



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DESARROLLO DE UNA PLATAFORMA MULTIMEDIA DE LIBRE ACCESO, PARA EL APRENDIZAJE DE LA LENGUA DE SEÑAS CHILENA Y DEL ESPAÑOL ESCRITO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL EN COMPUTACIÓN

NATALIA ANDREA ALARCÓN DÍAZ

PROFESOR GUÍA:

SERGIO OCHOA DE LORENZI

MIEMBROS DE LA COMISIÓN:

PATRICIO POBLETE OLIVARES

ALEJANDRO HEVIA ANGULO

SANTIAGO DE CHILE

2015

RESUMEN DE LA MEMORIA PARA OPTAR AL
TITULO DE INGENIERA CIVIL EN COMPUTACION
POR: NATALIA ANDREA ALARCÓN DÍAZ
FECHA: 10 DE JULIO, 2015
PROFESOR GUÍA: SERGIO OCHOA DELORENZI

DESARROLLO DE UNA PLATAFORMA MULTIMEDIA DE LIBRE ACCESO, PARA EL APRENDIZAJE DE LA LENGUA DE SEÑAS CHILENA Y DEL ESPAÑOL ESCRITO

Hoy en Chile, el currículo nacional para estudiantes sordos considera como idea básica de integración en la sociedad, el desarrollo pleno tanto de la Lengua de Señas chilena como del Español escrito, entendiendo la primera como una herramienta vehicular de comunicación y acceso al currículo, y la segunda, como herramienta básica para acceder a las exigencias académicas de la incorporación futura a los sistemas de educación superior o laboral.

Entre las tareas pendientes en cuanto a la integración de la cultura sorda se refiere, es fundamental aumentar el acceso a la educación. En este contexto se desarrolló la aplicación web llamada “Diccionario Multimedia INDESOR”, herramienta colaborativa que enseña y facilita el aprendizaje de ambas lenguas, tanto para sordos como para oyentes, y que busca además reforzar el desarrollo de las habilidades de lectura y escritura en niños con déficit auditivo pertenecientes al primer y segundo ciclo de enseñanza básica.

Esta aplicación consiste en un diccionario de libre acceso, que por cada palabra contiene su definición y un video de la seña que la representa. En torno a ésta, se creó una comunidad virtual de estudiantes, apoderados y profesionales del Instituto de la Sordera INDESOR, cuya capacidad de autogestión permite la mantención de información actualizada y de calidad dentro del sitio.

Los usuarios sordos y oyentes de esta comunidad son los encargados de subir, descargar y evaluar el material del diccionario, mientras que los usuarios expertos (profesionales del INDESOR) se encargan de la aprobación o rechazo del mismo. Estas interacciones están equilibradas de tal forma de asegurar el crecimiento en volumen de la base de datos, imponiendo un modelo de puntaje intuitivo para los usuarios de la aplicación.

El resultado de este trabajo, la creación de una plataforma web con cerca de 500 palabras, con una dinámica de interacción que permite a usuarios sordos y oyentes participar activamente dentro del sitio, cuyo potencial de crecimiento va en aumento. Así, desde la base de un diccionario, se genera un nuevo enfoque para el desarrollo de habilidades de comunicación, tanto para la comunidad sorda como para su entorno educativo.

A los niños del Instituto de la Sordera INDESOR, la creación de esta herramienta para el apoyo de su aprendizaje.

AGRADECIMIENTOS

A mi familia, por su pleno e incondicional apoyo en todos mis años de formación.

Al compañero de todos mis días, por darme el coraje para seguir adelante.

A mis amigos, porque sin ellos todo mi esfuerzo no habría bastado.

A mi profesor guía, por la oportunidad de trabajar con él y por creer en mis capacidades.

A la ciencia, la experiencia y al paso del tiempo, por su mirada cierta de que llegaría este día.

TABLA DE CONTENIDO

1. Introducción.....	1
1.1. Motivación.....	1
1.2. Objetivos de la memoria	2
1.3. Alcance del proyecto	3
1.4. Metodología	4
2. Investigación exploratoria	5
2.1. Diccionario Escolar Visual con Lengua de Señas	5
2.2. Proyecto Dicciseñas.....	6
2.3. Proyecto SueñaLetras.....	7
2.4. Duolingo.....	8
3. Diseño general de la solución	9
3.1. Requisitos generales.....	9
3.2. Arquitectura física de la solución	9
3.3. Arquitectura lógica de la solución	10
3.3.1. Componentes de la solución.....	12
3.3.2. Módulos de la solución	13
3.3.3. Modelo de datos de la solución.....	13
3.4. Arquitectura de una aplicación web	14
3.5. Herramientas de desarrollo.....	15
4. Implementación de la solución.....	17
4.1. Diseño e implementación de la API.....	17
4.1.1. Arquitectura de software: REST API	17
4.1.2. Metodología de programación: Test Driven Development.....	19
4.2. Diseño e implementación de la aplicación web Diccionario Multimedia INDESOR	19
4.2.1. Diseño general de las vistas de la aplicación	20
4.2.2. Login y registro de usuarios	20
4.2.3. Diccionario multimedia.....	25
4.2.4. Creación de archivos.....	28
4.2.5. Validación de palabras: Desafíos	32
4.2.6. Aceptación/rechazo de palabras	34
4.3. Diseño de actividades adicionales.....	38

4.3.1.	Diseño de material adicional	38
4.3.2.	Nueva modalidad para los desafíos	38
5.	Pruebas de usabilidad	40
5.1.	Resultados obtenidos	41
5.2.	Ajustes a la solución	42
6.	Conclusiones y trabajo futuro.....	43
7.	Bibliografía	45
Anexo 1:	Requisitos de usuario	46
Anexo 2:	Requisitos de software	49
Anexo 3:	Matriz de trazado - Requisitos de usuario vs requisitos de software	52
Anexo 4:	Pauta pruebas de usabilidad - usuarios sordos.....	53
Anexo 5:	Pauta pruebas de usabilidad - usuarios oyentes.....	55
Anexo 6:	Pauta pruebas de usabilidad – Usuarios expertos	57

TABLA DE FIGURAS

Figura 1: Diagrama de la plataforma multimedia	3
Figura 2: Diccionario Escolar Visual con Lengua de Señas	5
Figura 3: Ilustración de una palabra - Diccionario Escolar Visual con Lengua de Señas	6
Figura 4: Ilustración de una palabra - Proyecto Dicciseñas.....	6
Figura 5: Ilustración de la actividad "Completar una palabra" - Proyecto Sueñalettras.....	7
Figura 6: Ilustración de una actividad realizada correctamente – Proyecto Sueñalettras.....	7
Figura 7: Sección “Palabras” de la aplicación web de Duolingo.....	8
Figura 8: Esquema general de la arquitectura física del sistema	10
Figura 9: Diagrama componentes y módulos.....	11
Figura 10: Funciones principales de un usuario experto dentro de la plataforma	11
Figura 11: Funciones principales de un usuario sordo dentro de la plataforma	11
Figura 12: Funciones principales de un usuario oyente dentro de la plataforma	12
Figura 13: Componentes de la API	12
Figura 14: Módulos de la aplicación web	13
Figura 15: Modelo de datos de la API.....	14
Figura 16: Diagrama de una petición en Django	15
Figura 17: Sección “Cómo funciona” de la aplicación	20
Figura 18: Pantalla de ingreso a la aplicación	21
Figura 19: Menú de opciones para los usuarios sordos de la aplicación	23
Figura 20: Pantalla del cierre de sesión para un usuario sordo	23
Figura 21: Pantalla de registro de usuarios de la aplicación	24
Figura 22: Diccionario multimedia organizado alfabéticamente	26
Figura 23: Diccionario multimedia organizado según la configuración manual	26
Figura 24: Configuraciones manuales en dibujo vectorial	27
Figura 25: Estados de una palabra en la aplicación	28
Figura 26: Formulario de ingreso de una palabra al diccionario multimedia	29
Figura 27: Formulario de ingreso de una definición al diccionario multimedia	30
Figura 28: Desafío para un usuario sordo	32
Figura 29: Diagrama de un desafío de la aplicación.....	33
Figura 30: Listado de palabras en proceso de validación.....	35
Figura 31: Pantalla de revisión de una palabra	36

Figura 32: Juego tiro al blanco para la validación de la relación seña-palabra.....	38
Figura 33: Juego caza tesoros para la validación de la relación definición-palabra	39
Figura 34: Juego de <i>flashcards</i> para la validación de la relación seña-palabra	39
Figura 35: Detalle del diccionario multimedia con indicador de pago por palabras	42

1. INTRODUCCIÓN

En la actualidad en nuestro país se han desarrollado algunos proyectos [5, 6, 7] para mejorar el aprendizaje de la Lengua de Señas Chilena (que es distinta a la Lengua de Señas utilizada en el resto de los países, incluyendo países de habla hispana como Argentina o España), pero el enfoque está dirigido hacia el oyente. Es decir, en cómo un oyente puede aprender la Lengua de Señas, y no en cómo un sordo puede utilizar ese material para aprender Español o la misma Lengua de Señas.

El gran desafío de este trabajo es lograr generar una aplicación computacional que permita desarrollar una comunidad digital en torno a la Lengua de Señas, que sea autosustentable en el tiempo, y que tenga la capacidad de desarrollarse por sí misma (autorregularse), sin la necesidad de constante intervención por parte de moderadores expertos. Se busca además que la comunidad sea capaz de controlar colaborativamente la calidad del contenido didáctico que se comparte con terceros.

En particular, es necesaria la inclusión de técnicas de *crowdsourcing* [11] y *collective quality control* [12] para la mantención de “buena información” en el sitio. En otras palabras, es necesario generar mecanismos y espacios para que los mismos usuarios sean los encargados de proporcionar y validar la información tal que se mantenga fidedigna y actualizada.

En adición, es fundamental que la aplicación tenga los atributos funcionales y de calidad que los usuarios finales (niños, jóvenes y adultos, sordos y oyentes) esperan de una plataforma de estas características en términos del diseño de interfaces y usabilidad de la aplicación. Para ello, es necesario el trabajo en conjunto con personas ligadas a la Lengua de Señas, tanto con profesionales como con personas con déficit auditivo.

1.1. MOTIVACIÓN

La Comunidad de Sordos Chilena tiene dificultades para comunicarse con la comunidad oyente que la rodea, debido a las diferencias entre las formas de comunicarse que ellos emplean (Lengua de Señas vs Español). Esto genera dificultades en la inserción de estas personas en la sociedad.

Un problema derivado de lo anterior es que, dado que el Español escrito es la segunda lengua de estas personas (la Lengua de Señas es la primera), existe en la mayoría de los niños a nivel escolar la dificultad de aprender a leer y escribir en Español puesto que existen diferencias gramaticales y de modalidad: el Español es un lenguaje auditivo-oral mientras que la Lengua de Señas es viso-gestual. Esto dificulta tanto la enseñanza como el aprendizaje de los distintos contenidos en Español escrito para poder interactuar con el medio.

A su vez, para el oyente que interactúa con la comunidad de sordos, es difícil lograr una comunicación efectiva dado que, por una parte, no conoce la Lengua de Señas y por otra, no puede pretender transmitir todo lo que desea en Español (por ejemplo de manera escrita), puesto que el sordo no lo va a entender completamente.

Las problemáticas expuestas anteriormente surgen de la experiencia que se vive dentro del Instituto de la Sordera INDESOR [4], institución con la cual se han identificado además los siguientes problemas:

1. El material existente para el aprendizaje de la Lengua de Señas chilena es escaso, incompleto y en su mayoría protegido.
2. Existen pocos especialistas en la enseñanza de la Lengua de Señas en nuestro país.
3. Los cursos de Lengua de Señas chilena ofrecidos por distintas instituciones son pagados, básicos e intensivos, lo cual genera que la curva de aprendizaje de los alumnos sea mala puesto que, debido a la corta duración de estos cursos, hay una baja retención de información.
4. No hay validación de contenido, sólo autocontrol por parte de los profesionales de cada institución.

Finalmente, dado que el medio es en su mayoría oyente y no está preparado para interactuar con sordos, se genera una mayor segregación y aislación de estas personas en la sociedad.

1.2. OBJETIVOS DE LA MEMORIA

El objetivo principal de este trabajo de memoria es desarrollar una aplicación web con un diccionario multimedia de libre acceso, que comprenda la traducción de palabras entre Lengua de Señas y Español, y que permita aprender y/o reforzar el conocimiento de ambos idiomas. Los objetivos específicos que se desprenden del objetivo general, son los siguientes:

1. Diseñar e implementar una base de datos colectiva (*bootstrapping database*) de material pedagógico multimedia sobre la cual, mediante la aplicación web, los usuarios puedan subir, descargar y evaluar material. Cabe destacar que estas interacciones serán equilibradas imponiendo reglas que aseguren el crecimiento en volumen de la base de datos.
2. Diseñar una metodología que controle la calidad de la información, que funcione de manera colaborativa entre los usuarios de la aplicación (*collective quality control*).
3. Diseñar una metodología que motive a los usuarios del sitio a participar activamente en la inclusión de nueva información y validación de la misma (*crowdsourcing*). En particular, hay que considerar la simplicidad y eficacia para realizar estas labores, de modo que no se pierda el atractivo para los usuarios.
4. Diseñar e implementar un conjunto de servicios que permitan mantener un diccionario multimedia que por cada palabra contenga un video con la seña, la palabra en Español y su definición.

5. Diseñar una base de datos de frases y diálogos que permita a futuro crear actividades adicionales que motiven a los usuarios a aprender el idioma correspondiente. Para esto, se requiere inventar un sistema de puntaje que permita desbloquear este material para ampliar el conocimiento que la persona tiene de un lenguaje.

1.3. ALCANCE DEL PROYECTO

El proyecto de memoria presentado en este informe considera la creación de:

1. Una API (librería de funciones) que permita la alimentación y validación de contenidos a través de un conjunto de desafíos para los usuarios. Esta API deberá proveer funcionalidades tales como: permitir la carga y descarga de material en el servidor, administrar perfiles de usuario e implementar una moneda de cambio para las transacciones antes descritas con el fin de regular estos procesos.
2. Una aplicación web abierta que muestre un diccionario multimedia de Lengua de Señas chilena y Español escrito, cuyo contenido será: un video con una seña, la palabra en Español escrito representada por la seña y su definición. Se considera dejar implementado el diccionario con un mínimo de 300 palabras.
3. Una aplicación web (dentro de la aplicación anterior) que permita al INDESOR controlar el material disponible a ser mostrado. Esta aplicación permitirá revisar que la validación de contenido por parte de los usuarios se está realizando de forma correcta.

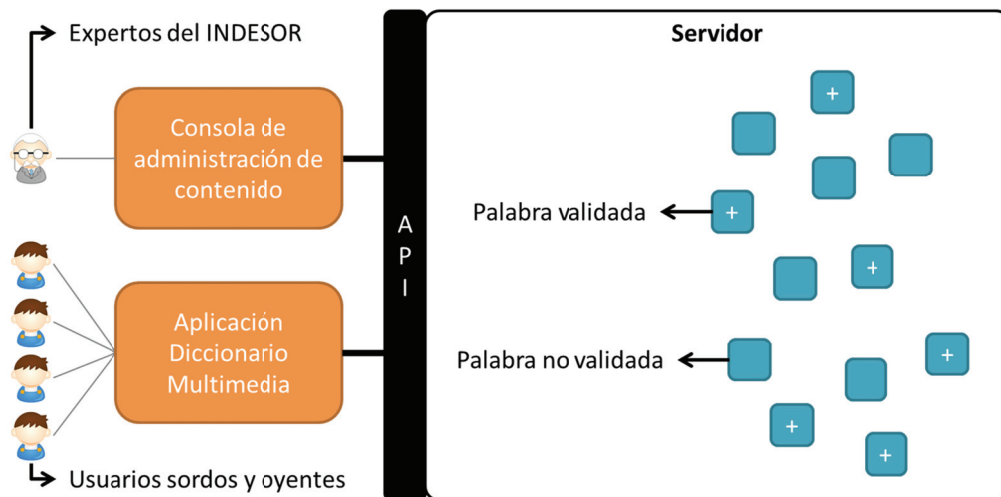


FIGURA 1: DIAGRAMA DE LA PLATAFORMA MULTIMEDIA

La figura 1 muestra como los usuarios, ya sean éstos sordos u oyentes, pueden acceder al diccionario multimedia. La aplicación por su parte, se conecta con la API para pedir el contenido almacenado en el servidor. A su vez, existe una aplicación web que permite a los expertos administrar el contenido.

1.4. METODOLOGÍA

El trabajo desarrollado se realizó con el apoyo de profesionales del Instituto de la Sordera INDESOR, cuyo representante y contacto fue el señor Jesús Gahona González, profesor diferencial de sordos. Éste abarcó las siguientes etapas:

1. **Investigación exploratoria:** Esta etapa contempló la búsqueda de aplicaciones o herramientas desarrolladas en Chile y en el extranjero, para detectar falencias y/o ideas a rescatar para hacer una buena aplicación.
2. **Diseño e implementación de una API que contiene mecanismos de alimentación y validación de contenidos:** Se utilizaron estrategias de *crowdsourcing* y *collective quality control*. La idea central de la aplicación es que sean los miembros de la comunidad (niños, apoderados y profesores del INDESOR) quienes suban material a la plataforma y que, mediante desafíos (preguntas al usuario dentro de la aplicación), se logre la validación del material.
3. **Diseño e implementación del diccionario multimedia:** Dado el apoyo brindado por el INDESOR, se comenzó con un material base para la implementación del diccionario. En esta etapa se desarrolló un prototipo con el fin de recibir la opinión y validación de los profesionales del INDESOR en el ámbito pedagógico del diseño.
4. **Diseño e implementación de una aplicación web:** Esta etapa consistió en el desarrollo de una aplicación web a través de la cual los usuarios finales pueden acceder al diccionario multimedia, debiendo registrarse primero en la aplicación. Además, este sitio provee a los usuarios expertos las funcionalidades necesarias para la validación final de las palabras a ser añadidas al diccionario.
5. **Diseño de actividades adicionales dentro de la aplicación.** A partir del material audiovisual con el que cuenta el INDESOR, se diseñó una base de datos sobre frases y diálogos que permite a los usuarios ampliar su conocimiento. Adicionalmente, se plantearon otros modelos de desafíos de mayor atractivo para sus usuarios.
6. **Integración de los componentes anteriores:** La integración dio lugar a un segundo prototipo validado en conjunto con el INDESOR con el fin de analizar si la plataforma creada cumple con los objetivos planteados y es potencialmente útil y eficaz para el aprendizaje tanto de la Lengua de Señas como del Español.
7. **Realización de pruebas de usuario:** La solución propuesta se validó con un conjunto de personas de la comunidad asociada al INDESOR (niños, apoderados y profesores). El objetivo principal fue determinar el nivel de aceptación de la plataforma web por parte de los usuarios y la efectividad de la misma como instrumento facilitador del aprendizaje.
8. **Ajustes de la aplicación web:** La aplicación se ajustó de acuerdo a los resultados obtenidos a partir de las pruebas de usuario realizadas a los integrantes de la institución que apoyó el desarrollo de este proyecto.

2. INVESTIGACIÓN EXPLORATORIA

En el contexto de realizar una aplicación para el aprendizaje de Español y Lengua de Señas destinada especialmente a la comunidad sorda, se realizó una revisión exhaustiva de las aplicaciones web que existen en el mercado nacional [5, 6, 7], así como de los cursos online que ofrecen distintas instituciones, referente a precio y contenido. Se llegó a la conclusión que no existe en Chile una herramienta colaborativa que permita aprender la Lengua de Señas.

A continuación se detallan 3 proyectos nacionales que se asemejan a un diccionario multimedia pero que contienen poco material y son proyectos que no han seguido desarrollándose en años, por tanto se mantienen incompletos. Luego, se describe un proyecto de renombre internacional que tiene el enfoque colaborativo que se le quiso dar a la aplicación realizada en este trabajo de memoria, pero que está orientado a ser un curso de idiomas y no implementa un diccionario como parte de sus herramientas para el aprendizaje.

2.1. DICCIONARIO ESCOLAR VISUAL CON LENGUA DE SEÑAS

Es un diccionario visual que muestra contenidos relacionados solamente con el área de Matemáticas y es limitado en las sub-áreas de esta materia. Su funcionalidad es enseñar vocabulario relacionado con la asignatura en cuestión mostrando una imagen, la palabra en Español escrito y un video en Lengua de Señas respectivamente. En las figuras 2 y 3 se visualizan las pantallas principales de este diccionario:



FIGURA 2: DICCIONARIO ESCOLAR VISUAL CON LENGUA DE SEÑAS

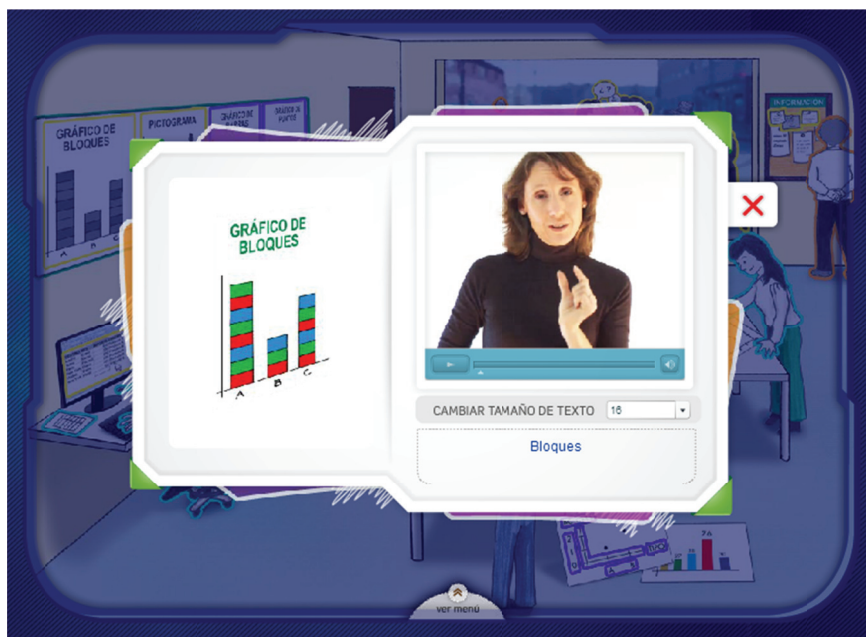


FIGURA 3: ILUSTRACIÓN DE UNA PALABRA - DICCIONARIO ESCOLAR VISUAL CON LENGUA DE SEÑAS

2.2. PROYECTO DICCISEÑAS

La figura 4 ilustra la aplicación Dicciseñas, la cual consiste en un diccionario visual de contenido estático que posee 102 palabras. Dado que la RAE (Real Academia Española) valida alrededor de 88.000 palabras para el Español, podemos concluir que ésta es una muy limitada herramienta para el aprendizaje.

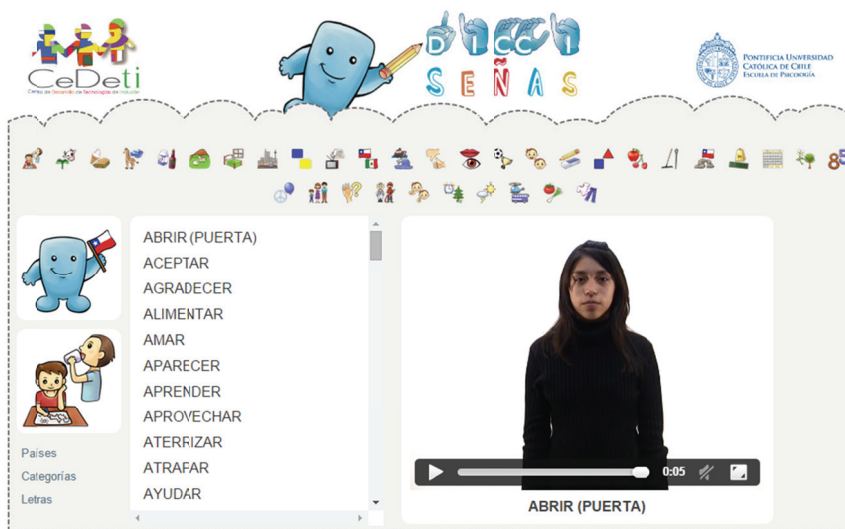


FIGURA 4: ILUSTRACIÓN DE UNA PALABRA - PROYECTO DICCISEÑAS

2.3. PROYECTO SUEÑALETRAS

Consiste en un conjunto de actividades tal que, sólo si se completa correctamente la palabra en Español señalada por la aplicación, aparece en la pantalla el video de la seña que la representa, así como se ilustra en las figuras 5 y 6. El problema de esta aplicación es que, además de ser limitada, está orientada sólo hacia oyentes que desean aprender la Lengua de Señas y por tanto es material que no aporta a la comunidad sorda.



FIGURA 5: ILUSTRACIÓN DE LA ACTIVIDAD "COMPLETAR UNA PALABRA" - PROYECTO SUEÑALETRAS

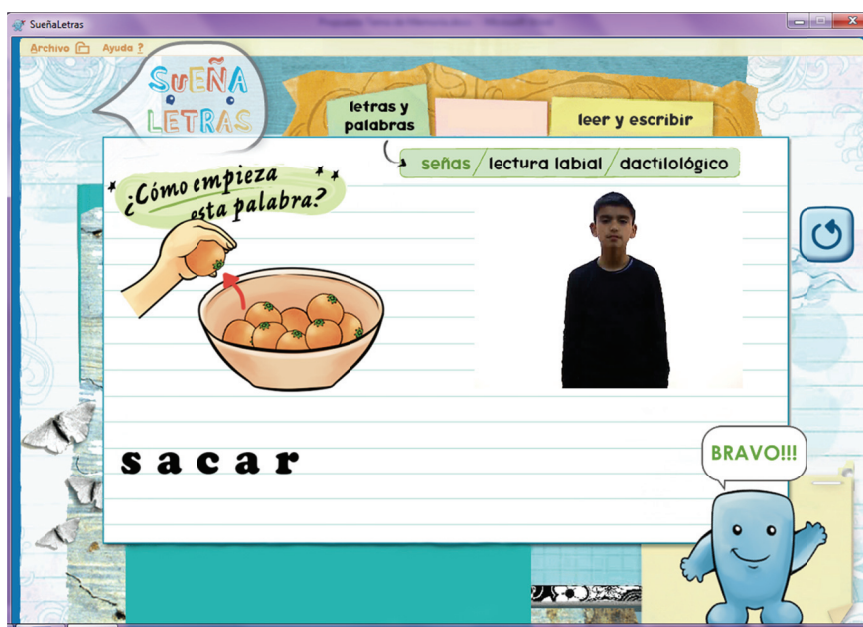


FIGURA 6: ILUSTRACIÓN DE UNA ACTIVIDAD REALIZADA CORRECTAMENTE – PROYECTO SUEÑALETRAS

2.4. DUOLINGO

En cuanto a proyectos internacionales, podemos encontrar, tanto en la web como para dispositivos móviles, cientos de aplicaciones que enseñan la Lengua de Señas. Sin embargo, dado que esta lengua no es universal, cada país desarrolla sus propias herramientas y no son útiles unas con otras pues se basan en la creación de contenido estático.

Debido a lo anterior surge la idea de desarrollar una herramienta colaborativa. Y, en este sentido, Duolingo [8] parece ser una buena comparación con lo realizado en este trabajo.

Duolingo es una aplicación gratuita destinada al aprendizaje de idiomas y a la traducción de textos basada en el concepto de *crowdsourcing*. En lo que al aprendizaje de idiomas se refiere, su diseño se estructura de acuerdo al formato de cursos en línea, los cuales son desarrollados colaborativamente por usuarios que demuestren su conocimiento en la materia en conjunto con profesionales que trabajan para la aplicación.

Cada curso consiste en una serie de actividades que permiten aprender palabras y frases según niveles de dificultad. Adicionalmente, tal como se puede apreciar en la figura 7, Duolingo cuenta con un apartado que permite revisar el vocabulario que se va estudiando a medida que se avanza en la aplicación. Esto podría considerarse similar al concepto de diccionario puesto que se muestra la traducción de dichas palabras, sin embargo, este material no constituye uno como tal, por considerarse acotado sólo a las palabras más relevantes trabajadas en las distintas actividades.



The screenshot shows the 'Palabras' section of the Duolingo web application. The header includes navigation tabs: Inicio, Palabras, Actividad, Foro, Inmersión. The user's profile 'LalSoong' is visible with a dropdown arrow, along with icons for a fire (0), a shield (22), and a bell (2). The main content area is titled 'Palabras que has aprendido en inglés' and shows '101 Palabras'. A table lists words with their grammatical categories, last practice date, and strength level. A tooltip for the word 'girls' shows the Spanish translation 'niñas' and 'chicas'. To the right, a 'Repaso espaciado' box explains the spaced repetition algorithm and shows a legend for strength levels: Bastante fuerte (4 bars), Bastante bueno (3 bars), Hora de practicar (2 bars), and Demasiado débil (1 bar).

Palabra	Categoría gramatical	Última práctica	Fuerza
vegetarian	Adjetivo	hace 7 meses	4 bars
you	Pronombre	hace 7 meses	3 bars
bird	Sustantivo	hace 7 meses	2 bars
the	Artículo	hace 7 meses	1 bar
girls	Sustantivo	hace 7 meses	3 bars
elephant	Sustantivo	hace 7 meses	2 bars
has	Verbo	hace 7 meses	1 bar
sandwich	Sustantivo	hace 7 meses	3 bars
book	Sustantivo	hace 7 meses	2 bars

FIGURA 7: SECCIÓN “PALABRAS” DE LA APLICACIÓN WEB DE DUOLINGO

3. DISEÑO GENERAL DE LA SOLUCIÓN

En esta sección se describen los requisitos, arquitectura y herramientas de software utilizadas en el desarrollo de la plataforma multimedia.

3.1. REQUISITOS GENERALES

Los principales requisitos considerados por la solución son los siguientes:

1. Las funcionalidades de la aplicación web son provistas por una API.
2. La API provee los mecanismos de validación de contenido.
3. Los puntajes para los usuarios que participen de la aplicación, ya sea aportando con material, descargando material o respondiendo desafíos, son gestionados por la API.
4. La aplicación web final tiene un diseño pensado principalmente para su uso por usuarios sordos, quienes requieren apoyo visual en todo momento al utilizar la aplicación.
5. La aplicación debe contar con 3 tipos de usuarios: sordos, oyentes y expertos.
6. La aplicación permite sólo a usuarios sordos subir videos representando palabras a la plataforma. Esto, por considerarse un acto de respeto a la cultura sorda el que el material en Lengua de Señas sea preparado por sordos.
7. La aplicación permite sólo a usuarios oyentes subir definiciones de palabras a la plataforma.
8. La aplicación permite sólo a usuarios expertos aceptar/rechazar palabras que se hayan agregado a la plataforma. Esto es, una vez que los usuarios sordos u oyentes incorporen nuevas palabras y a través de desafíos las hayan validado, son los profesionales los encargados de aceptar o rechazar esta validación. Esto asegura el control de calidad del contenido del diccionario.

El detalle de los requisitos de usuario y software asociados a la API y a la aplicación web, junto con la Matriz de Trazado respectiva, pueden verse en los Anexos 1, 2 y 3 del documento.

3.2. ARQUITECTURA FÍSICA DE LA SOLUCIÓN

La figura 8 muestra un esquema general de los componentes físicos que participan en la solución propuesta, así como la relación entre ellos. Estos componentes están separados según el modelo de 3 capas para aplicaciones web, el cual tiene como objetivo principal la separación de la lógica de negocio con la lógica de diseño de la aplicación, con el fin de permitir el desarrollo del sistema por niveles. Cada capa tiene las siguientes características:

1. **Capa de Presentación:** También llamada capa de usuario o interfaz gráfica. Es la encargada de comunicar y capturar la información del usuario. Se desarrolla sobre el navegador web y tiene un enlace sólo con el servidor de la aplicación.
2. **Capa de Negocio:** En ésta residen los programas que reciben las peticiones del usuario y envían sus respuestas tras este proceso. Aquí se establecen todas las reglas que debe seguir el sistema para su correcto funcionamiento. Esta capa se comunica con la capa de Presentación para interactuar con el usuario y con la capa de Datos para solicitar almacenar o recuperar datos de ésta.
3. **Capa de Datos:** En ésta residen los datos. Está formada por un gestor de base de datos que recibe solicitudes de almacenamiento o recuperación de información desde la capa de Negocio.

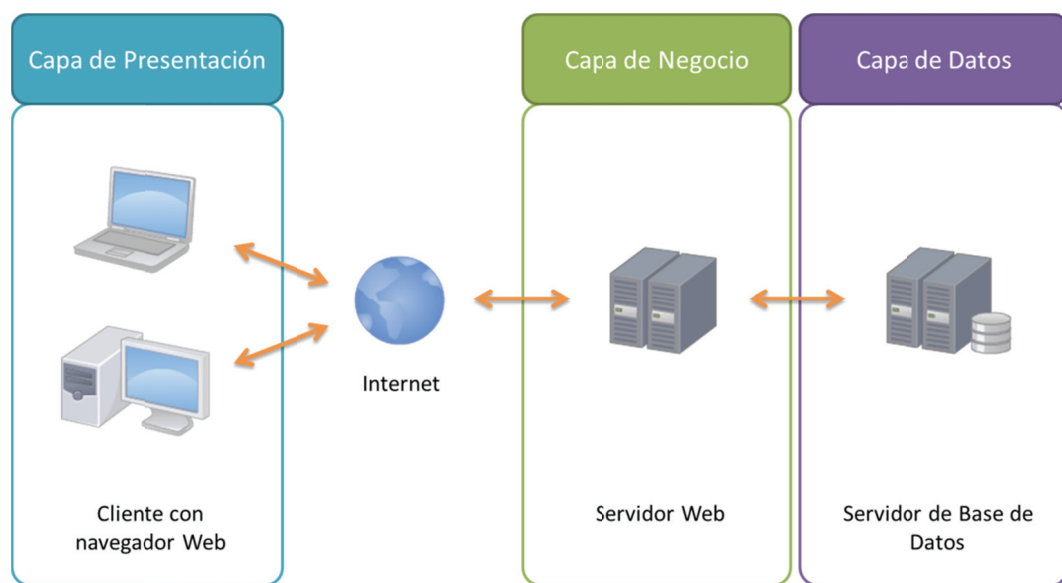


FIGURA 8: ESQUEMA GENERAL DE LA ARQUITECTURA FÍSICA DEL SISTEMA

Siguiendo el modelo descrito anteriormente, se requiere que el cliente/usuario acceda a los recursos del servidor a través del protocolo HTTP utilizando un navegador web compatible, tal como Internet Explorer (versión 7+), Google Chrome (versión 30+), Safari (versión 5+) y/o Mozilla Firefox (versión 4+). Por su parte, tanto la capa de negocio como la de datos residen sobre un único servidor perteneciente al Instituto de la Sordera INDESOR, el cual consta de un servidor web Apache Tomcat (versión 2.4.10), junto con el *framework* Django (versión 1.8) y un módulo MySQL (versión 5.5.34-cll-lve) para la gestión de la base de datos.

3.3. ARQUITECTURA LÓGICA DE LA SOLUCIÓN

La API desarrollada está organizada en componentes y módulos. Los componentes agrupan funcionalidades que dan soporte a los módulos, siendo éstos últimos entendidos como un conjunto de funcionalidades dirigidas a un perfil de usuario específico. La estructura de la aplicación se representa en la siguiente figura:

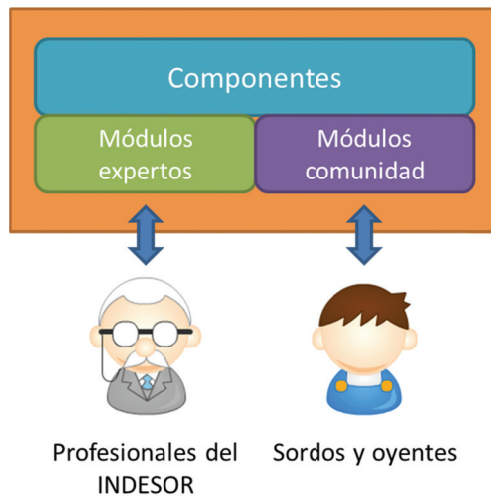


FIGURA 9: DIAGRAMA COMPONENTES Y MÓDULOS

Como se puede observar, la figura 9 muestra 2 tipos principales de usuarios que interactúan con la plataforma web:

1. Profesionales del INDESOR, encargados de aceptar/rechazar el contenido del diccionario multimedia.
2. Usuarios sordos y oyentes, que además de ser los usuarios a quienes está destinado el diccionario multimedia, son los encargados de subir material a la plataforma y de la validación de nuevas palabras a través de desafíos.

Las funciones principales de cada tipo de usuario sobre la plataforma se muestran en las figuras 10, 11 y 12:

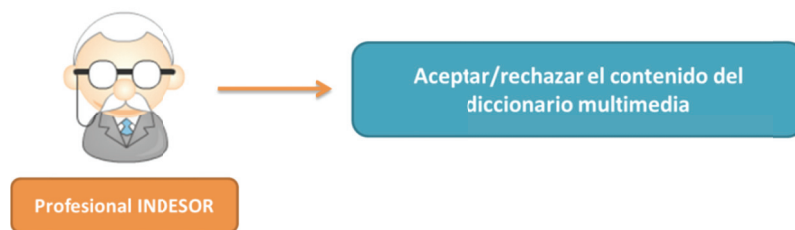


FIGURA 10: FUNCIONES PRINCIPALES DE UN USUARIO EXPERTO DENTRO DE LA PLATAFORMA

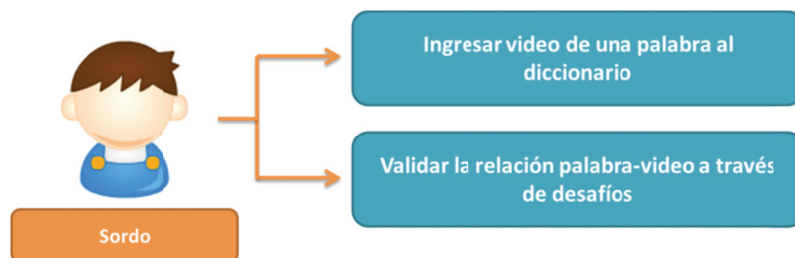


FIGURA 11: FUNCIONES PRINCIPALES DE UN USUARIO SORDO DENTRO DE LA PLATAFORMA

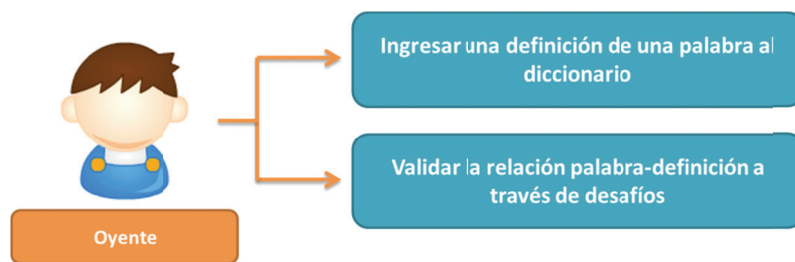


FIGURA 12: FUNCIONES PRINCIPALES DE UN USUARIO OYENTE DENTRO DE LA PLATAFORMA

3.3.1. COMPONENTES DE LA SOLUCIÓN

La figura 13 muestra los componentes de la API. A continuación se procede a describir sus funcionalidades:

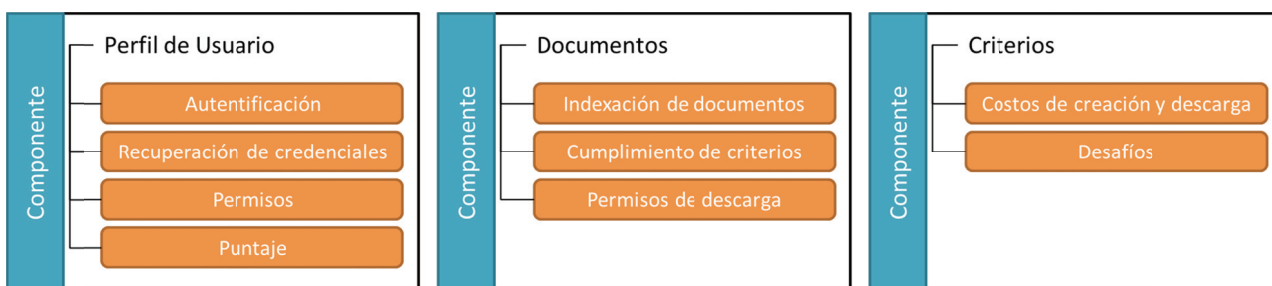


FIGURA 13: COMPONENTES DE LA API

1. **Componente Perfil de Usuario:** Este componente es el encargado de manejar principalmente la autenticación del usuario cuando ingresa a la plataforma y los permisos asociados a éste dependiendo si es Profesional del INDESOR, sordo u oyente, teniendo acceso a un conjunto determinado de funcionalidades respectivas a su perfil.
2. **Documentos:** Un documento en la aplicación está compuesto por un video, una palabra y su definición respectiva. De esta manera, un video y una definición pertenecen únicamente al criterio de la palabra que los define. En adición, esta componente maneja los permisos de descarga sobre cada documento que tienen los usuarios dependiendo si han desbloqueado o no su contenido.
3. **Criterios:** Cada palabra de la aplicación representa un criterio a ser preguntado en un desafío. En el caso de los usuarios sordos, un desafío consiste en preguntar qué texto en Español, de entre un listado de 4, representa la palabra que muestra un video de una seña. En el caso de los usuarios oyentes, éste consiste en preguntar qué definición, de entre un listado de 4, representa el texto en Español que se muestra. El cumplimiento de un desafío de manera correcta permite la validación del material, mientras el usuario gana puntos que le permiten desbloquear contenido dentro de la plataforma.

3.3.2. MÓDULOS DE LA SOLUCIÓN

Los componentes antes descritos brindan funcionalidades a los módulos de la aplicación web, detallados en la figura 14:

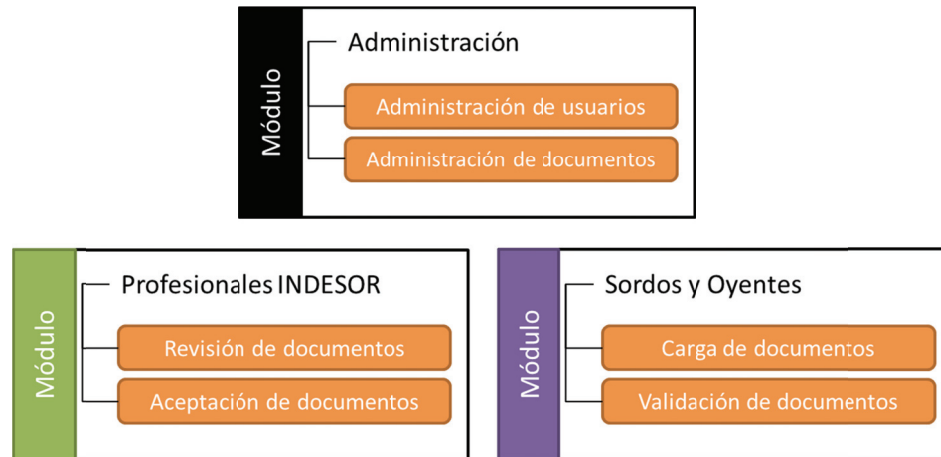


FIGURA 14: MÓDULOS DE LA APLICACIÓN WEB

El módulo Administración está destinado exclusivamente a aquel usuario que tenga el control sobre la API, ya sea para modificarla en caso de añadir, eliminar o cambiar funcionalidades. Por su parte, los módulos restantes describen las acciones a realizar por los distintos usuarios de la plataforma.

3.3.3. MODELO DE DATOS DE LA SOLUCIÓN

La figura 15 muestra el modelo de la base de datos relacional creada para la API de este proyecto. De acuerdo a este modelo, cabe destacar lo siguiente:

- La tabla de usuarios (*User*) utilizada viene por defecto en el *framework* utilizado. Es por ello que se creó una extensión de esa tabla llamada *Profile* que mantiene otros datos del usuario.
- Para un usuario se mantiene un registro de las palabras que ha descargado/visualizado dentro de la plataforma, de tal forma de poder acceder a éstas cuando lo desee (tabla *Log*).
- La aplicación mantiene un historial de los usuarios que han participado en la creación de una palabra en la tabla *Record*, con el fin de recompensarlos cuando una palabra es aceptada por un experto.
- La tabla *Fulfill* contiene la información que permite a los usuarios expertos decidir si una palabra se agrega o no al diccionario.
- Un documento puede estar compuesto por más de un archivo de tipo video puesto que una palabra puede ser representada por más de una señal.

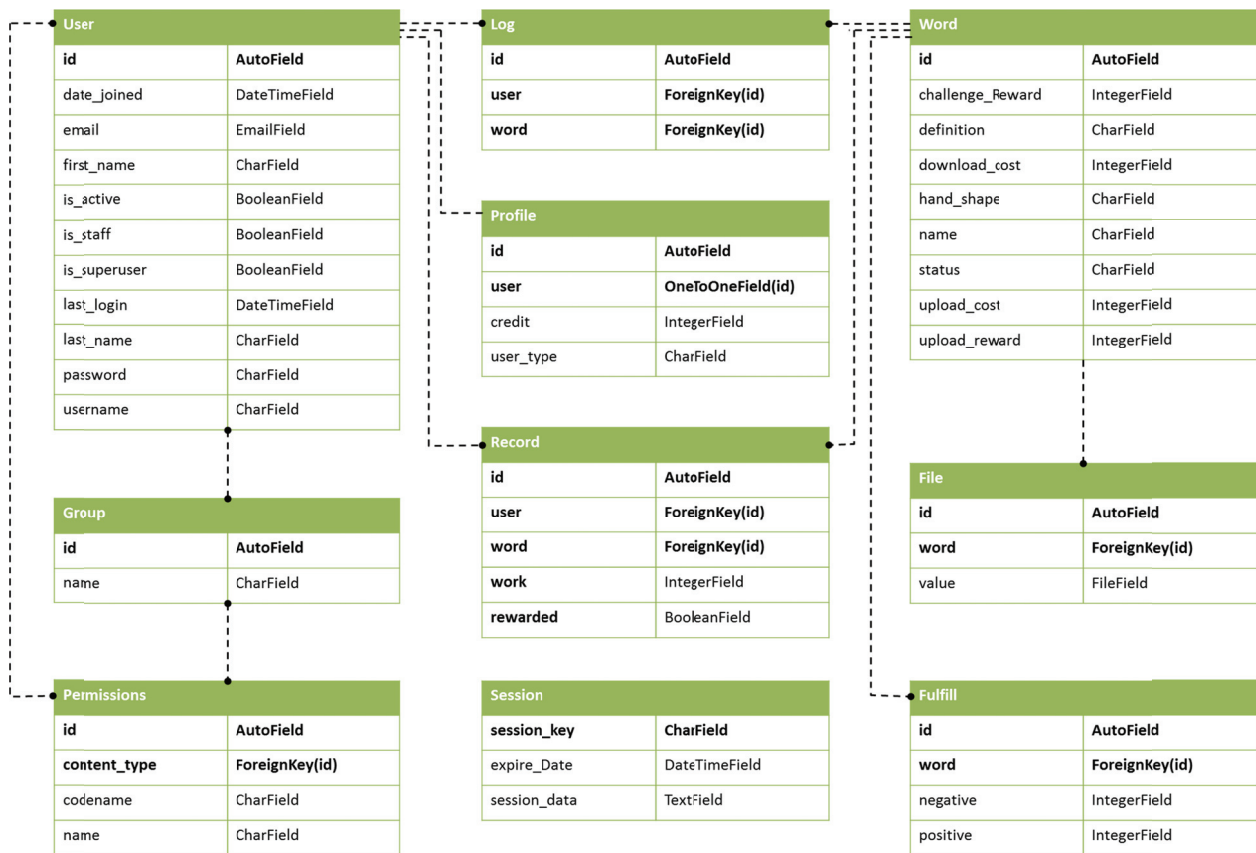


FIGURA 15: MODELO DE DATOS DE LA API

3.4. ARQUITECTURA DE UNA APLICACIÓN WEB

Se utilizó el patrón MVT (*Model-View-Template*) en el desarrollo de este proyecto. MVT es una arquitectura muy parecida a la conocida arquitectura MVC (*Model-View-Controller*), con la diferencia que en MVT la capa que actúa como controlador es la vista, y para presentar los datos al usuario final se ocupa un sistema de *templates* general para toda la aplicación, al contrario de las vistas acopladas al controlador de la arquitectura MVC convencional. Este patrón permite estructurar la aplicación en 3 capas:

- El **modelo** que representa los datos de la aplicación.
- El **template o plantilla** que despliega la presentación de los datos del modelo.
- La **vista** que maneja las peticiones realizadas por el cliente.

La separación que presenta la arquitectura MVT nos permite distribuir el trabajo entre el diseño gráfico y el desarrollo de la lógica de la aplicación, facilitando la reutilización y mantenimiento del código. Esta división en capas sitúa a dos de ellas por el lado del servidor, modelo y vista, mientras que la plantilla se encuentra asociada al cliente, correspondiente al navegador web. De esta forma, para desarrollar del lado del servidor es necesario conocer

lenguajes tales como Python y Django, mientras que para el desarrollo de las vistas es preciso manejar HTML, CSS y JavaScript.

Para el desarrollo de la aplicación se utilizó el *framework* MVT Django. Un *framework* es una estructura conceptual y tecnológica sobre la que un proyecto de software puede ser más fácilmente organizado e implementado.

El ciclo de una petición en Django se puede observar en la siguiente figura:

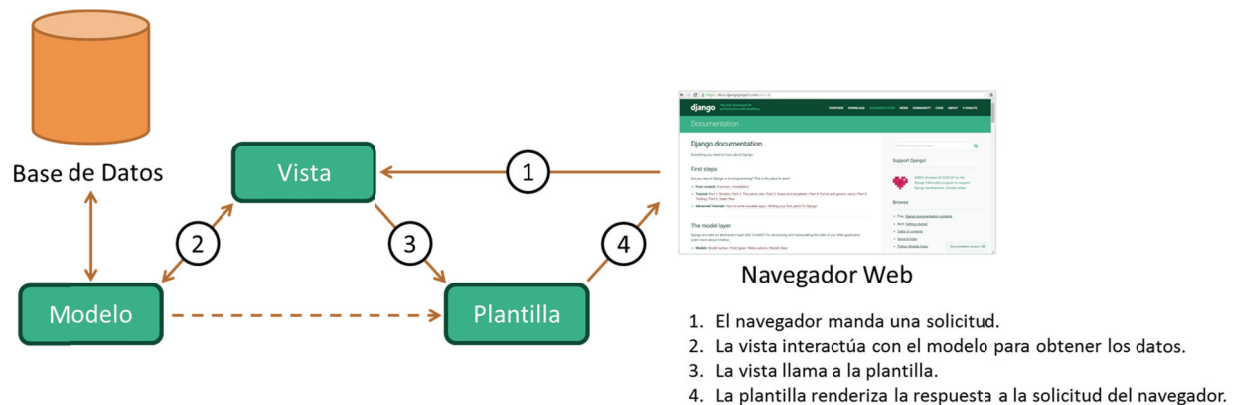


FIGURA 16: DIAGRAMA DE UNA PETICIÓN EN DJANGO

3.5. HERRAMIENTAS DE DESARROLLO

A continuación se presentan las herramientas utilizadas para la construcción de la API y de la aplicación web:

- **Ambiente de desarrollo:** Para el desarrollo de la API y de la aplicación web, se escogió el uso del lenguaje de programación Python en conjunto con el *framework* MVT Django v1.8.

Si bien existen otras herramientas de desarrollo en el mercado, se escogió la opción mencionada anteriormente dada la sintaxis compacta, ordenada, legible y simple de Python. Por otra parte, Django cuenta con una licencia BSD, lo que se traduce en poder utilizar, cambiar y extender su código fuente de manera gratuita. Esto ha fomentado el crecimiento de la comunidad en torno a este *framework* y por ende la disponibilidad de *plugins*, *templates*, librerías y soluciones a los problemas más comunes a los que se enfrenta un desarrollador al usar esta herramienta.

- **Servidor web:** Para el ambiente de producción, se utilizó el servidor del INDESOR.
- **Motor de base de datos:** Como motor de base de datos se escogió MySQL pues es la base de datos que trae el servidor, y además provee todas las funcionalidades necesarias para el desarrollo de la aplicación.

- **Control de versiones:** Para el control de versiones se utilizó Git, ya que ofrece todas las funcionalidades necesarias para mantener un desarrollo ordenado del software. En adición, se utilizó la metodología Git-Flow para facilitar la gestión del flujo de trabajo.

En adición, para la creación de la aplicación web, se utilizaron las siguientes tecnologías:

- **Bootstrap 3:** *Framework* CSS + Javascript utilizado para la construcción de las vistas.
- **JQuery:** Para la interactividad de las interfaces.

4. IMPLEMENTACIÓN DE LA SOLUCIÓN

En este capítulo se describe detalladamente la implementación de la solución expuesta en este documento.

4.1. DISEÑO E IMPLEMENTACIÓN DE LA API

Para la creación del diccionario multimedia implementado, se optó por el desarrollo de una API genérica que permite a la aplicación web final utilizar estrategias de *crowdsourcing* y *collective quality control* para la validación y alimentación de contenido. De esta forma, y dado que la aplicación final quedó en manos del INDESOR, cualquier cambio que quieran hacer en la aplicación web será transparente a los mecanismos implementados en esta API. Inclusive, si se desea en un futuro crear una aplicación móvil, proveer las funcionalidades de manera transparente al cliente facilitará este proceso.

4.1.1. ARQUITECTURA DE SOFTWARE: REST API

La API se desarrolló siguiendo la arquitectura de software REST (*Representational State Transfer*), la cual consiste en directrices y buenas prácticas para la creación de servicios web escalables.

Una REST API es una API accedida a través del protocolo HTTP, mediante direcciones web o URL's en las que se envían los datos de la consulta. Como respuesta a la consulta, se obtienen los datos en diferentes formatos, ya sea texto plano, XML, o en el caso de este proyecto, JSON. JSON (*JavaScript Object Notation*) es un formato estándar abierto, que utiliza texto legible para transmitir objetos de datos (pares atributo-valor) entre una aplicación y un servidor web.

La REST API tiene 3 niveles de calidad en su desarrollo:

1. Uso correcto de URL

Al desarrollar una aplicación web, las URL's permiten acceder a cada una de las páginas, secciones o documentos del sitio. Se denomina recurso a cada una de estos elementos pues es la información a la que se quiere acceder.

Las URL (*Uniform Resource Locator*) son un tipo de URI (*Uniform Resource Identifier*) que, además de permitir identificar de forma única el recurso, nos permite localizarlo para poder acceder a él o compartir su ubicación.

Una URL se estructura de la siguiente forma:

```
{protocolo}://{dominio}[:puerto(opcional)]/{ruta del recurso}?{consulta}
```

Existen reglas básicas para ponerle nombre a la URL de un recurso:

- Las URL deben ser únicas, no se debe tener más de una para identificar un mismo recurso.
- Los nombres de URL no deben implicar una acción, por lo tanto debe evitarse usar verbos en ellos. Por ejemplo, si se desea obtener la palabra de identificador 123 del diccionario multimedia, una URL válida para realizar esta acción es la siguiente:

```
HTTP://localhost:8000/palabras/123
```

- Las URL deben ser independientes de formato, luego, a pesar de saber que el formato de respuesta a una consulta es JSON, sería incorrecto poner en la URL esta extensión.
- Las URL deben mantener una jerarquía lógica. Esto es, si se desean obtener los videos de la palabra de identificador 123 (teniendo en cuenta que una palabra puede estar representada por más de una seña y por tanto tener asociada más de un video), el orden lógico para hacer esta consulta en la URL es:

```
HTTP://localhost:8000/palabras/123/videos
```

Mientras que una forma incorrecta de representar la misma consulta sería:

```
HTTP://localhost:8000/videos/palabras/123
```

- Los filtros de información de un recurso no se hacen en la URL. Para filtrar, ordenar o buscar información de un recurso, se debe hacer una consulta sobre la URL utilizando parámetros HTTP. Luego, si se desea obtener el listado de palabras validadas y ordenadas alfabéticamente, la URL correcta en este caso sería:

```
HTTP://localhost:8000/palabras?status=validada&orden=alfabetico
```

2. Uso correcto de HTTP

Para desarrollar una REST API, es importante tener claros los siguientes aspectos:

- Métodos HTTP: Para manipular los recursos, HTTP cuenta con los siguientes métodos:
 - GET: Para consultar y leer recursos.
 - POST: Para crear recursos.
 - PUT: Para editar recursos.
 - DELETE: Para eliminar recursos.
 - PATCH: Para editar partes concretas de un recurso.

- Códigos de estado: Cuando se realiza una operación, es necesario saber si dicha operación se ha realizado con éxito o en caso contrario, porqué ha fallado. Para ello, el protocolo HTTP provee una lista de códigos de respuesta y frases estándar asociadas, destinadas a dar una descripción corta del status de la solicitud.

3. Implementación de *Hypermedia*

El concepto de *Hypermedia* se refiere a la conexión mediante hipervínculos de la aplicación cliente con la API, permitiendo a dicho cliente despreocuparse por conocer de antemano el cómo acceder a los recursos. La API entonces provee una colección de vínculos para el cliente, permitiendo que éste cambie la aplicación de un estado a otro a través de éstos, haciendo de la aplicación una web navegable.

4.1.2. METODOLOGÍA DE PROGRAMACIÓN: TEST DRIVEN DEVELOPMENT

Para alcanzar un buen estándar de calidad en la construcción de la API, el proceso de desarrollo se llevó a cabo siguiendo la metodología Test Driven Development (TDD) que consiste en ciclos cortos de iteración, siguiendo la técnica de *refactoring* usando test unitarios. Este proceso se basa en la creación de uno o varios test iniciales que definen el comportamiento de la funcionalidad que se desea implementar. A continuación, se escribe el código más simple posible que sea capaz de pasar los test. Este proceso se repite refactorizando el código hasta alcanzar un nivel de código aceptable.

El uso de TDD permite construir software de alta calidad, ya que cada test creado actúa como un contrato de funcionamiento. De este modo, si un nuevo cambio introduce algún error en alguna funcionalidad ya existente, el problema se detecta al ejecutar los test correspondientes.

4.2. DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN WEB DICCIONARIO MULTIMEDIA INDESOR

El funcionamiento de la aplicación web se basa en el equilibrio de las acciones que pueden realizar los usuarios en la plataforma. Para ello, existe una moneda de cambio (👍) que permite al usuario:

- Pagar por visualizar palabras del diccionario. Este pago se hace efectivo sólo la primera vez que accede al contenido de una palabra.
- Pagar por subir material al diccionario. Esta limitante se creó con el fin de concientizar al usuario tal que suba material relevante y en cantidades razonables a la aplicación.

Además, el usuario gana puntaje:

- Respondiendo a desafíos correctamente.

- Si el material que subió fue aceptado por un experto.
- En el caso de ser un usuario experto, por aceptar/rechazar palabras.

Esta modalidad genera una dinámica que permite una intervención equitativa de los usuarios en la aplicación, generando además incentivos para que estos últimos participen activamente dentro de la plataforma.

A continuación se procede a detallar los principales aspectos de la implementación del diseño expuesto, mostrando algunas funcionalidades en pseudocódigo asociadas a las pantallas con mayor relevancia de la aplicación.

4.2.1. DISEÑO GENERAL DE LAS VISTAS DE LA APLICACIÓN

La aplicación web se desarrolló para su visualización tanto en equipos de escritorio y portátiles, como en dispositivos móviles. Esto gracias a la utilización del *framework* Bootstrap 3, el cual provee herramientas para la optimización de esta funcionalidad. En la figura 17 se muestra a la izquierda la vista desde un equipo de escritorio y a la derecha, la vista desde un dispositivo móvil.



FIGURA 17: SECCIÓN “CÓMO FUNCIONA” DE LA APLICACIÓN

En adición, se creó un diseño tal que en cada sección exista un video explicativo del contenido de la página que se está visitando, con el fin de asegurar el entendimiento por parte de los usuarios sordos de todos los textos, formularios y elementos generales de la aplicación.

4.2.2. LOGIN Y REGISTRO DE USUARIOS

El manejo de usuarios está implementado en base al uso de sesiones y al sistema de autenticación que provee Django. Este último consta de los siguientes componentes:

- **Usuarios:** Personas registradas en la aplicación web.

- **Permisos:** Valores binarios que indican si un usuario puede o no realizar una tarea determinada.
- **Grupos:** Forma genérica de aplicar permisos a más de un usuario.
- **Perfiles:** Mecanismo que permite extender los objetos de tipo usuario con campos adicionales.

En adición, este sistema implica tratar con 2 procesos:

- Verificar las credenciales del usuario, mediante la comprobación de su nombre de usuario y contraseña contra una tabla en la base de datos.
- Verificar que el usuario está autorizado a realizar una operación determinada, comprobando su acción en una tabla de permisos.

El ingreso de usuarios se realiza mediante un formulario al cual se accede desde el *header* de la aplicación, según se observa en la figura 18:

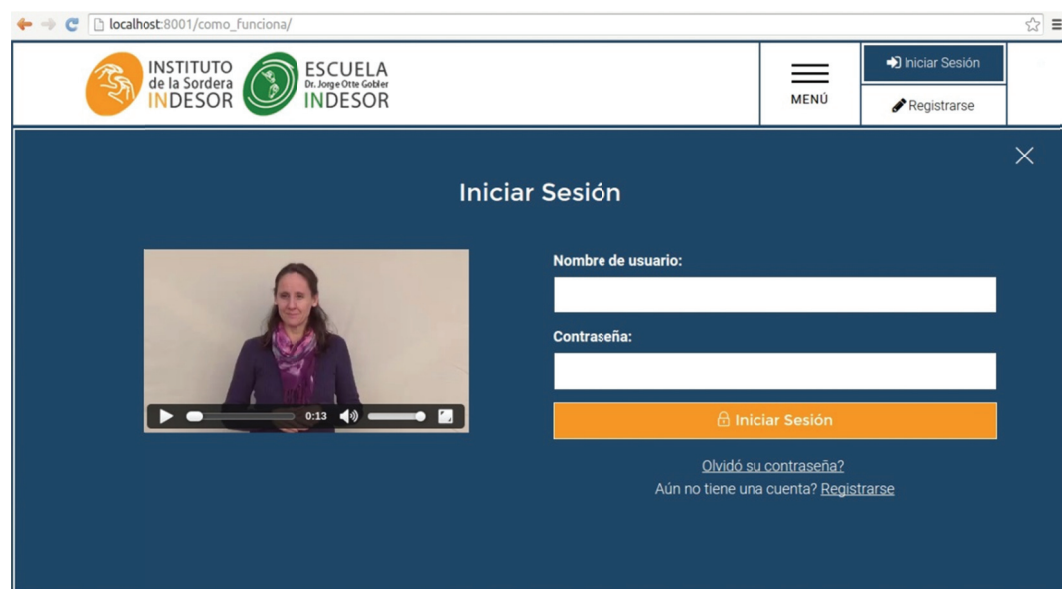


FIGURA 18: PANTALLA DE INGRESO A LA APLICACIÓN

Para el proceso de inicio de sesión, se hace una llamada a la función `log_in()` de la API a través del método `request.post` con los parámetros `username` y `password` provenientes del formulario. Si el `status` de la solicitud es `'ok'`, entonces el usuario está correctamente autenticado y se almacena su identificador en la sesión. El siguiente pseudocódigo muestra un extracto de esta función:

```
def log_in(request):
    params = {
        'username': request.POST['username'],
        'password': request.POST['password']
    }
```

```

r = requests.post('.../api/v1.0/users/login/', data = params)

if r.json()['status'] == 'ok':
    request.session['...'] = ...
    return HttpResponseRedirect(...)
else:
    return render_to_response(...)

```

La API por su parte, se encarga del proceso mismo de autenticación. Para ello, utiliza la función `authenticate()` con los parámetros `username` y `password` provistos en la llamada, la cual devuelve un objeto de tipo `User` si la contraseña es correcta para el nombre de usuario ingresado. En caso de fallar la autenticación, la función `authenticate()` devolverá `None`.

A continuación se muestra un extracto de la función `log_in()` de la API:

```

def log_in(request):
    response = {}

    user = authenticate(
        username = request.POST['username'],
        password = request.POST['password'])

    if user is not None:
        response['status'] = 'ok'
        ...
    else:
        response['status'] = 'error'
        response['error'] = 'Username or password incorrect'

    return
    HttpResponse(json.dumps(response), content_type='application/json')

```

Una vez autenticado el usuario, se procede a determinar su perfil para cargar el menú de opciones en la vista correspondiente, de acuerdo a los permisos que se le han establecido. Por ejemplo, un usuario sordo sólo tiene permisos para ejecutar las acciones de subir videos y completar desafíos dentro de la aplicación, tal como se observa en la figura 19.

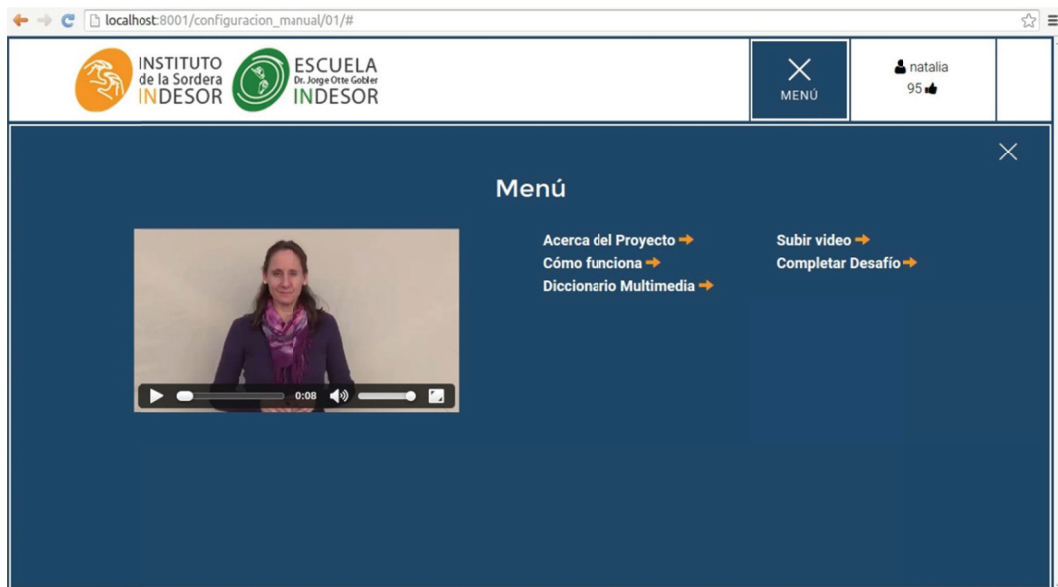


FIGURA 19: MENÚ DE OPCIONES PARA LOS USUARIOS SORDOS DE LA APLICACIÓN

Para el cierre de sesión es necesario llamar a la función `log_out ()` dentro de una vista de la aplicación web (figura 20), cuya única función es limpiar los datos de la sesión del usuario y redirigirlo a una página que informe que la acción se ha ejecutado correctamente. Esta función se muestra en el siguiente pseudocódigo:

```
def log_out (request) :
    auth.logout (request)
    return HttpResponseRedirect (...)
```



FIGURA 20: PANTALLA DEL CIERRE DE SESIÓN PARA UN USUARIO SORDO

Por su parte, la creación de usuarios se hace a través de una vista (figura 21) que solicita los datos que la aplicación requiere según los atributos de la clase `User` que provee el sistema de autenticación de Django. Los principales atributos de esta clase son: `first_name`, `username`, `email` y `password`.

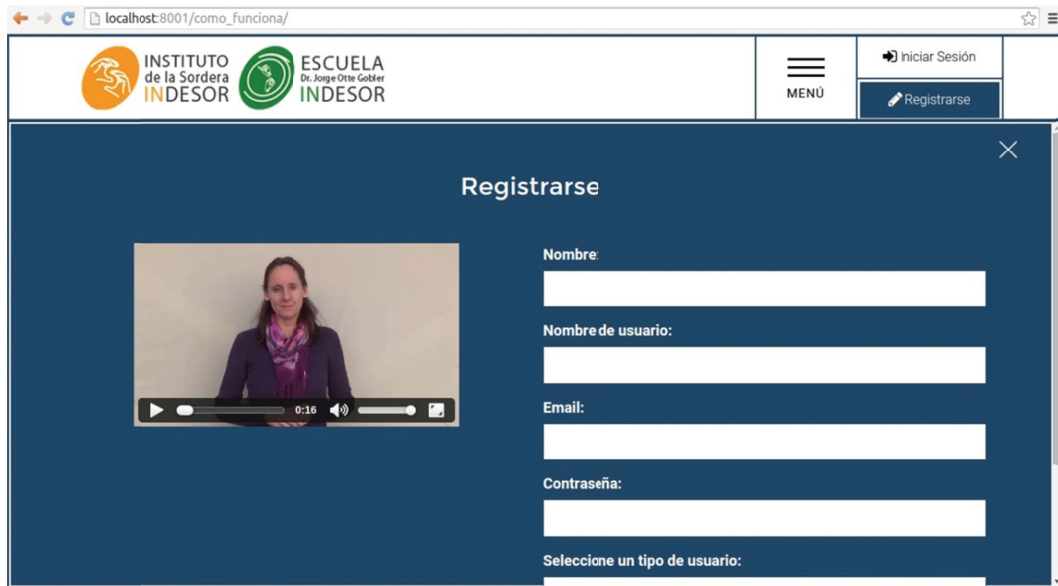


FIGURA 21: PANTALLA DE REGISTRO DE USUARIOS DE LA APLICACIÓN

La función `sign_up()` de la aplicación web se encarga de recoger la información introducida por el usuario a través del formulario de registro. Si se produce alguna excepción al registrar al usuario, tal como el ingreso de un nombre de usuario existente en la base de datos, el usuario es entonces informado. A continuación se muestra un extracto de esta función:

```
def sign_up(request):
    params = {
        'first_name': request.POST['first_name'],
        'username': request.POST['username'],
        'email': request.POST['email'],
        'password': request.POST['password'],
        'user_type': request.POST['user_type']}

    r = requests.post('.../api/v1.0/users/', data = params)

    if r.json()['status'] == 'ok':
        request.session['...'] = ...
        return HttpResponseRedirect(...)
    else:
        if r.json()['error'] == "Username already exists":
            ...
```

```
return render_to_response(...)
```

La llamada a la API genera la creación misma del usuario en la base de datos. Para ello, se utiliza la función `create_user()` del sistema de autenticación, creando una instancia de la clase `User`. La función de la API que hace el llamado a `create_user()` es la siguiente:

```
def sign_up(request):
    response = {}
    first_name = request.POST['first_name']
    username = request.POST['username']
    email = request.POST['email']
    password = request.POST['password']
    user_type = request.POST['user_type']

    #Validaciones
    ...

    user = User.objects.create_user(
        first_name = first_name,
        username = username,
        email = email,
        password = password)

    profile = Profile.objects.create(...)

    response['status'] = 'ok'
    return
    HttpResponse(json.dumps(response), content_type='application/json')
```

Si el objeto `User` ha sido satisfactoriamente creado en la base de datos, entonces la API responde con un `status 'ok'` a la solicitud del usuario y la aplicación web se encarga de iniciar la sesión de éste, de acuerdo a su perfil y permisos asociados dentro de la plataforma.

4.2.3. DICCIONARIO MULTIMEDIA

En el marco de facilitar la usabilidad de la aplicación web para los usuarios sordos, el Diccionario Multimedia INDESOR contempla la creación de vistas que permiten la búsqueda de contenido ya sea según el orden alfabético de las palabras, o según la configuración manual de las señas que las representan, entendiéndose esto último como un modo de organizar el contenido basado en la forma que toman las manos al momento de realizar una seña. Las figuras 22 y 23 muestran las vistas en la aplicación según los 2 tipos de organización de las palabras dentro del diccionario:

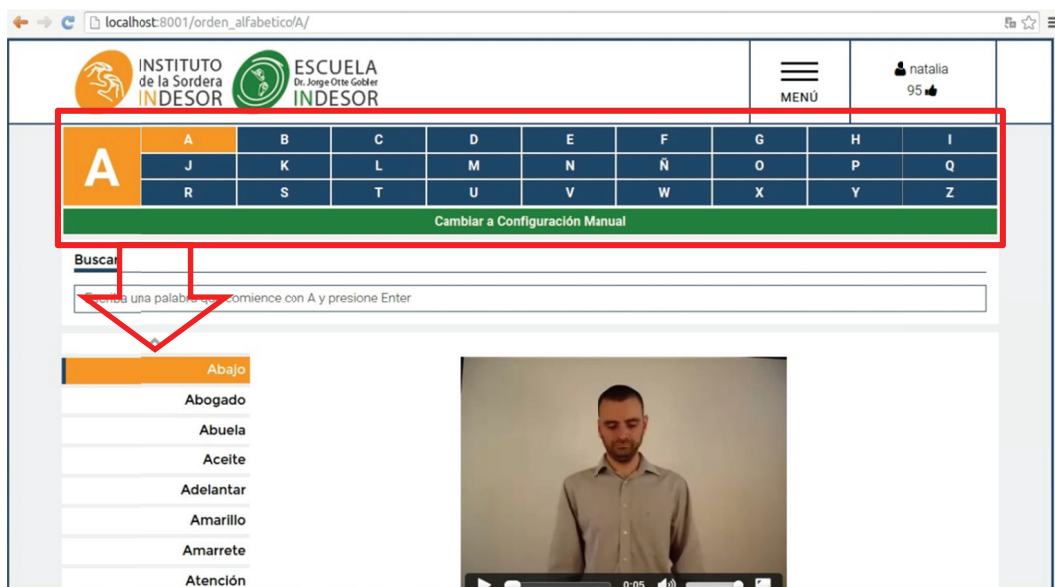


FIGURA 22: DICCIONARIO MULTIMEDIA ORGANIZADO ALFABÉTICAMENTE

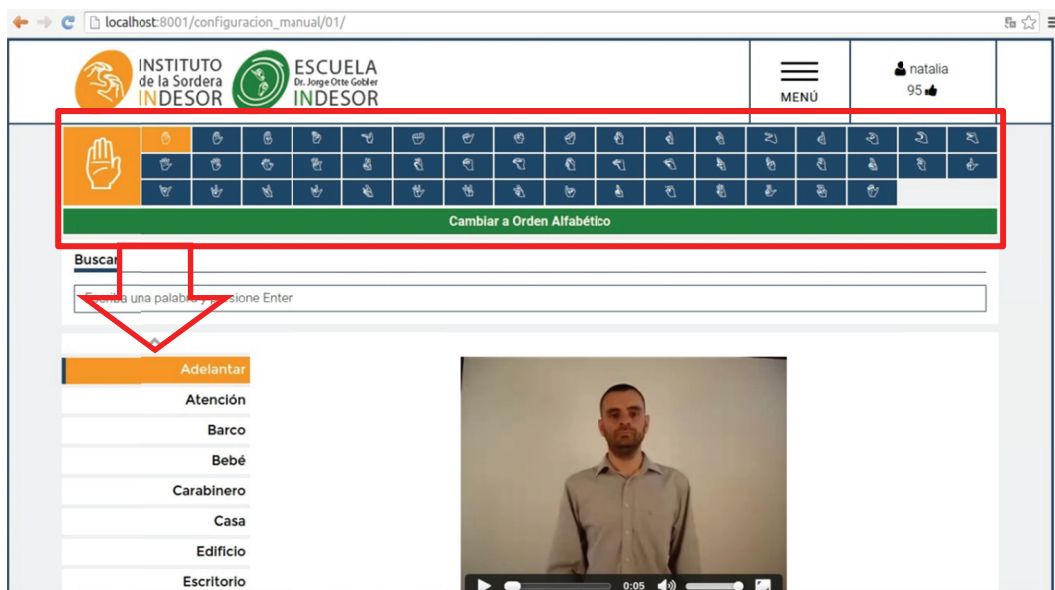


FIGURA 23: DICCIONARIO MULTIMEDIA ORGANIZADO SEGÚN LA CONFIGURACIÓN MANUAL

Un estudio realizado en el departamento de Educación Diferencial de la Universidad Metropolitana de Ciencias de la Educación [2], muestra que existen 49 configuraciones manuales dentro de la Lengua de Señas chilena, las cuales se dibujaron vectorialmente y se transformaron a una fuente de íconos para su uso en el presente trabajo (figura 24).



FIGURA 24: CONFIGURACIONES MANUALES EN DIBUJO VECTORIAL

Para obtener el listado de palabras validadas para una configuración manual, la aplicación llama a la API con la siguiente función, la cual recibe la configuración manual como parámetro:

```
def words(request, hand_shape):
    r = requests.get('.../api/v1.0/words/validated/?handshape=' + hand_shape)
    list_of_words = r.json()['data']
    return render_to_response(...)
```

El request, mediante una función GET, obtiene todas las palabras solicitadas, siendo la API la encargada de entregar junto con la palabra, la definición y los videos asociados a ésta en la variable data. El siguiente extracto de código muestra esta función:

```
def words(request):
    response = {}
    handshape = request.GET['handshape']

    words = Word.objects.filter(hand_shape = handshape).filter(status =
        'VA').order_by('name')

    response['status'] = 'ok'
    words_array = []

    for i in words:
        word = {}
        word['name'] = i.name
        word['definition'] = i.definition
```

```

files = File.objects.filter(word = i)
files_array = []
for file in files:
    files_array.append(file.value.url)
word['files'] = files_array
words_array.append(word)

response['data'] = words_array
return
HttpResponse(json.dumps(response), content_type='application/json')

```

El caso para obtener las palabras asociadas a una letra del abecedario es simétrico.

4.2.4. CREACIÓN DE ARCHIVOS

Por respeto a la cultura sorda, un requisito fundamental sobre el diccionario multimedia es que sean sólo usuarios sordos quienes suban videos con señas que representan palabras a la plataforma. Debido a esto, y con el fin de crear un rol distinto y de un nivel de importancia equivalente para los usuarios oyentes, es que sólo estos últimos tienen permitido subir definiciones.

Lo anterior crea un flujo con distintos estados para una palabra dentro de la aplicación, el cual se muestra en la siguiente figura:

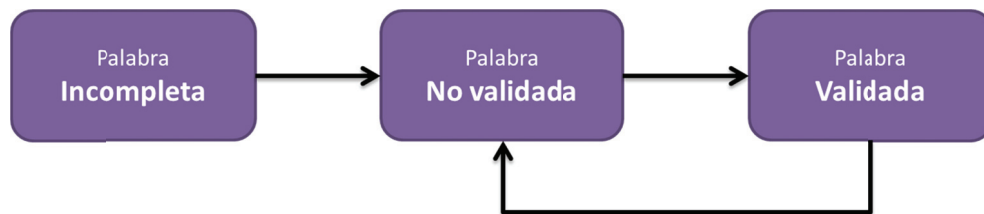


FIGURA 25: ESTADOS DE UNA PALABRA EN LA APLICACIÓN

Los usuarios tienen la siguiente participación dentro de este flujo:

1. Un usuario sordo crea una palabra, subiendo su nombre, configuración manual y video con la seña que la representa. Una vez realizada esta acción, la palabra queda en el estado “Incompleta”.
2. Un usuario oyente toma la palabra anterior y le agrega la definición. En este momento, la palabra pasa a un estado en el que ya se le han agregado todas sus propiedades y está a la espera de ser validada por los usuarios. Ahora se encuentra entonces en el estado “No validada”.
3. Una vez que los usuarios (sordos y oyentes) han validado la palabra a través de desafíos, un usuario experto se encarga de revisar esta validación y, de estar

correcta, añade la palabra al diccionario. Ocurrido lo anterior, la palabra queda en estado “Validada” y recién en este punto es visible en el diccionario de la aplicación.

4. Un usuario sordo tiene además la opción de subir un video a una palabra previamente validada (puesto que una palabra puede ser representada por más de una seña). De ocurrir esto, la palabra vuelve a quedar en estado “No validada”, a la espera que los usuarios validen la nueva seña que se ha subido.

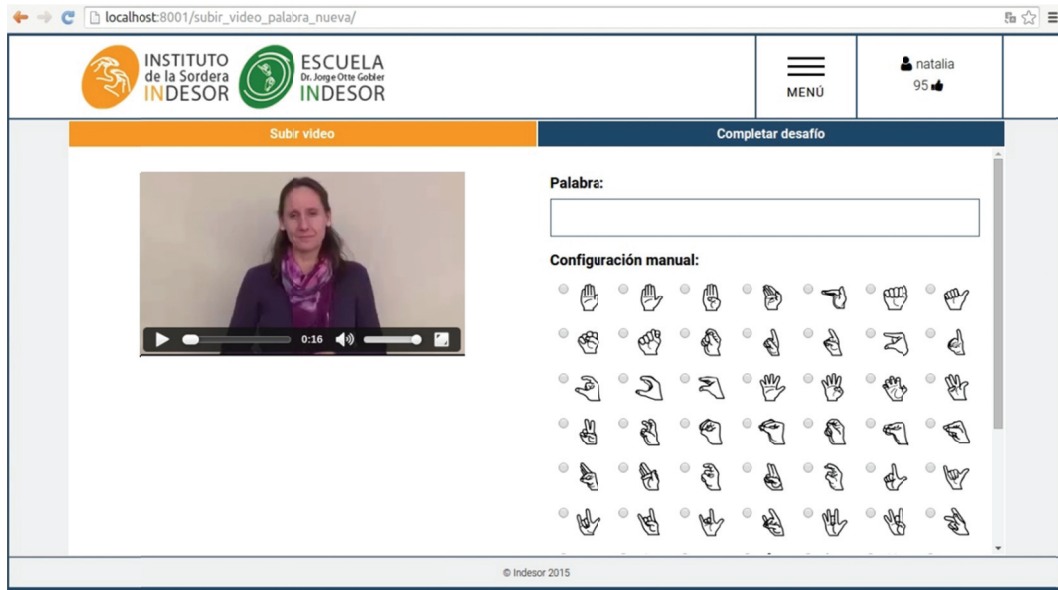


FIGURA 26: FORMULARIO DE INGRESO DE UNA PALABRA AL DICCIONARIO MULTIMEDIA

Para subir una palabra nueva al diccionario, la aplicación web recibe del formulario (figura 26) los parámetros `word` y `hand_shape` junto con el video con la seña y, a través de la siguiente función, los envía a la API:

```
def upload_video_new_word(request):
    word_name = request.POST['word']
    hand_shape = request.POST['hand_shape']
    for filename, file in request.FILES.iteritems():
        files = {'file': file}

    params = {
        'word_name': word_name,
        'hand_shape': hand_shape}

    r = requests.post('.../api/v1.0/word/files_post/', data=params,
                     files=files)

    if r.json()['status'] == 'ok':
        result = "The file has been uploaded."
```

```
return render_to_response(...)
```

La función principal de la API es almacenar el video en el servidor luego de crear la palabra correspondiente. Un extracto de esta función se muestra a continuación:

```
def upload_video_new_word(request):
    response = {}
    word_name = request.POST['word_name']
    hand_shape = request.POST['hand_shape']
    word = Word(
        name = word_name,
        hand_shape = hand_shape,
        status = 'IN'
    )
    word.save()

    for filename, file in request.FILES.iteritems():
        File.objects.create(word=word,value=file)

    response['status'] = 'ok'
    return
    HttpResponse(json.dumps(response), content_type='application/json')
```

Una vez que un usuario sordo ha creado una palabra, el usuario oyente puede subir la definición de ésta. Para ello, completa el formulario que se muestra en la figura 27:

The screenshot shows a web browser window at localhost:8001/subir_definicion/. The page has a header with two logos: 'INSTITUTO de la Sordera INDESOR' and 'ESCUELA Dr. Jorge Otte Gótzler INDESOR'. To the right of the logos are a 'MENÚ' button and a user profile for 'daniela' with 100 likes. The main content area is split into two sections: 'Subir definición' (orange background) and 'Completar desafío' (dark blue background). Under 'Subir definición', there is a video player showing a woman speaking. Under 'Completar desafío', there is a form with a 'Palabra:' field containing 'Fotografiar', a 'Definición:' text area, and a 'Subir Definición' button. Below the text area, there is a note: 'Se sugiere que la definición provenga del diccionario de la RAE.' The footer of the page shows '© Indesor 2015'.

FIGURA 27: FORMULARIO DE INGRESO DE UNA DEFINICIÓN AL DICCIONARIO MULTIMEDIA

La particularidad de este formulario es que debe cargar la lista de todas las palabras cuyo estado es “Incompleta”. Para ello, como se puede observar en el siguiente pseudocódigo, la función de la aplicación web debe hacer una primera llamada a la API con el método GET. Luego, se hace una segunda llamada con el método POST para subir la definición a la base de datos.

```
def upload_definition(request):
    words = requests.get('.../api/v1.0/words/?status=IN')
    list_of_words = words.json()['data']

    word_name = request.POST['word']
    word_definition = request.POST['definition']
    params = {
        'word_name': word_name,
        'word_definition': word_definition}

    r = requests.post('.../api/v1.0/word/definition_post/', data=params)

    if r.json()['status'] == 'ok':
        result = "The word definition has been uploaded."

    return render_to_response(...)
```

La API por su parte, busca la palabra en cuestión, le añade su definición y cambia su estado a “No validada”, tal como se puede apreciar en el siguiente extracto de código:

```
def upload_definition(request):
    response = {}
    word_name = request.POST['word_name']
    word_definition = request.POST['word_definition']

    word = Word.objects.get(name = word_name)
    word.definition = word_definition
    word.status = 'NV'
    word.save()

    response['status'] = 'ok'
    return
    HttpResponse(json.dumps(response), content_type='application/json')
```

Una vez que la palabra tiene un estado “No validada”, son los usuarios sordos y oyentes los encargados de validar colectivamente que las relaciones palabra-video y palabra-definición respectivamente sean correctas.

4.2.5. VALIDACIÓN DE PALABRAS: DESAFÍOS

El concepto de validación colectiva en este proyecto está basado en reCAPTCHA [9]. Este último es un sistema de reconocimiento de texto en imágenes que emplea la prueba desafío-respuesta para determinar cuándo un usuario es humano, mientras se mejora la digitalización de textos. En nuestro caso, se genera la validación de contenido al mismo tiempo que los usuarios ganan puntaje en la aplicación, sin estar a priori conscientes de este proceso.

La validación de palabras la hacen los usuarios sordos y oyentes mediante desafíos (figura 28). Un desafío consiste en realizar 2 preguntas sobre videos con señas a usuarios sordos o 2 preguntas sobre definiciones a usuarios oyentes.

En el caso de los usuarios sordos, la primera pregunta consiste en mostrar un video con una seña de una palabra con estado “Validada” (palabra conocida) y 4 opciones de palabras tal que una de ellas corresponde a la seña en cuestión. Dado que la palabra está validada dentro de la aplicación, esta pregunta se utiliza para decidir si el usuario respondió o no correctamente al desafío (pregunta de puntuación).

La segunda pregunta consiste en mostrar un video con una seña de una palabra con estado “No validada” (palabra desconocida) y 4 opciones de palabras, tal que una de ellas es la que el usuario que subió el video a la aplicación dijo que le correspondía. Esta pregunta es la que genera una validación, pues permite al sistema acumular las respuestas de los usuarios con el fin de validar colectivamente la relación palabra-video, dándole un voto positivo a esta relación si es que la respuesta corresponde a la información que fue ingresada a la aplicación, y un voto negativo en caso de lo contrario (pregunta de validación).



FIGURA 28: DESAFÍO PARA UN USUARIO SORDO

La figura 29 muestra el diagrama de un desafío de la aplicación. En este proceso cabe destacar que:

- El sistema sólo confía en usuarios que responden correctamente a la primera pregunta. Esto es, si un usuario respondió incorrectamente a la pregunta de puntuación, entonces la respuesta a la segunda pregunta no será considerada dentro de la votación.
- El sistema entrega las preguntas de puntuación y validación de manera aleatoria al usuario, de tal forma que éste no pueda saber de antemano qué pregunta le dará puntaje en la aplicación.

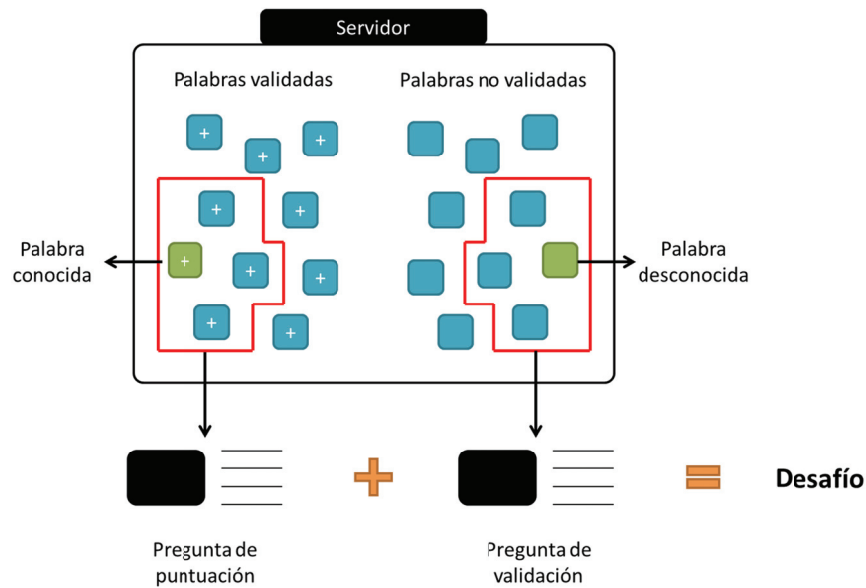


FIGURA 29: DIAGRAMA DE UN DESAFÍO DE LA APLICACIÓN

Para generar un desafío a un usuario sordo, la aplicación, a través de la siguiente función, hace una petición de tipo GET a la API:

```
def challenge_video(request):
    r = requests.get('.../api/v1.0/word/challenge_video/')
    challenge = r.json()
    return render_to_response(...)
```

La API entrega en formato JSON la palabra conocida junto a los nombres de otras 3 palabras validadas para las opciones de la pregunta de puntuación y la palabra desconocida junto a los nombres de otras 3 palabras no validadas para la pregunta de validación. A continuación se puede ver el pseudocódigo de la función que retorna un desafío al usuario:

```
def challenge_video(request):
    response = {}
    va_words = Word.objects.filter(status = 'VA').order_by('?')[:4]
```

```

nv_words = Word.objects.filter(status = 'NV').order_by('?')[:4]
#Palabra conocida
known_word = {}
options_known_word = []
for word in va_words:
    known_word ['id'] = word.pk
    known_word ['name'] = word.name
    known_word ['video'] = word.file
    options_known_word.append(word.name)
shuffle(options_known_word)
known_word['options'] = options_known_word

#Palabra desconocida
unknown_word = {}
options_unknown_word = []
for word in nv_words:
    unknown_word['id'] = word.pk
    unknown_word['name'] = word.name
    unknown_word['video'] = word.file
    options_unknown_word.append(word.name)
shuffle(options_unknown_word)
unknown_word['options'] = options_unknown_word

#armar respuesta
response['status'] = 'ok'
response['known_word'] = known_word
response['unknown_word'] = unknown_word

return
HttpResponse(json.dumps(response), content_type='application/json')

```

Para el caso de los usuarios oyentes, los desafíos siguen la misma lógica, pero en vez de mostrar videos se muestran definiciones, validándose entonces la relación palabra-definición.

4.2.6. ACEPTACIÓN/RECHAZO DE PALABRAS

Una vez que los usuarios sordos y oyentes han validado colectivamente tanto la definición como los videos de una palabra, son los usuarios expertos los encargados de revisar esta validación para aceptar o rechazar la palabra en el diccionario. Para ello, la aplicación web le pide a la API el listado de todas aquellas palabras en estado “No Validada” a través de una petición GET:

```

def words_to_validate(request):
    r = requests.get('.../api/v1.0/words/non_validated/')
    list_of_words = r.json()['data']
    return render_to_response(...)

```

La API retorna en la variable data los nombres de estas palabras, junto con la cantidad de votos positivos y negativos correspondientes, tal como se muestra a continuación:

```

def words_to_validate(request):
    response = {}
    words = Word.objects.filter(status = 'NV').order_by('name')
    words_array = []
    for i in words:
        word = {}
        word['name'] = i.name
        word['positive'] = i.positive
        word['negative'] = i.negative
        words_array.append(word)

    response['status'] = 'ok'
    response['data'] = words_array
    return
    HttpResponse(json.dumps(response), content_type='application/json')

```

La vista que muestra esta información al experto se puede observar en la siguiente figura:

Palabra	Votos positivos	Porcentaje votos positivos	Votos negativos	Porcentaje votos negativos
Decir	27	100,0%	0	0,0%
Empanada	25	100,0%	0	0,0%
Adornos	23	100,0%	0	0,0%
Ahora	19	95,0%	1	5,0%
Ayer	19	100,0%	0	0,0%
Futuro	19	95,0%	1	5,0%
Después	18	100,0%	0	0,0%
Bonita	17	100,0%	0	0,0%
Fotografiar	17	100,0%	0	0,0%
Agua	16	80,0%	4	20,0%

FIGURA 30: LISTADO DE PALABRAS EN PROCESO DE VALIDACIÓN

Como se puede observar, el usuario experto ve una lista con todas las palabras pendientes de pasar a estado “Validada”. En este caso, las palabras se muestran ordenadas en forma descendente de acuerdo a la cantidad de votos positivos que éstas tienen.

El objetivo de ordenarlas de esta manera es permitir al experto revisar sólo aquellas palabras que cumplan con cierto criterio de aceptación impuesto por él mismo, tal que aquellas palabras que no cumplan con ese criterio puedan seguir en estado “No validada” para que usuarios sordos y oyentes las validen. Por ejemplo, un experto podría querer revisar sólo aquellas palabras cuya cantidad de votos positivos sea mayor o igual a 18 y cuyo porcentaje de votos positivos sea 100%. Sin embargo, este usuario tiene la libertad de aceptar/rechazar palabras que no hayan sido validadas, esto es, que no tengan puntaje asociado.

Para aceptar o rechazar una palabra, el experto debe seleccionarla del listado, lo cual lo lleva a una nueva pantalla en la aplicación que le muestra toda su información (la palabra misma, su definición, configuración manual y video con la seña asociada).



FIGURA 31: PANTALLA DE REVISIÓN DE UNA PALABRA

En esta ventana (figura 31), el experto tiene la posibilidad de editar parte de la información en caso de encontrar errores, pudiendo modificar la definición, configuración manual y/o eliminar aquel video que no haya sido validado siempre y cuando éste no sea el primero que le fue agregado. Esto porque el usuario experto no tiene permitido agregar videos a la aplicación, luego si elimina el primer y único video asociado, entonces se pierde la relación palabra-seña. Por lo tanto, si una palabra contiene sólo una seña y ésta no le corresponde, entonces el experto debe rechazar la palabra en la aplicación.

Desde el momento en que el experto considera que la palabra y toda su información es correcta, entonces la acepta en el diccionario, haciéndola visible para todos los usuarios.

El siguiente código de la aplicación web es el que realiza la acción en cuestión, haciendo una llamada a la API con los datos del usuario y con la opción escogida (aceptar o rechazar), dejando a la API la tarea de procesar esta respuesta.

```
def validate_word(request):
    word = request.GET['palabra']
    answer = request.GET['respuesta']

    params = {
        'word': word,
        'answer': answer,
        'username' : request.session["username"],
        'password' : request.session["password"]}

    r = requests.post('.../api/v1.0/word/validate_word/', data=params)
    return HttpResponseRedirect(...)
```

Si la respuesta obtenida por la API es “Aceptar”, entonces la palabra cambia su estado a “Validada” y si es “Rechazar”, entonces elimina la palabra de la base de datos, tal como se puede observar en la siguiente función:

```
def validate_word(request):
    response = {}
    user = authenticate(username = request.POST['username'],
                        password = request.POST['password'])

    word = Word.objects.get(name = request.POST['word'])
    answer = request.POST['answer']

    if answer == 'Accept':
        word.status = 'VA'
        word.save()
    else:
        word.delete()

    response['status'] = 'ok'
    HttpResponseRedirect(json.dumps(response), content_type='application/json')
```

Tras este proceso de revisión por parte del experto, se concluye el flujo de una palabra en la aplicación.

4.3. DISEÑO DE ACTIVIDADES ADICIONALES

A continuación se procede a explicar el diseño de nuevas actividades a ser incorporadas en la aplicación a partir del diseño actual de la API desarrollada en este proyecto.

4.3.1. DISEÑO DE MATERIAL ADICIONAL

Actualmente, la aplicación por definición contempla sólo la inclusión de palabras en el diccionario aun cuando este modelo es fácilmente extensible a la incorporación de frases y diálogos. Estos últimos, permiten reforzar la lectura y escritura en usuarios sordos, así como la traducción entre ambos idiomas para los usuarios oyentes.

Para ello, se contempló en el diseño de la API el poder agregar funcionalidades para la creación de una sección de material adicional dentro de la aplicación web, la cual permite la incorporación de estos elementos (tanto en Español escrito como en Lengua de Señas) sólo por parte de los usuarios expertos, tal que no exista validación de este contenido por parte del resto de los usuarios. Esto, dada la necesidad de mantener un mayor control sobre el nuevo contenido en la plataforma y con el fin de poder organizar el material de acuerdo a niveles de dificultad, para que un usuario pueda ir desbloqueando niveles a medida que avanza en el aprendizaje de los niveles anteriores.

4.3.2. NUEVA MODALIDAD PARA LOS DESAFÍOS

En lo que a la validación de contenido se refiere, hasta ahora ésta se ha planteado a través de desafíos basados en simples preguntas de alternativas. Sin embargo, la información entregada a la aplicación por la API (1 video o definición más 4 palabras) puede ser traducida en actividades lúdicas de mayor atractivo para los usuarios, especialmente para los niños del INDESOR.

A continuación, las figuras 32 y 33 muestran 2 ejemplos de juegos que se pueden crear siguiendo el modelo de desafío antes expuesto:

1. **Tiro al blanco:** consiste en disparar un dardo que contiene un video con una seña o una definición al tablero que contiene la palabra correspondiente.



FIGURA 32: JUEGO TIRO AL BLANCO PARA LA VALIDACIÓN DE LA RELACIÓN SEÑA-PALABRA

2. **Caza tesoros:** consiste en seleccionar el cofre de la palabra correspondiente a la definición o seña que se muestra.

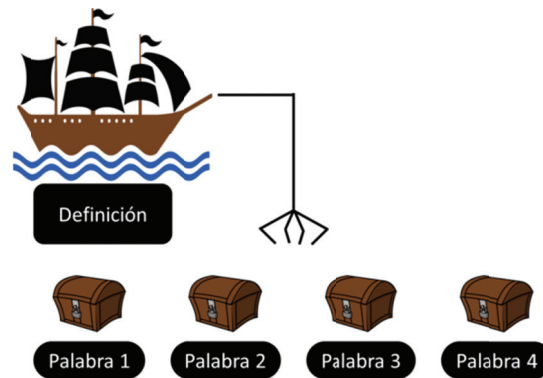


FIGURA 33: JUEGO CAZA TESOROS PARA LA VALIDACIÓN DE LA RELACIÓN DEFINICIÓN-PALABRA

Adicionalmente, la API provee otro mecanismo de validación que puede ser traducido en otro tipo de actividades. Este mecanismo consiste en entregar a la aplicación web la información de 4 palabras en vez de una, tal que ésta pueda generar juegos basados en el concepto de *flashcards*, en los cuales el usuario debe emparejar las señas o definiciones con las palabras respectivas. Lo anterior se ilustra en la siguiente figura:



FIGURA 34: JUEGO DE FLASHCARDS PARA LA VALIDACIÓN DE LA RELACIÓN SEÑA-PALABRA

Estos juegos son sólo demostrativos de lo que se puede hacer para validar contenido de manera lúdica, teniendo siempre presente que cada uno de ellos debe tener 2 etapas o preguntas, tal que de manera aleatoria, en una de ellas se deben dar al usuario palabras previamente validadas (pregunta de puntuación) y en la otra palabras no validadas (pregunta de validación).

5. PRUEBAS DE USABILIDAD

Las pruebas de usabilidad corresponden a una serie de técnicas aplicadas para evaluar la calidad del funcionamiento de una aplicación, en términos de que tan bien cumple con el propósito para la que fue creada. Esto tiene que ver principalmente con el diseño de las interfaces y las respuestas de la aplicación a las interacciones de los usuarios.

En este proyecto se aplicó el método de evaluación de usabilidad por observación en conjunto con el método *thinking aloud*, ambos descritos por J. Nielsen en su libro Usability Engineering [13]. En conjunto, estos métodos consisten en solicitar a un participante que por primera vez tiene acceso a la aplicación que realice, en lo posible sin ayuda, una serie de tareas preestablecidas, mientras va relatando en voz alta sus impresiones, pensamientos y opiniones sobre lo que está sucediendo al momento de realizar estas tareas. Por su parte, el evaluador anota en una pauta previamente elaborada los resultados del usuario (cumplimiento o fallo de la tarea realizada) junto con posibles observaciones que puedan surgir en el proceso referentes a cómo el usuario interactuó con la aplicación.

Para realizar las pruebas de usabilidad, la aplicación se instaló en un servidor de prueba, con un total de 50 palabras agregadas al diccionario. En adición, se entregaron videos y definiciones a los usuarios para las tareas correspondientes a la carga de material.

La muestra utilizada fue de 20 usuarios, de los cuales 5 eran niños sordos pertenecientes a octavo básico, 5 adultos oyentes apoderados de dichos estudiantes y 10 profesionales del INDESOR, 5 sordos y 5 oyentes. A estos usuarios se les hizo una charla introductoria a las características de la aplicación antes de comenzar con las pruebas de usabilidad.

A continuación se describen a grandes rasgos las tareas realizadas de acuerdo al perfil de cada participante:

1. Registro en la aplicación (todos).
2. Exploración del diccionario multimedia (todos).
3. Carga de videos (usuarios sordos).
4. Carga de definiciones (usuarios oyentes).
5. Realización de desafíos (usuarios sordos y oyentes).
6. Aceptación/rechazo de palabras en el diccionario (usuarios expertos).

Las pautas de observación utilizadas correspondientes a cada tipo de usuario se pueden ver en los Anexos 4,5 y 6 de este documento.

5.1. RESULTADOS OBTENIDOS

Los resultados más importantes obtenidos luego de realizar las pruebas de usabilidad son los siguientes:

1. Usuarios sordos:

- El 100% de los participantes revisó los videos explicativos la primera vez que se enfrentó a la aplicación, entendiendo claramente que la finalidad de éstos era apoyar o sustituir el proceso de lectura de los textos y formularios.
- El registro, inicio y cierre de sesión fue intuitivo para los usuarios, quienes comentaron que tales acciones eran muy similares a las que realizaban al revisar su correo electrónico o redes sociales.
- Dos de cinco niños mostraron dificultades para cambiar la modalidad del diccionario, preguntando qué debían hacer para ello.
- Ningún niño tuvo problemas para entender que en el menú de opciones se debe utilizar la flecha naranja para ir a una sección en la plataforma. Esto gracias a que atendieron al video que explicaba tal acción.
- Las tareas de subir videos a la aplicación, ya sea de palabras nuevas o existentes, fueron completadas exitosamente por el 100% de los usuarios.
- Concibiendo los desafíos como una actividad para ganar puntaje en la aplicación, los niños se mostraron entusiasmados por completar más de uno de éstos, entendiendo claramente, de acuerdo a los videos mostrados, cuándo respondían correctamente y cuándo fallaban en sus respuestas.

2. Usuarios oyentes:

- Uno de cinco adultos no logró por sí mismo cambiar la modalidad del diccionario.
- Dos de cinco adultos observaron que, al revisar palabras en el diccionario, no existe forma de saber por cuáles palabras se ha pagado y por cuáles no.
- Al contrario de los niños sordos, el 100% de los oyentes tuvo dificultad para ir de una sección a otra en la aplicación a través del menú de opciones. Esto, puesto que no lograron identificar con claridad que el texto es utilizado para mostrar videos a los niños sordos y la flecha naranja para cambiarse de página dentro del sitio.
- Dada la indicación de en lo posible subir definiciones provenientes del diccionario de la RAE a la aplicación, un usuario observó la posibilidad de incluir un link al diccionario online de esta institución.

3. Usuarios expertos:

- Sólo el 40% de los expertos logró entender por sí mismo que dentro del listado de palabras en proceso de validación, éstas pueden ser ordenadas o filtradas.
- El 80% de los expertos observó que, al momento de editar una palabra en el proceso de validación, hacía falta un botón para cancelar esta acción.
- A pesar de entender que sólo los usuarios sordos deben ser los encargados de subir videos a la aplicación, los usuarios expertos solicitaron poder subir palabras para la utilización de este material en sus clases.

5.2. AJUSTES A LA SOLUCIÓN

Los resultados obtenidos indican que en general los usuarios están conformes con el diseño y usabilidad de las interfaces. Los problemas detectados son en general de baja incidencia y se considera que se solucionan dando las indicaciones pertinentes al usuario.

Los problemas de mayor envergadura, tales como el agregar una indicación de pago por palabras en el diccionario (figura 35) y la falta del botón para cancelar la edición de una de éstas por parte de un experto, fueron solucionados.

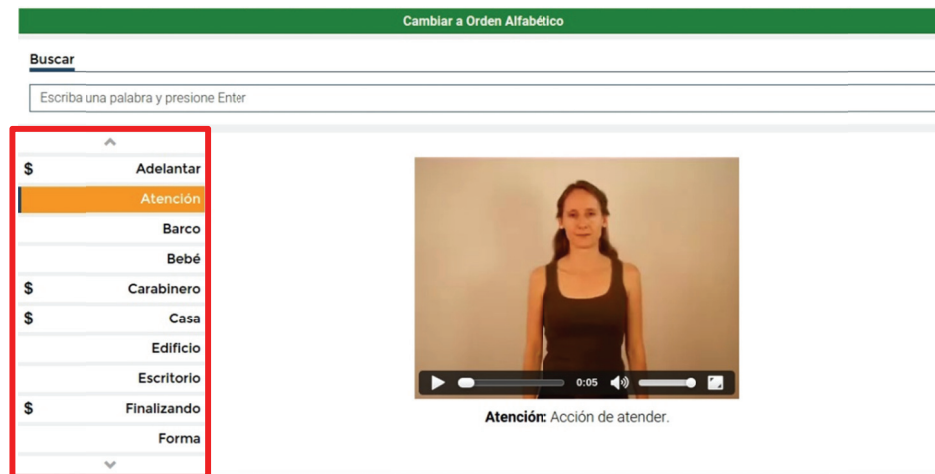


FIGURA 35: DETALLE DEL DICCIONARIO MULTIMEDIA CON INDICADOR DE PAGO POR PALABRAS

Finalmente, como respuesta a la solicitud que hicieron los usuarios expertos de poder subir material a la plataforma, se planteó la opción de crear un nuevo perfil de usuario que permita realizar esta acción, con el fin de no entorpecer el funcionamiento actual de la aplicación. Sin embargo, este cambio queda propuesto como trabajo futuro en la plataforma.

6. CONCLUSIONES Y TRABAJO FUTURO

La principal motivación que guio el desarrollo de este trabajo, fue la falta de herramientas computacionales orientadas a la comunidad de personas con déficit auditivo de nuestro país, que permitan no sólo el desarrollo de sus habilidades de comunicación, sino que también contribuyan a la integración de ésta en su entorno social. En este sentido, se puede argumentar que la aplicación web desarrollada, “Diccionario Multimedia INDESOR”, representa un avance en el ámbito formativo de sus usuarios, al potenciar los planes curriculares de apoyo existentes para el aprendizaje, tanto de la Lengua de Señas chilena como del Español escrito.

Los objetivos planteados para este proyecto llevaron a la realización de un diseño de interfaces acorde con la inclusión de sordos y oyentes dentro de la aplicación, cuya lógica de funcionamiento se basa en la participación activa de los usuarios dentro de la misma. Esto es, se creó un modelo de alimentación del diccionario que hace que los propios usuarios sean los encargados de subir material, cuya calidad es evaluada de manera colectiva a través de actividades, que a su vez, generan aprendizaje significativo.

El perfil colaborativo que tiene la aplicación, invita a los usuarios sordos y oyentes a ser parte de la evolución de la misma, haciéndolos responsables de la mejora continua de la calidad de la información dentro del sitio. De ahí la importancia que tuvo la creación de una comunidad virtual en torno al proyecto dentro del Instituto de la Sordera INDESOR, quienes participaron activamente en la definición de los aspectos pedagógicos de la solución propuesta, aportando en la validación de ésta en las distintas etapas del proyecto. Más aún, en la actualidad la aplicación cuenta con una base de material que incluye 500 videos en Lengua de Señas chilena desarrollados por los profesionales sordos de esta institución.

Pese a las ventajas que trajo la concepción del proyecto al interior de una comunidad educativa específica, se espera que la herramienta construida amplíe sus beneficios a otras comunidades del país. Sin embargo, para lograr lo anterior, es fundamental hacer cambios en la aplicación que permitan regular de manera más estricta la participación de los usuarios dentro de la misma. Así, el estándar de calidad del diccionario multimedia se mantendrá alto y confiable para sus usuarios. Por ejemplo, la carga de material a la plataforma se podría restringir a usuarios sordos específicos del INDESOR, de manera que usuarios externos a esta institución sólo puedan acceder al contenido del diccionario y completar desafíos.

En este mismo sentido, y dado el potencial de crecimiento que tiene esta herramienta, se vuelve necesario implementar mecanismos de compresión de videos, los cuales por tiempo no fueron considerados en esta etapa del proyecto. A este respecto, actualmente sólo existe una restricción sobre el tamaño de los archivos que pueden ser subidos a la plataforma.

Por su parte, la posibilidad de agregar material adicional a la aplicación, da pie para considerar a futuro la creación de actividades que enseñen las reglas y principios gramaticales de la Lengua de Señas chilena en contraste con aquellos que rigen la estructura de las oraciones en Español. No obstante, para lograr aprendizaje significativo, es imprescindible el trabajo de expertos, tanto en la estructuración de estas actividades según niveles de dificultad (debido a la complejidad que conlleva la enseñanza de la gramática de una lengua), como en la creación de

material de apoyo que explique las nociones básicas de las diferencias gramaticales de ambos idiomas. Esto genera una carga excesiva de trabajo para estos usuarios, considerando la cantidad limitada de profesionales en la materia.

El problema planteado se resuelve parcialmente involucrando a estudiantes sordos del segundo ciclo de enseñanza básica o de un nivel superior, quienes están capacitados para incorporar en la aplicación videos con frases en Lengua de Señas y su traducción al Español como material base de estas nuevas actividades, el que deberá seguir el mismo modelo colaborativo de validación planteado en este trabajo. Aun así, el diseño del contexto en el que este material pueda ser utilizado seguirá estando en manos de los expertos.

Los puntos mencionados anteriormente permiten concluir que las oportunidades de mejora del software construido en este trabajo de título son variadas. Pese a ello, el nivel de logro alcanzado fue satisfactorio para sus usuarios, quienes validaron en la práctica, con gran éxito, las funcionalidades principales creadas en esta versión de la aplicación.

7. BIBLIOGRAFÍA

1. Currículum Nacional, Ministerio de Educación, Gobierno de Chile, <http://www.curriculumnacional.cl/>. Fecha de última visita: 30/06/2015.
2. “La estructura sublexical de la Lengua de Señas chilena”, D. Adamo, I. Cabrera, P. Lattapiat y X. Acuña, 1988.
3. Diccionario Bilingüe Lengua de Señas Chilena-Español, Universidad Metropolitana de Ciencias de la Educación, edición 2009.
4. Instituto de la Sordera Indesor, <http://www.institutodelasordera.cl>. Fecha de última visita: 30/06/2015.
5. Diccionario Escolar Visual con Lengua de Señas, Instituto de la Sordera INDESOR en conjunto con el Ministerio de Educación, Gobierno de Chile, <http://www.institutodelasordera.cl/enlaces/web/web2/web>. Fecha de última visita: 30/06/2015.
6. “Proyecto Dicciseñas”, Centro de Desarrollo de Tecnologías de Inclusión CEDETI, Pontificia Universidad Católica de Chile, <http://diccisenas.cedeti.cl>. Fecha de última visita: 30/06/2015.
7. “Proyecto Sueñalettras”, Centro de Desarrollo de Tecnologías de Inclusión CEDETI, Pontificia Universidad Católica de Chile, <http://www.cedeti.cl/software-educativo/suenaletras>. Fecha de última visita: 30/06/2015.
8. “Duolingo”, <https://es.duolingo.com/>. Fecha de última visita: 30/06/2015.
9. “reCAPTCHA: Human-Based Characters Recognition via Web Security Measures”, L. von Ahn, B. Maurer, C. McMillen, D. Abraham y M. Blum, 2008.
10. “Motivations to Participate in Online Communities”, C. Lampe, R. Wash, A. Velazquez y E. Ozkaya, 2010.
11. “Toward an Integrated Crowdsourcing Definition”, E. Estellés-Arolas, F. González-Ladrón-de-Guevara, 2012.
12. “Moderated Online Communities and User-Generated Content”, J. Chen, H. Xu, A. Whinston, 2009.
13. Usability Engineering, J. Nielsen, edition 1993.

ANEXO 1: REQUISITOS DE USUARIO

En esta sección se presentan los requisitos de usuario asociados a la API y a la aplicación web desarrolladas.

RU001: Validación de contenido	
Descripción:	Se deben desarrollar desafíos para que los usuarios puedan validar el contenido de la aplicación.
Tipo:	Funcional

RU002: Subir videos	
Descripción:	Se debe permitir sólo a los usuarios sordos subir videos a la plataforma.
Tipo:	Funcional/Restricción

RU003: Subir texto	
Descripción:	Se debe permitir sólo a los usuarios oyentes subir la definición de una palabra a la plataforma.
Tipo:	Funcional/Restricción

RU004: Descargar material	
Descripción:	No está permitido que los usuarios descarguen a sus equipos el material disponible en la web. Esto sin embargo puede cambiar con el tiempo.
Tipo:	Funcional/Restricción

RU005: Perfiles de usuario	
Descripción:	Debe existir un perfil especial para los expertos (profesionales del INDESOR) en la aplicación dado que sólo éstos podrán administrar el contenido de ésta.
Tipo:	Funcional

RU006: Criterio-palabra	
Descripción:	Cada palabra representa un criterio en la API a ser preguntado en los desafíos. Esto es, cada palabra cumple sólo con 1 criterio por tanto un trío video-palabra-definición es positivo para la palabra en Español que representa y negativo para todas las otras palabras.
Tipo:	Funcional

RU007: Creación de desafíos	
Descripción:	Deben existir 2 tipos de desafíos: 1. Preguntar qué texto en Español de entre un listado de 4, representa la palabra que muestra un video de una seña. 2. Preguntar qué definición de entre un listado de 4, representa el texto en Español que se muestra.
Tipo:	Funcional

RU008: Moneda	
Descripción:	La aplicación debe contar con un tipo de moneda para las transacciones de subir, descargar/visualizar y validar palabras.
Tipo:	Funcional

RU009: Costo por subir material	
Descripción:	La aplicación debe contemplar un costo por subir material.
Tipo:	Funcional

RU010: Costo por descargar material	
Descripción:	La aplicación debe contemplar un costo por descargar/visualizar material.
Tipo:	Funcional

RU011: Premio por validación	
Descripción:	La aplicación debe contemplar un valor a dar a un usuario cuyo material subido haya sido validado.
Tipo:	Funcional

RU012: Premio por desafío	
Descripción:	La aplicación debe contemplar un valor a dar a un usuario por responder correctamente un desafío.
Tipo:	Funcional

RU013: Creación de un diccionario web multimedia	
Descripción:	Se debe crear una aplicación web que contenga un diccionario visual con palabras en Español y videos respectivos en Lengua de Señas chilena.
Tipo:	Funcional

RU014: Orden de las palabras según configuración manual o alfabéticamente	
Descripción:	Las palabras del diccionario deben poder mostrarse ordenadas según su configuración manual o alfabéticamente, siendo la configuración manual la opción por defecto. Esto para favorecer la búsqueda tanto para sordos como para oyentes.
Tipo:	Funcional

RU015: Interfaz amigable	
Descripción:	La interfaz de la aplicación web debe ser amigable para su uso por personas sordas. En particular, se deben incluir videos explicativos en todas las secciones del sitio.
Tipo:	Funcional

ANEXO 2: REQUISITOS DE SOFTWARE

Los requisitos de software asociados a los requisitos de usuario descritos anteriormente se presentan a continuación:

RS001: Validación de contenido	
Descripción:	La API debe tener mecanismos para que los distintos usuarios validen el contenido de la plataforma final. Estos mecanismos se traducen en implementar funcionalidades que retornen una lista de videos y/o textos para ser preguntados en los desafíos, según corresponda a cada usuario.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS002: Carga y descarga de material	
Descripción:	La API debe implementar funcionalidades para subir y bajar el material del servidor. Éste último debe almacenar texto y videos.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS003: Perfiles de usuario	
Descripción:	La API debe considerar distintos perfiles de usuario, distinguiendo entre expertos (profesionales del INDESOR, encargados de administrar el contenido), sordos y oyentes.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS004: Almacenamiento de criterios de evaluación	
Descripción:	La API debe crear y almacenar un registro de los criterios que se van creando a medida que se suben palabras al diccionario. Cada palabra representa un criterio a ser preguntado en un desafío.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS005: Desafíos	
Descripción:	La API debe tener mecanismos para mostrar desafíos a los usuarios según su perfil. Para esto, deben existir funcionalidades tales que: <ol style="list-style-type: none"> 1. Para sordos retorne un video y una lista de 4 palabras/criterios de entre los cuales sólo 1 representa el video que se muestra. 2. Para oyentes retorne una palabra y una lista de 4 definiciones de entre las cuales sólo 1 representa la palabra que se muestra.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS006: Implementación de puntajes	
Descripción:	La API debe implementar, almacenar y mostrar en la interfaz web el valor que tiene un usuario disponible de acuerdo a la moneda de cambio.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS007: Administración de palabras en la aplicación web	
Descripción:	Debe existir una sección en la aplicación web que permita a los expertos administrar las palabras que se ingresan al diccionario multimedia. Esto es, cuando una palabra cumple con cierto porcentaje de validación, es el experto el encargado de revisar que la validación hecha por parte de los usuarios esté correcta. De ser así, añade la palabra al diccionario.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS008: Aplicación web Diccionario Multimedia INDESOR	
Descripción:	Debe crearse una aplicación web que contenga el diccionario multimedia Lengua de Señas-Español escrito.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Funcional

RS009: Botón para cambiar el orden de las palabras	
Descripción:	En la vista del diccionario multimedia debe haber un botón que permita cambiar el orden de las palabras entre orden por configuración manual y orden alfabético.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Interfaz/Funcional

RS010: Interfaz de la aplicación web	
Descripción:	La interfaz de la aplicación web debe ser sencilla y contener videos explicativos por cada sección que contenga.
Prioridad:	Crítica
Estado:	Cumple
Tipo:	Interfaz

ANEXO 3: MATRIZ DE TRAZADO - REQUISITOS DE USUARIO VS REQUISITOS DE SOFTWARE

	RS001	RS002	RS003	RS004	RS005	RS006	RS007	RS008	RS009	RS010
RU001	X									
RU002		X								
RU003		X								
RU004		X								
RU005			X				X			
RU006				X						
RU007					X					
RU008						X				
RU009						X				
RU010						X				
RU011						X				
RU012						X				
RU013								X		
RU014									X	
RU015										X

ANEXO 4: PAUTA PRUEBAS DE USABILIDAD - USUARIOS SORDOS

INSTRUCCIONES

- Solicitar al participante que en lo posible intente decir lo que piensa mientras usa la aplicación.
- Anotar en cada ítem si logro realizar la acción por sí mismo. En caso de requerir ayuda por parte del observador, dejar constancia de ello en los comentarios.

PARÁMETRO A OBSERVAR	OBJETIVO LOGRADO (SÍ/NO)	COMENTARIOS
REGISTRO E INICIO DE SESIÓN		
Desde la página inicial, llegar al formulario para registrarse como usuario sordo en la aplicación.		
Crearse una cuenta como usuario sordo en la aplicación.		
Cerrar sesión en la aplicación.		
Iniciar sesión en la aplicación.		
EXPLORACIÓN DEL DICCIONARIO MULTIMEDIA		
Ir al diccionario multimedia organizado alfabéticamente y volver luego al diccionario organizado según configuración manual.		
Utilizar el buscador para encontrar una palabra en el diccionario multimedia organizado según configuración manual.		

SUBIR ARCHIVOS A LA APLICACIÓN

Desde el diccionario multimedia, llegar al formulario para ingresar un video con una seña a la aplicación.

Subir en la aplicación un video de una palabra nueva.

Subir en la aplicación un video de una palabra existente.

DESAFÍOS

Desde el diccionario multimedia, llegar a la sección que permite completar un desafío.

Completar un desafío.

ANEXO 5: PAUTA PRUEBAS DE USABILIDAD - USUARIOS OYENTES

INSTRUCCIONES

- Solicitar al participante que en lo posible intente decir lo que piensa mientras usa la aplicación.
- Anotar en cada ítem si logro realizar la acción por sí mismo. En caso de requerir ayuda por parte del observador, dejar constancia de ello en los comentarios.

PARÁMETRO A OBSERVAR	OBJETIVO LOGRADO (SÍ/NO)	COMENTARIOS
REGISTRO E INICIO DE SESIÓN		
Desde la página inicial, llegar al formulario para registrarse como usuario oyente en la aplicación.		
Crearse una cuenta como usuario oyente en la aplicación.		
Cerrar sesión en la aplicación.		
Iniciar sesión en la aplicación.		
EXPLORACIÓN DEL DICCIONARIO MULTIMEDIA		
Ir al diccionario multimedia organizado según configuración manual y volver luego al diccionario organizado alfabéticamente.		
Utilizar el buscador para encontrar una palabra en el diccionario multimedia organizado alfabéticamente.		

SUBIR ARCHIVOS A LA APLICACIÓN

Desde el diccionario multimedia, llegar al formulario para ingresar una definición a la aplicación.

Subir en la aplicación una definición de una palabra.

DESAFÍOS

Desde el diccionario multimedia, llegar a la sección que permite completar un desafío.

Completar un desafío.

ANEXO 6: PAUTA PRUEBAS DE USABILIDAD – USUARIOS EXPERTOS

INSTRUCCIONES

- Solicitar al participante que en lo posible intente decir lo que piensa mientras usa la aplicación.
- Anotar en cada ítem si logro realizar la acción por sí mismo. En caso de requerir ayuda por parte del observador, dejar constancia de ello en los comentarios.

PARÁMETRO A OBSERVAR	OBJETIVO LOGRADO (SÍ/NO)	COMENTARIOS
REGISTRO E INICIO DE SESIÓN		
Iniciar sesión en la aplicación.		
Cerrar sesión en la aplicación.		
EXPLORACIÓN DEL DICCIONARIO MULTIMEDIA		
Ir al diccionario multimedia organizado según configuración manual y volver luego al diccionario organizado alfabéticamente.		
Utilizar el buscador para encontrar una palabra en el diccionario multimedia organizado alfabéticamente.		
ACEPTAR ARCHIVOS PREVIAMENTE VALIDADOS		
Desde el menú, llegar a la sección que permite ver las palabras en proceso de validación.		
Aceptar una palabra.		