamiento de lenguaje natural. A partir de ella se crean ramas hijas por funcionalidad a desarrollar.
      - ◇ API-PLN-funcionalidad: *API-PLN-sujeto*
  - **Web:** Rama principal del cliente que contiene el código más reciente y en funcionamiento. A partir de ella se crean ramas hijas por funcionalidad a desarrollar:

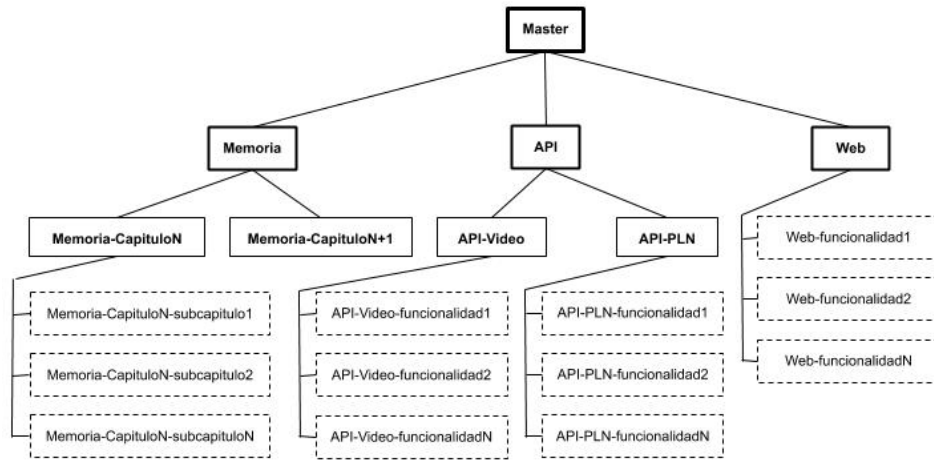


Figura 4.1: Estructura “Git” utilizada en el proyecto.

- Web-funcionalidad: *Web-llamadaFetch*

Cada vez que se termina con el desarrollo en las ramas hijas se hace un merge a cada una de ramas principales para tener la versión mas reciente de cada rama. Una vez revisadas estas ramas, se actualiza la rama Master con la última versión.

## Capítulo 5

# Text2LSE

En este capítulo se describe el trabajo desarrollado a lo largo de este TFG. Text2LSE es una aplicación web (ver Figura 5.1) que permite traducir textos escritos en castellano a LSE en formato texto LSE, vídeo e imagen en tiempo real. Text2LSE es una aplicación pública, y se puede acceder a ella desde la siguiente url:

```
https://holstein.fdi.ucm.es/tfg-text2lse
```

Text2LSE sigue una arquitectura SOA y todos los servicios implementados están disponibles para todo el mundo de manera gratuita. Los vídeos e imágenes utilizados en la traducción a LSE son los recursos ofrecidos en el catálogo de LSE de ARASAAC.

En la sección 5.1 se muestra la arquitectura de la aplicación web y cómo se comunica ésta con los servicios web. En la sección 5.2 se explican en profundidad los servicios web desarrollados, tanto los utilizados en la aplicación web, como los desarrollados para aumentar las posibilidades de obtención de información para futuros desarrolladores. Por último, en la sección 5.3 se detalla la aplicación web, tanto su diseño como su funcionalidad.

### 5.1. Arquitectura

Text2LSE es una aplicación de traducción de castellano a LSE basada en servicios web, de manera que el código desarrollado sea fácilmente reutilizable. La arquitectura utilizada para el desarrollo de este proyecto es la arquitectura de cliente-servidor, es decir, un cliente muestra una web, la cual hace peticiones al servidor, que almacena los servicios web y devuelve la respuesta al cliente.

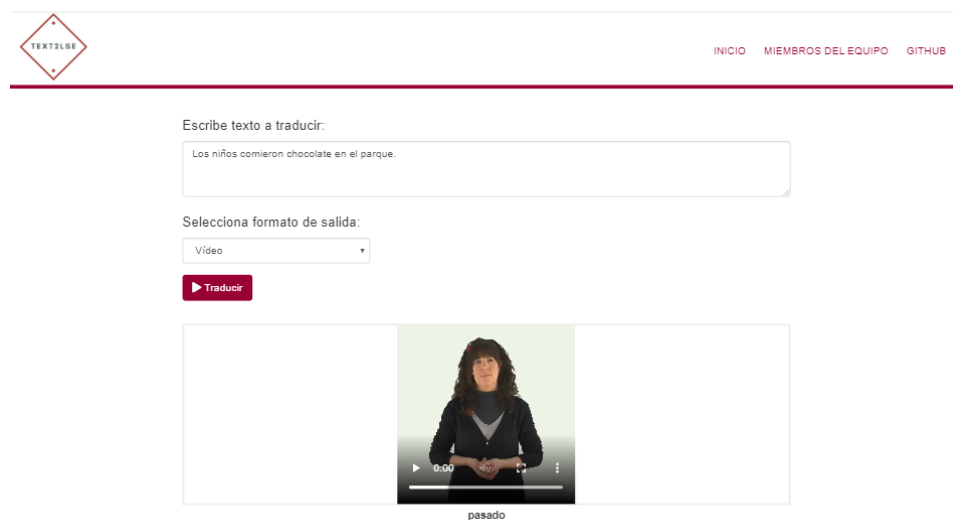


Figura 5.1: Aplicación web Text2LSE

Se ha optado por desarrollar una aplicación web debido a que ésta es accesible desde cualquier dispositivo que disponga de un navegador, ya sea un teléfono móvil, una tablet o un ordenador, sean de la marca y tamaño que sean. Para que se pueda ver de manera correcta en todos esos dispositivos, la aplicación web se ha desarrollado siguiendo un diseño responsive, es decir, que su visualización se adapte dependiendo de las dimensiones de la pantalla del dispositivo desde el cual se esté accediendo. La aplicación web ha sido desarrollada usando html, css y javascript.

Respecto a la parte del servidor, se han desarrollado siete servicios web distintos: dos para devolver de manera rápida el vídeo y la imagen en LSE de una sola palabra, otro para devolver el texto traducido a texto en LSE, dos para devolver el texto LSE adaptado a los vídeos e imágenes de ARASAAC, y otros dos para devolver el texto traducido a video e imágenes en LSE.

Se ha utilizado un proxy inverso para poder acceder tanto a la página web como a los servicios web a través desde un mismo punto de acceso. Un Proxy inverso es un método de redireccionamiento del tráfico a partes específicas de una infraestructura concreta (IONOS, 2020). Las principales finalidades para las que se usa este tipo de servidores son:

- **Anonimización:** el proxy recibe todas las llamadas al servidor y se encarga de filtrarlas y redirigirlas como se haya configurado previamente. De esta manera, desde fuera del servidor no se va a poder obtener ningún tipo de información del servidor ni de los servicios que estén en

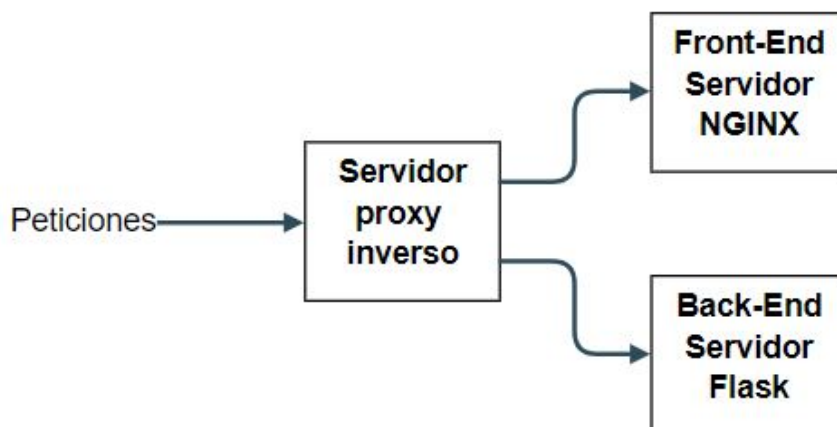


Figura 5.2: Esquema Proxy Inverso

él, solo se podrá obtener información del proxy.

- **Protección y cifrado:** al utilizar un proxy inverso se tiene la posibilidad de instalar sistemas de control y filtros de paquetes que protegen al servidor, aumentando así la seguridad del sistema.
- **Balanceo de carga:** permite redirigir las distintas solicitudes entrantes por varios servidores, permitiendo repartir la carga de trabajo para no sobrecargar ningún servidor o equilibrar la carga en el caso de que falle uno de ellos.
- **Caché:** el proxy se puede configurar para que sea capaz de almacenar las respuestas del servidor temporalmente para ofrecer una mayor velocidad de respuesta. De esta forma, si se recibe una solicitud cuya respuesta la tiene almacenada el proxy en su caché, se manda la respuesta de manera inmediata haciendo que no reciba tanta carga de procedimiento el back-end.
- **Compresión:** un proxy inverso también se puede utilizar como compresor de datos, tanto entrantes como salientes.

En este proyecto, se ha configurado el servidor Nginx como servidor proxy inverso con el fin de redirigir las distintas solicitudes al servidor correspondiente. Se han especificado una serie de rutas para diferenciar qué llamadas de usuario redireccionar a la página web y cuáles a los servicios web. Esta estructura se puede observar en la Figura 5.2.

## 5.2. Back-End

En esta sección se explican con detalle los servicios web desarrollados, su implementación y cómo utilizan de los recursos LSE de ARASAAC. La API donde están estos servicios es pública<sup>1</sup> y en los siguientes apartados se indica la URL para acceder a cada servicio.

### 5.2.1. Recursos LSE

Todas las imágenes y vídeos LSE que usa Text2LSE se han descargado del catálogo de recursos de ARASAAC<sup>2</sup>, que cuenta con 4.138 imágenes en formato jpg, y 4.102 vídeos en formato mp4. Estos recursos están almacenados en dos carpetas alojadas en el propio servidor, y cada uno de ellos se llama igual que el signo al que representa. Por ejemplo, para el signo *AGUA*, existen los ficheros *agua.jpg* y *agua.mp4*. Además, para poder traducir correctamente sustantivos con género femenino y sustantivos con número plural, necesitamos añadir a la traducción el signo *FEMENINO* y el signo *PLURAL* respectivamente, que no se incluyen en el catálogo de recursos de ARASAAC. Sin embargo, el signo *MUJER* está en el catálogo y es equivalente al signo *FEMENINO*, por lo que hemos copiado los ficheros *mujer.jpg* y *mujer.mp4* y, cambiando el nombre de las copias, hemos creado los ficheros *femenino.jpg* y *femenino.mp4*. Lo mismo sucede con el signo *OTRO* y el signo *PLURAL*, son equivalentes, por lo que hemos copiado los ficheros *otro.jpg* y *otro.mp4* para crear los ficheros *plural.jpg* y *plural.mp4*.

### 5.2.2. Servicio web de traducción de palabra a vídeo LSE

Este servicio permite obtener el vídeo del signo en LSE correspondiente a una palabra en lenguaje natural. Para poder acceder a este servicio, se debe hacer una llamada GET a la API, indicando la palabra que se desea traducir a LSE de la siguiente forma:

```
https://holstein.fdi.ucm.es/tfg-text2lse/video/<palabra>
```

En el parámetro “*palabra*” se debe indicar la palabra para la que se desea obtener su traducción en LSE.

Se utilizará el parámetro de entrada para construir una ruta concatenando dicho parámetro con la ruta del directorio de vídeos del servidor. Esta ruta se utilizará para comprobar la existencia del vídeo correspondiente. En

<sup>1</sup><https://holstein.fdi.ucm.es/tfg-text2lse>

<sup>2</sup><http://www.http://www.arasaac.org/descargas.php>

caso de que exista, se usará la función “*sendfile*” de Flask para devolverlo como respuesta, indicando como parámetro la ruta donde se aloja y su formato (mp4). En caso de que el vídeo buscado no exista, se sigue el mismo procedimiento, pero sustituyendo la ruta por la del vídeo de error. Este flujo lo podemos observar en la Figura 5.3.

Por ejemplo, para obtener el vídeo en LSE de la palabra “*agua*” se debe realizar la siguiente llamada:

```
https://holstein.fdi.ucm.es/tfg-text2lse/video/agua
```

### 5.2.3. Servicio web de traducción de palabra a imagen LSE

Este servicio permite obtener la imagen del signo en LSE correspondiente a una palabra en lenguaje natural. Para poder acceder a este servicio, se debe hacer una llamada GET a la API, indicando la palabra que se desea traducir a LSE de la siguiente forma:

```
https://holstein.fdi.ucm.es/tfg-text2lse/imagen/<palabra>
```

En el parámetro “*palabra*” se debe indicar la palabra de la que se desea obtener su traducción en LSE.

Se utilizará el parámetro de entrada para construir una ruta concatenando dicho parámetro con la ruta del directorio de imágenes del servidor. Esta ruta se utilizará para comprobar la existencia de la imagen correspondiente. En caso de que exista, se usará la función “*sendfile*” de Flask para devolverlo como respuesta, indicando como parámetro la ruta donde se aloja y su formato (jpg). En caso de que la imagen buscada no exista, se sigue el mismo procedimiento, pero sustituyendo la ruta por la de la imagen de error, la cual la podemos observar en la Figura 5.4. Este flujo lo podemos observar en la Figura 5.5.

Por ejemplo, para obtener la imagen en LSE de la palabra “*coche*” se debe realizar la siguiente llamada:

```
https://holstein.fdi.ucm.es/tfg-text2lse/imagen/coche
```

La respuesta a la llamada del ejemplo devolverá la imagen de la Figura 5.6.

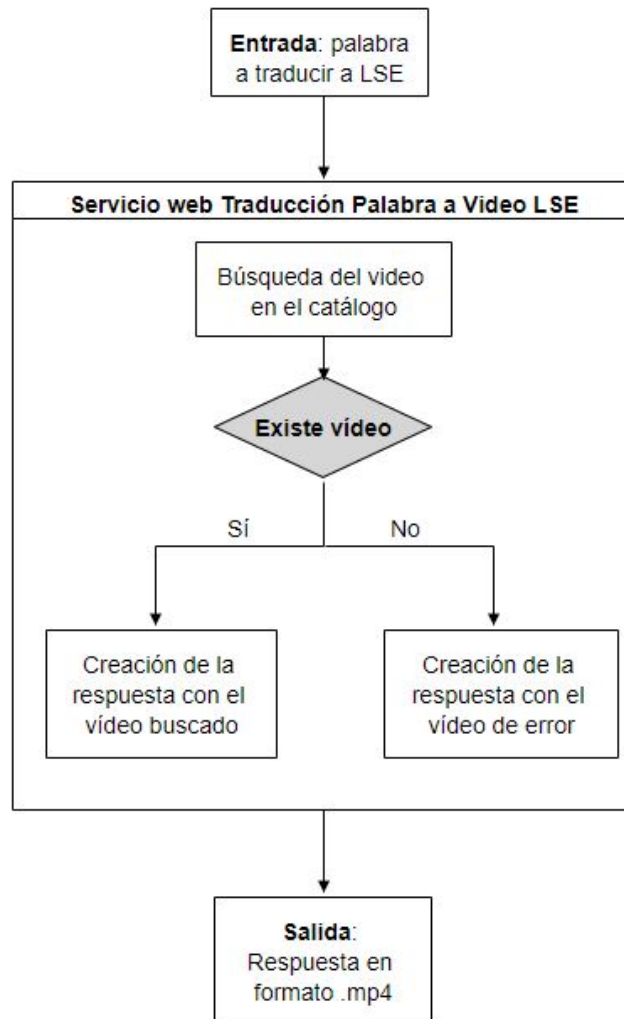


Figura 5.3: Flujo del servicio de Traducción de una palabra a vídeo en LSE



Figura 5.4: Imagen de error devuelta en la traducción de una palabra a LSE

#### 5.2.4. Servicio web para Traducción de texto en castellano a texto en LSE

Este servicio implementa la funcionalidad de traducción de un texto en castellano a LSE en formato texto. Como la LSE tiene una estructura gramatical distinta al castellano es necesario obtener una traducción en LSE en formato texto que sirva de base para los demás servicios implementados, ya que su salida en formato texto les servirá como entrada.

Por ejemplo la frase “*El niño comió chocolate*” en texto LSE sería “*niño parque jugar*”. Para traducirlo a formato imagen o formato vídeo habría que buscar las imágenes o vídeos (niño parque jugar) con los correspondientes servicios de imagen y vídeo.

Para poder acceder a este servicio, se debe realizar una petición POST a la API en la siguiente URL:

```
https://holstein.fdi.ucm.es/tfg-text2lse/TextoLSE
```

Los datos de entrada en la petición POST deben tener la siguiente estructura en JSON donde el parámetro “texto” se debe incluir el texto que se desea traducir a LSE.

```
{ “Texto” : “<texto>” }
```

La salida tiene la misma estructura en formato JSON:

```
{ “Texto” : “<texto>” }
```

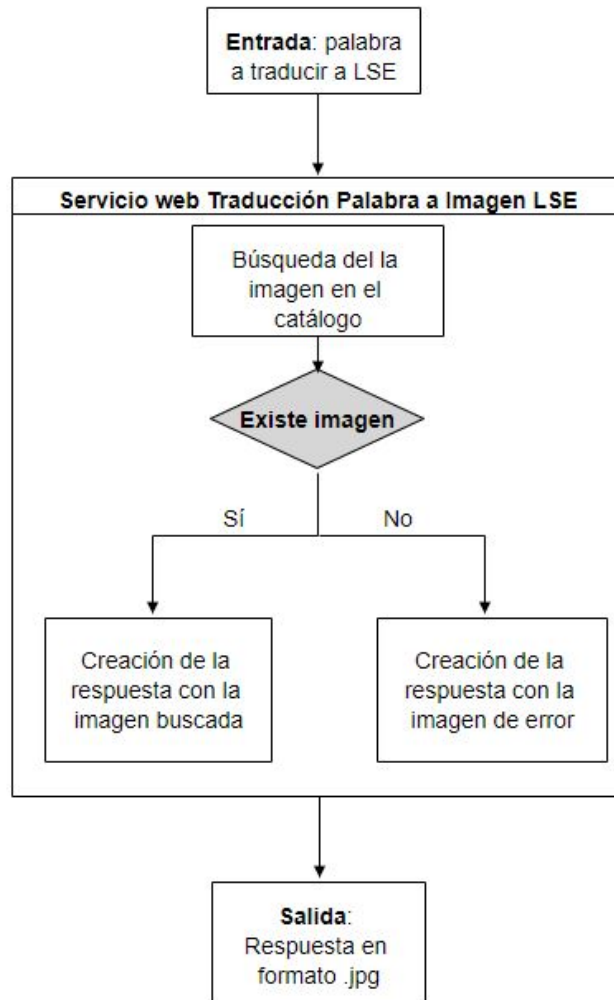


Figura 5.5: Flujo del servicio de Traducción de una palabra a imagen en LSE



Figura 5.6: Imagen devuelto por el servicio de traducción de la palabra “*coche*” a LSE

Un ejemplo de uso sería:

Entrada: { “Texto” : “El niño bebe agua.” }

Salida: { “Texto” : “niño agua beber.” }

Para que el servicio web sea capaz de traducir una frase a texto LSE, se debe que realizar un procesamiento previo del texto en castellano para poder adaptarlo a la Lengua de Signos Española. Este procesamiento es realizado con un sistema de reglas, que filtran qué palabras deben de ser utilizadas y cuáles no, al igual que determina su orden en la oración en LSE, puesto que la estructura gramatical de la LSE es diferente al castellano, tal y como se contó en la sección 2.3.1.2. La elección a la hora de desarrollar el Procesamiento del Lenguaje Natural para nuestro proyecto estaba entre un sistema basado en reglas o mediante el uso de aprendizaje automático. La opción de desarrollar un aprendizaje automático no era viable debido a la dificultad y la falta de ejemplos en castellano para entrenar el sistema. Por este motivo se eligió el sistema basado en reglas, ya que era más fácil de implementar y, aunque puede limitar la capacidad de traducción de frases que no cumplan dichas reglas, cumple con el objetivo marcado en el proyecto, que es traducir frases simples a LSE.

Este servicio web lo primero que hace es recopilar información de la frase para luego realizar un análisis sintáctico y morfológico, y así poder ordenarla respetando la estructura de la LSE. En las siguientes subsecciones se explicará con más detalle cada cada uno de los pasos del flujo de este servicio (Ver Figura: 5.7)

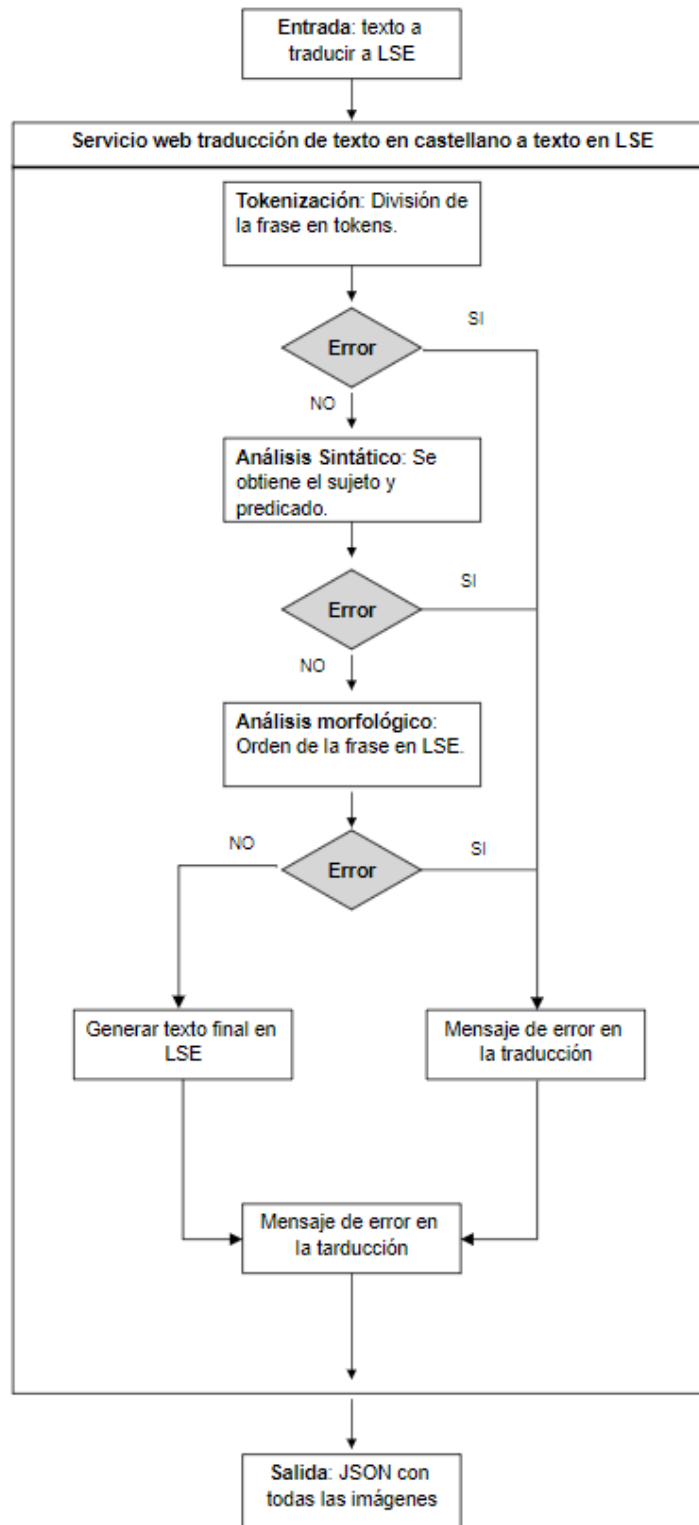


Figura 5.7: Flujo servicio de traducción de texto en castellano a texto en LSE

### 5.2.4.1. Tokenización

Para poder realizar el análisis sintáctico y morfológico de la oración se necesita obtener toda la información posible de la oración. Para ello, se realizará con Spacy el proceso de tokenización, el cual divide la oración en una lista de palabras denominadas *tokens*, donde cada token contiene la siguiente información de cada palabra:

- **Dependencia:** Indica la relación de dependencia sintáctica dentro de la oración, es decir, qué función tiene la palabra dentro de la oración. Las funciones principales son sujeto, raíz (que es la palabra de la que dependen todas las demás) u objeto. En la Figura 5.8 se puede ver un ejemplo donde “yo” es el sujeto, el verbo “comer” la raíz y “chocolate” el objeto.

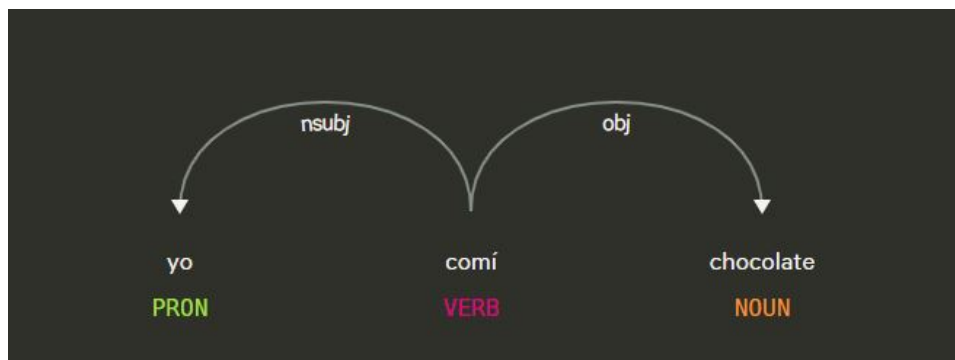


Figura 5.8: Dependencias de una frase

- **Tags:** Indica el género, número, el tiempo verbal y la persona de cada token.
- **Part of speech:** Indica el tipo de palabra en función de su uso en la oración, tales como nombres, adjetivos, adverbios y verbos.
- **Lemma:** Es la palabra que encabeza un artículo de un diccionario o de una enciclopedia. Por ejemplo, el lemma de “jugábamos” es “jugar”.

Por ejemplo, los tokens de la frase “*Los niños merendaron chocolate*” serían los que aparecen en la Figura 5.9. Como se puede ver, cada token tiene su información detallada. Por ejemplo, de los tokens “*los*” y “*niños*” la información obtenida sería:

- **Los:** la dependencia indica que es un determinante y los tags que es artículo determinado, masculino y plural.
- **niños:** la dependencia indica que es el sujeto de la oración, los tags que es masculino y plural y el lemma es niño.

```

LOS:
[DEPENDENCIA (det) --- TAGS {DET__Definite=Def|Gender=Masc|Number=Plur|PronType=Art} --- POS (DET) --- LEMMA (Los)]
NIÑOS:
[DEPENDENCIA (nsubj) --- TAGS {NOUN__Gender=Masc|Number=Plur} --- POS (NOUN) --- LEMMA (niño)]
MERENDARON:
[DEPENDENCIA (ROOT) --- TAGS {VERB__Mood=Ind|Number=Plur|Person=3|Tense=Past|VerbForm=Fin} --- POS (VERB) --- LEMMA (merendar)]
CHOCOLATE:
[DEPENDENCIA (obj) --- TAGS {NOUN__Gender=Masc|Number=Sing} --- POS (NOUN) --- LEMMA (chocolate)]
.:

```

Figura 5.9: Tokens de la frase “*Los niños merendaron chocolate*”

Durante el proceso de tokenización también se obtiene un árbol de dependencias entre palabras. Para entender cómo funcionan las dependencias se va a utilizar como ejemplo la oración “*Ayer los niños merendaban chocolate*”, donde sus dependencias se muestran en la Figura 5.10

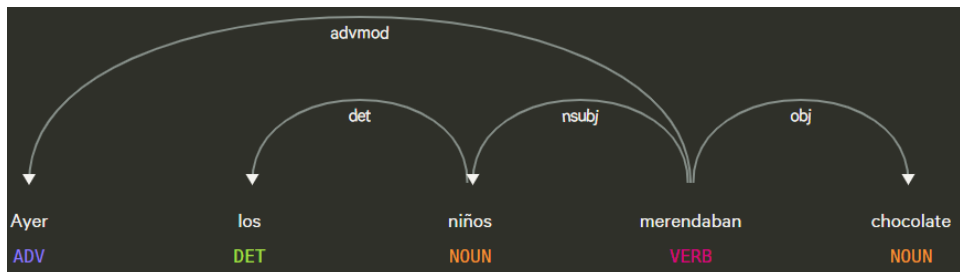


Figura 5.10: Displacy grafo de dependencias

Como se puede apreciar en el diagrama del token, de la palabra “*merendaban*”, que es la raíz de la oración, dependen tres tokens (Ayer, niños y chocolate), y de ellos también dependen otros y así sucesivamente. Estas dependencias se traducen en un árbol, donde el ROOT es la raíz, como podemos ver en la Figura 5.11:

#### 5.2.4.2. Análisis sintáctico

Una vez separada la oración en tokens, hay que diferenciar qué palabras forman parte del sujeto y cuáles del predicado. Para ello hay que recorrer el árbol de dependencias mediante una función recursiva implementada por nosotros, empezando desde el ROOT, que es el token raíz del cual dependen todos los demás. Para obtener el sujeto, se busca el token que contenga la dependencia “nsubj”. A continuación, se obtienen los hijos del token “nsubj” y se guardan como parte del sujeto, mientras que los tokens restantes, es decir, aquellos que no pertenecen al sujeto, formarían parte del predicado. En el ejemplo anterior “*Ayer los niños merendaban chocolate*” el **sujeto** sería “*Los, niños*” y el **predicado** “*Ayer, merendaban y chocolate*”.

El resultado de los tokens pertenecientes al sujeto y al predicado se alma-

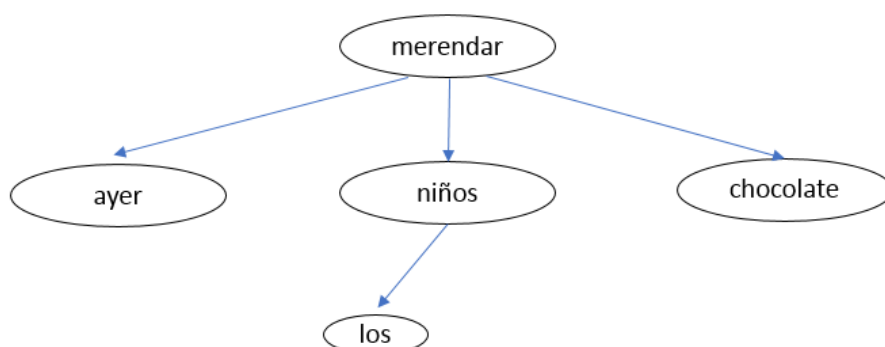


Figura 5.11: Árbol de la oración “Los niños merendaron chocolate”

cenan en una lista. Para el ejemplo anterior las dos listas resultantes serían:

- **Sujeto:** [Los, niños]
- **Predicado:** [Ayer, merendaban, chocolate]

#### 5.2.4.3. Análisis morfológico

La Lengua de Signos Española tiene una estructura gramatical diferente al castellano. Para comenzar con la traducción se toma como punto de partida la estructura más simple en LSE:

TIEMPO + SUJETO + OBJETO + VERBO.

Como se explica en la sección 2.3.1.2 en la LSE las palabras pueden estar estructuradas de forma diferente con respecto a la oración original en castellano.

Para conseguir la traducción de la frase original en castellano a LSE se necesita un análisis morfológico que permita detectar:

- **Determinantes:** Es importante detectar estos elementos, ya que los determinantes se omiten a la hora de hacer la traducción. Por ejemplo, “*El niño come carne*” se traduce a LSE como “*NIÑO CARNE COMER*”.
- **Verbos copulativos:** Los verbos copulativos ser, estar o parecer se omiten a la hora de la traducción a texto LSE. Por ejemplo, “*yo soy bajo*” se traduce como “*YO BAJO*”.
- **Poseivos:** Los determinantes poseivos se sustituyen por pronombres personales. Por ejemplo, “*Mi niño es bajo*” se traduce como “*YO NIÑO BAJO*”.

- **Adjetivos:** Al traducir una frase a LSE, los adjetivos siempre van en masculino. Por ejemplo, “*Mi mamá es fea*” se traduce como “*YO MAMÁ FEO*”.
- **Temporalidad:** En la LSE, la temporalidad va al principio de la oración. Por ejemplo, “*Yo comí chocolate ayer*” se traduce como “*AYER YO CHOCOLATE COMER*”. Sin embargo, no siempre hay adverbios de tiempo en la oración que indiquen la temporalidad. En ese caso, el tiempo viene determinado por el tiempo verbal. Por ejemplo, “*Yo comeré chocolate*” se traduce como “*FUTURO YO CHOCOLATE COMER*”.

Una vez añadidas y quitadas las palabras que correspondan del sujeto y del predicado, la frase se tiene que ordenar según la siguiente estructura: tiempo + sujeto + objeto + verbo. Así el resultado final de la frase usada como ejemplo (“Los niños ayer merendaban chocolate”) sería:

- **Sujeto:** [niños]
- **Predicado:** [Ayer, merendar, chocolate]

El texto en LSE final que devolvería el servicio en este caso sería: “AYER NIÑOS CHOCOLATE MERENDAR”

### 5.2.5. Servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de vídeos de ARASAAC

Este servicio implementa la funcionalidad de traducción de un texto en castellano a LSE en formato texto en función de los vídeos existentes en el catálogo de ARASAAC. Para poder acceder a este servicio, se debe realizar la siguiente petición POST a la API:

```
https://holstein.fdi.ucm.es/tfg-text2lse/TextoLSEVideos/
```

Los datos a enviar en la petición POST deben tener la siguiente estructura en JSON:

```
{ "Texto" : "<texto>" }
```

En el parámetro “*texto*” se debe incluir el texto que se desea traducir a LSE. Este servicio utilizará este parámetro para hacer una llamada al servicio de traducción de texto en LN a texto LSE, con el fin de obtener la traducción a texto LSE. Una vez obtenida esta traducción, se realizarán una serie de

comprobaciones y transformaciones por cada palabra que sea un sustantivo, con la finalidad de adaptarlas al catálogo de vídeos de ARASAAC. Para comenzar, se comprueba si existe vídeo para ese sustantivo, y en caso de que exista se pasa a realizar la misma comprobación para la siguiente palabra. Si por el contrario no existe el vídeo, se eliminan los morfemas de género y número, y se vuelve a comprobar si existe el vídeo. Si tras este cambio sigue sin existir el vídeo, se deja la palabra sin modificar y se pasa a comprobar la siguiente. Si existe vídeo tras el cambio, se comprueba si la palabra era femenina y si es así se añade a continuación la palabra “*FEMENINO*”. Después se comprueba si la palabra estaba en plural y si es así se añade a continuación la palabra “*PLURAL*”. Al realizar estas comprobaciones con todas las palabras, se crea la respuesta en formato JSON y se devuelve. El flujo de este servicio lo podemos observar en la Figura 5.12.

Por ejemplo, para traducir el texto “*Mis tías comen chocolate*” a LSE en formato texto en función de los vídeos existentes, se debe realizar la llamada POST indicada anteriormente con el siguiente JSON:

```
Entrada: { “Texto” : “Mis tías comen chocolate” }
```

Al no existir el vídeo para la palabra “*tías*”, busca el vídeo de esa palabra sin morfemas de género y número, que en este caso es la palabra “*tío*”. Como “*tías*” es plural y femenino, se añaden las palabras “*FEMENINO*” para indicar el femenino y “*PLURAL*” para indicar el plural. Para finalizar, se crea la respuesta en formato JSON y se devuelve. La salida en este caso sería:

```
Salida: { “Texto” : “YO TÍO FEMENINO PLURAL CHOCOLATE  
COMER” }
```

### 5.2.6. Servicio web de traducción de texto en LN a texto LSE adaptado a las imágenes del catálogo de ARASAAC

Este servicio implementa la funcionalidad de traducción de un texto en castellano a LSE en formato texto en función de las imágenes existentes en el sistema. Para poder acceder a este servicio, se debe realizar la siguiente petición POST a la API:

```
https://holstein.fdi.ucm.es/tfg-text2lse/textoImagen/
```

Los datos a enviar en la petición POST deben tener la siguiente estructura en JSON:

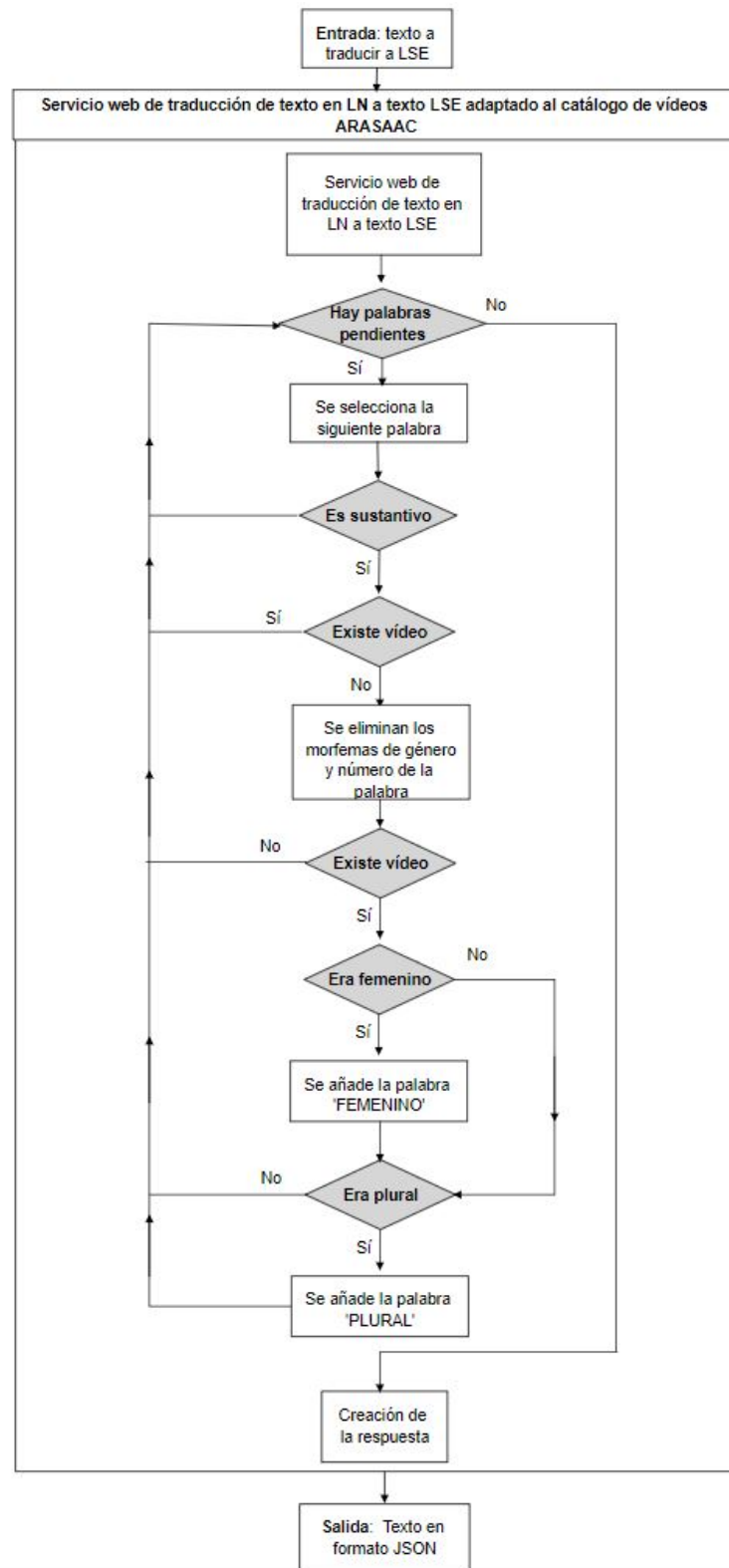


Figura 5.12: Flujo servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de vídeos de ARASAAC

```
{ "Texto" : "<texto>" }
```

En el parámetro *texto* se debe incluir el texto que se desea traducir a LSE. Al recibir este parámetro, este servicio lo utilizará para hacer una llamada al servicio de traducción de texto en LN a texto LSE, con el fin obtener la traducción a texto LSE. Una vez obtenida esta traducción, se realizarán una serie de comprobaciones y transformaciones por cada palabra que sea un sustantivo, con la finalidad de adaptarlas al catálogo de imágenes de ARA-SAAC. Para comenzar, se comprueba si existe la imagen para ese sustantivo, y en caso de que exista se pasa a comprobar la siguiente palabra. Si por el contrario no existe la imagen, se eliminan los morfemas de género y número, y se vuelve a comprobar si existe la imagen. Si tras este cambio sigue sin existir la imagen, se deja la palabra sin modificar y se pasa a comprobar la siguiente. Si existe la imagen tras el cambio, se comprueba si la palabra era femenina y si es así se añade a continuación la palabra *FEMENINO*. Después se comprueba si la palabra estaba en plural y si es así se añade a continuación la palabra *PLURAL*. Al realizar estas comprobaciones con todas las palabras, se crea la respuesta en formato JSON y se devuelve. El flujo de este servicio lo podemos observar en la Figura 5.13.

Por ejemplo, para traducir el texto *Los caballos son rápidos* a LSE en formato texto en función de las imágenes existentes, se debe realizar la llamada POST indicada anteriormente con el siguiente JSON:

```
Entrada: { "Texto" : "Los caballos son rápidos" }
```

Al no existir la imagen para la palabra *caballos*, busca la imagen de esa palabra sin morfemas de género y número, que en este caso es la palabra *caballo*. Como *caballos* es plural, se añade la palabra *PLURAL* para indicar el plural. Para finalizar, se crea la respuesta en formato JSON y se devuelve. La salida en este caso sería:

```
Salida: { "Texto" : "CABALLO PLURAL RÁPIDO" }
```

### 5.2.7. Servicio web para traducción de texto a LSE (vídeo)

Este servicio implementa la funcionalidad de traducción de un texto en castellano a LSE en formato video. Para poder acceder a este servicio, se debe realizar la siguiente petición POST a la API:

```
https://holstein.fdi.ucm.es/tfg-text2lse/video/
```

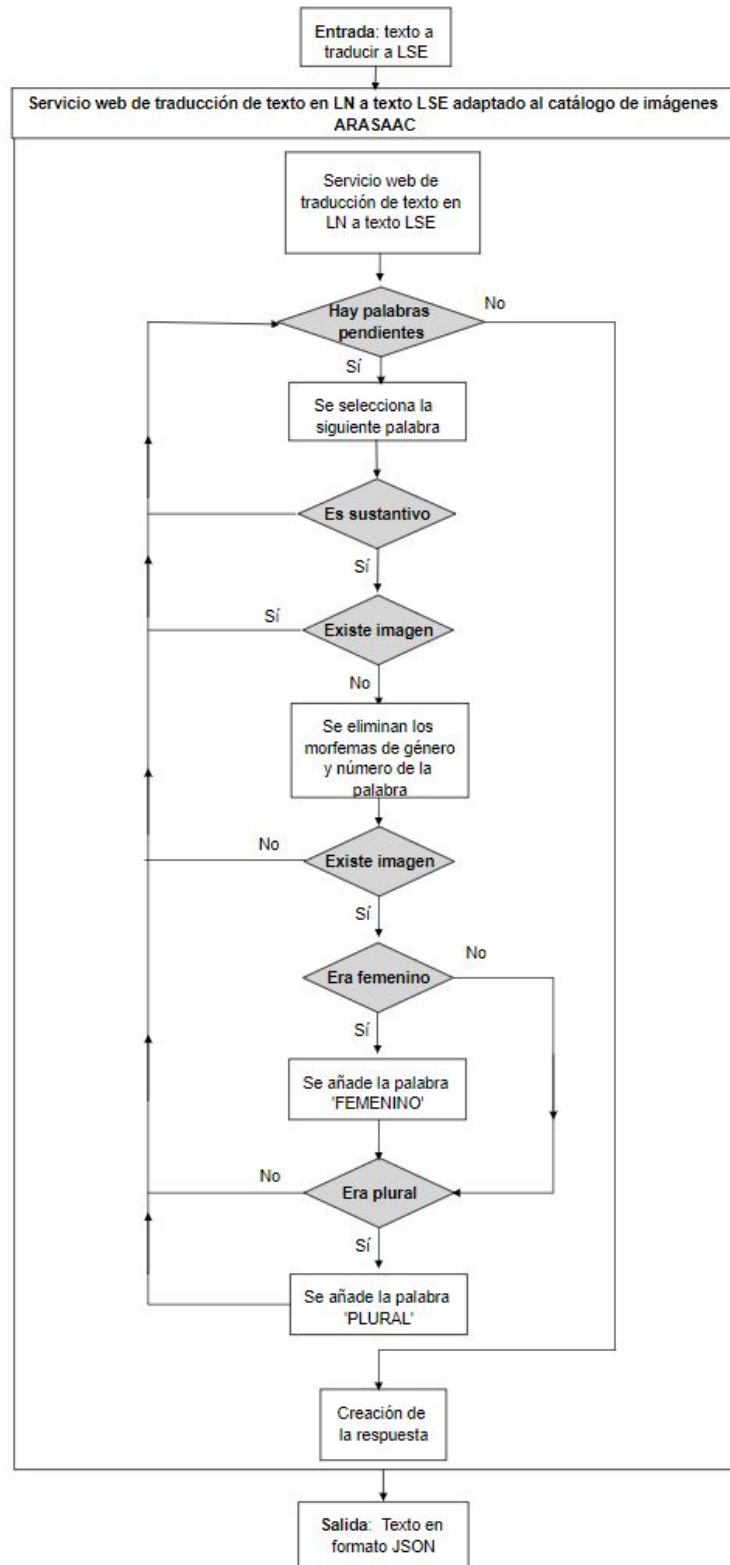


Figura 5.13: Flujo servicio web de traducción de texto en LN a texto LSE adaptado al catálogo de imágenes de ARASAAC

Los datos a enviar en la petición POST deben tener la siguiente estructura en JSON:

```
{ "Texto" : "<texto>" }
```

En el parámetro *texto* se debe incluir el texto que se desea traducir a LSE. Al recibir este parámetro, este servicio lo utilizará para hacer una llamada al servicio de traducción de texto en LN a texto LSE adaptado al catálogo de vídeos de ARASAAC, con el fin obtener la traducción a texto LSE. Una vez obtenida esta traducción, por cada palabra se llamará al servicio de traducción de palabra a vídeo LSE, que devolverá el video correspondiente o el vídeo de error en caso de que no exista el video buscado. Al finalizar con todas las palabras se utilizará la librería Moviepy<sup>3</sup> de Python para concatenar todos vídeos recibidos en uno solo, que se devolverá como respuesta. El flujo de este servicio lo podemos observar en la Figura 5.14.

Por ejemplo, para obtener el vídeo en LSE del texto *“Los niños comen chocolate”*, se debe realizar la llamada POST indicada anteriormente con el siguiente JSON:

```
{ "Texto" : "Los niños comen chocolate" }
```

La respuesta recibida será un único vídeo en formato mp4 con la traducción deseada.

### 5.2.8. Servicio web para traducción de texto a LSE (imagen)

Este servicio implementa la funcionalidad de traducción de un texto en castellano a LSE en formato imagen. Para poder acceder a este servicio, se debe realizar la siguiente petición POST a la API:

```
https://holstein.fdi.ucm.es/tfg-text2lse/imagenes/
```

Los datos a enviar en la petición POST deben tener la siguiente estructura en JSON:

```
{ "Texto" : "<texto>" }
```

En el parámetro *texto* se debe incluir el texto que se desea traducir a LSE. Al recibir este parámetro, este servicio lo utilizará para hacer una llamada al servicio de traducción de texto en LN a texto LSE adaptado al

<sup>3</sup><https://zulko.github.io/moviepy/index.html>

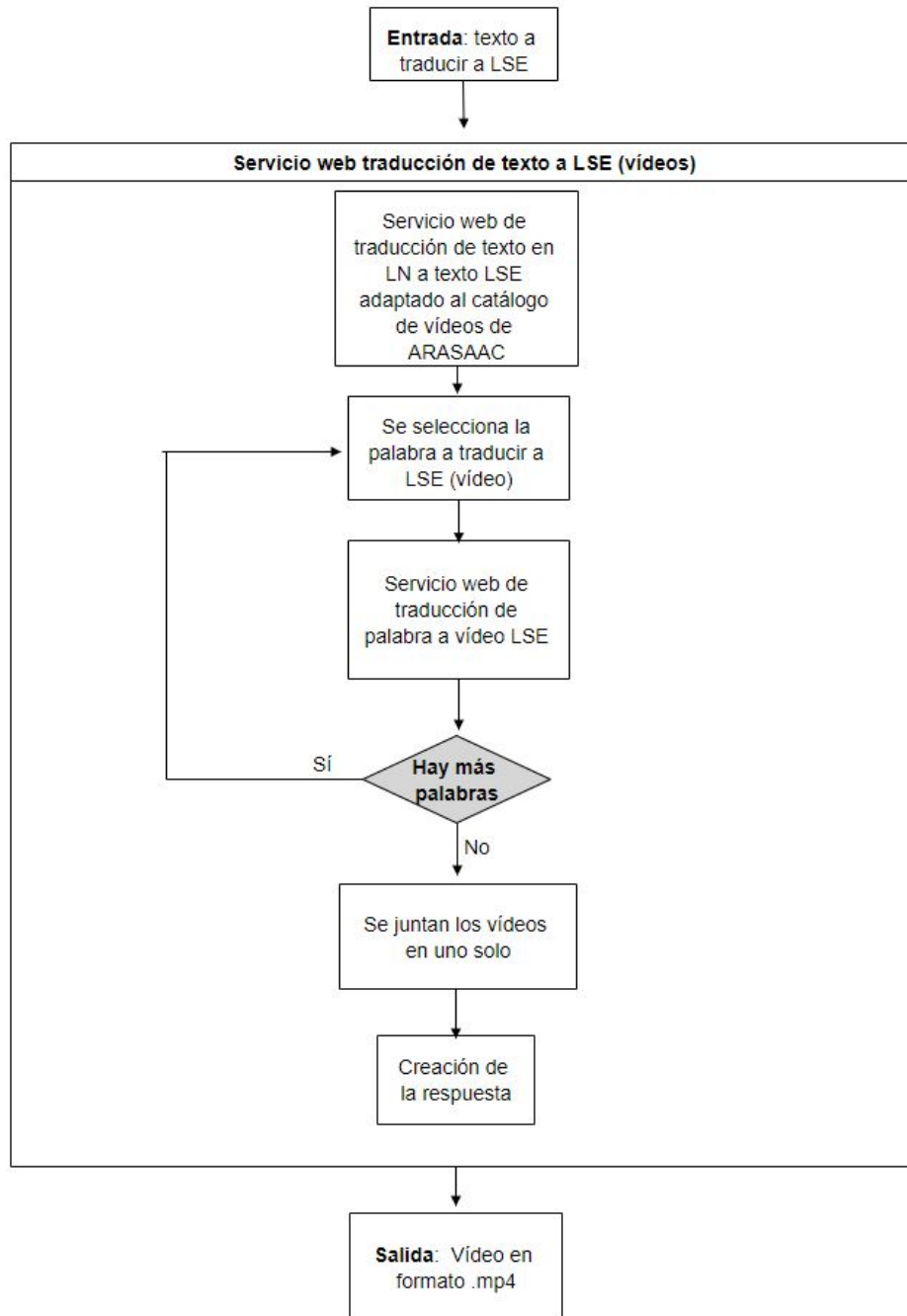


Figura 5.14: Flujo del servicio web para traducción de texto a LSE (vídeo)

catálogo de imágenes de ARASAAC, con el fin obtener la traducción a texto LSE. Una vez obtenida esta traducción, por cada palabra se incluirá de manera ordenada en un array la ruta necesaria para obtener dicha imagen mediante una petición al servicio de traducción de palabra a imagen LSE. Al finalizar con todas las palabras, se añadirán dicho array a un JSON que se devolverá como respuesta. El flujo de este servicio lo podemos observar en la Figura 5.15.

Por ejemplo, para obtener las imágenes en LSE del texto *“Mañana yo iré al médico”*, se debe realizar la llamada POST indicada anteriormente con el siguiente JSON:

```
{ "Texto" : "Mañana yo iré al médico" }
```

La respuesta recibida será un JSON que contendrá un array con todas las rutas de las imágenes, como podemos ver a continuación en la Figura 5.16.

### 5.3. Front-End

En esta sección se explica en detalle la aplicación web desarrollada, cuya finalidad es proporcionar a los usuarios una interfaz sencilla donde puedan introducir el texto que desean traducir y obtener la traducción del texto en LSE, tanto en imágenes como en vídeo. En la Figura 5.1 se muestra la interfaz de la aplicación, que consta de un input donde el usuario podrá introducir el texto en castellano que desee traducir, un desplegable donde podrá seleccionar el formato de salida y un botón para realizar la traducción.

Los formatos de salida disponibles en el desplegable son los siguientes:

- **Traducción a vídeo:** tras seleccionar este formato y pulsar el botón “traducir”, se realizará una llamada Fetch al servicio web de traducción de texto en lenguaje natural a texto LSE (ver sección 5.2.5), pasando por el servidor proxy, que a su vez redirigirá la petición a la API. Una función Fetch se encargará de recibir el texto resultante y de llamar por cada una de esas palabras al servicio que devuelve el video LSE correspondiente. Posteriormente se incrustan todos los vídeos en orden en el código HTML, de tal manera que se reproduce uno detrás de otro. Así, se podrá visualizar por pantalla un vídeo en el que se podrá ver a un intérprete de ARASAAC realizando los signos LSE correspondientes. En la Figura 5.17 se puede observar el flujo para la traducción en este formato. Se ha decidido hacerlo así en lugar de llamar directamente servicio de traducción de texto a LSE (vídeo), explicado en la sección 5.2.7, porque el tiempo de respuesta de ese servicio web es

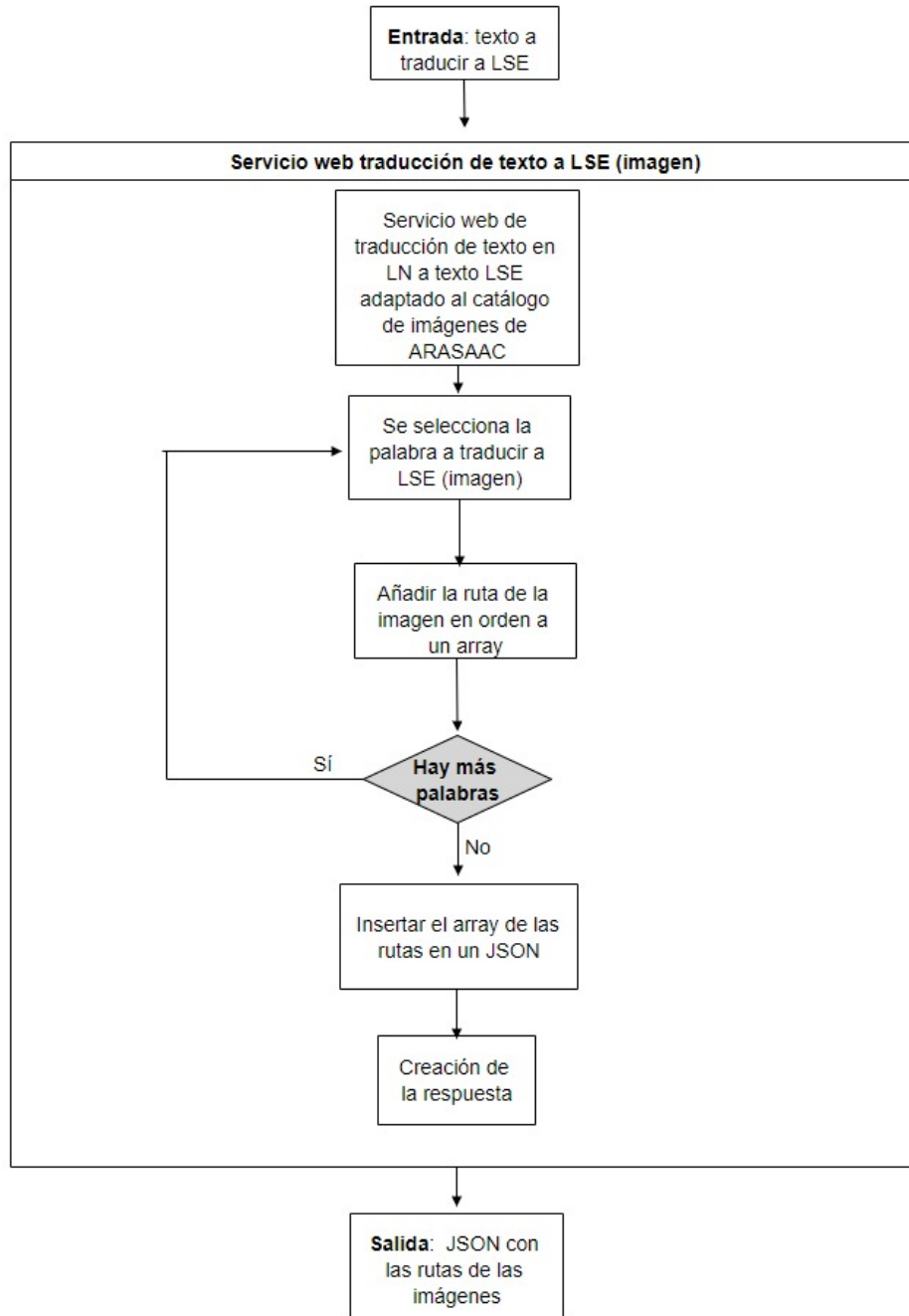


Figura 5.15: Flujo del servicio web para traducción de texto a LSE (imagen)

```
{
  "rutas": [
    "https://holstein.fdi.ucm.es/tfg-text2lse/imagen/mañana",
    "https://holstein.fdi.ucm.es/tfg-text2lse/imagen/yo",
    "https://holstein.fdi.ucm.es/tfg-text2lse/imagen/médico",
    "https://holstein.fdi.ucm.es/tfg-text2lse/imagen/ir"
  ]
}
```

Figura 5.16: JSON devuelto por el servicio web para traducción de texto a LSE (imagen)

largo y supone una gran carga de trabajo para el servidor concatenar vídeos mp4 y devolverlos como respuesta.

- **Traducción a imágenes:** de la misma manera que en el caso de los vídeos, tras seleccionar este formato y pulsar el botón “traducir”, se realizará una llamada Fetch al servicio web para traducción de texto a LSE (imagen), pasando por el servidor proxy, que a su vez redirigirá la petición a la API. Una función Fetch se encargará de recibir el JSON resultante, que contendrá un array con las URLs necesarias para obtener las imágenes. Posteriormente se utilizan dichas URLs para incrustar todas las imágenes en orden en el código HTML, mostrando así la traducción deseada. En la Figura 5.18 se puede observar el flujo para la traducción a imágenes.
- **Traducción a texto LSE:** tras seleccionar este formato y pulsar el botón “traducir” se llamará a una función Fetch que realizará una petición al servicio de traducción de texto en castellano a texto en LSE. En caso de éxito aparecerá por pantalla dicho texto traducido a texto LSE. En la Figura 5.19 se puede observar el flujo para esta traducción.

Por otro lado, la interfaz también cuenta con un menú en la parte superior, donde además de a la página principal podemos acceder a una página con información sobre los miembros del equipo y de los tutores, y a otra con información sobre la aplicación.

Toda la aplicación se ha implementado siguiendo un diseño responsive, haciendo uso de Bootstrap 3, para garantizar que la herramienta sea accesible desde cualquier dispositivo. En la Figura 5.20 podemos observar la interfaz vista desde un dispositivo móvil.

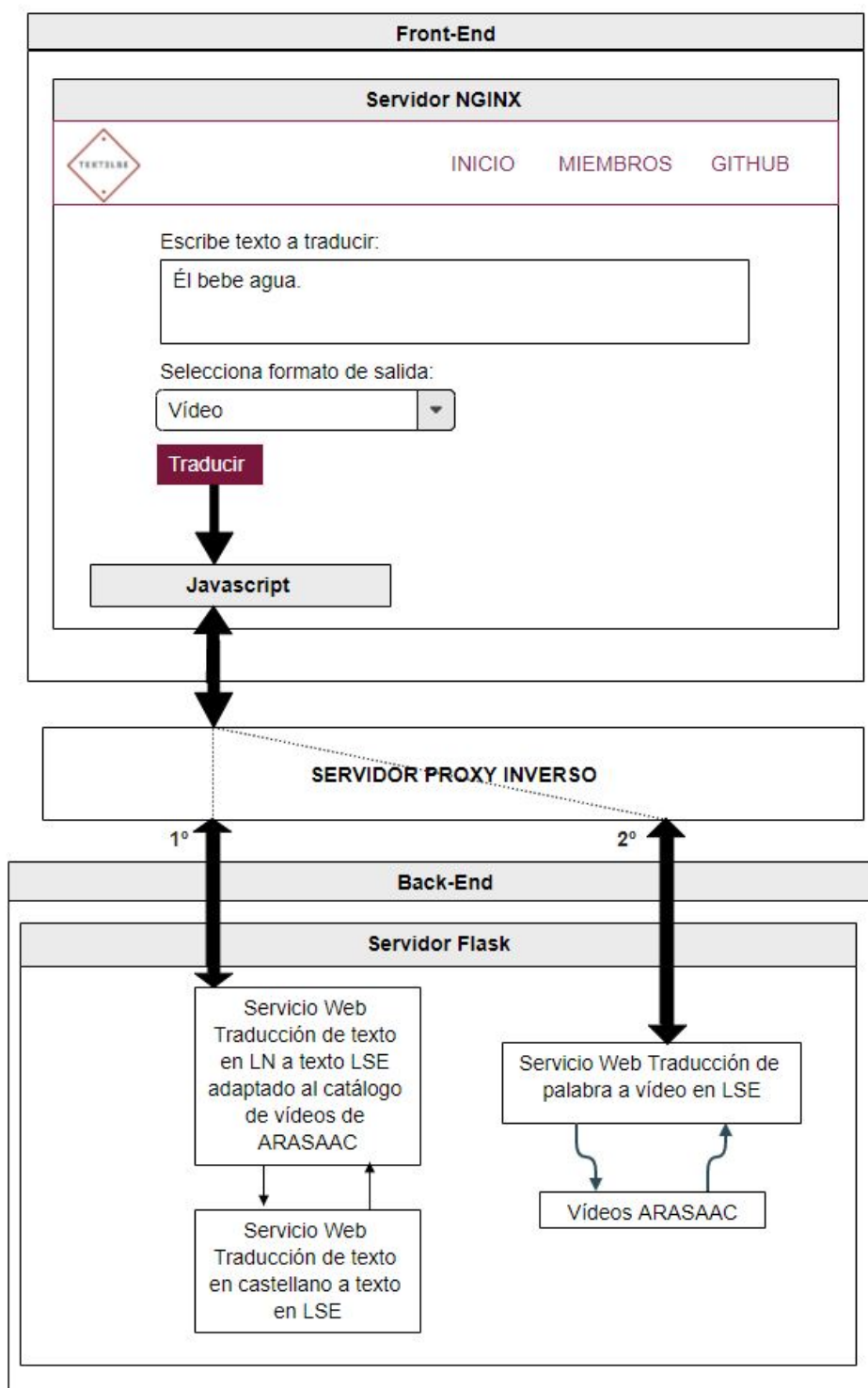


Figura 5.17: Flujo de la aplicación web al seleccionar la traducción en formato vídeo.

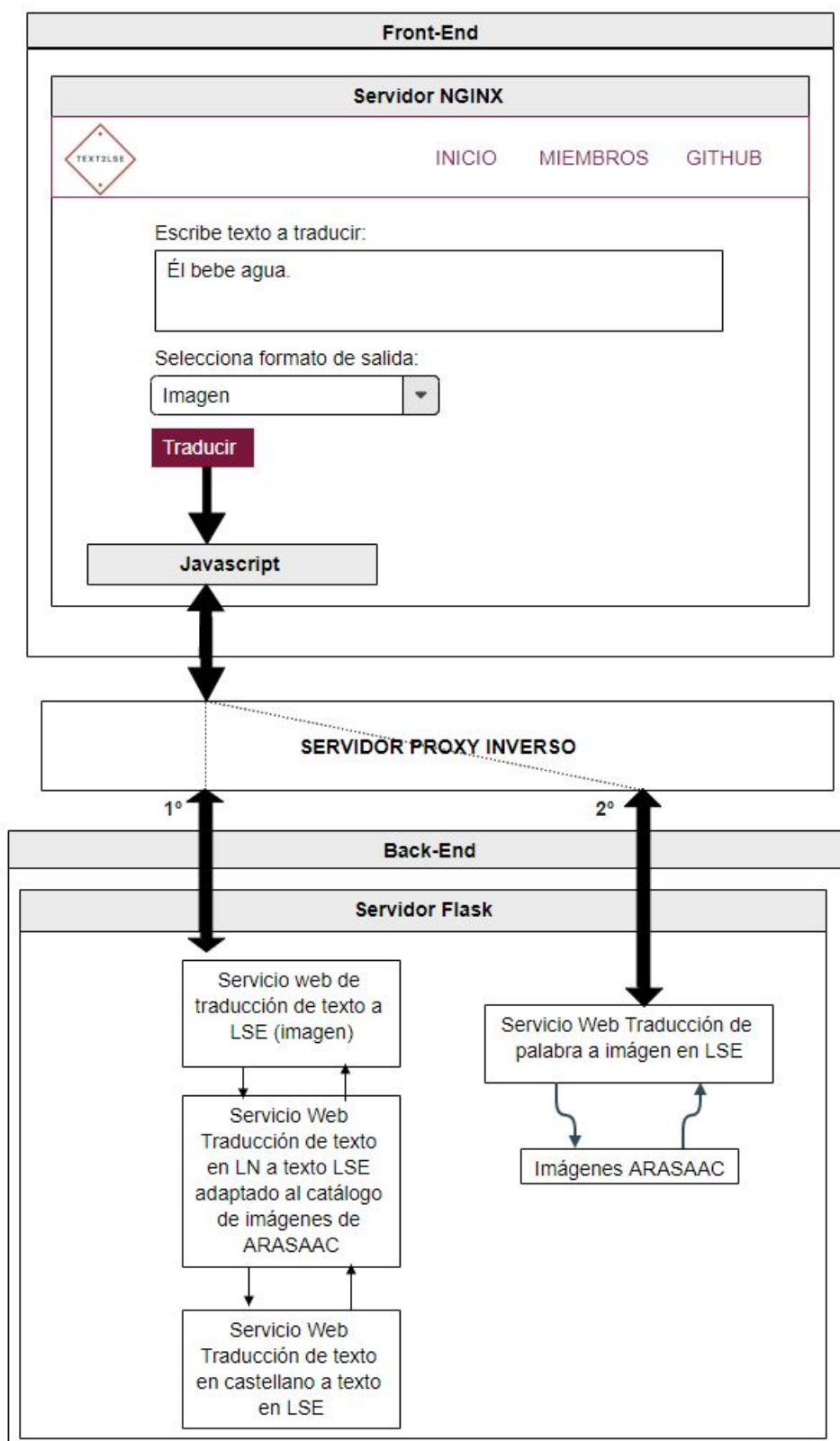


Figura 5.18: Flujo de la aplicación web al seleccionar la traducción en formato imagen.

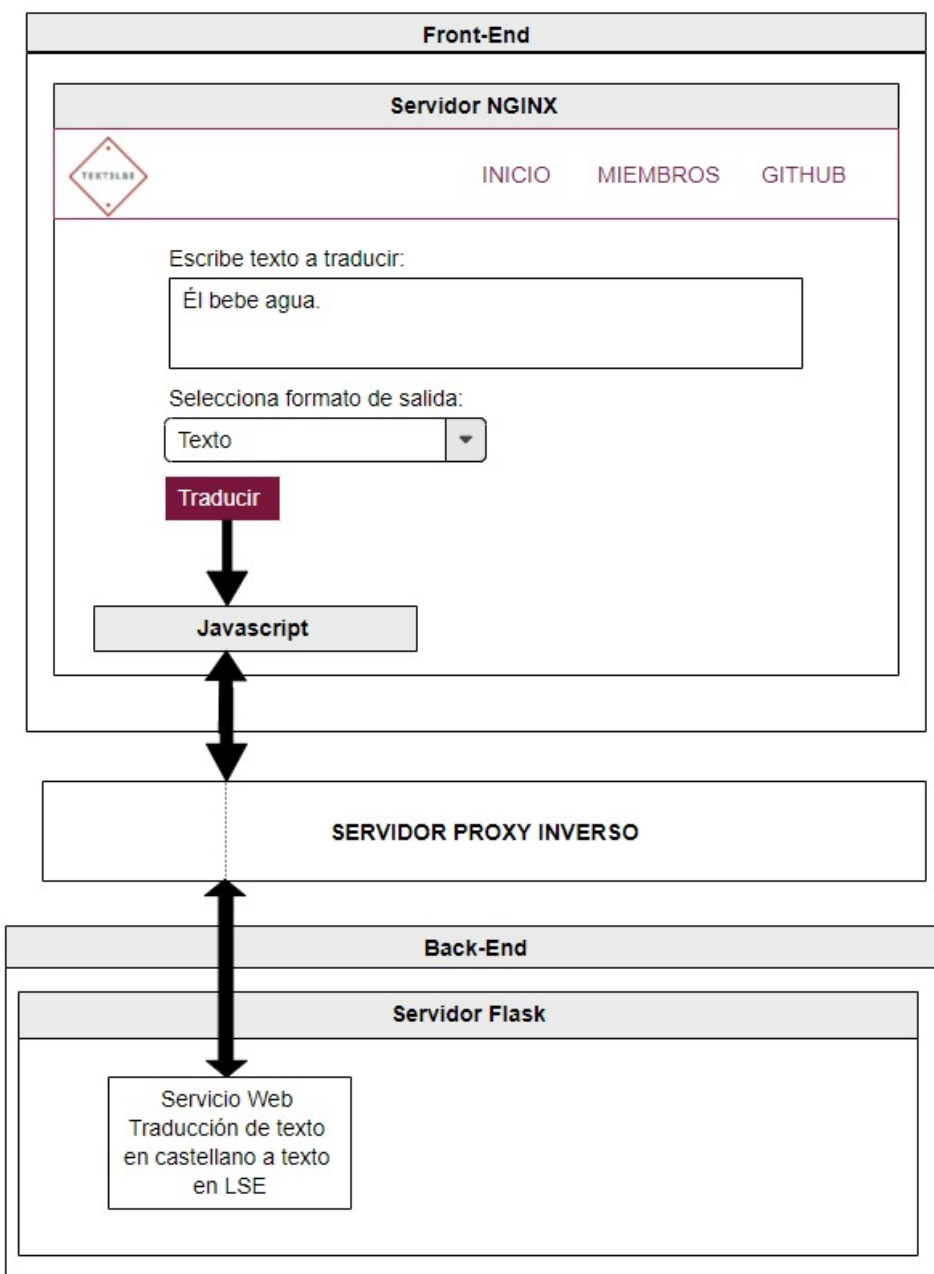


Figura 5.19: Flujo de la aplicación web al seleccionar la traducción en formato texto.

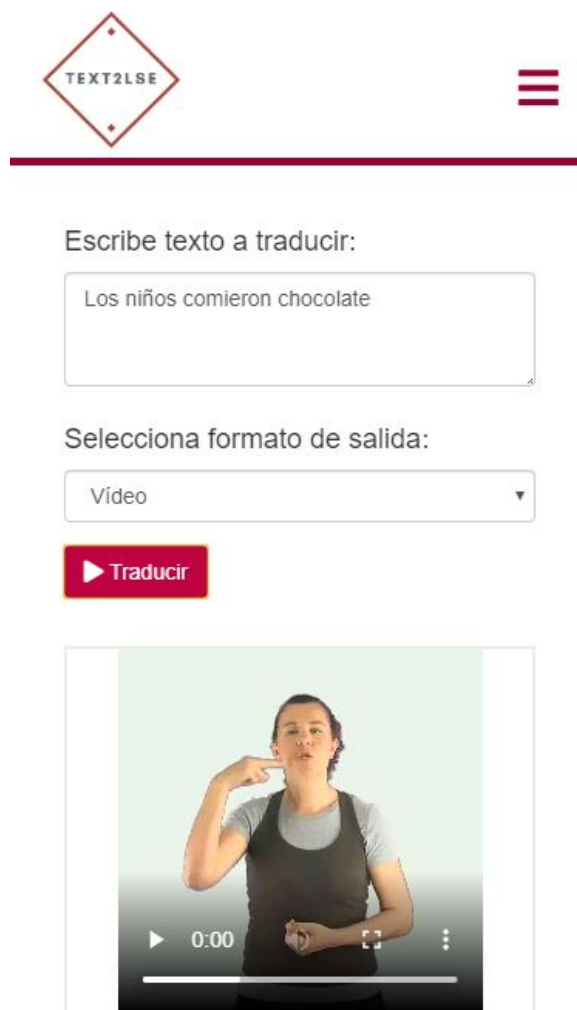


Figura 5.20: Aplicación Web vista desde un dispositivo móvil



## Capítulo 6

# Evaluación

Al terminar la fase de desarrollo, comenzó la fase de evaluación, cuya finalidad era comprobar la corrección del trabajo realizado y la precisión y alcance de las traducciones, para la cual se usarán métricas y se realizará un análisis cualitativo. En este capítulo se muestra el proceso de dicha evaluación para poder determinar qué tipo de oraciones se logra traducir y con qué nivel de fiabilidad. En la Sección 6.1 se explica el diseño de la evaluación realizada, en la Sección 6.2 se presentan los resultados y en la Sección 6.3 se realiza el análisis y discusión de los resultados obtenidos.

### 6.1. Diseño de la evaluación

Hemos enfocado la fase de evaluación en la precisión de la traducción a LSE, dejando fuera, por falta de tiempo, aspectos de usabilidad de la aplicación web desarrollada. Para llevar a cabo la evaluación, se creó un corpus de evaluación compuesto por 5 textos, con un total de 137 oraciones. Todas las frases del corpus fueron traducidas a LSE (en formato imagen y vídeo) y fueron enviadas a un experto en LSE para su evaluación.

A la hora de seleccionar los textos del corpus de evaluación se tuvieron en cuenta las aplicaciones que podría tener nuestra herramienta de traducción de texto a LSE: traducción de subtítulos, traducción de transcripciones de megafonías de estaciones de tren o aeropuertos, etc. Teniendo en cuenta todo esto, hemos seleccionado como textos para la evaluación las instrucciones de seguridad de un avión, un capítulo de una serie infantil, un anuncio de televisión y un trailer de una película, de los que se realizó una transcripción manual para obtener todas las oraciones que contenían y que fueron usadas para la evaluación. También se ha añadido un corpus de oraciones sencillas, ya que los otros corpus cuentan con oraciones muy complejas y nuestro traductor se centra en la traducción de oraciones más simples (con estructuras del tipo SUJETO + VERBO + COMPLEMENTOS). A continuación expli-

camos en detalle los textos seleccionados.

1. **Corpus de oraciones de instrucciones de seguridad de un vuelo**<sup>1</sup>: Este corpus está formado por oraciones complejas que componen los subtítulos de un video real de las instrucciones de seguridad de un avión. Consta de oraciones compuestas con varios verbos y con varios sustantivos, tanto en el sujeto como en el predicado. Está formado por 54 oraciones que podemos observar en el Apéndice A. Algunas de las oraciones que componen este corpus son:

- Les damos la bienvenida a este vuelo de Iberia en nuestro nombre y en el de Madrid.
- Antes de despegar tenemos que darles unas instrucciones de seguridad.
- Es importante que presten atención.

2. **Corpus de oraciones de un capítulo de la serie infantil Pepa Pig**<sup>2</sup>: Corpus compuesto por 34 oraciones simples, en las que se incluyen construcciones específicas del español. En el Apéndice B encontramos las oraciones que componen este corpus. Algunas de estas oraciones son:

- Yo soy Pepa.
- Este es mi hermano pequeño.
- A Pepa le encanta saltar en los charcos.

3. **Corpus de subtítulos del trailer de la película Togo**<sup>3</sup>: Compuesto por 13 oraciones. En este corpus hay oraciones simples con estructura SUJETO + VERBO + OBJETO, y oraciones más complejas que incluyen varios verbos y diversos tipos de complementos. El texto en su totalidad lo podemos encontrar en el Apéndice C. Algunas de estas oraciones son:

- Mi negocio son los perros.
- Solo un hombre y un perro pueden hacer esa carrera.
- Yo siempre pensé que vivía para los trineos.

---

<sup>1</sup><https://www.youtube.com/watch?v=rFohg-g1VCU>

<sup>2</sup><https://www.youtube.com/watch?v=B1rgTUWqWzI>

<sup>3</sup><https://www.youtube.com/watch?v=42ubb5b1s9o>

4. **Corpus de subtítulos de un anuncio de la Covid-19<sup>4</sup>:** Conjunto de 11 oraciones con distintos niveles de complejidad. Hay desde oraciones más simples hasta más complejas con más de un verbo y diversidad de complementos. Algunas de las frases de este corpus cuentan con lenguaje metafórico y oraciones hechas específicas del castellano. El texto que compone este corpus lo encontramos en el Apéndice D. Algunas de las oraciones que lo componen son:

- Quédate en casa.
- Mantén una distancia de dos metros.
- Evita tocar cualquier superficie si es necesario.

5. **Corpus de oraciones simples:** Conjunto de 25 oraciones, desde oraciones simples con estructura SUJETO + VERBO + COMPLEMENTOS, hasta oraciones a las que se le va añadiendo complejidad con diversos complementos (temporalidad, preposiciones, etc) y varios sustantivos dentro del sujeto. Podemos encontrar todas las oraciones que componen este corpus en el Apéndice E. Algunas de estas oraciones son:

- Él bebe agua.
- Mis tíos irán al supermercado en coche.
- Mi hermana y mi tía están de vacaciones en la playa.

## 6.2. Resultados de la Evaluación

En el Apéndice F pueden verse todas las traducciones de las oraciones del corpus de evaluación en formato imagen, cuya traducción es similar en formato vídeo. Hemos decidido clasificar los resultados en tres grupos: oraciones traducidas correctamente, oraciones cuya traducción no es exacta pero se entiende su significado y oraciones erróneas. En la Tabla 6.1 se detallan los resultados obtenidos en la evaluación de cada corpus incluyendo únicamente las oraciones traducidas correctamente. La tabla contiene el número de oraciones que componen el corpus, número de oraciones correctas y el porcentaje de aciertos. A continuación, en la tabla 6.2 se muestra el número de oraciones no traducidas a la perfección, pero cuyo significado se comprende, y el porcentaje que representa. Además, en la tabla 6.3 se hace referencia a la suma de ambos grupos, mostrando el número y porcentaje de oraciones entendibles, que incluye tanto las oraciones traducidas a la perfección como las no exactas.

---

<sup>4</sup>[https://www.youtube.com/watch?v=oSncDP\\_vC3Q](https://www.youtube.com/watch?v=oSncDP_vC3Q)

<b>Corpus</b>	<b>Nº oraciones</b>	<b>Nº aciertos</b>	<b>% acierto</b>
Instrucciones de avión	54	0	0 %
Serie infantil	34	0	0 %
Trailer de película	13	0	0 %
Anuncio	11	0	0 %
Oraciones simples	25	9	36 %

Tabla 6.1: Resultados de la evaluación: traducciones correctas

<b>Corpus</b>	<b>Nº oraciones</b>	<b>Nº no exactas</b>	<b>% no exactas</b>
Instrucciones de avión	54	0	0 %
Serie infantil	34	1	2.94 %
Trailer de película	13	2	15.38 %
Anuncio	11	0	0 %
Oraciones simples	25	6	24 %

Tabla 6.2: Resultados de la evaluación: traducciones no exactas

<b>Corpus</b>	<b>Nº oraciones</b>	<b>Nº entendibles</b>	<b>% entendible</b>
Instrucciones de avión	54	0	0 %
Serie infantil	34	1	2.94 %
Trailer de película	13	2	15.38 %
Anuncio	11	0	0 %
Oraciones simples	25	15	60 %

Tabla 6.3: Resultados de la evaluación: traducciones entendibles

Como podemos observar en estos resultados, el corpus de oraciones sencillas cuenta con un porcentaje de traducciones entendibles bastante alto (60 %), mientras que el resto de corpus cuentan con mayor margen de mejora. Esto es debido a que en estos últimos la complejidad de las oraciones es más elevada, produciendo así una serie de errores que analizaremos en la sección 6.3. En esta sección también veremos que una parte importante de las traducciones erróneas se debe a la falta de imágenes o vídeos LSE, ya que más del 33 % de los signos que se buscan en la biblioteca no se encuentran.

### 6.3. Análisis de los Resultados

En este apartado se han analizado los resultados en las oraciones que no se han traducido a la perfección (traducciones erróneas y traducciones no exactas), agrupándolas según el tipo de error que se ha obtenido. De cada una de estas oraciones únicamente se ha extraído el error más grave, ya que el resto de errores (si los hubiera), podrían haber sido provocados por este. Por ejemplo, la traducción que proporciona Text2LSE de “*Yo bebo agua*” es “*YO BEBO AGUA*”, cuando debería ser “*YO AGUA BEBER*”. Esto se debe a un fallo del analizador, que no reconoce la palabra “*bebo*” como verbo, lo que además provoca que no se encuentre el recurso para ese signo. Por ello, esta frase está únicamente en la categoría “*Fallo del analizador Spacy*”, y no en “*Falta de imágenes o vídeos LSE*”. Para realizar la clasificación se han analizado las traducciones en formato imagen, ya que el resultado es similar al formato vídeo, y se ha revisado tanto los signos como el texto que los acompaña con el fin de determinar de qué tipo de error se trata. Para el resto de casos hemos agrupado los resultados dependiendo del tipo de error obtenido. En la Tabla 6.4 podemos observar el número de fallos de cada tipo y su porcentaje de error.

- **Género y número:** El sistema si no encuentra el signo de un sustantivo busca esa palabra sin morfemas de género y número, y si así encuentra el signo se añade a continuación los signos “*FEMENINO*” y “*PLURAL*” en caso de que el sustantivo fuese femenino y plural. Pero hay ocasiones en las que no haría falta añadirlo. Por ejemplo, en la oración “*y disponen de rampas o balsas de evacuación*” se traduce “*rampas*” como “*RAMPA FEMENINO PLURAL*”, pero en este caso al ser la palabra “*rampa*” femenino no haría falta añadir el signo “*FEMENINO*”.
- **Cambio de significado:** El significado de la oración ha cambiado al realizar la traducción a LSE. Por ejemplo, la oración “*Mi negocio son los perros*” se traduce como “*YO NEGOCIO PERRO PLURAL*”. Sin embargo, en este caso la traducción correcta sería “*YO NEGOCIO PERRO*”, ya que el signo “*PLURAL*” se entiende en referencia a negocio y no a perro, por lo que el significado de la traducción cambia a que su negocio son los perros y otros negocios más.
- **Preposiciones:** La aplicación no tiene en cuenta todas las particularidades de las preposiciones en la LSE. Por ejemplo, al traducir la oración “*Le ganó a todos los demás*” se elimina la preposición “*a*”, obteniendo “*PASADO DEMÁS TODOS GANAR*”. Según las normas de la LSE dicha preposición sí debería de aparecer en la traducción final, ya que eliminarla cambia el sentido de la traducción a “*Todos los demás ganaron*”.



Figura 6.1: Representación del signo “por favor” en LSE <sup>5</sup>

- **Negación:** Text2LSE no ha contemplado el orden de aparición de la negación en una oración. Por eso en oraciones, como por ejemplo, “Él no es un perro de trineo” realiza la traducción “ÉL PERRO NO TRINEO”, cuando debería ser “ÉL PERRO TRINEO NO”, ya que en LSE la negación se coloca al final de la oración.
- **Particularidad LSE:** En LSE existen frases hechas que se dicen con un único signo concreto, por lo que es incorrecto realizar la traducción de la frase hecha palabra por palabra. Por ejemplo, la oración “Por favor” es una oración hecha y en LSE se traduce como un solo signo, como podemos ver en la Figura 6.1.
- **Estructuras compuestas:** En esta primera versión de Text2LSE no se han tenido en cuenta en el desarrollo las estructuras más complejas, como pueden ser verbos compuestos, pasados perfectos u oraciones compuestas con varios verbos. Esto se puede observar en la oración “Pe-pa ha encontrado un charco pequeño” cuya traducción proporcionada por la aplicación es “PASADO PEPA HA CHARCO PEQUEÑO ENCONTRAR”. En este caso, al no tener en cuenta la aparición de un verbo compuesto, la traducción es errónea, ya que se debería eliminar la palabra “HA”.
- **Polisemia:** El sistema no tiene en cuenta la polisemia de las palabras, es decir, que una misma palabra pueda tener varios significados, por lo que en la traducción en estos casos se pueden mostrar signos que correspondan a un significado que no es el adecuado para la frase. Por

<sup>5</sup><http://www.fundacioncnse.org>

ejemplo, en la oración “*Encontraron la cura*” el signo “*CURA*” buscado debe ser el de cura referente a sanación, y no el de “*CURA*” referente a sacerdote.

- **Fallo del analizador Spacy:** Text2LSE ha utilizado el analizador Spacy, el cual no tiene un porcentaje de acierto del 100 %. Por tanto, puede haber ocasiones en que el análisis realizado por Spacy no sea correcto y esto lleve a nuestro sistema a fallar en la traducción. Esto lo podemos comprobar en oraciones como “*Solo es barro*”, en la que Spacy nos dice que “*barro*” es un sustantivo cuando en realidad es un verbo (“*barrer*”), lo que hace que nuestro sistema traduzca la frase como “*SOLO BARRER*”, que es incorrecta.
- **Falta de imágenes o vídeos LSE:** Para realizar las traducciones se han utilizado los recursos LSE de ARASAAC, los cuales no incluyen signos para todas las palabras del castellano. Por ejemplo, al traducir la frase “*El corazón de un superviviente*”, vemos que la traducción que hace nuestro sistema a LSE (“*CORAZÓN SUPERVIVIENTE*”) es correcta, sin embargo, el catálogo de recursos LSE de ARASAAC no incluye el signo para la palabra “*SUPERVIVIENTE*”, por lo que la traducción a vídeo o imagen que realiza nuestra aplicación deja una imagen o vídeo de error para esa palabra, quedando la traducción incompleta.
- **Nombres propios:** En la LS no existen signos para los nombres propios, sino que estos se deletrean. En el desarrollo de la aplicación no se han tenido en cuenta los nombres propios, por lo que Text2LSE muestra una imagen o vídeo de error cuando se encuentra con uno.

Como podemos observar en la Tabla 6.4, las oraciones con un margen de mejora más amplio son las que contienen estructuras compuestas, que suponen un 32 % del total de errores obtenidos. Esto se debe a que en el desarrollo de la aplicación nos hemos centrado en la traducción de oraciones con estructuras simples como primera aproximación al problema planteado. Otro aspecto a mejorar serían las particularidades de la LSE, cuya solución sería introducir reglas concretas en el código para tratar todas estas particularidades de forma correcta. A pesar de estos errores en las traducciones, los resultados de la evaluación han sido muy satisfactorios. Teniendo en cuenta que actualmente no existe ninguna aplicación capaz de traducir castellano a LSE en tiempo real. Text2LSE en un futuro puede ser de gran ayuda para el colectivo de personas sordas, ya que actualmente traduce un número de oraciones considerable, que fácilmente se puede ampliar trabajando en los problemas anteriormente mencionados.

<b>Tipos de fallos</b>	<b>Nº fallos / Total fallos</b>	<b>% fallos</b>
Estructuras compuestas	42 / 129	32.55 %
Particularidad LSE	35 / 129	27.13 %
Fallo del analizador	21 / 129	16.27 %
Falta de imágenes o vídeos LSE	11 / 129	8.52 %
Preposiciones	8 / 129	6.20 %
Nombres propios	4 / 129	3.10 %
Negación	3 / 129	2.32 %
Género y número	2 / 129	1.55 %
Cambio de significado	2 / 129	1.55 %
Polisemia	1 / 129	0.77 %

Tabla 6.4: Oraciones traducidas incorrectamente por cada tipo de error

Como hemos visto, la falta de imágenes y vídeos para muchas palabras del corpus de evaluación hace que el porcentaje de acierto de nuestro sistema sea muy bajo. Por ello, creemos que es interesante realizar también una evaluación de la precisión de la traducción a texto LSE de nuestro sistema. En la Tabla 6.5 se muestran las traducciones perfectas y su porcentaje con respecto al número total de frases por corpus. La Tabla 6.6 de la misma manera refleja las traducciones que se entienden pero no son perfectas, y finalmente en la Tabla 6.7 se pueden observar los resultados en cuanto a traducciones entendibles (suma de perfectas y no exactas) siguiendo este mismo criterio.

<b>Corpus</b>	<b>Nº oraciones</b>	<b>Nº aciertos</b>	<b>% acierto</b>
Instrucciones de avión	54	7	12.96 %
Serie infantil	34	4	11.76 %
Trailer de película	13	1	7.69 %
Anuncio	11	1	9.09 %
Oraciones simples	25	11	44 %

Tabla 6.5: Resultados de la evaluación: traducciones correctas en texto LSE

<b>Corpus</b>	<b>N° oraciones</b>	<b>N° no exactas</b>	<b>% no exactas</b>
Instrucciones de avión	54	14	25.92 %
Serie infantil	34	7	20.58 %
Trailer de película	13	3	23.07 %
Anuncio	11	2	18.18 %
Oraciones simples	25	8	32 %

Tabla 6.6: Resultados de la evaluación: traducciones no exactas en texto LSE

<b>Corpus</b>	<b>N° oraciones</b>	<b>N° entendibles</b>	<b>% entendible</b>
Instrucciones de avión	54	21	38.88 %
Serie infantil	34	11	28.94 %
Trailer de película	13	4	30.76 %
Anuncio	11	3	27.27 %
Oraciones simples	25	19	76 %

Tabla 6.7: Resultados de la evaluación: traducciones entendibles en texto LSE

Como podemos observar en estos resultados, el corpus de oraciones sencillas cuenta con un porcentaje de traducciones entendibles bastante más alto (76 %) que el resto de corpus, cuyos porcentajes rondan el 20 o el 30 %. Esto es debido a que en estos últimos la complejidad y riqueza de vocabulario de las oraciones es más elevada. En comparación con los resultados globales, llegamos a la conclusión de que añadiendo más recursos a la biblioteca de recursos LSE de ARASAAC, los porcentajes de éxito aumentan de manera notable, por lo que merece la pena trabajar en este sentido.

Por otro lado, se han analizado todas las traducciones que cuyo texto LSE es entendible, haciendo un recuento de los signos que se buscan y viendo qué porcentaje de signos no se encuentran en el catálogo de ARASAAC, con el fin de dar una idea de qué parte de fallos son debidos a la falta de signos en el catálogo. En la tabla 6.8 se muestra el número de signos buscados en el catálogo de ARASAAC, el número de signos que no se encuentran y el porcentaje que este representa.

Corpus	Nº signos buscados	Nº signos no encontrados	% error
Instrucciones de avión	101	53	52.47 %
Serie infantil	44	17	38.63 %
Trailer de película	14	5	35.71 %
Anuncio	14	5	35.71 %
Oraciones simples	86	7	8.13 %

Tabla 6.8: Resultados de la evaluación: signos buscados y signos no encontrados

Como podemos observar en la Tabla 6.8, la cantidad de signos que se buscan pero no están en el catálogo representan un porcentaje considerable del total, llegando incluso a ser más de la mitad en el caso del corpus de instrucciones de avión y superando el 33 % en el conjunto de todos los corpus. Esto quiere decir que oraciones que Text2LSE es capaz de traducir correctamente acaban siendo erróneas por culpa de un catálogo insuficiente, lo que refuerza la conclusión de que merece la pena trabajar en añadir más recursos al catálogo actual.

## Capítulo 7

# Trabajo individual

### 7.1. Sara Vegas Cañas

Lo primero de todo fue una investigación profunda de la discapacidad auditiva y de la Lengua de Signos Española por parte de los tres integrantes de este proyecto, de esta manera todos nos concienciamos de las dificultades que tienen todas las personas sordas para comunicarse en su día a día. Realicé una investigación de las herramientas que tienen estas personas a su disposición, tanto posibles traductores a LSE, como bancos de vídeos e imágenes. Encontré una variedad de aplicaciones, las cuales probé para poder ver hasta donde llegaban las soluciones actuales para el problema de comunicación de las personas sordas.

Cuando fijamos los objetivos de nuestro TFG, empezamos con la investigación de las posibles tecnologías que podríamos usar en el proyecto. Mi investigación en este tema se centró en servicios web, sobre todo de tipo REST, Python y en el Procesamiento del Lenguaje Natural, probando diferentes librerías, como Spacy y NLTK, para poder elegir la que más se ajustase a nuestras necesidades.

Una vez que terminó la primera parte de investigación, nos reunimos para juntar todas nuestras conclusiones. Con éstas comenzamos a desarrollar la motivación y objetivos de la memoria, junto con todos los apartados referentes a la Lengua de Signos, encargándome sobre todo de las Reglas de la LSE y de los bancos de vídeos e imágenes LSE.

A continuación, comenzamos a crear una primera base de la aplicación. Lo primero fue realizar la preparación del entorno local en Debian 10, utilizando Nginx para alojar la aplicación web, y Flask para ejecutar la API, preparando el modo debug para el posterior proceso de desarrollo de código. Construimos una web básica, en la cual podíamos escribir un texto y enviar-

lo a la API a través de JavaScript, mediante jQuery y Ajax, y una primera estructura de la API REST en Python. En este apartado me encargue junto con mi compañero Alejandro del desarrollo de una funcionalidad javascript capaz de lanzar una petición a la API con el texto y que esta nos devolviese un vídeo mp4. Más adelante decidimos cambiar las llamadas AJAX por llamadas Fetch.

Respecto a los servicios web, me dediqué a estructurar la función que devuelve un vídeo con varios signos y de desarrollar las llamadas a las funciones que devuelven diferente tipo de información en formato JSON.

Respecto a PLN, en un principio me centré en desarrollar un algoritmo recursivo que recorriera un árbol de dependencias desde el verbo a partir de una frase, complementándolo con el tratamiento de artículos y colocando las palabras en orden de LSE. Junto con mis compañeros realizamos una estructura y desarrollo de la traducción con dos partes bien diferenciadas, el orden y la morfología.

Para poder observar de manera sencilla el funcionamiento de estos servicios, decidimos realizar un nuevo diseño de la página web que fuera totalmente responsive, en el cual participamos los tres miembros del proyecto.

Además de todo el trabajo técnico, participé en la redacción de la memoria. Me ocupé junto con mis compañeros de los capítulos 1 y 2, que desarrollamos de forma conjunta. En el capítulo 3 me encargué de la redacción de la sección de Google Collaboratory. En cuanto al capítulo 4, donde explicamos la metodología de trabajo seguida en este TFG, desarrollé las secciones de Trello y Reuniones. En el capítulo 5 me ocupé de la introducción, la sección de arquitectura y las de los tres servicios que componen la traducción a LSE en formato vídeo. Finalmente, junto con mi compañero Alejandro, desarrollé el capítulo de evaluación, y además me encargué de la creación de los apéndices. Por otro lado, trabajé en la corrección de todos estos apartados a medida que nuestros tutores nos fueron dando feedback.

## 7.2. Alejandro Torralbo Fuentes

Al comienzo del proyecto, los tres integrantes del grupo realizamos una fase de investigación, en la que adquirimos conocimientos sobre la discapacidad auditiva y la Lengua de Signos, y nos pusimos al día en los problemas a los que se enfrenta el colectivo de personas sordas para poder orientar nuestro proyecto a solucionar dichos problemas. A continuación, realizamos una búsqueda de recursos que pudiéramos utilizar en el desarrollo de nuestra

aplicación, encontrando bancos de imágenes y videos de Lengua de Signos Española como la de ARASAAC, entre otras herramientas.

Una vez recopilada la información necesaria, comenzamos con la instalación y preparación del entorno de desarrollo local: una máquina virtual con Debian 10, en la que instalamos un servidor para alojar el cliente con Nginx, y una API desarrollada en Python y ejecutada mediante Flask. Me encargué de configurar el entorno Flask, activando el modo debug para poder empezar a depurar el código de la API, e instalé la herramienta Postman para la realización de pruebas y el controlador de versiones Git, para trabajar en el código de forma conjunta con el resto de compañeros.

A partir de este punto, comenzamos con el desarrollo de la aplicación. Me encargué de desarrollar una web responsive con HTML, CSS y Javascript, alojada en el servidor Nginx. Junto con mi compañera Sara, conseguimos implementar una función en Ajax capaz de realizar una petición POST a la API, enviando el texto introducido por el usuario y recibiendo un vídeo en formato mp4 para mostrarlo en la web. Posteriormente mi compañera Sara y yo cambiamos estas llamadas Ajax por llamadas Fetch. Además, me encargué de que la API fuera capaz de juntar varios vídeos en función del texto recibido, y prepararlo para enviarlo a la página web. Posteriormente los tres integrantes decidimos estructurar la API y dividir esta funcionalidad en varias funciones diferentes, de modo que el usuario interesado en utilizar nuestros microservicios cuente con varias opciones de llamada. Mis compañeros Miguel y Sara se encargaron de dividir esta función. Una vez estructurado todo, los tres compañeros conjuntamente participamos en el desarrollo y pruebas de los microservicios de PLN.

Junto con el trabajo técnico, también participé en la redacción y corrección de la memoria. Los tres compañeros comenzamos desarrollando conjuntamente el capítulo 1 y 2. Además, desarrollé el apartado de Flask del capítulo 3 y metodologías de trabajo en el capítulo 4. En cuanto al capítulo 5, escribí la sección Front-End, la introducción de la sección Back-End y los servicios de traducción LSE en imágenes. Finalmente, junto con mi compañera Sara, redacté el capítulo de evaluación. A medida que fueron llegando las correcciones de nuestros tutores, me encargué de la corrección del capítulo 1 y parte del capítulo 2, junto con todas las secciones que yo mismo redacté en el resto de capítulos.

### 7.3. Miguel Rodríguez Cuesta

El comienzo del proyecto fue un proceso sobre todo de investigación donde cada miembro del grupo buscó información de distinta variedad relacionada con el mundo de la Lengua de Signos, para después realizar una puesta en común y empezar a redactar la memoria. Personalmente durante este período he ido recopilando información sobre qué es la LSE, quién la usa o como es su estructura. También investigué sobre la comunidad sorda y sus problemas de adaptación en la sociedad actual y discapacidad auditiva.

La redacción de la memoria ha ido dividiéndose entre los tres integrantes del equipo. Al principio se desarrolló conjuntamente el capítulo 1 y 2, ya que teníamos que poner en común lo investigado de la LSE y redactar las motivaciones que nos llevaron a hacer este proyecto. Después se dividió la memoria en apartados y cada miembro se encargaba de redactar su parte, además de incorporar las correcciones de los tutores.

Una vez bien definido el reparto de la memoria, en el capítulo 2 me encargué de redactar el apartado de las aplicaciones de traducción de texto a LSE y el Procesamiento de Lenguaje Natural mientras que en el capítulo 3 y 4 mi aportación ha sido la redacción de los apartados Servicios Web y control de versiones respectivamente. De cara a la parte final de la memoria he ayudado a redactar el capítulo 5 junto con mis compañeros, escribiendo la sección de los Servicios web para Traducción de texto en castellano a texto en LSE y traducción de texto a LSE (imagen) y también me he encargado de redactar el capítulo 8 teniendo en cuenta los consejos de mis compañeros, ya que al tratarse de las conclusiones de un proyecto es necesario poner en común ideas de los tres miembros.

En cuanto al desarrollo de la aplicación, al principio se hizo un trabajo de investigación de manera individual para aprender como funcionan las herramientas que se iban a utilizar en el proyecto. Una vez investigado se configuró de forma conjunta el control de versiones y el servidor de la UCM para poder subir la versión final del proyecto, además de configurar cada uno su entorno virtual para poder trabajar de forma independiente.

Una vez hecho el trabajo de investigación y configuración, mi aportación ha sido desarrollar el PLN junto con mis dos compañeros y personalmente crear algunos servicios, como los Servicios web de traducción de una palabra a LSE, tanto en vídeo como en imagen, además de ayudar en otros servicios. También ha sido mi responsabilidad la salida en caso de error de la traducción, sustituyendo el modal de error anteriormente implementado por los tres integrantes del equipo, por una imagen o vídeo indicando que no existe ese

recurso en nuestra librería.

Para el desarrollo de la Web responsive, he participado de manera conjunta y activa con mis compañeros en el diseño de la aplicación para que sea totalmente adaptable a todos los dispositivos.



## Capítulo 8

# Conclusiones y Trabajo Futuro

En este capítulo se presentarán las conclusiones finales del TFG, así como las posibles mejoras que se podrían realizar como trabajo futuro si se continúa con el proyecto.

### 8.1. Conclusiones

En la sociedad actual la comunicación es esencial en la vida diaria de las personas, ya que permite transmitir información e intercambiar opiniones y sentimientos unos con otros, lo que es algo imprescindible como seres humanos. A día de hoy la sociedad no está completamente adaptada a las necesidades de todos, pues hay muchas personas que tienen distintas discapacidades: cognitivas, físicas, visuales, auditivas... Algunas de las personas con estas discapacidades a veces pueden sentirse un poco apartadas del resto por no tener acceso completo a la información, como el caso de una persona con discapacidad auditiva a la hora de escuchar mensajes de megafonía en una estación de tren.

Las personas con discapacidad auditiva disponen de una gran cantidad de posibilidades de comunicarse con los demás, ya que la gran mayoría sabe leer y escribir. Además, tienen una lengua propia, la Lengua de Signos, que les permite expresar emociones y sentimientos a la hora de comunicarse. Para ayudar a las personas con discapacidad auditiva, vimos necesario desarrollar una herramienta que fuera capaz de traducir texto a Lengua de Signos Española (LSE), ya que podría tener muchas utilidades y facilitar el día a día de las personas de este colectivo. Algunas de estas utilidades podrían ser el aprendizaje de la LSE, traducción de subtítulos de películas o un texto informativo como las instrucciones de un avión a LSE.

El objetivo principal de este TFG era desarrollar una aplicación capaz de traducir cualquier texto en castellano a LSE en formato vídeo, imagen o



Figura 8.1: Oración con tiempo verbal y posesivos en LSE traducida por Text2LSE

texto LSE, y para ello se ha desarrollado Text2LSE.

Text2LSE actualmente es capaz de traducir frases simples que siguen la estructura TIEMPO + SUJETO + OBJETO + VERBO, además de tener en cuenta un gran número de reglas gramaticales de la LSE, siendo alguna de ellas:

- **Añadir el tiempo verbal:** Detecta el tiempo verbal de cada frase y añade el signo correspondiente ya sea “FUTURO” o “PASADO” como se puede ver en la Figura 8.1.
- **Cambiar los determinantes posesivos:** Los determinantes que indican posesión los sustituye por pronombres personales, como se puede ver también en la Figura 8.2.
- **Género y Número:** Detecta el género y número de cada sustantivo e indica el signo de “PLURAL” en caso del número o “FEMENINO” en caso del género, como se puede ver en la Figura 8.3.
- **Adjetivos:** Detecta todos los adjetivos de la oración y los pasa a masculino singular, como por ejemplo la Figura 8.4.

Aunque queda mucho trabajo por hacer, nos gustaría destacar que al comienzo de este TFG se hizo un análisis de distintas aplicaciones similares



Figura 8.2: Oración con tiempo verbal y posesivos en LSE traducida por Text2LSE



Figura 8.3: Oración con género y número en LSE traducida por Text2LSE



Figura 8.4: Oración con adjetivos en LSE traducida por Text2LSE

en el mercado, pero no encontramos ninguna aplicación que realizara traducciones de texto a LSE en tiempo real, por lo que podemos afirmar que aún no siendo perfecto el resultado, este trabajo es una buena base para la traducción de lenguaje natural a Lengua de Signos Española.

Otro aspecto importante del trabajo que hemos desarrollado ha sido crear una API pública <sup>1</sup> con los servicios web desarrollados, de tal forma que sea fácil de utilizar en otros proyectos.

Aparte de la traducción, el propósito era crear también una aplicación con una interfaz intuitiva y accesible para todos desde cualquier dispositivo. Se tenía la intención de hacer una evaluación de la interfaz con usuarios reales, pero debido a la Covid-19 no ha sido posible realizar dicha evaluación. Aún así, teniendo en cuenta las opiniones de los tutores en este sentido y siguiendo sus consejos de diseño, creemos que la interfaz es sencilla e intuitiva y cumplimos el objetivo marcado.

El desarrollo de Text2LSE nos ha permitido aplicar muchos conocimientos adquiridos durante el Grado de Ingeniería Informática en un proyecto grande y con una utilidad social, no solo para el ámbito académico. Algunas asignaturas que han sido muy importantes para poder realizar este proyecto han sido:

<sup>1</sup><https://holstein.fdi.ucm.es/tfg-text2lse/>

- **Aplicaciones Web:** Nos ha servido mucho a la hora de realizar el Front-end ya que nos aportó grandes conocimientos sobre el desarrollo web con HTML, CSS y Javascript que han sido muy útiles para este TFG.
- **Tecnología de la Programación:** Gracias a esta asignatura aprendimos a realizar un proyecto en Java. Aunque el TFG esté hecho en Python, lo importante de esta asignatura es que nos aportó un gran conocimiento a la hora de programar, por lo que aprender un nuevo lenguaje como Python no ha sido tan complicado.
- **Estructura de Datos Algorítmicos:** De esta asignatura lo que más nos ha ayudado ha sido saber como utilizar la recursión a la hora de programar, ya que ha sido un pilar fundamental en el PLN.
- **Ética, Legislación y Profesión** Es una asignatura que nos ha enseñado como usar las licencias con las que proteger nuestro trabajo, así como aprender a utilizar el software libre y referenciar bien toda la información utilizada en el proyecto.

Además, el proyecto nos ha permitido desarrollar muchos conocimientos nuevos que no se han cursado en la carrera, como por ejemplo: Servicios Web, Proxys, PLN, Python...

## 8.2. Trabajo Futuro

Con el objetivo de mejorar algunas funcionalidades de la aplicación de Text2LSE de cara a obtener una aplicación más completa y con una mayor cobertura, pensamos que se podría marcar como trabajo futuro lo siguiente:

- **Traducción de nombres propios:** En la LSE los nombres propios se pueden traducir usando el signo de cada letra hasta completar el nombre, o utilizar un signo propio que defina a la persona. Actualmente la aplicación detecta el nombre propio pero no encuentra traducción, ya que no hay imágenes o vídeos para ese nombre, como se puede ver en la Figura 8.5. La solución sería utilizar el signo de cada letra del nombre propio como se puede ver en la Figura 8.6
- **Mejorar el Género y Número:** Como se indica en la sección 6.3, una mejora sería detectar cuando añadir el signo que indica Femenino o el signo que indica el Plural, ya que hay veces que no es necesario añadirlo, por ejemplo en la palabra “rampas” cuya traducción sería el signo “*RAMPAS*” (Figura 8.7). Sin embargo, la aplicación lo traduce como “RAMPAS + FEMENINO + PLURAL”, como se puede ver en la Figura 8.8.



Figura 8.5: Ejemplo de mala traducción de un nombre propio realizada por Text2LSE



Figura 8.6: Ejemplo de buena traducción de un nombre propio realizada por Text2LSE



Figura 8.7: Traducción correcta de la palabra rampas realizada por Text2LSE



Figura 8.8: Traducción incorrecta de la palabra rampas realizada por Text2LSE



Figura 8.9: Ejemplo de mala traducción de tiempos compuestos realizada por Text2LSE

- Añadir la negación:** Actualmente el sistema de reglas no contempla la negación de las oraciones, por lo que sería un gran avance añadir una regla que permita traducir bien una oración negativa en LSE.
- Reconocimiento de tiempos compuestos:** La aplicación actualmente no es capaz de reconocer los tiempos compuestos, como pueden ser *“han comido”* o *“estaban comiendo”*, por lo que sería importante ampliar la parte del PLN de Text2LSE para que este tipo de verbos se traduzcan de manera correcta. En la Figura 8.9 se puede ver que nuestra aplicación intenta traducir el verbo haber y no encuentra resultado. La opción correcta sería la de la Figura 8.10.
- Tratamiento de frases hechas:** Contemplar la identificación de frases hechas que tienen un signo propio y no requiere una traducción palabra a palabra. Por ejemplo la frase *“Secar el pelo”* se traduce con un solo signo, como se puede ver en la Figura 8.11. Sin embargo, nuestra aplicación realiza la traducción como si fuera una oración normal (Ver Figura 8.12).



Figura 8.10: Ejemplo de buena traducción de tiempos compuestos realizada por Text2LSE



Figura 8.11: Signo en LSE para la frase hecha “Secar el pelo”



Figura 8.12: Signo en LSE para la frase hecha “Secar el pelo” realizado por Text2LSE

- **Traducción de oraciones compuestas:** La versión actual de Text2LSE permite traducir frases simples, pero no traduce frases compuestas, como por ejemplo “*Los niños fueron al parque para jugar con sus amigos*”. En esta oración se puede apreciar que hay dos oraciones más simples bien distinguidas: “*Los niños fueron al parque*” y “*(Los niños) jugar con sus amigos*”. Una posible mejora sería realizar un conjunto de reglas que permitan traducir estas frases compuestas.
- **Mejora en el PLN del Sistema:** El PLN utilizado en el proyecto ha sido mediante un sistema de reglas utilizando la herramienta Spacy. Este sistema no es capaz de traducir frases que no estén contempladas en las reglas previamente programadas. Un cambio interesante y muy importante sería cambiar el sistema basado en reglas por uno basado en aprendizaje automático, que sea capaz de aprender por sí mismo a traducir cualquier tipo de frase a base de entrenamiento.



## Chapter 8

# Conclusions and Future Work

In this chapter, the final conclusions of the project are presented as well as possible improvements that could be applied as future work.

### 8.1. Conclusions

In today's society, communication is essential in people daily life, since it allows for conveying information and exchanging opinions and feelings with each other, which is something that is essential as human beings. Nowadays, society is not completely adapted to everyone's needs, since there are many people who have different disabilities: cognitive, physical, visual, auditory... People with disabilities may sometimes feel a little isolated due to a lack of full access to information, as in the case of a person with a hearing impairment when listening to public address messages at a train station.

Hearing impaired people have many possibilities to communicate with others since the vast majority can read and write, but they also have a language of their own that is the Sign Language, which allows them to express emotions and feelings when communicating. In order to help hearing impaired people, we saw the need to develop a tool that would be capable of translating text into the Spanish Sign Language (SSL), since it could have many uses and facilitate day-to-day life to people from this group. Some of these utilities could be learning about SSL, translating to SSL subtitles, informative text, such as an airplane safety guide to SSL...

The main goal of this project is to develop an application capable of translating any text written in spanish to SSL in video, image or SSL text format. Text2LSE tool has been developed for this purpose.

Tex2LSE is able to translating simple sentences with the following structure: TIME + SUBJECT + OBJECT + VERB. In addition, the translation



Figure 8.1: Verbal tense and possessive sentence in SSL translated by Text2LSE

carried out by our system allows incorporating a large number of Sign Language rules.. Some of this rules are:

- **Additional verbal tenses:** The tool detects the verbal tense of each sentence and adds the corresponding sign for both future (“FUTURO”) or past (“PASADO”), as it can be seen in Figure 8.1.
- **Changes in possessive determinants:** The determinants that indicate possession are replaced by personal pronouns, as it can be seen in Figure 8.2.
- **Gender and number:** The tool detects the gender and number of each noun and shows the signs “PLURAL” in the case of the number or woman (“MUJER”) in the case of gender, as it can be seen in Figure 8.3.
- **Adjectives:** It detects every adjective in the sentence and turns them to singular masculine, such as Figure 8.4.

Although there is a lot of work to be done. We would like to highlight the fact that at the beginning of this project an analysis was made to similar applications on the market, but we did not find any application that carried out text translations to SSL in real time. That’s by we can say that, though the result is not perfect, this work provides a good basis for the translation



Figure 8.2: Verbal tense and possessive sentence in SSL translated by Text2LSE



Figure 8.3: Sentence with gender and number in SSL translated by Text2LSE



Figure 8.4: Sentence with adjectives in SSL translated by Text2LSE

from natural language to Spanish Sign Language.

Another important aspect of our work is the creation of a public API<sup>1</sup> with the developed web services, in such a way that it is easy to use them in other projects.

Apart from translation, the purpose was also to create an application with an intuitive interface accessible to everyone from any device. The initial plan was to carry out an evaluation of the interface with real users, but due to the Covid-19 crisis such an evaluation was not possible. Even so, taking into account the opinion of our tutors and following their design advice, we believe that the interface is simple and intuitive and we meet our goal.

The development of Text2LSE has allowed us to apply a great amount of knowledge acquired during the Computer Engineering Degree in a large project with a social utility, not only in the academic field. Some subjects that have been very important to carry out this project are:

- **Web applications:** This subject helped us when doing the Front-end, since it provided us with great knowledge about web development with HTML, CSS and Javascript, which was very useful for this project.
- **Programming Technology:** Thanks to this subject we learned to program a project in Java. Although the project is programmed in

<sup>1</sup><https://holstein.fdi.ucm.es/tfg-text2lse/>



Figure 8.5: Example of bad translation of a proper name made by Text2LSE

Python, the important thing about this course is that it provided us with great knowledge when programming, so learning a new language like Python was not very complicated.

- **Algorithmic Data Structure:** What helped us most about this subject was the recursion we learned in it, since it has been essential to develop the NLP part of the project.
- **Ethics, Legislation and Profession:** This is a subject that taught us how to use the licenses with which to protect our work, as well as learning to use free software and to correctly reference all the information used in the project.

Furthermore, this project has allowed us to acquire new knowledge that was not taught in the degree, such as: Web Services, Proxies, NLP, Python...

## 8.2. Future Work

To improve the accuracy of the translations provided by Text2LSE, the following aspects should be addressed as future work:

- **Proper names translation:** In the SSL, proper names can be translated using either the sign of each letter to complete the name or use a proper sign that defines the person. Currently, the application detects the proper name but cannot find a translation, since there are no images or videos for that name, as it can be seen in Figure 8.5. The solution would be to use the sign of each letter of the proper name, as it can be seen in Figure 8.6
- **Improving Gender and Number:** As stated in section 6.3, the detection of when to add the sings that indicate feminine or plural



Figure 8.6: Example of a good translation of a proper name made by Text2LSE



Figure 8.7: Correct translation of the word “rampas” by Text2LSE

would be an improvement, since sometimes it is not necessary to add then, for example in the word “rampas” whose translation would be the sign “*RAMPAS*” (Figure 8.7). However, the application translates it as “*RAMPAS + FEMENINO + PLURAL*”, as it can be seen in Figure 8.8

- **Add negation processing:** Currently, the rule system does not contemplate negative sentences, so it would be a great advance to add a rule that allows a negative sentence to be correctly translated in SSL.
- **Recognition of compound verbal tenses:** The application is currently not capable of recognizing compound tenses, such as “han comido” or “estaban comiendo”, so it would be important to expand the NLP part of Text2LSE so that these types of verbs will be correctly translated. In Figure 8.9 it can be seen that our application tries to translate the verb “to have” and finds no result. The correct option would be the one in Figure 8.10
- **Sentences that do not require NLP:** It contemplates the identi-



Figure 8.8: Incorrect translation of the word “rampas” by Text2LSE



Figure 8.9: Example of bad translation of compound times made by Text2LSE



Figure 8.10: Example of good translation of compound times made by Text2LSE



Figure 8.11: Sign in SSL for the phrase “Secar el pelo”



Figure 8.12: Sign in SSL for the phrase “Secar el pelo” made by Text2LSE

fication of phrases made that have their own sign and do not require a word-by-word translation. For example, the sentence “Secar el pelo” is translated with a single sign, as it can be seen in Figure 8.11. However, our application performs the translation as if it were a normal sentence (See Figure 8.12)

- **Translation of compound sentences:** The current version of Text2LSE has been implemented to translate simple sentences not more complex sentences such as “Los niños fueron al parque para jugar con sus amigos”. The example sentence is made up of two simple phrases: “Los niños fueron al parque” and “(Los niños) jugar con sus amigos”. Developing a set of rules to translate compound sentences would be an improvement.

- **Improve the NLP system incorporating machine learning:** An improvement would be to change the current rule-based NLP system to one that uses machine learning.



## Apéndice A

# Corpus de oraciones de instrucciones de seguridad de un vuelo

1. Hola, les damos la bienvenida a este vuelo de Iberia en nuestro nombre y en el de Madrid.
2. Antes de despegar tenemos que darles unas instrucciones de seguridad.
3. Es importante que presten atención.
4. Durante el despegue y aterrizaje.
5. Los dispositivos electrónicos deberán permanecer desenchufados y en modo avión.
6. Después podrán utilizarlos durante todo el vuelo.
7. Excepto cuando la tripulación les pida que los apaguen.
8. Recuerden que este avión dispone de conexión Wifi.
9. Y que podrán conectarse a Internet cuando se lo comuniquemos.
10. Si han traído equipaje de mano.
11. Por favor, colóquelo en los compartimentos situados encima de sus butacas o debajo de sus asientos delanteros.
12. Dejando despejados los pasillos y salidas de emergencia.
13. Durante el vuelo podrán ponerse cómodos.
14. Pero durante el despegue y aterrizaje.

15. Por favor, pongan sus asientos en posición vertical.
16. Y mantengan su mesa plegada.
17. Por su seguridad.
18. Les recomendamos que mantengan su cinturón abrochado y visible durante todo el vuelo.
19. Y siempre que la señal luminosa lo indique.
20. Para abrocharlo.
21. Inserte la trabilla en su enganche correspondiente.
22. Para soltarlo.
23. Simplemente levanten la lengüeta del enganche.
24. Este avión cuenta con ocho puertas y dos ventanas de salida, 4 puertas a cada lado del avión y una ventana sobre cada ala.
25. Todas están señalizadas con la palabra EXIT.
26. Y disponen de rampas o balsas de evacuación.
27. Debajo de sus asientos encontrarán un chaleco salvavidas.
28. Sáquenlo de la bolsa.
29. Y para ponérselo.
30. Introduzcan la cabeza por la abertura.
31. Y pasen la cinta de atado por detrás de su cintura.
32. Enganchando la fijación de la hebilla de un extremo a otro.
33. Y tirando de la cinta para ajustárselo.
34. Para inflarlo.
35. Solo tienen que tirar con fuerza del tirador del plástico rojo.
36. O soplar por el tubo.
37. Y recuerden.
38. Nunca deben inflar el chaleco dentro del avión.
39. En caso de despresurización de la cabina.
40. Se abrirá automáticamente un compartimento sobre sus asientos.

41. Que contiene máscaras de oxígeno.
42. Tire de la suya.
43. Colóquela sobre su nariz y boca.
44. Y respire con normalidad.
45. Después preste ayuda a quién pueda depender de usted.
46. Recuerden.
47. Que para evitar poner en peligro la seguridad de este vuelo.
48. No está permitido fumar.
49. O usar dispositivos de liberación de nicotina en ningún caso.
50. Y no olviden.
51. Que tienen más información en las instrucciones de seguridad.
52. Que encontrarán en la bolsa delantera de sus asientos.
53. Muchas gracias por su atención.
54. Esperamos que disfruten de un feliz vuelo.



## Apéndice B

# Corpus de oraciones de un capítulo de una serie infantil

1. Yo soy Pepa.
2. Este es mi hermano pequeño.
3. Esta es mamá Pig.
4. Este es papá Pig.
5. Hoy está lloviendo.
6. Así que Pepa no puede jugar fuera.
7. Ha dejado de llover.
8. ¿Podemos salir a jugar?
9. Salid un rato.
10. A Pepa le encanta saltar en los charcos.
11. Me encanta saltar en los charcos.
12. Si vas a pisar charcos debes ponerte las botas de agua.
13. Lo siento.
14. A George también le gusta saltar en los charcos.
15. Si vas a saltar en los charcos, debes ponerte tus botas.
16. A Pepa le gusta cuidar de su hermano pequeño.
17. Vamos a buscar más charcos.

18. Pepa y George se lo están pasando muy bien.
19. Pepa ha encontrado un charco pequeño.
20. Ese charco sí que es grande.
21. George quiere ser el primero en saltar en el gran charco.
22. Tengo que ver si es seguro para ti.
23. Parece seguro para tí.
24. Solo es barro.
25. Vamos a enseñárselo a Papi.
26. Adivina qué hemos estado haciendo.
27. Déjame pensar.
28. Habéis estado saltando en los charcos de barro.
29. Hemos saltado en los charcos de barro.
30. Mirad cómo os habéis puesto.
31. No pasa nada.
32. Sólo es barro.
33. Voy a limpiaros antes de que os vea mamá.
34. Podemos jugar todos juntos.
35. Pepa y George llevan sus botas de agua.

## Apéndice C

# Corpus de subtítulos de un trailer de una película

1. Mi negocio son los perros.
2. El corazón de un superviviente.
3. Le ganó a todos los demás.
4. Él no es un perro de trineo.
5. Es un ganador.
6. Lo que tienen nuestros niños no es una epidemia, es una sentencia de muerte.
7. Encontraron la cura.
8. Solo un hombre y un perro pueden hacer esa carrera.
9. Tiene 12 años.
10. Es demasiado viejo.
11. Pienso que no los vamos a poder encontrar después de deshielo.
12. Es el momento de descubrir quienes somos.
13. Yo siempre pensé que vivía para los trineos.



## Apéndice D

# Corpus de subtítulos de un anuncio

1. Quédate en casa.
2. Mantén una distancia de dos metros.
3. No te toques la cara.
4. Lávate las manos con frecuencia.
5. Evita tocar cualquier superficie si es necesario.
6. No saludes dando la mano.
7. No des besos ni abrazos.
8. Puedes saludar con la mirada o de palabra.
9. Mantén tu teléfono móvil limpio.
10. Si tienes síntomas.
11. Quédate en casa y aíslate en tu habitación.



## Apéndice E

### Corpus de frases simples

1. Él bebe agua.
2. Yo soy morena.
3. Ellos parecen contentos.
4. María compra galletas.
5. Yo bebo agua.
6. Yo como carne.
7. Ellos compraron aceite.
8. Él venderá refrescos
9. Ella ganó la carrera.
10. Yo haré la cama.
11. Mi tía y mi hermana beberán agua.
12. Ella y él comieron arroz.
13. María y Javier nadan.
14. Mi hermana y mi tía están de vacaciones en la playa.
15. Los niños comerán chocolate negro.
16. Las niñas jugarán en el parque.
17. Mis tíos irán al supermercado en coche.
18. Los niños jugarán con la pelota.
19. Los ganadores recibirán un trofeo de oro.

20. Él pasa ante la casa todos los días.
21. Él pasea ante la casa por la mañana.
22. Mañana yo iré al médico.
23. Él comió mucho en la fiesta.
24. Él se comió la hamburguesa aquí.
25. Mi tía conduce muy despacio.

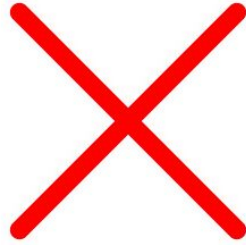
## Apéndice F

# Traducción de los corpus mediante Text2LSE

### F.1. Corpus de oraciones de instrucciones de seguridad de un vuelo

1. Hola, les damos la bienvenida a este vuelo de Iberia en nuestro nombre y en el de Madrid.



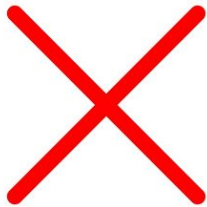


MADRID

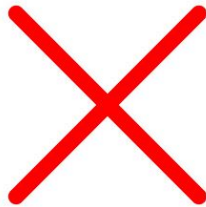


DAR

2. Antes de despegar tenemos que darles unas instrucciones de seguridad.



ANTES



DARLES



QUE



INSTRUCCIONES



SEGURIDAD



DESPEGAR

3. Es importante que presten atención.



IMPORTANTE



QUE



ATENCIÓN



PRESTAR

4. Durante el despegue y aterrizaje.



5. Los dispositivos electrónicos deberán permanecer desenchufados y en modo avión.



6. Después podrán utilizarlos durante todo el vuelo.



7. Excepto cuando la tripulación les pida que los apaguen.



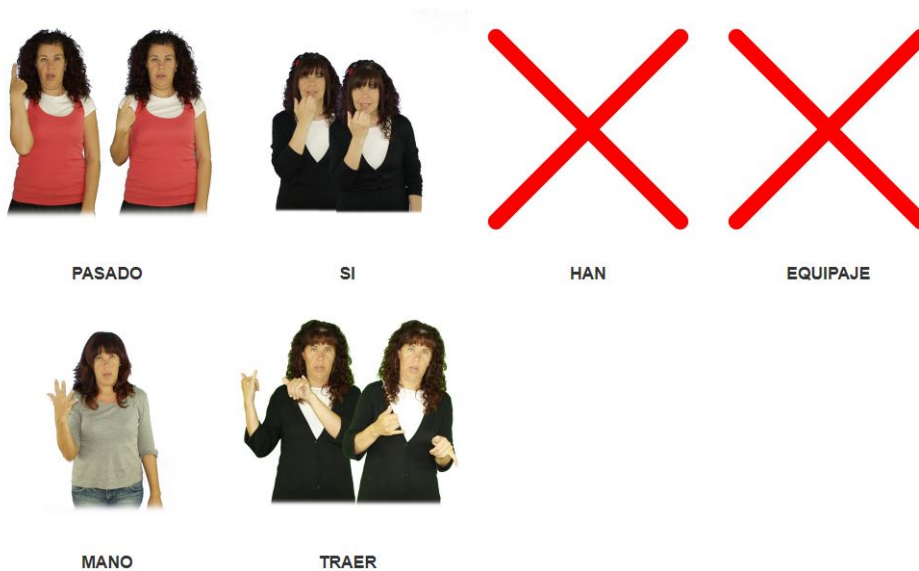
8. Recuerden que este avión dispone de conexión Wifi.



9. Y que podrán conectarse a Internet cuando se lo comunicemos.



10. Si han traído equipaje de mano.



11. Por favor, colóquelo en los compartimentos situados encima de sus butacas o debajo de sus asientos delanteros.





12. Dejando despejados los pasillos y salidas de emergencia.



13. Durante el vuelo podrán ponerse cómodos.



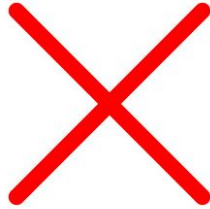
14. Pero durante el despegue y aterrizaje.



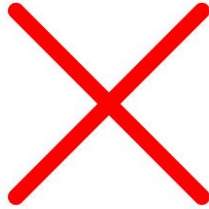
15. Por favor, pongan sus asientos en posición vertical.



POR



FAVOR



ASIENTOS



ELLOS



POSICIÓN



VERTICAL



PONER

16. Y mantengan su mesa plegada.



MESA



ÉL



PLEGAR

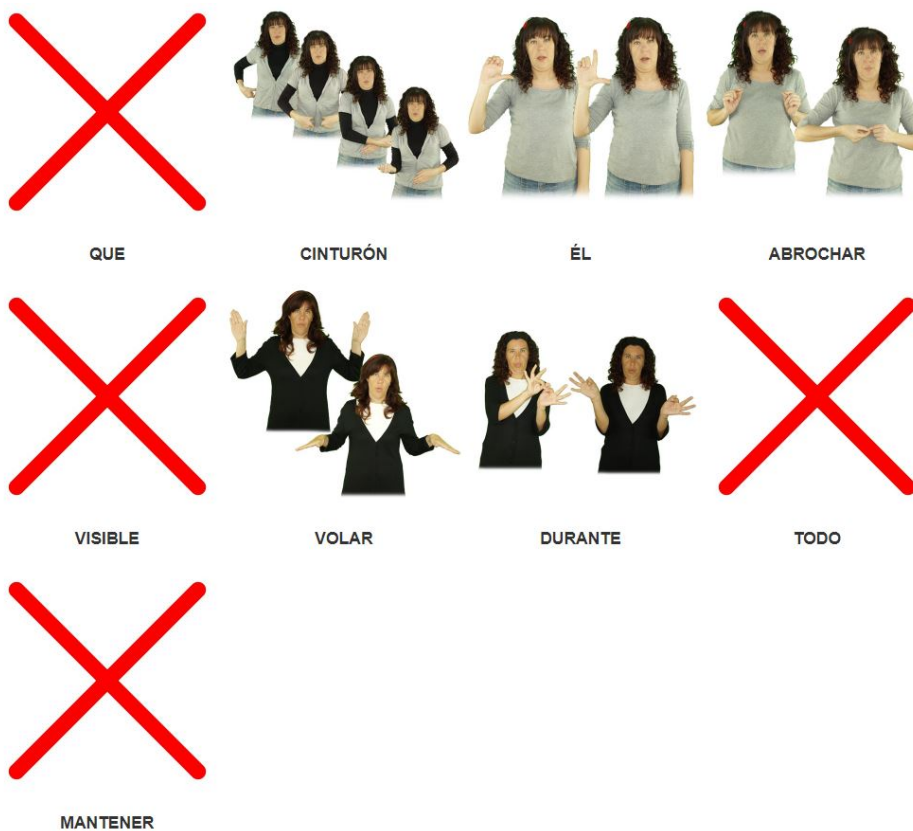


MANTENER

17. Por su seguridad.



18. Les recomendamos que mantengan su cinturón abrochado y visible durante todo el vuelo.



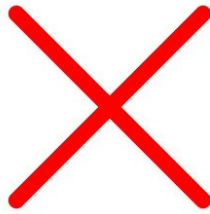
19. Y siempre que la señal luminosa lo indique.



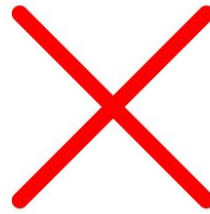
SIEMPRE



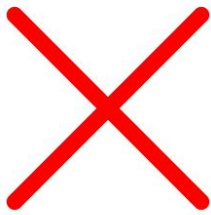
SEÑAL



LUMINOSO



QUE



INDICAR

20. Para abrocharlo.

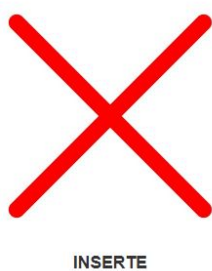


ABROCHARLO



PARA

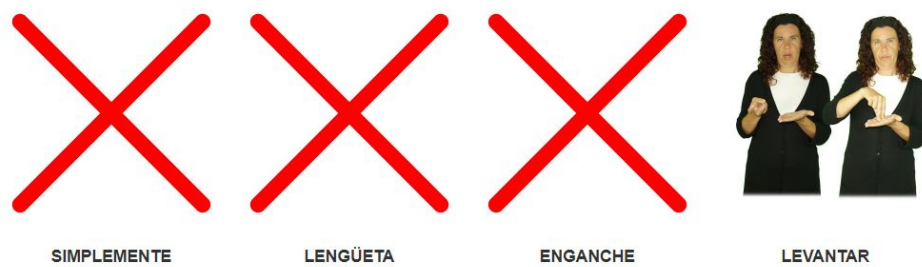
21. Inserte la trabilla en su enganche correspondiente.



22. Para soltarlo.

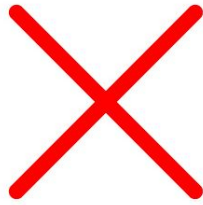


23. Simplemente levanten la lengüeta del enganche.



24. Este avión cuenta con ocho puertas y dos ventanas de salida, 4 puertas a cada lado del avión y una ventana sobre cada ala.

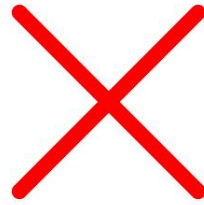
AVIÓN	ESTE	PUERTA	FEMENINO
PLURAL	CON	OCHO	VENTANA
FEMENINO	PLURAL	DOS	SALIR
FEMENINO	PUERTA	FEMENINO	PLURAL



4



LADO



CADA



AVIÓN



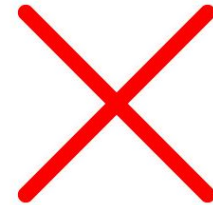
VENTANA



ALA



SOBRE



CADA



CONTAR

25. Todas están señalizadas con la palabra EXIT.



26. Y disponen de rampas o balsas de evacuación.



27. Debajo de sus asientos encontrarán un chaleco salvavidas.



28. Sáquenlo de la bolsa.



29. Y para ponérselo.



PONÉRSELO

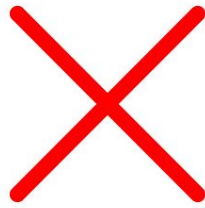


PARA

30. Introduzcan la cabeza por la abertura.



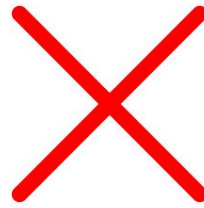
CABEZA



ABERTURA



POR



INTRODUZCAN

31. Y pasen la cinta de atado por detrás de su cintura.



32. Enganchando la fijación de la hebilla de un extremo a otro.



33. Y tirando de la cinta para ajustárselo.



34. Para inflarlo.



35. Solo tienen que tirar con fuerza del tirador del plástico rojo.



36. O soplar por el tubo.



37. Y recuerden.



**RECORDAR**

38. Nunca deben inflar el chaleco dentro del avión.



**NUNCA**



**DEBEN**



**CHALECO**



**DENTRO**



**AVIÓN**



**INFLAR**

39. En caso de despresurización de la cabina.



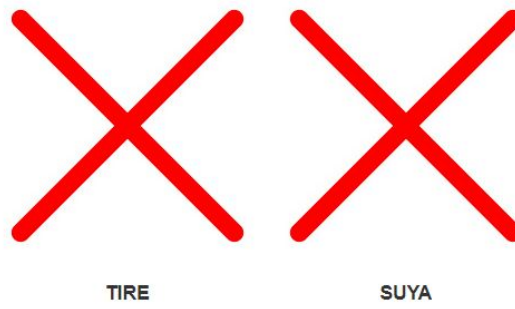
40. Se abrirá automáticamente un compartimento sobre sus asientos.



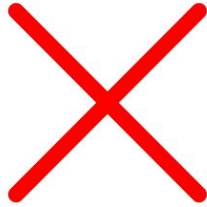
41. Que contiene máscaras de oxígeno.



42. Tire de la suya.



43. Colóquela sobre su nariz y boca.



COLÓQUELA



NARIZ



SOBRE



ÉL



BOCA

44. Y respire con normalidad.



RESPIRAR



NORMALIDAD



CON

45. Después preste ayuda a quién pueda depender de usted.



DESPUÉS



QUIÉN



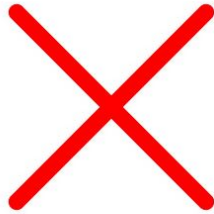
AYUDAR



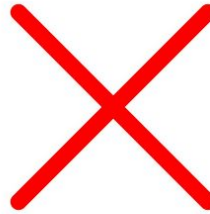
FEMENINO



PUEDA

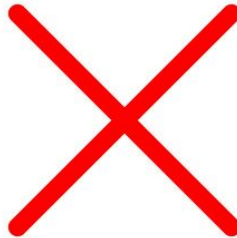


USTED



DEPENDER

46. Recuerden.



RECUERDEN

47. Que para evitar poner en peligro la seguridad de este vuelo.



48. No está permitido fumar.



49. O usar dispositivos de liberación de nicotina en ningún caso.



50. Y no olviden.





53. Muchas gracias por su atención.



54. Esperamos que disfruten de un feliz vuelo.

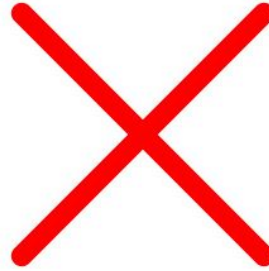


## F.2. Corpus de oraciones de un capítulo de una serie infantil

1. Yo soy Pepa.



YO



PEPA

2. Este es mi hermano pequeño.



ESTE



HERMANO



YO



PEQUEÑO

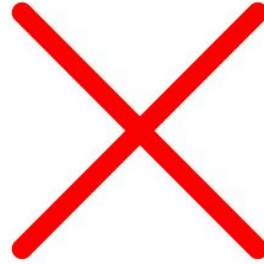
3. Esta es mamá Pig.



**ESTA**



**MAMÁ**



**PIG**

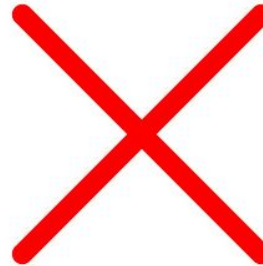
4. Este es papá Pig.



**ESTE**



**PAPÁ**



**PIG**

5. Hoy está lloviendo.



6. Así que Pepa no puede jugar fuera.



7. Ha dejado de llover.



HA



DEJADO



LLOVER

8. ¿Podemos salir a jugar?



PODEMOS

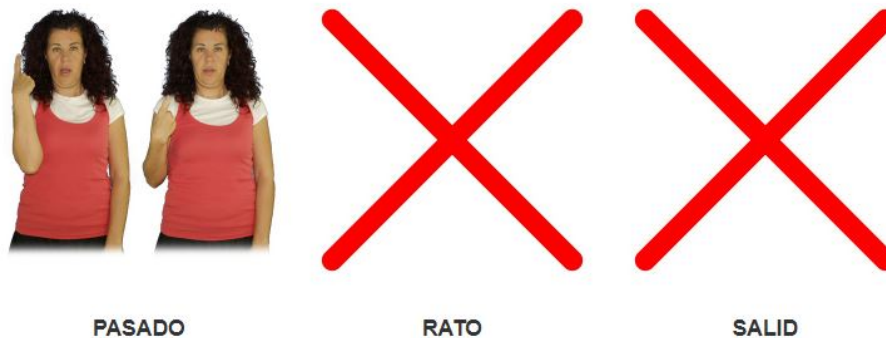


SALIR



JUGAR

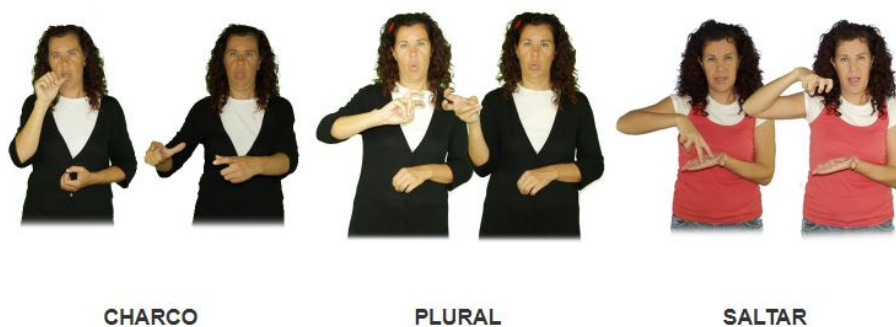
9. Salid un rato.



10. A Pepa le encanta saltar en los charcos.



11. Me encanta saltar en los charcos.



12. Si vas a pisar charcos debes ponerte las botas de agua.



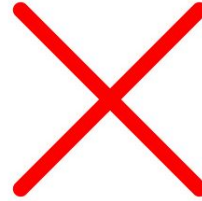
SI



CHARCO



PLURAL



DEBES



BOTAS

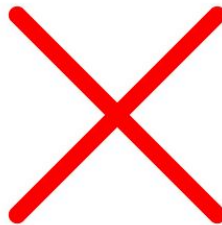


AGUA



IR

13. Lo siento.

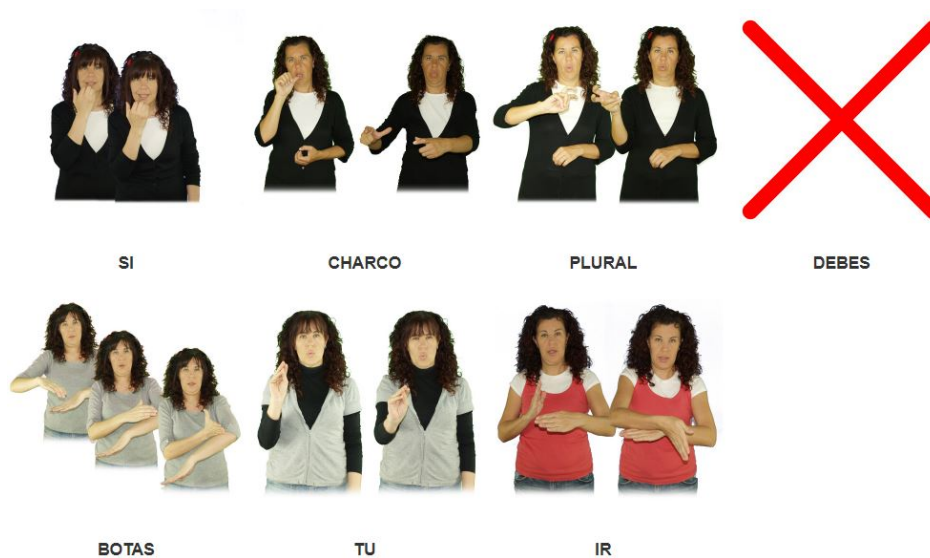


SENTIR

14. A George también le gusta saltar en los charcos.



15. Si vas a saltar en los charcos, debes ponerte tus botas.



16. A Pepa le gusta cuidar de su hermano pequeño.



PEPA



HERMANO



ÉL



PEQUEÑO



CUIDAR

17. Vamos a buscar más charcos.



VAMOS



CHARCO



PLURAL



MÁS



BUSCAR

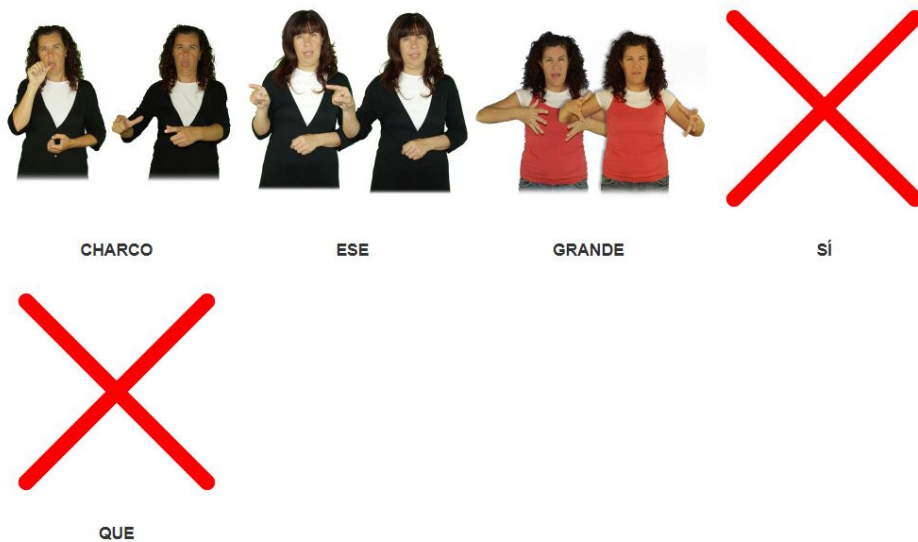
18. Pepa y George se lo están pasando muy bien.



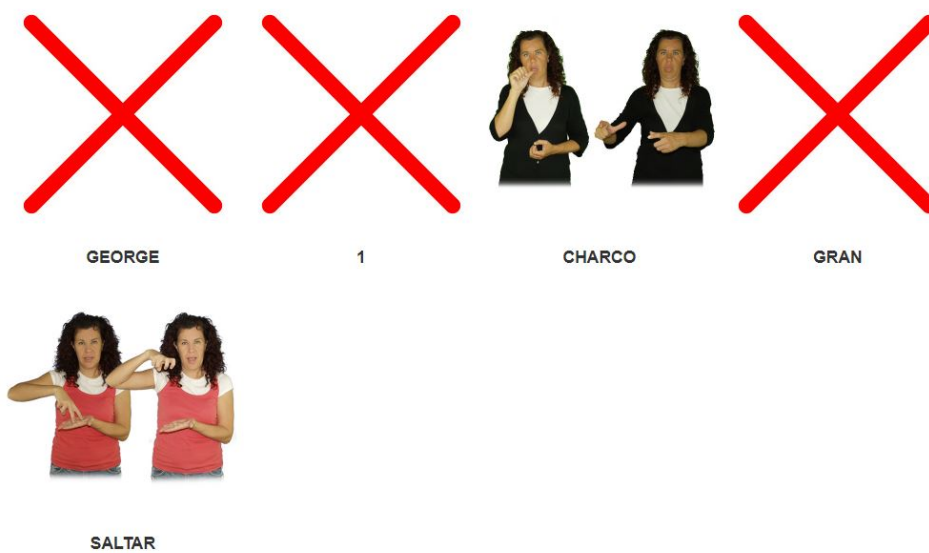
19. Pepa ha encontrado un charco pequeño.



20. Ese charco sí que es grande.



21. George quiere ser el primero en saltar en el gran charco.



22. Tengo que ver si es seguro para ti.



23. Parece seguro para tí.



24. Solo es barro.



SOLO

BARRER

25. Vamos a enseñárselo a Papi.

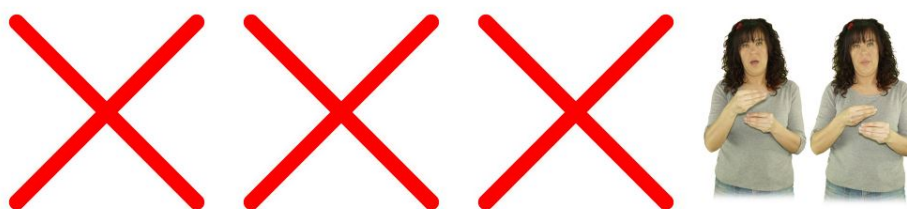


ENSEÑARSELO

PAPI

VAMOS

26. Adivina qué hemos estado haciendo.



QUÉ

HEMOS

ESTADO

HACER

27. Déjame pensar.



DÉJAME

28. Habéis estado saltando en los charcos de barro.



HABÉIS



ESTADO



CHARCO



PLURAL



BARRER

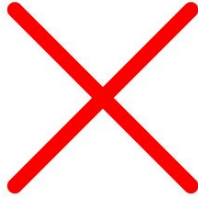


SALTAR

29. Hemos saltado en los charcos de barro.



PASADO



HEMOS



CHARCO



PLURAL



BARRER

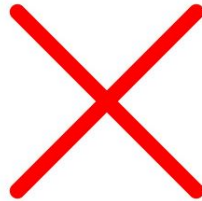


SALTAR

30. Mirad cómo os habéis puesto.



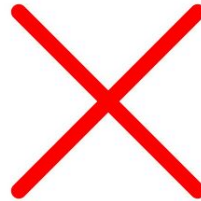
PASADO



MIRAD



CÓMO



HABÉIS



PONER

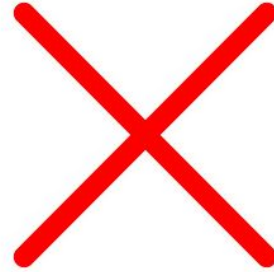
31. No pasa nada.



**NADA**



**NO**

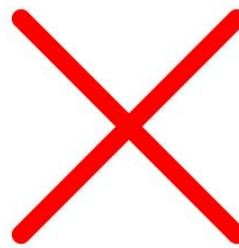


**PASO**

32. Sólo es barro.

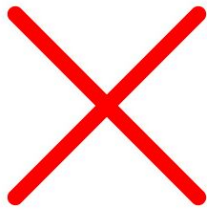


**BARRER**



**SÓLO**

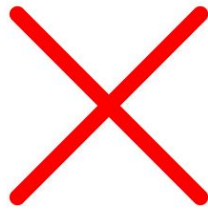
33. Voy a limpiaros antes de que os vea mamá.



ANTES



LIMPIAROS



QUE



MAMÁ



VER

34. Podemos jugar todos juntos.



PODEMOS



JUNTO

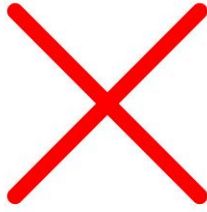


TODOS

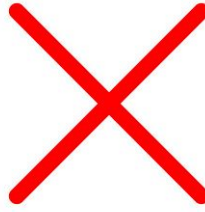


JUGAR

35. Pepa y George llevan sus botas de agua.



PEPA



GEORGE



BOTAS



ELLOS



AGUA



LLEVAR

### F.3. Corpus de subtítulos de un trailer de una película

1. Mi negocio son los perros.



YO



NEGOCIO



PERRO



PLURAL

2. El corazón de un superviviente.



CORAZÓN



SUPERVIVIENTE

3. Les ganó a todos los demás.



PASADO



DEMÁS



TODOS



GANAR

4. Él no es un perro de trineo.



5. Es un ganador.



6. Lo que tienen nuestros niños no es una epidemia, es una sentencia de muerte.



7. Encontraron la cura.



8. Solo un hombre y un perro pueden hacer esa carrera.



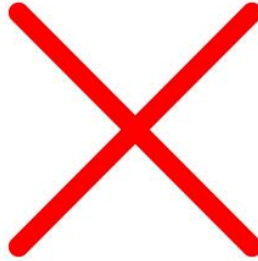
9. Tiene 12 años.



10. Es demasiado viejo.



VIEJO



DEMASIADO



ES

11. Pienso que no los vamos a poder encontrar después del deshielo.



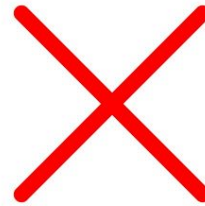
DESPUÉS



QUE



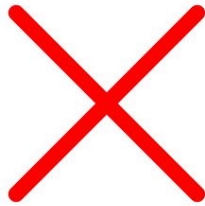
NO



VAMOS



PODER



DESHIELO



ENCONTRAR

12. Es el momento de descubrir quienes somos.



13. Yo siempre pensé que vivía para los trineos.



## F.4. Corpus de subtítulos de un anuncio

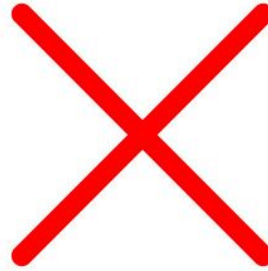
1. Quédate en casa.



PASADO



CASA



QUÉDATE

2. Mantén una distancia de dos metros.



DISTANCIA



METRO



PLURAL



DOS



MANTÉN

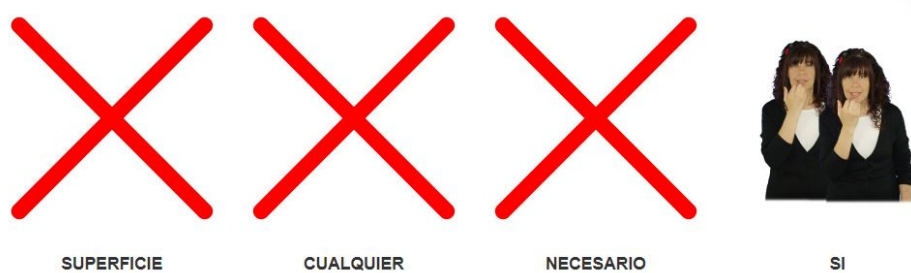
3. No te toques la cara.



4. Lávate las manos con frecuencia.



5. Evita tocar cualquier superficie si es necesario.



6. No saludes dando la mano.



**NO**



**MANO**



**DAR**

7. No des besos ni abrazos.



**BESOS**

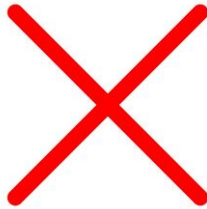


**NO**



**ABRAZOS**

8. Puedes saludar con la mirada o de palabra.



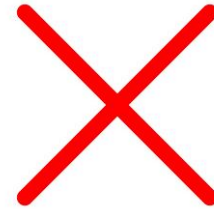
PUEDES



MIRAR



FEMENINO



CON



PALABRA



SALUDAR

9. Mantén tu teléfono móvil limpio.



PASADO



TELÉFONO



TU



MÓVIL



LIMPIAR



MANTÉN

10. Si tienes síntomas.



11. Quédate en casa y aíslate en tu habitación.



## F.5. Corpus de frases simples

1. Él bebe agua.



**ÉL**



**AGUA**



**BEBER**

2. Yo soy morena.



**YO**



**MORENO**

3. Ellos parecen contentos.



**ELLOS**



**CONTENTO**

4. María compra galletas.



**MARÍA**



**GALLETAS**



**COMPRAR**

5. Yo bebo agua.



**YO**



**BEBO**



**AGUA**

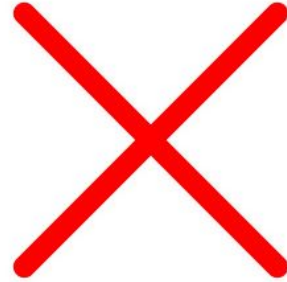
6. Yo como carne.



YO



CARNE



COMO

7. Ellos compraron aceite.



PASADO



ELLOS



ACEITE



COMPRAR

8. Él venderá refrescos



FUTURO

ÉL

REFRESCO

PLURAL



VENDER

9. Ella ganó la carrera.



PASADO

ELLA

CARRERA

GANAR

10. Yo haré la cama.



FUTURO

YO

CAMA

HACER

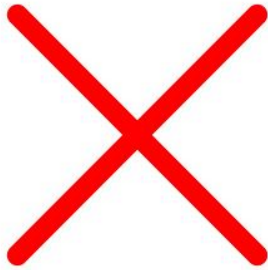
11. Mi tía y mi hermana beberán agua.



12. Ella y él comieron arroz.



13. María y Javier nadan.



MARÍA



JAVIER



NADAR

14. Mi hermana y mi tía están de vacaciones en la playa.



YO



HERMANA



YO



TÍO



FEMENINO



VACACIONES



PLAYA

15. Los niños comerán chocolate negro.



FUTURO



NIÑOS



CHOCOLATE



NEGRO



COMER

16. Las niñas jugarán en el parque.



FUTURO



NIÑO



FEMENINO



PLURAL



PARQUE



JUGAR

17. Mis tíos irán al supermercado en coche.



18. Los niños jugarán con la pelota.



19. Los ganadores recibirán un trofeo de oro.



FUTURO



GANADORES



TROFEO



ORO



RECIBIR

20. Él pasa ante la casa todos los días.



ÉL



CASA



ANTE



DÍA



PLURAL



TODOS



PASO

21. Él pasea ante la casa por la mañana.



MAÑANA



POR



ÉL



CASA



ANTE



PASEAR

22. Mañana yo iré al médico.



MAÑANA



YO



MÉDICO



IR

23. Él comió mucho en la fiesta.



PASADO

ÉL

MUCHO

FIESTA



COMER

24. Él se comió la hamburguesa aquí.



PASADO

ÉL

HAMBURGUESA

AQUÍ



COMER

25. Mi tía conduce muy despacio.



YO



Tío



FEMENINO



DESPACIO



MUY



CONDUCIR



# Bibliografía

- ARASAAC. *Qué son los SAAC*. Disponible en <http://www.arasaac.org/aac.php>, 2019.
- DISCAPNET. *Discapacidad auditiva*. Disponible en <https://www.discapnet.es/areas-tematicas/salud/discapacidades/auditivas/discapacidad-auditiva>, 2019.
- EDUARDO SOSA. *Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (Parte I)*. Disponible en [http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento\\_del\\_lenguaje\\_natural\\_revisin\\_del\\_estado\\_actual\\_bases\\_tericas\\_y\\_aplicaciones\\_parte\\_i.html](http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html), 1997.
- FERNÁNDEZ, A. *La cantidad a manos llenas. La expresión de la cuantificación en la Lengua de Signos Española*. Fundación CNSE, 2008.
- IBM CORPORATION. *Qué es un servicio web*. Disponible en [https://www.ibm.com/support/knowledgecenter/es/SSMKHH\\_9.0.0/com.ibm.etools.mft.doc/ac55710\\_.html](https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55710_.html), 2014.
- IEBSCHOOL. *Qué es Kanban*. Disponible en <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/>, 2013.
- IEBSCHOOL. *Metodologías Ágiles*. Disponible en [https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/#metodologias\\_agiles](https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/#metodologias_agiles), 2019.
- IONOS. *Servidor Proxy Inverso*. Disponible en <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-proxy-inverso/>, 2020.
- JUAN IGNACIO BAGNATO. *Introducción al Procesamiento del Lenguaje Natural*. Disponible en <https://www.aprendemachinelearning.com/procesamiento-del-lenguaje-natural-nlp/>, 2018.

- KANBANIZE. *Qué es un límite Kanban. Por qué lo necesita. Disponible en <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-limite-wip/>*, 2020.
- KANBANTOOL. *Metodologías Ágiles. Disponible en <https://kanbantool.com/es/metodologia-kanban>*, 2019.
- MANUEL ROSA MONCAYO, J. *Qué es REST. Conoce su potencia. Disponible en <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>*, 2018.
- MUÑOZ, F. y ET.AL., A. *Apuntes de Lingüística de la Lengua de Signos Española*. Fundación CNSE, 2002.
- OPENWEBINARS. *Qué es Flask. Disponible en <https://openwebinars.net/blog/que-es-flask/>*, 2019.
- TORRES, S. *Sistemas alternativos de comunicación..* Ed. Aljibe, 2001.
- WIKIS FDI. *Qué es la licencia BSD. Disponible en [http://wikis.fdi.ucm.es/ELP/Licencia\\_BSD](http://wikis.fdi.ucm.es/ELP/Licencia_BSD)*, 2017.