



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODE.JS

Fernando Antonio Hernández Gramajo
Asesorado por el Ing. Carlos Gustavo Alonzo

Guatemala, febrero de 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODE.JS

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

FERNANDO ANTONIO HERNÁNDEZ GRAMAJO
ASESORADO POR EL ING. CARLOS GUSTAVO ALONZO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. Luis Fernando Espino Barrios
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODE.JS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha noviembre 2020.

Fernando Antonio Hernández Gramajo

Guatemala, 04 de Noviembre de 2021

Ingeniero

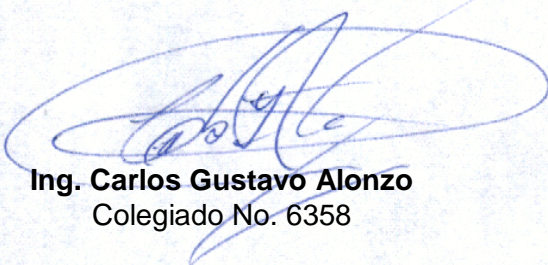
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **Fernando Antonio Hernández Gramajo** con carné **201403624** y CUI **2966 25671 0101** titulado "**Diccionario web de lenguaje de señas desarrollado en NodeJS**", lo he revisado y luego de corroborar que el mismo se encuentra concluido y que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Gustavo Alonzo
Colegiado No. 6358



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 12 de noviembre de 2021

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **FERNANDO ANTONIO HERNÁNDEZ GRAMAJO** con carné **201403624** y CUI **2966 25671 0101** titulado **“DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODEJS”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

LNG.DIRECTOR.011.EICCSS.2022

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODE.JS**, presentado por: **Fernando Antonio Hernández Gramajo**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”

The image shows a handwritten signature in blue ink over an official circular stamp. The stamp contains the text 'UNIVERSIDAD DE SAN CARLOS DE GUATEMALA' at the top, 'DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS' in the center, and a small emblem at the bottom. The signature is written over the stamp and extends upwards.

Ing. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, febrero de 2022

Facultad de Ingeniería

Decanato
24189101-
24189102
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.044.2022

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DICCIONARIO WEB DE LENGUAJE DE SEÑAS DESARROLLADO EN NODE.JS**, presentado por: **Fernando Antonio Hernández Gramajo**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Aurelia Anabela Cordova Estrada

Decana

Guatemala, febrero de 2022

AACE/gaoc

ACTO QUE DEDICO A:

- Dios** Por ser una importante influencia en mi carrera, entre otras cosas.
- Mis padres** Héctor Hernández García y María Teresa Gramajo Gómez. Por estar ahí siempre y ser ese apoyo cuando yo más lo necesite.
- Mis hermanos** Wagner Alexander y Hector Armando Hernández Gramajo, por todos los buenos momentos que hemos pasado juntos y el apoyo que me han dado, este logro es de ustedes también.
- Mi novia** Joselyne Pahola Ponsoy Santos, por estar siempre apoyándome en toda decisión que he tomado desde que empezamos la carrera.
- Mi abuela** Sofía Catalina Gómez Tepe, que en paz descanse, siempre te recordaré y llevaré conmigo todo lo que me enseñaste.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser la casa de estudios donde pude hacer realidad este logro.
Facultad de Ingeniería	Por lo importante que fue en mi formación como profesional.
Mis amigos de la Facultad	Joel Daniel Pérez Camey, Brandon Bredly Álvarez López, Yoselin Annelice Lemus López, Nery Gonzalo Gálvez Gómez y Rubén Emilio Osorio Sotorro por acompañarme a lo largo de estos años de carrera en la formación como profesionales.
Carlos Gustavo Alonzo	Por su ayuda desinteresada en todo momento y por ser un excelente catedrático.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN.....	XVII
OBJETIVOS.....	XIX
INTRODUCCIÓN	XXI
1. EL LENGUAJE DE SEÑAS	1
1.1. ¿Qué es el lenguaje de señas?	1
1.1.1. Definición	1
1.1.2. Origen.....	2
1.1.3. Tipos de lenguaje de señas	3
1.1.4. Sordera	5
1.1.5. Causas de la pérdida de audición y la sordera ...	6
1.1.5.1. Causas congénitas	6
1.1.5.2. Causas adquiridas	7
1.2. ¿Por qué aprender lenguaje de señas?	8
1.2.1. Dificultades en la formación académica	9
1.2.2. Dificultades en la vida diaria	10
1.2.3. Beneficios de aprender lenguaje de señas.....	10
1.2.4. Avances del lenguaje de señas en Guatemala.	11
1.2.5. Apoyo de asociaciones	12
1.2.5.1. Asorgua	13
1.2.5.1.1. Misión	13
1.2.5.1.2. Visión.....	13

1.2.5.1.3.	Reseña histórica	14
------------	-----------------------	----

2.	ALTERNATIVAS EN LA ACTUALIDAD PARA APRENDER LENGUAJE DE SEÑAS	15
2.1.	Tecnologías para el aprendizaje de personas sordas	15
2.1.1.	Aporte de empresas	17
2.1.2.	Realidad aumentada	17
2.1.3.	Aportes nacionales	19
2.2.	Mercado objetivo de aplicaciones de lenguaje de señas ..	19
2.2.1.	Personas sordas	20
2.2.2.	Familiares de personas sordas	20
2.2.3.	Docentes	21
2.2.4.	Interesados en aprender el lenguaje de señas.	23
2.2.5.	Instituciones nacionales	24
3.	PROPUESTA DE APLICACIÓN DE DICCIONARIO WEB	27
3.1.	Arquitectura propuesta	27
3.1.1.	Descripción del problema	27
3.1.2.	Descripción de la solución	28
3.1.3.	Tecnologías seleccionadas para desarrollo	29
3.1.3.1.	Lenguaje de programación	29
3.1.3.2.	<i>Framework</i>	30
3.1.3.3.	Motor de plantillas	31
3.1.3.4.	<i>Docker</i>	31
3.1.3.5.	<i>Docker-compose</i>	32
3.2.	Descripción de módulos en la aplicación	32
3.2.1.	Módulo de inicio	33
3.2.2.	Módulo de registro	34
3.2.3.	Módulo de diccionario	35

3.2.4.	Módulo ingresar video	36
3.2.5.	Módulo administrador	37
3.2.6.	Módulo ver video	39
3.2.7.	Panel de usuario	40
4.	IMPLEMENTACIÓN DE PROPUESTA DE APLICACIÓN.....	43
4.1.	Descripción de la implementación	43
4.1.1.	Estructura del proyecto.....	43
4.1.2.	Manejo de versiones de código	46
4.1.2.1.	Configuración de repositorio en línea.....	47
4.1.2.2.	Enlace de repositorio local.....	50
4.1.3.	Configuraciones del ambiente desarrollo de aplicación.....	51
4.1.3.1.	Descripción <i>dockerfile</i>	51
4.1.3.2.	Descripción <i>docker-compose.yml</i> ..	53
4.1.3.3.	Iniciar ambiente de desarrollo.....	56
4.1.4.	Base de datos.....	56
4.1.4.1.	Descripción de base de datos.....	57
4.1.4.2.	Descripción de <i>docker- compose.yml</i>	57
4.1.4.3.	Iniciar de base de datos de pruebas	59
4.1.4.4.	Modelo entidad relación.....	60
4.1.4.5.	<i>Script sql</i> con esquema de la base de datos.....	60
4.1.4.6.	Descripción de tablas	62
4.2.	Descripción de módulos principales en la aplicación.....	66
4.2.1.	Conexión con base de datos	66

4.2.2.	Manejo de las rutas de la aplicación	68
4.2.3.	Ingreso de nuevos usuarios	70
4.2.4.	Inicio de sesión.....	72
4.2.5.	Creación de palabras nuevas en el diccionario	74
4.2.6.	Ver palabras de usuario	75
4.2.7.	Ver palabras en revisión como administrador ..	77
4.2.8.	Ver una palabra en el diccionario	81
4.3.	Consideraciones de despliegue de aplicación	83
4.3.1.	Palabras iniciales para el diccionario	83
4.3.1.1.	Palabra número 1	84
4.3.1.2.	Palabra número 2.....	84
4.3.1.3.	Palabra número 3.....	85
4.3.1.4.	Palabra número 4.....	86
4.3.1.5.	Palabra número 5.....	86
4.3.1.6.	Palabra número 6.....	87
4.3.1.7.	Palabra número 7.....	88
4.3.1.8.	Palabra número 8.....	88
4.3.1.9.	Palabra número 9.....	89
4.3.1.10.	Palabra número 10.....	90
4.3.2.	Configuración para despliegue en producción .	90
4.3.2.1.	Instalación de <i>docker</i>	90
4.3.2.2.	Descarga de código en servidor....	91
4.3.2.3.	Iniciar de base de datos en producción.....	92
4.3.2.4.	Iniciar aplicación en producción	92
CONCLUSIONES.....		95
RECOMENDACIONES		97

BIBLIOGRAFÍA.....	99
APÉNDICES	105
ANEXOS.....	107

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Pantalla de ingreso a la aplicación.....	33
2.	Pantalla de registro en la aplicación.....	35
3.	Pantalla de índice de diccionario.....	36
4.	Formulario de ingreso de video nuevo	37
5.	Panel de administrador	38
6.	Visualización de video	40
7.	Videos subidos por el usuario	41
8.	Estructura general de proyecto	43
9.	Página de inicio <i>github</i>	48
10.	Creación de repositorio	48
11.	Comandos importantes para el enlace de repositorio	49
12.	Agregar repositorio remoto al proyecto	50
13.	Realizar cambios nuevos al repositorio en el servidor	50
14.	Contenido de <i>dockerfile</i> para la aplicación.....	52
15.	Contenido de <i>docker-compose.yml</i> para la aplicación	54
16.	Iniciar contenedores de <i>docker</i> para aplicación	56
17.	Contenido de <i>docker-compose.yml</i> para la base de datos.....	57
18.	Iniciar contenedor de <i>docker</i> de base de datos de pruebas.....	59
19.	Modelo entidad relación para la aplicación	60
20.	Descripción de esquema de base de datos	61
21.	Configuración para conexión con base de datos	67
22.	Archivo de rutas para el manejo de usuarios	68
23.	Archivo de rutas para el manejo de palabras	69

24.	Archivo de rutas para el manejo de comentarios	69
25.	Manejo de un nuevo usuario.....	70
26.	Manejo de nuevo usuario en bases de datos	70
27.	Uso de recurso para un nuevo usuario	71
28.	Vista del registro usuarios.....	71
29.	Manejo de inicio de sesión de un usuario	72
30.	Manejo de inicio de sesión en base de datos	72
31.	Uso de recurso para inicio de sesión	73
32.	Vista de inicio de sesión	73
33.	Manejo de una palabra nueva	74
34.	Manejo de una palabra nueva con base de datos	74
35.	Uso de recurso para creación de palabra nueva	75
36.	Vista de creación de palabra nueva.....	75
37.	Manejo de listar palabras subidas por usuario.....	76
38.	Manejo de listar palabras por usuario en base de datos.....	76
39.	Uso de recurso para listar palabras de usuario	77
40.	Vista de listado de palabras de un usuario	77
41.	Manejo de listar palabras a revisión.....	78
42.	Manejo de listar palabras a revisión en base de datos	78
43.	Uso de recurso para listar palabras a revisión.....	79
44.	Vista de listado de palabras a revisión.....	79
45.	Manejo de listado de diccionario.....	79
46.	Manejo de listado de diccionario desde la base de datos.....	80
47.	Uso de recurso para listar el diccionario	80
48.	Vista de diccionario.....	81
49.	Manejo de una palabra	81
50.	Manejo de una palabra en base de datos.....	82
51.	Uso de recurso para obtener una palabra	82
52.	Vista de una palabra del diccionario	83

53.	Video corto para buenos días	84
54.	Video corto para buenas noches.....	85
55.	Video corto para bien	85
56.	Video corto para cómo estas.....	86
57.	Video corto para computadora portátil	87
58.	Video corto para excelente.....	87
59.	Video corto para hola	88
60.	Video corto para mucho gusto	89
61.	Video corto para permiso	89
62.	Video corto para regular.....	90
63.	Instalación de <i>docker</i> y <i>docker-compose</i>	91
64.	Clonación de repositorio de aplicación.....	91
65.	Inicio de base de datos en ambiente de producción	92
66.	Inicio de aplicación en ambiente de producción.....	92
67.	Salida de resultado de iniciar aplicación	93

TABLAS

I.	Descripción de general de carpetas y archivos del proyecto	44
II.	Descripción de elementos sobre el control de versiones	46
III.	Descripción de comando en <i>dockerfile</i> para la aplicación.....	52
IV.	Descripción de comandos en <i>docker-compose.yml</i>	55
V.	Descripción de <i>docker-compose.yml</i> para base de datos.....	58
VI.	Descripción de entidades en modelo	62
VII.	Entidad usuario	63
VIII.	Entidad palabra.....	64
IX.	Entidad comentario.....	65

LISTA DE SÍMBOLOS

Símbolo	Significado
Js	Extensión de archivo JavaScript.
yml	Extensión de un archivo en formato YAML.
TCP	Protocolo de control de transmisión.
HTTP	Protocolo de transferencia de hipertexto.
DNS	Sistema de nombres de dominio.

GLOSARIO

Despliegue	Son todas las actividades que hacen que un sistema de software esté disponible para su uso.
<i>Docker</i>	Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.
<i>Docker-compose</i>	Es una herramienta para definir y ejecutar aplicaciones <i>Docker</i> de varios contenedores. Con <i>Compose</i> , usa un archivo YAML para configurar los servicios de su aplicación. Luego, con un solo comando, crea e inicia todos los servicios desde su configuración.
<i>Express</i>	Es un <i>framework web</i> transigente, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS.
<i>Framework</i>	Es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas.

Git	Es un software de control de versiones diseñado por <i>Linus Torvalds</i> , pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.
Github	Es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores.
JavaScript	Es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web.
Microsoft	Es una compañía multinacional, que diseña y comercializa programas informáticos y dispositivos electrónicos.
Node.js	Es un entorno de ejecución de JavaScript orientado a eventos asíncronos.
Npm	Es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.
Repositorio	Es un lugar en donde el código de una aplicación, de un programa cualquiera está almacenado y desde donde se puede distribuir.

Software

Sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

YAML

Es un formato para guardar objetos de datos con estructura de árbol. Sus siglas significan: *Ain't Markup Language* (YAML no es otro lenguaje de marcado).

RESUMEN

La lengua de señas en Guatemala es en parte practicada en su gran mayoría por personas sordas, lo que dificulta mucho la comunicación de este grupo en específico con personas no practicantes de este lenguaje o no perteneciente a la comunidad de lengua de señas. En los últimos años el impacto que tiene la lengua de señas en el país ha ido en aumento, como sociedad se tiene que aprender lo básico para que una persona se pueda comunicar con alguien perteneciente a la comunidad sorda, entonces uno de los primeros pasos aparte de conocer ciertas palabras iniciales en el lenguaje es practicar.

Para una práctica más amena ya sea solo con uno mismo o entre personas que practican este lenguaje, se propone una aplicación de lenguaje de señas el cual será un diccionario en donde las personas puedan subir videos de palabras o frases cortas referentes a este lenguaje.

Dicha aplicación es desarrollada en el lenguaje Node.js, como una plataforma web de apoyo en la cual se tendrá un diccionario formado por cada palabra o frase corta que los usuarios aporten al mismo, cada palabra del diccionario tendrá un video asociado y con la posibilidad de dejar comentarios con la meta de dar retroalimentación sobre la palabra la cual un usuario en específico quiera mejorar, logrando así que se practique y también se logre enriquecer el contenido del diccionario para futuras personas, que quieran tener un referente de cómo practicar la palabras de este lenguaje.

OBJETIVOS

General

Desarrollar una aplicación web tipo diccionario sobre lenguaje de señas.

Específicos

1. Proporcionar una galería de distintas palabras sobre lenguaje de señas.
2. Las personas interesadas en aprender lenguaje de señas puedan ver videos demostrativos de palabras pertenecientes.
3. Proporcionar a los usuarios un medio el cual retroalimentará a las personas sobre cómo se debe practicar el lenguaje de señas.

INTRODUCCIÓN

Dado a el interés de las personas en aprender o simplemente conocer más del lenguaje de señas en Guatemala, se necesita una forma de aprender este lenguaje de manera más práctica, a lo largo de los capítulos primero a conocer más a las personas sordas, así también por los avances que se tienen hoy en día en el ámbito de la tecnología y luego a conocer como estará construida la aplicación de diccionario desde su desarrollo hasta su despliegue y como finalidad se presenta una aplicación tipo diccionario el cual ayuda a las personas a conocer palabras de uso diario de lengua de señas, por medio de palabras que suban los mismos usuarios de manera en la que poco a poco se pueda ir enriqueciendo el contenido del diccionario y que más personas conozcan, practiquen y se les facilite el acceso a esta información.

1. EL LENGUAJE DE SEÑAS

1.1. ¿Qué es el lenguaje de señas?

Es la manera principal de comunicación entre las personas sordas o de una persona sorda hacia una persona oyente.

1.1.1. Definición

La lengua de señas es la lengua natural de las personas sordas y es como cualquier otra, posee y cumple todas las leyes lingüísticas y se aprende dentro de la comunidad de usuarios a quienes facilita resolver todas las necesidades comunicativas y no comunicativas propias del ser humano, social y cultural.

La lengua de señas es una lengua que también está al alcance de las personas oyentes siempre que se sumerjan en el mundo del sordo. (Sordo-a, con Mayúscula, es una convención utilizada para designar a las personas que se identifican con la lengua, costumbres, valores, tradiciones, entre otros de la comunidad sorda. Escrito en minúscula está referido a la perspectiva médica-clínica). En consecuencia, los oyentes que cumplan estas condiciones también pueden pertenecer a ese grupo lingüístico-cultural.

La lengua de señas o lengua de signos, como ya se ha dicho anteriormente es un lenguaje natural de expresión y configuración gesto-espacial y percepción visual, gracias a la cual las personas sordas pueden establecer un canal de comunicación con su entorno social, ya sea conformado por otros individuos sordos o por cualquier persona que conozca la lengua de señas

empleada. Mientras que con el lenguaje oral la comunicación se establece en un canal vocal-auditivo, el lenguaje de señas lo hace por un canal gesto-viso-espacial.

1.1.2. Origen

Hoy en día las lenguas de señas se utilizan casi exclusivamente entre personas sordas, el uso de las señas como sistema de comunicación es tan antiguo en la historia de la humanidad como el de las lenguas orales, o incluso más, y también han sido y continúan siendo empleadas por comunidades de oyentes.

De hecho, se cree que algunas tribus indígenas de Norteamérica utilizaban las señas como método de comunicación entre distintas etnias que no compartían la misma lengua.

En el siglo XVI, Jerónimo Cardano (médico de Padua, en la Italia norteña), proclamó que las personas sordas podrían hacerse entender por medio de combinaciones escritas de símbolos, asociándolos con las cosas a que ellos se referían.

En el año 1620, Juan de Pablo Bonet, realiza una publicación considerada como el primer tratado moderno de fonética y logopedia, en el que se proponía un sistema de enseñanza oral para las personas sordas, mediante el uso de señas alfabéticas.

Así se divulga en Europa, y posteriormente en el mundo, el llamado alfabeto manual, destinado a mejorar la comunicación de personas mudas, sordas y sordomudas. En dicha obra aparece un abecedario ilustrado mediante

grabados calcográficos de los signos de las manos, que representan las letras del alfabeto.

La edición fue traducida a distintos idiomas, y Charles Michel de l'Épée, publica en el siglo XVIII su alfabeto, sobre la base del divulgado por Bonet. Este nuevo alfabeto es básicamente el que ha llegado hasta la actualidad, siendo conocido internacionalmente como alfabeto manual español.

En 1817 Gallaudet fundó la primera escuela norteamericana para sordos, en Hartford, Connecticut, y Clerc se convirtió en el primer maestro sordo de lengua de señas de los Estados Unidos.

Pronto las escuelas para personas sordas comenzaron a aparecer en varios estados y un total de veintidós escuelas se habían establecido a lo largo de la nación en el año 1865.

1.1.3. Tipos de lenguaje de señas

Por lo general, el lenguaje de señas que las familias desarrollan con sus niños sordos es bastante incompleto. La comunicación les es aún bastante difícil. Pero hay gente que se ha unido para crear lenguajes de señas mucho más completos. Hay cientos de diferentes lenguajes de señas, pero hay 3 tipos principales:

- Lenguajes de señas nacionales y regionales. En casi todos los países, la gente sorda ha desarrollado su propia lengua de señas, con la que puede aprender a comunicarse tan bien y casi tan rápidamente como la gente que oye. Diferentes señas de las manos representan diferentes objetos, acciones e ideas. La estructura (gramática) de estas lenguas es diferente

a la del lenguaje hablado y por lo tanto es difícil que la aprenda la gente que oye. Estas lenguas las prefieren las personas que nacen sordas. El lenguaje de señas americano, usado en los Estados Unidos (E.U.) y Canadá, y el lenguaje de señas mexicano son algunos ejemplos.

- Lenguajes de señas basados en lenguajes hablados. Estos lenguajes tienen la misma organización y gramática que la lengua que se habla en el área. Son más fáciles de aprender para las personas que oyen y para las que se volvieron sordas cuando ya habían aprendido a hablar. A veces usan la primera letra (deletreo con los dedos) de una palabra como parte de la seña. Son más difíciles de aprender para los niños que no saben leer, pero, por otro lado, pueden hacer que sea más fácil y divertido aprender a leer. Algunos ejemplos son el lenguaje de señas en inglés y el lenguaje de señas en español.
- Deletreo con los dedos. Cada palabra se deletrea con las señas que representan las letras del alfabeto. Este método de escritura en el aire es lento pero preciso. Es más fácil que lo aprendan las personas que ya saben leer y escribir. En inglés, los británicos usan un sistema de dos manos y los americanos un sistema de una mano. Trate de aprender el sistema que se use más en su país.

Muchas personas sordas combinan estos 3 sistemas. Para comunicarse con otras personas sordas usan sobre todo el primer método, con las personas que oyen o con un intérprete usan el segundo método y deletrean con los dedos las palabras difíciles.

La discapacidad se resume en un gran número de diferentes limitaciones funcionales que se registran en las poblaciones de todos los países del mundo.

La discapacidad puede ser una deficiencia física, intelectual o sensorial, una dolencia que requiera atención médica o una enfermedad mental.

1.1.4. Sordera

La sordera o hipoacusia es toda reducción de las capacidades auditivas, es la privatización completa o parcial del sentido del oído; esta puede ser debida alteraciones del aparato de transmisión de los sonidos (oído externo, oído medio) o alteraciones de las vías nerviosas.

El oído puede percibir desde sonidos, apenas audibles, hasta sonidos muy fuertes, así como diferenciar el volumen y la distancia e identificar la dirección de una fuente sonora con mucha exactitud.

La audición es un proceso en el que las ondas sonoras se convierten en señales eléctricas, que luego el nervio auditivo envía del oído al cerebro. La capacidad de oír depende del correcto funcionamiento de la estructura del oído, del nervio auditivo y del área del cerebro encargada de recibir e interpretar los sonidos.

El concepto común y corriente de sordera es el de un trastorno auditivo total o grave. Al trastorno auditivo en el que únicamente se requiere que otra persona hable un poco más fuerte, se llama pérdida auditiva.

Sordera y pérdida auditiva implican una pérdida en la sensibilidad del oído, probablemente en el sistema auditivo periférico. Ambos caen en la misma dimensión, y surge el problema de considerar en dónde debe trazarse la línea entre débil auditivo y sordo.

Se estima que 360 millones de personas en el mundo viven con un grado de hipoacusia que les genera algún tipo de discapacidad (hipoacusia en rango moderado), siendo el 91 % de estos casos en adultos y 56 % en hombres. Esto representa el 5,3 % de la población mundial. Sin embargo, se estima que hasta el 15 % de la población adulta tiene algún grado de pérdida auditiva. Este porcentaje es mucho mayor si se valoran las personas mayores de 65 años, llegando hasta un tercio de esta población. Este dato cobra mayor interés ya que se estima que el número de personas de 65 o más años crecerá entre un 18 % - 50 % en el período 2010-2020 en todas las regiones del mundo.

De acuerdo con el grado de agudeza auditiva la sordera puede ser:

- Falta de audición (sordera total o pura).
- Deficiencia auditiva (oye parcialmente con dificultades en el habla).
- De transmisión o conductiva (Problemas físicos en el oído externo o el oído medio impidiendo el paso del sonido).
- De percepción (lesiones en el oído interno).
- Mixta (lesiones tanto de transmisión como de percepción).

1.1.5. Causas de la pérdida de audición y la sordera

Las causas de pérdida de audición y sordera pueden ser congénitas o adquiridas.

1.1.5.1. Causas congénitas

Las causas congénitas pueden determinar la pérdida de audición en el momento del nacimiento o poco después. La pérdida de audición puede

obedecer a factores hereditarios y no hereditarios, o a complicaciones durante el embarazo y el parto, entre ellas:

- Rubéola materna, sífilis u otras infecciones durante el embarazo.
- Bajo peso al nacer.
- Asfixia del parto (falta de oxígeno en el momento del parto).
- Uso inadecuado de ciertos medicamentos.
- Ictericia grave durante el período neonatal, que puede lesionar el nervio auditivo del recién nacido.

1.1.5.2. Causas adquiridas

Las causas adquiridas pueden provocar la pérdida de audición a cualquier edad.

- Enfermedades infecciosas, como la meningitis, el sarampión y la parotiditis.
- La infección crónica del oído.
- La presencia de líquido en el oído (otitis media).
- El uso de algunos medicamentos, como los empleados para tratar las infecciones neonatales, el paludismo, la tuberculosis y distintos tipos de cáncer.
- Los traumatismos craneoencefálicos o de los oídos.
- La exposición al ruido excesivo, por ejemplo, en entornos laborales en los que se trabaja con maquinaria ruidosa o se producen explosiones.
- La exposición a sonidos muy elevados durante actividades recreativas, como el uso de aparatos de audio personales a un volumen elevado durante períodos prolongados de tiempo, o en bares, discotecas, conciertos y acontecimientos deportivos.

- El envejecimiento, en concreto la degeneración de las células sensoriales.
- La obstrucción del conducto auditivo producida por cerumen o cuerpos extraños.

En los niños, la otitis media crónica es una causa común de pérdida de audición.

1.2. ¿Por qué aprender lenguaje de señas?

Los niños comienzan comunicarse pronto. O al menos, claro, lo intentan. Su comprensión del lenguaje y sus habilidades motoras se desarrollan más rápido que el habla. El aprendizaje de los más pequeños empieza por la imitación, de ahí que aprender lengua de señas sea más natural. Muchos padres podrían pensar que este tipo de comunicación interfiere con el desarrollo natural del lenguaje hablado, pero tan solo lo complementa, a fin de que el proceso comunicativo sea más claro.

Enseñar, conocer y entenderse con soltura en lengua de señas es fundamental para la inclusión, pues sin ella, las personas con discapacidad auditiva pierden el acceso a la información y a la interacción diaria con amigos o seres queridos. Pero el conocimiento de esta es limitado, y las barreras, muy altas para quienes tan solo ir al médico suponen una carrera de obstáculos. Por suerte, existen muchas iniciativas para garantizar la inclusión: programas de formación para funcionarios y docentes, cursos gratuitos a jóvenes, incluso aplicaciones móviles para practicar de forma individual.

1.2.1. Dificultades en la formación académica

Los niños con discapacidad auditiva tienen problemas a la hora de asistir a escuelas convencionales a no ser que se les proporcione la ayuda y el apoyo adecuados. Pueden llegar a tener problemas de integración y tienden a sentarse al fondo de la clase para pasar desapercibidos.

Muchos niños no participan en clase porque no puede oír lo que el profesor o los demás alumnos dicen. No siempre piden a los compañeros que repitan lo que han dicho y algunos incluso le dicen al profesor que no necesitan micrófonos especiales a pesar de que la amplificación de sonido les facilitaría la escucha.

Los niños discapacitados auditivos quieren ser como sus compañeros con audición normal y se sienten incómodos si se llama la atención sobre su problema auditivo. Todo esto repercute en la vida social del niño. Muchos se apartan y prefieren no participar en las actividades escolares. No atender los problemas de estos niños suele manifestarse en cansancio y dolores de cabeza cuando llegan a casa de la escuela.

Incluso una pérdida de audición leve no diagnosticada en niños en edad escolar puede afectar negativamente en el proceso de aprendizaje y en consecuencia planteará algún tipo de dificultad de aprendizaje en la escuela. Repercutirá en las tareas escolares si el niño con pérdida de audición emplea energía adicional para intentar oír al profesor, tomar apuntes y procesar todo lo que oye al mismo tiempo. La pérdida de audición en niños suele pasar desapercibida porque se tiende a pensar que los problemas escolares se deben a falta de concentración o atención.

1.2.2. Dificultades en la vida diaria

La pérdida de audición ocasiona problemas de comunicación que pueden tener efectos importantes en la vida cotidiana y generar sensación de soledad, aislamiento y frustración, sobre todo, en las personas mayores.

Las complicaciones psicológicas más habituales son: vergüenza, culpabilidad e ira; pena; problemas de concentración; preocupación y frustración; ansiedad y desconfianza; inseguridad y autocrítica y baja autoestima/confianza en sí mismo.

En cuanto a las complicaciones sociales las más frecuentes son aislamiento y retraimiento; pérdida de atención; distracción/falta de concentración; problemas en el trabajo; problemas para participar en la vida social; problemas de comunicación; pérdida de intimidad o problemas sexuales.

Respecto a las consecuencias físicas de la sordera puede provocar cansancio; cefalea; vértigo; tensionar la musculatura; producir estrés; problemas de alimentación y/o sueño; problemas estomacales o aumentar la presión sanguínea.

1.2.3. Beneficios de aprender lenguaje de señas

Muchas personas, sea por algún defecto de nacimiento, un accidente o malformaciones congénitas, no pueden hacer uso de sus cinco sentidos, aquellos que nos permiten percibir el mundo de cierta manera. Específicamente a las personas que padecen hipoacusia (incapacidad parcial o total de poder percibir los sonidos, comúnmente llamada sordera), el lenguaje de señas les permite interactuar de manera eficaz con otros.

Aprender este lenguaje basado en el uso de manos y gestos para describir conceptos y elaborar oraciones ofrece una gran oportunidad para generar un mundo más incluyente y accesible y para crear puentes con más personas. Además, así como cada idioma tiene ciertos sonidos y maneras de pronunciar que lo hacen único y especial, la belleza de este lenguaje radica en que depende del sentido por medio del cual percibimos más del 90 % de lo que expresan los demás: la vista.

A través de la observación puede ayudarnos a conocer mejor a nuestro interlocutor, ya que los gestos nos permiten notar, por ejemplo, los estados de ánimo, la sinceridad de lo expresado e incluso los sentimientos que a veces ni siquiera necesitamos transmitir mediante palabras.

El lenguaje de señas puede ser una excelente herramienta para comunicarnos con los niños que todavía no aprenden a hablar. Una de las principales causas de frustración en los padres es no saber qué necesita o qué quiere su hijo cuando llora. Afortunadamente, a los pequeños les encanta imitar a los demás y a ello se suma que sus habilidades motoras se desarrollan más rápidamente que su capacidad de hablar, por lo que no les resulta nada difícil aprender el lenguaje de señas. De hecho, un especialista en desarrollo infantil descubrió que los bebés que tenían padres sordos se comunicaban con ellos a una edad mucho más temprana desde los 6 meses que los hijos de padres sin problemas de audición.

1.2.4. Avances del lenguaje de señas en Guatemala

El organismo legislativo aprobó el martes 28 de enero de 2020 el Decreto Legislativo No. 3-2020 que hace énfasis a la Ley que reconoce la Lengua de

Señas de Guatemala (Lensegua), la cual busca la inclusión de las personas con discapacidad visual y auditiva.

Este decreto permitirá que las personas sordas o sordociegas tengan derecho a acceder a la enseñanza de la Lengua de Señas como primera lengua.

La ley establece lo siguiente:

- Este idioma deberá implementarse en establecimientos públicos y privados, así como en la televisión nacional.
- El Consejo Nacional para la Atención de las Personas con Discapacidad (Conadi) asesorará al Ministerio de Educación para la aprobación de materiales que se utilizarán para los cursos de lengua de señas que serán impartidos en el país.
- Toda persona sorda tendrá derecho a acceder a la enseñanza de la lengua de señas sin ningún tipo de discriminación.
- Cada 23 de septiembre se conmemorará el Día de la Lengua de Señas de Guatemala¹.

La Ley de Lengua de Señas de Guatemala (Lensegua) entrará en vigencia seis meses después de su publicación en el Diario de Centroamérica.

1.2.5. Apoyo de asociaciones

Existen asociaciones que apoyan de manera educativa a personas que posean discapacidad o bien a población en general que necesiten o quieran conocer el lenguaje.

¹ La nueva Ley que reconoce la Lengua de Señas en Guatemala, enero 2020. <https://www.guatemala.com/noticias/sociedad/nueva-ley-que-reconoce-lengua-de-senas-guatemala-enero-2020.html>. Consulta: septiembre 2020.

1.2.5.1. Asorgua

La Asociación de Sordos de Guatemala (Asorgua) es una institución que apoya a las personas con discapacidad auditiva, en el área educativa, salud y laboral; tomando en cuenta sus necesidades individuales para adecuarle ayuda y así poder realizar acciones de reactivación social.

De acuerdo con datos de ASORGUA, alrededor de 240 mil personas viven con discapacidad auditiva en Guatemala. Entre las acciones importantes promovidas por el movimiento de personas sordas en coordinación con el CONADI, se encuentra aprobación de la ley de Lengua de Señas de Guatemala, normativa que promueve la enseñanza y aprendizaje de la Lengua de Señas.²

1.2.5.1.1. Misión

Somos una institución que impulsa acciones para las personas con discapacidad auditiva, con el propósito de trabajar en programas y proyectos en áreas de educación, salud, situación laboral y recreación, para el beneficio de este sector de la población, generando así su integración social.

1.2.5.1.2. Visión

Somos una institución representativa de personas con discapacidad auditiva a nivel nacional facilitadora de procesos para el apoyo mutuo y la solidaridad que permita el logro de una vida plena a las personas sordas.

² CONADI. *Promueven aprendizaje de Lengua de Señas.* <https://conadi.gob.gt/web/2021/01/12/promueven-aprendizaje-de-lengua-de-senas/>. Consultado: septiembre 2020.

1.2.5.1.3. Reseña histórica

A partir del año 1968, sus primeras reuniones las efectuaban en el Parque Central de la Ciudad de Guatemala. Cuatro años después en 1972, se inició el Club de sordos en la zona uno de la ciudad capital en la 18 Calle y 12 Avenida zona 1. Posteriormente en 1978, se formó una comisión dentro del Club, encargada de la elaboración de los estatutos. En 1990 se retomó el esfuerzo de la legalización; en 1992 se terminaron los Estatutos. Dos años después en 1994 se reconoció la personalidad jurídica como Asociación de Sordos de Guatemala. Actualmente se encuentra funcionando legalmente, con estructura física y administrativa en la 2a. Calle "A" 10-44 zona 10 de esta ciudad capital, instalaciones proporcionadas por el Comité Pro Ciegos y Sordos de Guatemala³.

³ SAMAYOA URIZAR, Rubí María. *Cultura y psicología de una comunidad sorda*. <http://www.repositorio.usac.edu.gt/6547/1/T%2013%20%282972%29.pdf>. Consultado: septiembre 2020.

2. ALTERNATIVAS EN LA ACTUALIDAD PARA APRENDER LENGUAJE DE SEÑAS

2.1. Tecnologías para el aprendizaje de personas sordas

El asunto de las tecnologías para las personas con discapacidad, es algo reciente, el cual ha tenido un avance significativo en relación con otros campos tecnológicos, es decir; si comparamos los años que han empleado otras áreas humanas de desarrollo tecnológico, las personas con discapacidad recientemente han tenido un mayor beneficio en el disfrute de tecnologías de apoyo para su vida independiente.

En lo que respecta a las diferentes discapacidades existentes, si encontramos diferencias, por ejemplo, las tecnologías de apoyo para personas con discapacidad visual han tenido un avance exponencial, en segundo lugar; las tecnologías de apoyo para personas con movilidad restringida o parcial, (parálisis cerebral y otras), en tercer lugar; las personas con discapacidad auditiva, y por último; las personas con discapacidad cognitiva.

Las personas con discapacidad auditiva, en general, mucha de la tecnología que se desarrolla para su vida independiente, es en su mayor parte experimental; por ejemplo, en Costa Rica y EEUU se han diseñado prototipos de un guante que funciona conectado a una computadora, el cual lee los movimientos del lenguaje de señas y los interpreta de forma escrita en el monitor de la computadora.

El guante representa un gran avance para las personas con discapacidad auditiva, ya que podrían utilizar la computadora y por medio de internet y otras aplicaciones, comunicarse de forma escrita con personas de todo el mundo, así también, los servicios públicos podrían emplear este tipo de tecnología para comprender y dar un servicio de mayor calidad y facilitar el uso de servicios.

Este guante, es todavía un prototipo, a pesar que ya tiene algún tiempo de su publicación, debido principalmente a los costos de este tipo de tecnología, lo cual es el problema que experimentan las personas con discapacidad y las instituciones relacionadas. Es una realidad que la tecnología para personas con discapacidad es costosa, lo cual hace que la misma no sea muy accesible económicamente hablando para las personas con discapacidad e instituciones.

Entre otras tecnologías de apoyo para personas con discapacidad auditivas están las que permiten la comunicación, como las tablas de comunicación; ya sea por pictogramas, grafemas o símbolos, audífonos e implantes, que permiten potenciar el nivel auditivo en personas con baja audición, cámaras o webcam, para comunicarse con otras personas con el lenguaje de señas, telefonía con capacidad de video llamada, para comunicarse con lenguaje de señas, o por medio de mensajes de texto, despertadores o alarmas, con adaptaciones que por medio de señales de luces o vibración, indican o llaman la atención a la persona para despertar o recordarle alguna tarea o asunto programado por la misma; entre otros tipos de herramientas tecnológicas adaptadas o diseñadas para una función o necesidad de la persona con discapacidad auditiva.

Las tecnologías de apoyo para las personas con discapacidad cada vez van avanzando más, así que, en un futuro no muy lejano, podremos ver y conocer

muchas más tecnologías que nos podrían beneficiar en múltiples tareas y actividades de una forma increíble.

2.1.1. Aporte de empresas

Microsoft se ha asociado con el Instituto Técnico Nacional para la Sordera del RIT, para probar el uso de la tecnología de habla y lenguaje impulsada por IA de Microsoft para apoyar a los estudiantes en el aula con sordera o dificultades para escuchar.

Jenny Lay-Flurrie, jefa de accesibilidad en Microsoft, quien es sorda, comentó que el proyecto piloto con el RIT muestra el potencial de IA para impulsar a la gente con discapacidades, en especial a aquellas con sordera. Los subtítulos brindados por Microsoft *Translator* brindan otra capa de comunicación que, junto con el lenguaje de señas, podría ayudar a la gente.

“El lenguaje es el motor de la evolución humana. Mejora la colaboración, la comunicación y el aprendizaje. Al tener los subtítulos en el aula del RIT, ayudamos a todos a aprender mejor y a comunicarse mejor”⁴, comentó Xuedong Huang, asociado técnico y jefe del grupo de habla y lenguaje para IA e Investigación en Microsoft.

2.1.2. Realidad aumentada

Un equipo de científicos del *Motion Light Lab* de la Universidad Gallaudet, de Washington, EE. UU., desarrolló una aplicación con cuentos para niños con

⁴ ROACH, John, News Center Corp. *Tecnología IA ayuda a estudiantes con sordera a aprender.* <https://news.microsoft.com/es-xl/features/tecnologia-ia-ayuda-a-estudiantes-con-sordera-a-aprender/>. Consultado: septiembre 2020.

sordera. La herramienta de aprendizaje bilingüe educa a los menores en el Lenguaje de señas americano (ASL, en inglés) y les ayuda a mejorar sus habilidades de lectura y escritura.

El libro infantil utiliza un personaje virtual tridimensional creado como un sistema de captura de movimiento. Este método logra que la figura tenga las expresiones y el ritmo natural del ASL que no se puede obtener con una animación a computadora.

La aplicación se basa en: mirar, leer y aprender. En el primero, pueden conocer la historia relatada por un cuentacuentos en lenguaje de señas. Luego, pueden avanzar al modo de lectura que tiene textos al final de cada página.

Si el menor tiene dificultad para comprender el contenido, puede activar la interpretación de señas. Mientras que el modo aprender, contiene un diccionario con vocabulario del cuento y su traducción al lenguaje de señas.

Melissa Malzkuhn, fundadora del proyecto, explicó por medio de una intérprete, que la cantidad de personas con sordera es un problema global por la falta de recursos. Nuestra meta es apoyar el desarrollo temprano porque la adquisición del lenguaje es vital entre los cero a cinco años, manifestó la profesional que lidera a un equipo de personas con sordera.⁵

Existen alrededor de 300 lenguajes de señas en el mundo. Debido a que el ASL solamente se practica en EE. UU. y algunas partes de Canadá, el ML2 trabaja con otros países para recrear la aplicación en otros idiomas.

⁵ PAREDES, Luisa. ELPERIÓDICO. *“App” para alfabetizar a niños con sordera.* <https://elperiodico.com.gt/nacionales/2019/04/23/app-para-alfabetizar-a-ninos-con-sordera/>. Consultado: setiembre 2020.

Actualmente han traducido seis libros a los lenguajes de señas usados en Rusia, Noruega, Alemania, Tailandia, entre otros. Además, han realizado animaciones para leyendas o cuentos populares para promover un sentido de identidad en los menores.

2.1.3. Aportes nacionales

Kitsord es una aplicación para aprender lenguaje de señas, cuya finalidad es romper la barrera que existe entre las personas que tienen deficiencias auditivas y quienes no las tienen.

- El aprendizaje se efectúa por lecciones empezando por el abecedario.
- Glosario integrado para buscar rápidamente alguna seña.
- Lleva el control de tus estadísticas por lección para identificar las palabras en las que te has equivocado y de esta manera practicarlas inmediatamente.
- Cuenta con el apoyo de Benemérito Comité Pro Ciegos y Sordos de Guatemala, por lo que es muy confiable.
- Es 100 % gratuito.

2.2. Mercado objetivo de aplicaciones de lenguaje de señas

Las diferentes razones por las que una persona puede llegar a interesarse en aprender lenguaje de señas son diversas, siempre a favor de enriquecer su conocimiento y ayudar a las personas con discapacidad se integren de mejor forma a la sociedad.

2.2.1. Personas sordas

Hacer realidad los derechos de las personas sordas comienza con asegurar que los niños sordos tengan acceso a la educación en lenguaje de señas. Los niños sordos tienen derecho a una educación de calidad, como el resto de niños, en un lenguaje y un entorno que maximice su potencial. En todo el mundo, a los niños y los jóvenes sordos a menudo se les niega la educación, incluso en lengua de señas. Existe una carencia de profesores con una buena formación en el lenguaje de señas y, en muchos casos, los padres no saben que sus hijos tienen derecho a ir a la escuela y que pueden aprender si reciben el apoyo adecuado.

2.2.2. Familiares de personas sordas

Debido a que los padres generalmente no han tenido contacto con otras personas sordas, desconocen la lengua de señas y la importancia que tiene en el desarrollo de su hijo. Estimulados frecuentemente por los médicos, tratan de comunicarse con los niños desde el español en su forma oral, descuidando, sin saberlo, un factor importante: el niño sordo no oye y, por lo tanto, la comprensión del mensaje sonoro no será posible.

En caso contrario cuando los padres sean sordos utilizan entre ellos y con su hijo sordo la lengua de señas, el niño desarrolla espontáneamente el lenguaje ya que estar en contacto con una lengua accesible le posibilita un proceso comunicativo natural.

La interacción social y lingüística de los padres sordos con sus hijos sordos es semejante y tan compleja como aquella de los padres e hijos oyentes.

Este proceso de interacción contribuye a que el niño no solo ponga en funcionamiento sus capacidades lingüísticas y comunicativas, sino que también vaya formando una identidad social propia: el niño sordo es capaz de asimilar la realidad que lo rodea, es decir, socializarse y conocer el mundo.

2.2.3. Docentes

La lengua de señas es una expresión natural basada a través de movimientos y expresiones del cuerpo que llevan a cabo las personas sordas para comunicar, aprender y expresar lo que sienten. Es de gran importancia que esta lengua sea implementada en la educación básica a través de estrategias educativas y que los docentes puedan poner en práctica los conocimientos básicos de la lengua de señas, con el fin de que los alumnos puedan desempeñar sus habilidades y conocimientos. En el trayecto de su vida escolar podrán ver un amplio panorama sobre las oportunidades que se presentarán al dominar esta lengua y se logre dar una interacción e incorporación con las personas a su alrededor y puedan ser beneficiados a través de sus conocimientos ya que ambos alumnos podrán dominar la lengua oral y la lengua de señas.

Es indispensable saber el uso correcto para que puedan tener un aprendizaje adecuado de dicha lengua, así como la lengua verbal posee reglas de la misma manera la lengua de señas. Sin embargo, en las escuelas de educación básica no se ha implementado el aprendizaje de comunicación de señas y verbal debido a la falta de servicios que sean de apoyo para dicha educación.

Así mismo, la lengua de señas hará posible que los alumnos de educación básica, generen conocimientos de aprendizaje como todo alumno sin distinción

alguna, contribuyendo al desarrollo de capacidades para la lectura, la escritura y las matemáticas, que facilitarían la comunicación en edades tempranas.

La práctica de este lenguaje, cada día más recurrente, favorece la inclusión escolar, social y laboral de dicha población.

Es considerada una forma de comunicación visual y gestual a la que también se le conoce como Lensegua o LSG. Utiliza gestos del rostro, movimientos corporales y de ambas manos para la composición y expresión de ideas y conceptos.

Aunque es ampliamente conocido alrededor del mundo, no es un lenguaje universal, ya que varía en cada país. Aparte de la lengua de señas, las personas con problemas auditivos deben tener acceso a un lenguaje integrado (oral y global), lo que significa que también aprenden a hablar y a leer los labios. Se dice que este lenguaje es más antiguo que las lenguas habladas, ya que antes que se formalizaran y desarrollaran los dialectos, los antiguos seres humanos transmitían ideas a través de señas y expresiones.

En la página de la ONU sobre la Convención Universal de los Derechos de las Personas con Discapacidad, se cita a la lengua de señas y se menciona que posee todas las propiedades y complejidades propias de cualquier lengua natural oral.

El martes 28 de enero de 2020, el Congreso de la República de Guatemala aprobó un decreto que reconoce la Lensegua como una lengua oficial en nuestro país. Esto permite que las personas con deficiencia auditiva tengan derecho a aprender el lenguaje de señas como primera lengua y que este se implemente en establecimientos públicos, privados y televisión.

Es importante que el docente conozca esta lengua para incluir a los niños sordos que posiblemente tenga en el aula,

Para aprender el lenguaje de señas hay varias opciones: cursos impartidos por instituciones y universidades, videos y ejemplos en el portal del Mineduc, en la página de temas de Educación especial de la Digeesp (Dirección de Educación Especial) y con el manual de uso público de Asorgua (Asociación de Sordos de Guatemala).

El trabajo de la inclusión comienza con los maestros: al acoplarnos a las necesidades se les da una oportunidad que no tenían en el pasado. No olvides conmemorar el Día de la lengua de señas de Guatemala cada 23 de septiembre y ayudar a crear un aula y una sociedad más inclusiva⁶.

2.2.4. Interesados en aprender el lenguaje de señas

La lengua de señas no tiene un costo, más aún si se sabe que la mejor forma de aprenderla, no es asistiendo a cursos donde la imparten, más si se sabe que la forma más pura es conviviendo con sordos, con la comunidad que la transmite tal cual, de la forma más limpia y correcta que hay, sin deformaciones ni agregados que luego los oyentes hacen. La lengua de señas no requiere de horas de aprendizaje, no requiere inscripciones y mensualidades costosas, ni riesgo de hacer inversiones, para que años después se diga que los resultados fueron menores a los esperados.

Cuando las familias pasan su vida buscando alternativas que modifique la condición de su hijo, pero en ello se les va la estabilidad económica, se sufre.

⁶ Redacción Educativa. *Inclusión escolar: Lenguaje de señas en Guatemala*. <https://www.hagomitarea.com/recursos/docentes/inclusion-escolar-lenguaje-de-senas-en-guatemala/>. Consultado: septiembre 2020.

Millones de centros terapéuticos hacen negocios con las ilusiones de padres de familia, promesas van y vienen, la realidad es que solo los que tienen un poder adquisitivo cómodo, los que tienen el tiempo suficiente para el extenuante seguimiento terapéutico y los que tienen acceso a información actualizada son los que logran de mejor forma el desarrollo de la lengua oral.

La lengua de señas tiene la característica de ser noble, de ser accesible, de no distinguir clases sociales ni estratos económicos ni académicos. La lengua de señas se adquiere de forma natural en cualquier chico sordo, sin importar si puede contar con una terapia oral (que también cabe aclarar no en todos lados es eficiente porque no en todos lados verdaderos profesionales la dan).

2.2.5. Instituciones nacionales

El Benemérito Comité Pro Ciegos y Sordos de Guatemala, es una Institución privada no lucrativa, social y educativa, fundada el 3 de diciembre de 1945.

El Congreso de la República de Guatemala le otorgó el título y dignidad de Benemérita Institución el 27 de octubre de 1998. Por su encomiable labor que ha beneficiado ampliamente, sin discriminación alguna a la población guatemalteca.

El Benemérito Comité está regido por la Asamblea General y dirigida por una Junta Directiva, la cual está integrada por once profesionales guatemaltecos, quienes prestan sus servicios ad-honorem. Actualmente la Presidencia está a cargo de la licenciada María de los Ángeles de Rueda.

Está conformado por 36 programas y servicios, los cuales se hacen posibles por el trabajo de 642 colaboradores y gracias a los fondos que genera la venta de billetes de Lotería Santa Lucía. El Benemérito Comité es una Institución que ofrece servicios

médicos, educativos y de rehabilitación los cuales son el producto de una planificación integral⁷.

⁷ Benemérito Comité Pro-ciegos y sordos de Guatemala. *¿Quiénes Somos?*
<https://prociegossordos.org.gt/QuienesSomos.php>. Consultado: septiembre 2020.

3. PROPUESTA DE APLICACIÓN DE DICCIONARIO WEB

La manera más eficiente de apoyar a las personas interesadas en aprender lenguaje de señas desde el ámbito tecnológico es con una solución de *software*.

3.1. Arquitectura propuesta

Desde el momento que se tiene la idea de hacer un software capaz de proponer una solución óptima al problema que se plantea más adelante, se consideró utilizar las tecnologías más adecuadas para el desarrollo de la aplicación.

3.1.1. Descripción del problema

Las personas interesadas en aprender lenguaje de señas muchas veces no practican lo necesario siquiera para decir una frase corta, por lo cual, nunca se terminan de iniciar en dicho lenguaje más allá de una palabra o un saludo, los materiales de estudio se pueden encontrar en internet, pero aún, así lo único que hace falta a la persona interesada es practicar, de esta manera el interesado se ira metiendo cada vez más a aprender no solo letras, sino, palabras del lenguaje para ir reforzando.

Con esta idea en mente, seria de mucho beneficio para las personas interesadas tener una plataforma tipo comunidad en donde se posteen videos cortos de personas practicantes con mucho tiempo o no del lenguaje las cuales aparte de subir videos puedan retroalimentar a otras que se inician en el lenguaje,

todo esto con el objetivo de dejar un historial de videos tipo diccionario para que otras personas que visiten el mismo puedan saber palabras de uso diario en lenguaje de señas.

3.1.2. Descripción de la solución

La aplicación estará enfocada en ser un diccionario de videos, los cuales contendrán palabras de uso diario y si se llega a crecer el mismo también frases u oraciones cortas, en donde los usuarios puedan postear sus propios videos de palabras, con el beneficio de recibir retroalimentación de otras personas, usuarios del diccionario, acerca de los mismos, logrando así que los videos que permanezcan en la página sean de calidad para futuras visitas y un apoyo a que las personas practiquen de una mejor manera los movimientos las manos y/o el cuerpo en algunas palabras para lograr que su lenguaje de señas sea entendible y puedan así iniciar la práctica al lenguaje de señas de una manera mucho más natural.

Por supuesto que los videos que se suban a la aplicación tienen que pasar por un filtro en el cual se revisará que el contenido del mismo sea exclusivo del lenguaje de señas y no de otra índole o videos que puedan dañar el contenido del diccionario, como palabras o frases ofensivas, por lo que se tendrá un panel de administrador encargado de aceptar o rechazar, si se diera el caso, los videos en cuestión, y así estos sean incluidos en el diccionario.

Los videos en su mayoría serán entre 5 y 15 segundos para que no se pierda el enfoque de una palabra o una frase muy corta y facilitar así el análisis del mismo y de la practica el en usuario quien es el que sube el video, así también los comentarios de retroalimentación se plantea un mínimo de entre 15 y 50

palabras para que la crítica u opinión sea objetiva con el afán de transmitir verdadera retroalimentación a la persona que subió el video.

Todos los videos que pasen por el filtro y se publiquen como parte del diccionario, estarán listados formando parte del mismo y se pondrán encontrar buscando por la letra inicial de la palabra a la cual representa el video en el listado.

El diccionario para que no empiece sin contenido se tiene contemplado que en un inicio cuente con 10 a 15 palabras las cuales serán a elección propia, para que en el lanzamiento de la aplicación sean una base para visualizar, en un inicio de cómo deberían ser los próximos videos subidos por usuarios o que en el mejor de los casos sirvan de guía.

En la plataforma se podrá reproducir el video en su totalidad, solo con entrar en el mismo, sin restricciones ya que uno de los propósitos del diccionario es llegar a la mayoría de personas.

3.1.3. Tecnologías seleccionadas para desarrollo

Toda herramienta seleccionada ha sido probada con anterioridad asegurando así que la misma no vaya presentar complicaciones en el lapso de desarrollo y en posteriores publicaciones en la nube.

3.1.3.1. Lenguaje de programación

Se decidió usar Node.js, el cual es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Puede considerarse como una máquina virtual rápida y de gran calidad, además muchas

de las capacidades de Node.js para peticiones y respuestas (entrada y salida de datos) son realmente ligeras y potentes, por consecuencia pudiéndole sacar el máximo para las aplicaciones basadas en altas tasas de transacciones.

Este lenguaje soporta varios protocolos los cuales se listan los siguientes: HTTP, DNS y TCP, todos a la vez, además uno de los beneficios de Node.js, la capacidad de atender a clientes con sus peticiones el cual se puede ir modificando según la aplicación vaya creciendo.

Gran facilidad para crear aplicaciones de lado del servidor con muchas bibliotecas y/o *frameworks* de terceros para instalar al proyecto el cual puede facilitar aún más el desarrollo del mismo, logrando escribir código más sencillo y fácil de entender dado que se apega a ciertos estándares de codificación impuestos para el uso de los mismos.

3.1.3.2. Framework

El *framework* seleccionado para trabajar la aplicación es *Express*, este es el más conocido y por ende el más usado para crear aplicaciones que necesiten desarrollar aplicaciones con el formato de peticiones en formato HTTP.

Proporciona las siguientes facilidades para desarrollo en las que se pueden encontrar:

- Manejo de todos los verbos HTTP existentes con diferentes rutas definidas por el programador.
- Tiene varias opciones para generar vistas para el usuario final.
- Capacidad de agregar procedimientos intermediarios para verificación o un mejor manejo de las peticiones que llegan hacia el servidor.

- Manejo de diferentes tipos de respuesta dependiendo el tipo de las peticiones hechas hacia el servidor.

Express como tal es bastante sencillo de implementar en cualquier proyecto, sumado a todo lo que otros programadores han hecho por mejorarlo, hay muchos otros complementos para manejo de bases de datos, sesiones y así como para las peticiones con cabeceras las cuales pueden indicar el tipo de contenido en la petición, hasta parámetros de autenticación como los *tokens* o *cookies* de sesión o en el caso de las respuestas identificar el tipo de las mismas y hasta su contenido personalizado.

3.1.3.3. Motor de plantillas

El motor de plantillas seleccionado dentro de todo lo que ofrece *Express* es EJS el cual posee una sintaxis muy sencilla para el maquetado HTML, se configura muy rápido y es eficiente a la hora de crear varias vistas que necesiten de renderizar ciertos elementos traídos desde una petición del servidor.

Las vistas se apoyan por medio de la definición de rutas de Node.js para funcionar de tal manera que se puede navegar por las diferentes vistas por medio de su ruta asignada desde el servidor, todo esto de manera muy sencilla y fácil de entender para un programador.

3.1.3.4. Docker

Para la parte del despliegue de la aplicación no importando su sistema operativo se usará *docker* para empaquetar la aplicación en un entorno en el cual no nos interesa la naturaleza del sistema operativo, de tal manera que todo lo que la aplicación necesite descargar para su funcionamiento no interfiera con

otros ambientes en el sistema operativo y que se haga de forma automática para la generación de una imagen de *docker* la cual posteriormente podremos iniciar con algunos parámetros más desde el servidor creando así su contenedor.

Se usarán *dockerfiles* para desarrollar este apartado los cuales son archivos destinados a contener información relevante de la aplicación como en que lenguaje está escrita, que se necesita instalar para funcionar, los puertos los cuales necesita para comunicarse desde donde este, los comandos para una configuración inicial y el arranque de la aplicación.

3.1.3.5. *Docker-compose*

Se necesitará algo más que solo el archivo de *dockerfile* para que todo quede completamente configurado y sea mucho más fácil el despliegue de la aplicación, para eso existe *docker-compose* que se encarga de interpretar un archivo el cual contiene todos los parámetros utilizados para crear la imagen de *docker* de tal manera que con solo un comando se pueda iniciar la aplicación desde el servidor, creando así no solo la imagen de *docker* sino también su contenedor.

3.2. Descripción de módulos en la aplicación

Explicación y bosquejo por cada uno de los módulos importantes en la aplicación, detallando las posibles acciones que se podrán realizar dentro de la misma.

3.2.1. Módulo de inicio

Módulo el cual contará con una caja de texto para introducir el correo con el cual creó su cuenta y una caja de texto para la contraseña de dicha cuenta en la aplicación, cuando todos estos valores sean válidos se puede proceder a presionar el botón en la parte inferior al formulario.

Dicho módulo solo será visible por el usuario si este no ha iniciado sesión con anterioridad en la aplicación desde su navegador en caso contrario se le enviará al módulo de diccionario, para evitar inconsistencias en las credenciales guardadas como parte de su sesión.

Para cerrar la sesión del usuario en curso basta con presionar el botón ubicado en la parte superior derecha de la pantalla el cual enviará al usuario al módulo de diccionario, pero esta vez sin posibilidad de subir videos o comentar los mismos.

Figura 1. **Pantalla de ingreso a la aplicación**

El diagrama muestra un formulario de inicio de sesión con el título "Login". Hay dos campos de texto, cada uno etiquetado como "Text Field". El primer campo contiene el texto "correo" y el segundo campo contiene el texto "contraseña". Debajo de los campos hay un botón rectangular azul con el texto "Ingresar" en blanco.

Fuente: elaboración propia, empleando figma.com.

3.2.2. Módulo de registro

Módulo el cual se encargará de crear una cuenta de usuario a cualquier persona que quiera formar parte de los usuarios los cuales pueden contribuir al diccionario subiendo sus propios videos y comentando para la retroalimentación de otros, así mismo el módulo cuenta con un formulario que la persona debe llenar con los campos de la caja de texto de correo, una caja de texto para la contraseña y una caja de texto para su confirmación de contraseña y verificar que sean las mismas, al momento que todos estos campos estén llenos se podrá crear la cuenta presionando el botón ubicado en la parte posterior del formulario.

Cuando se presione dicho botón, pueden pasar 2 escenarios los cuales se detallan a continuación:

- Cuando el usuario crea una cuenta con una dirección de correo ya existente en la aplicación, en el cual se le notificará al usuario que ya existe un usuario usando dicha cuenta de correo, por lo que se le pedirá que intente crear una cuenta con una dirección de correo diferente.
- Cuando al usuario se le permitió crear su cuenta, por lo que será enviado a la página de módulo de diccionario.

Se debe tomar en cuenta que cuando la persona llena el formulario para crear una cuenta de usuario y este se le permite crear la misma, pasará a estar con sesión iniciada dentro de la aplicación por lo que si lo que se necesita ingresar con otro usuario o simplemente cerrar sesión, de la misma forma que pasa con el ingreso a la aplicación (Módulo inicio) deberá presionar el botón ubicado en la parte superior derecha para cerrar la sesión dentro del navegador que se esté usando, de tal manera que se enviará al usuario al módulo de diccionario pero esta vez sin acceso a subir videos o comentar los mismos.

Figura 2. **Pantalla de registro en la aplicación**

The image shows a registration form titled "Registro". It consists of three text input fields stacked vertically. The first field is labeled "correo" and contains the text "correo". The second field is labeled "contraseña" and contains the text "contraseña". The third field is labeled "confirmar contraseña" and contains the text "confirmar contraseña". Below the input fields is a blue button with the text "Crear Cuenta".

Fuente: elaboración propia, empleando figma.com.

3.2.3. **Módulo de diccionario**

Pantalla con la funcionalidad principal de la aplicación la cual es listar todas las palabras del diccionario en forma amigable con un índice con cada letra del abecedario, de tal forma que se vea ordenado por secciones y además que sea en orden alfabético descendente.

El diccionario tendrá una barra para descender en el diccionario, pudieron regresar al principio en cualquier momento, facilitando así su búsqueda por alguna palabra que se esté buscando, pero no se esté seguro como se escriba.

Cada una de las palabras en el diccionario tendrá un botón para ver dicho video el cual enviará al usuario al módulo de ver video, no importando si la persona que visita la aplicación tiene una sesión activa o solo quiere entrar a ver el video de dicha palabra en cuestión, sin crear cuenta.

Se podrá regresar a esta pantalla desde cualquier parte de la aplicación presionando el botón de diccionario en la barra superior de la aplicación.

Figura 3. **Pantalla de índice de diccionario**



Fuente: elaboración propia, empleando figma.com.

3.2.4. Módulo ingresar video

Módulo el cual se encargará de la carga de videos nuevos hacia el diccionario, que pasarán por el proceso de aceptación del administrador, este módulo contará con un formulario para la carga de una nueva palabra, los campos que se necesitan son por consiguiente una caja de texto para el nombre de la palabra nueva y un botón que abrirá una ventana de búsqueda de archivos de la computadora para indicarle cuál es el video a cargar en el diccionario, luego se mostrará el nombre del archivo cargado a la par del botón de su búsqueda y por último, presionar el botón en la parte inferior del formulario para crear la nueva

palabra y esta sea enviada a revisión, para posteriormente ser mostrada como una palabra más en el diccionario.

Solo se podrán cargar videos cortos a la aplicación, esto ligado al propósito del diccionario que es mostrar videos demostrativos de una palabra solamente, por lo que no deberían durar más de unos pocos segundos.

Figura 4. **Formulario de ingreso de video nuevo**

Formulario de ingreso de palabra nueva

Text Field

Text Field

Fuente: elaboración propia, empleando figma.com.

3.2.5. **Módulo administrador**

Módulo el cual se mostrará cuando se ingresa a la aplicación con una cuenta de usuario administrador, la siguiente desplegará únicamente un listado de todas la palabras dispuestas a revisión que han subido todos los usuarios para que se tomen en cuenta en formar parte del diccionario, cada una de las propuestas de nuevas palabras contará con un nombre de la palabra y una serie

de botones los cuales son para visualizar el video que se ha subido, aceptar la nueva palabra o rechazar la nueva palabra.

Cuando un video se acepta o rechaza este pasa a eliminarse de esta lista de propuestas y luego se continuará con notificar al usuario que su video si califica y pasará a formar parte del diccionario si y solo si se aceptó o a notificar al usuario que su video no califica para el diccionario en dado caso se rechace.

La opción de ver video enviará a una nueva pantalla del módulo con el único propósito de visualizar el video de manera que se pueda tomar una decisión, con respecto al estado de la palabra nueva, luego de ver el video, en la parte inferior del mismo se proporcionará un botón que al presionarlo enviará al panel administrador, para tomar la decisión de aceptarlo o rechazarlo.

Para salir del panel de administrador basta únicamente con presionar el botón de cerrar sesión ubicado en la parte superior derecha del módulo y por consiguiente enviará a la página de diccionario de la aplicación.

Figura 5. **Panel de administrador**



Fuente: elaboración propia, empleando figma.com.

3.2.6. Módulo ver video

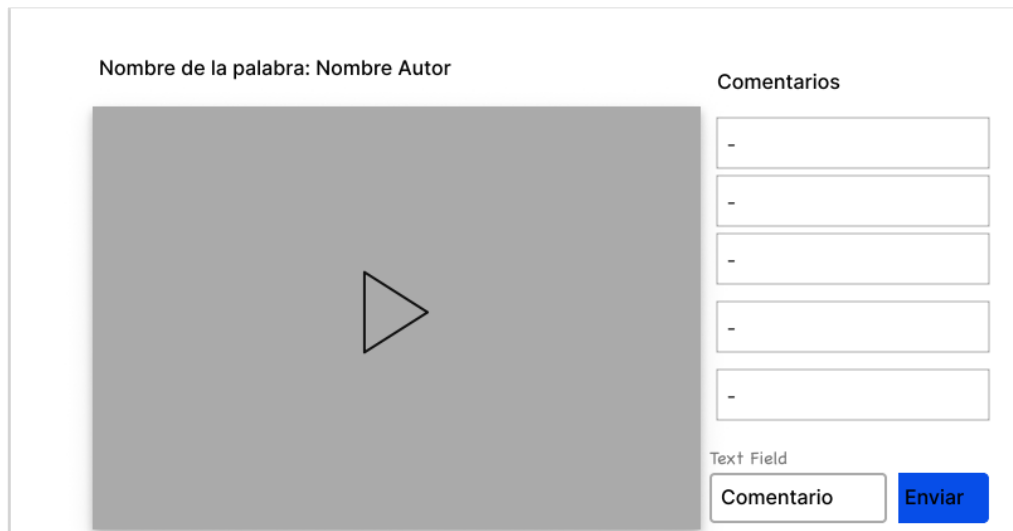
Módulo que servirá para mostrar el contenido del video de la palabra seleccionada en el diccionario, esta funcionalidad podrá hacerse aun si el usuario no está registrado en la aplicación, pudiendo reproducir o dar pausa al video de ser necesario.

En la parte superior se mostrará información como cuál es la palabra la cual se practica en el video y el nombre del usuario quien sube la misma.

Todo aquel usuario que este con sesión activa en el momento de ver el video podrá comentar aspectos del mismo si así lo desea, en caso contrario solo sea un visitante sin haberse autenticado en la aplicación no tendrá disponible esta funcionalidad y no se mostrará, además evitará mostrar los comentarios de retroalimentación de otros usuarios hacia el video, esto con el afán de incitar al visitante a unirse a la comunidad.

Se podrá regresar al diccionario en cualquier momento solo presionando la opción de diccionario en la parte superior de la pantalla.

Figura 6. **Visualización de video**



Fuente: elaboración propia, empleando figma.com.

3.2.7. **Panel de usuario**

El módulo de panel de usuario es parecido en cierta manera al panel de administrador, con la diferencia que este presentará un listado solo de videos propios subidos por el usuario, el cual indicará por video un botón de ver video, el cual enviará al usuario al módulo de ver video y un indicador de estado del video con las siguientes opciones:

- Video pendiente de revisión: este estado se mostrará si el usuario ha subido el video, pero este no ha sido revisado por un administrador.
- Video aceptado: este estado se mostrará si el video ha sido revisado por un administrador y este fue aceptado para formar parte del diccionario.
- Video rechazado: este estado se mostrará si el video ha sido rechazado por no cumplir con los lineamientos requeridos para agregar un video al diccionario.

Figura 7. **Videos subidos por el usuario**

Panel

Mi videos

- Palabra nueva 1	▶ Estado del video
- Palabra nueva 2	▶ Estado del video
- Palabra nueva 3	▶ Estado del video
- Palabra nueva 4	▶ Estado del video

Fuente: elaboración propia, empleando figma.com.

4. IMPLEMENTACIÓN DE PROPUESTA DE APLICACIÓN

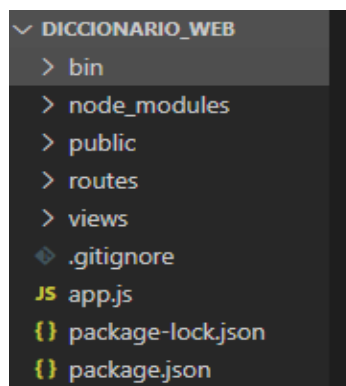
4.1. Descripción de la implementación

La aplicación de una manera muy generalizada está basada en el modelo vista controlador el cual define una clara estructura para el manejo de la base de datos, para el manejo de los controladores y por último una estructura para mostrar las páginas en la aplicación.

4.1.1. Estructura del proyecto

El proyecto esta creado con el generador de aplicaciones del *framework* seleccionado para programar en Node.js, el cual es *Express*, de tal manera que, al terminar de generarse todas las carpetas y archivos básicos del proyecto, su estructura desde la carpeta raíz del mismo es la siguiente:

Figura 8. Estructura general de proyecto



Fuente: elaboración propia, empleando la aplicación recortes.

Dependiendo de la versión de Node.js que se esté usando pueden variar el nombre de las carpetas, pero en general todos los proyectos generados deberían tener la misma estructura inicial, la cual se detalla en la siguiente tabla:

Tabla I. **Descripción de general de carpetas y archivos del proyecto**

Nombre de archivo/carpeta	Descripción
bin	Carpeta que contiene un archivo de nombre: <code>www</code> , el cual en su contenido posee una serie de configuraciones al momento de iniciar la aplicación.
<i>node_modules</i>	Carpeta designada para almacenar todas las dependencias que se instalan al proyecto que este necesita para iniciar a funcionar.
<i>public</i>	Carpeta encargada de almacenar todos aquellos archivos estáticos en el proyecto como los archivos de JavaScript, hojas de estilos (CSS) o en su defecto imágenes que la página web necesitará en ciertos momentos.
<i>routes</i>	Carpeta que contiene archivos de JavaScript, encargados de enlazar la lógica de programación del proyecto con las rutas las cuales empleará para acceder a dicha lógica de programación.

Continuación de la tabla I.

Nombre de archivo/carpeta	Descripción
<i>views</i>	Carpeta que contiene archivos con extensión.ejs los cuales son utilizados para mostrar al usuario los diferentes módulos programados en la aplicación de manera visual en el navegador.
<i>app.js</i>	Es el archivo principal del proyecto en el cual se crean todas las posibles funciones de petición al usuario, ya sean solo de extracción o ingreso de datos o bien para mostrar en el navegador según la ruta una página en formato HTML, así como la declaración de todas las variables globales utilizadas en todo el proyecto como lo son la conexión a la base de datos o la configuración de los archivos estáticos en la carpeta <i>public</i> .
<i>package.json</i>	Archivo el cual especifica en detalle las dependencias instaladas en la carpeta node_modules, así como su versión y si solo se instalaron por motivos de desarrollo o motivos de producción.

Continuación de la tabla I.

Nombre de archivo/carpeta	Descripción
<i>Package-lock.json</i>	Archivo el cual almacena información importante de la instalación de todas las dependencias del proyecto almacenadas en la carpeta <code>node_modules</code> .

Fuente: elaboración propia.

4.1.2. Manejo de versiones de código

El proyecto será almacenado en un servidor de control de versiones el cual se encargará de almacenar todos los cambios que se le hagan al código del proyecto, para esto en la creación del proyecto por medio del generador de aplicaciones de *Express*, se tiene que indicar que se usará como control de versiones de código la herramienta git y se crearán los siguientes elementos al proyecto:

Tabla II. **Descripción de elementos sobre el control de versiones**

Nombre de carpeta/archivo	Descripción
.git	Carpeta creada automáticamente para el control de versiones en git, contiene generalmente archivos de historial y bitácoras de los cambios al código.

Continuación de la tabla II.

Nombre de carpeta/archivo	Descripción
<i>.gitignore</i>	Archivo el cual especifica en su contenido las carpetas y los archivos a los cuales no se les dará seguimiento en el control de versiones del código.

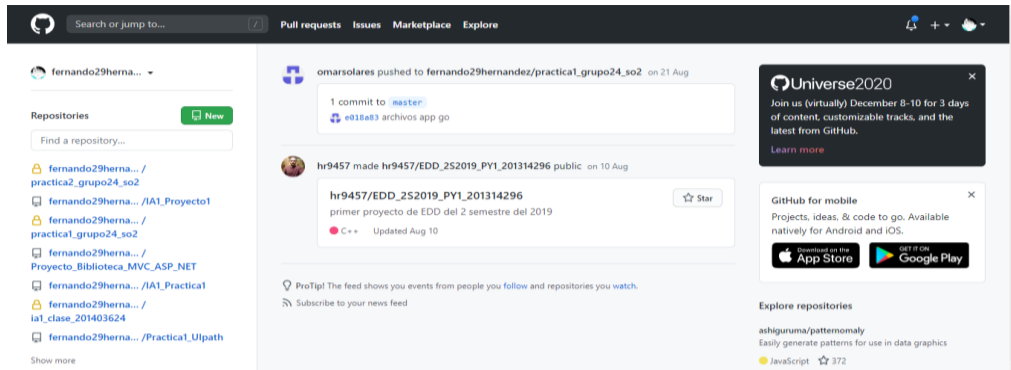
Fuente: elaboración propia.

Desde este punto a la carpeta contenedora del proyecto se le conocerá como repositorio del proyecto, por el momento de manera local (ubicado en nuestra computadora), por la simple razón de estar bajo el control de versiones de git y monitorear todos los posibles cambios a los archivos contenidos en esta carpeta.

4.1.2.1. Configuración de repositorio en línea

Se tiene que crear el repositorio en un servidor de versiones, para este caso el seleccionado será github, como se muestra en la siguiente imagen:

Figura 9. Página de inicio *github*



Fuente: elaboración propia, empleando la aplicación recortes.

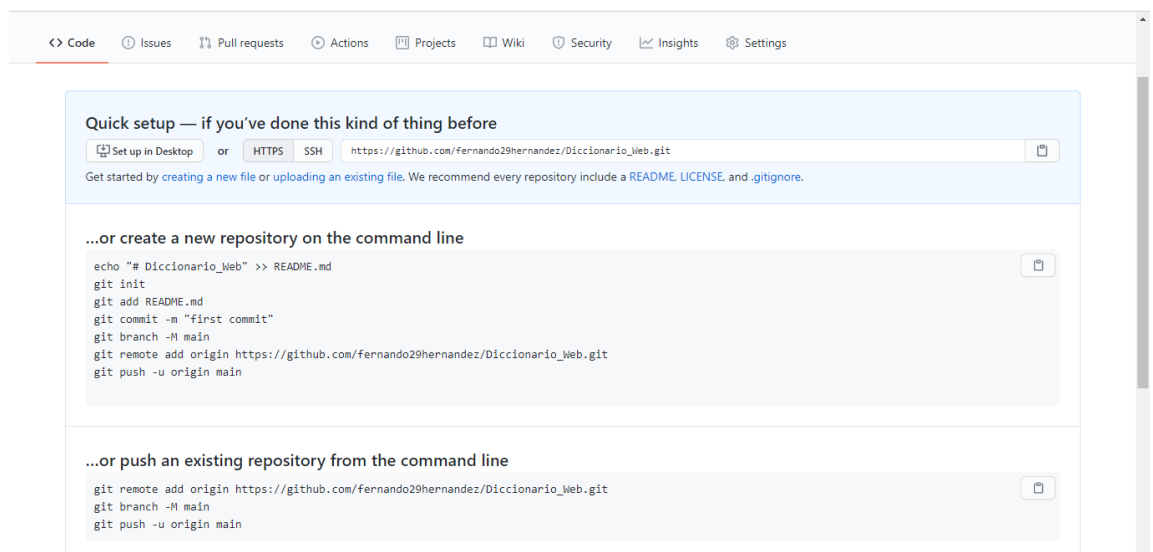
Estado en esta página lo que se tendrá que hacer es presionar el botón verde con la palabra: New, la cual se enviara a otra página mostrándose el siguiente formulario a llenar:

Figura 10. Creación de repositorio

Fuente: elaboración propia, empleando la aplicación recortes.

Como se muestra se solicitan los campos de nombre del proyecto, si será público o privado (solo nosotros tendremos acceso), una descripción y entre otras configuraciones que para este proyecto se dejarán tal cual están por defecto. Después de haber llenado todos los campos requeridos se enviará a la página siguiente mostrándose información importante para enlazar el repositorio en este servidor de versiones con el código que se tiene en la computadora de manera local.

Figura 11. **Comandos importantes para el enlace de repositorio**



Fuente: elaboración propia, empleando la aplicación recortes.

De esta última, se necesita solo el comando que aparece dentro del área celeste que servirá para decirle al repositorio local la dirección web del repositorio creado en el servidor.

4.1.2.2. Enlace de repositorio local

Como último paso para la comunicación del repositorio local con el que está en el servidor en línea, necesita ingresar los siguientes comandos en una consola de sistema en el sistema operativo, y ubicar en la carpeta en la cual se tiene el código del proyecto, que es el único lugar donde funcionará la configuración antes mencionada.

Figura 12. **Agregar repositorio remoto al proyecto**

```
git remote add origin https://github.com/fernando29hernandez/Diccionario_Web.git
```

Fuente: elaboración propia, empleando la aplicación recortes.

De esta forma ya se tienen enlazados ambos repositorios y cada vez que se requiera subir los nuevos cambios del código escrito localmente al repositorio en el servidor, basta con escribir el siguiente comando en consola de sistema operativo:

Figura 13. **Realizar cambios nuevos al repositorio en el servidor**

```
git push origin master
```

Fuente: elaboración propia, empleando la aplicación recortes.

Todo lo anterior configurado sirve para hacer un despliegue de la aplicación de manera más sencilla dado a que solo se necesitará tener la ubicación del repositorio en línea para descargar el código en el servidor donde esté funcionando la aplicación del diccionario.

4.1.3. Configuraciones del ambiente desarrollo de aplicación

Se explica el propósito de los archivos involucrados para el despliegue de la aplicación de la manera más rápida y sin inconvenientes, lo que puede llegar a facilitar futuras actualizaciones dentro del servidor en la nube en donde se esté alojando la aplicación.

4.1.3.1. Descripción *dockerfile*

Se explican todas las líneas escritas en el archivo el cual se encarga de crear una imagen de *docker* en donde se estará ejecutando un servidor de web con todo el código dentro de la imagen.

Figura 14. Contenido de *dockerfile* para la aplicación

```

FROM node:10-alpine

# Creación de carpeta de proyecto dentro de contenedor
RUN mkdir /app
WORKDIR /app
# Copia de archivo package.json
COPY package.json ./
# Instalación de dependencias de proyecto
RUN npm install
RUN npm install http-errors
RUN npm install -g nodemon
RUN npm install --save multer
# Puerto expuesto para comunicación fuera del contenedor
EXPOSE 3000
# comando de inicio a la aplicación
CMD [ "nodemon","start" ]

```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Tabla III. Descripción de comando en *dockerfile* para la aplicación

Comando	Descripción
FROM node:10-alpine	Imagen de <i>docker</i> sobre el lenguaje a instalar describiendo su versión.
RUN mkdir /app	Creación de carpeta en la cual estará el código de la aplicación dentro del contenedor de <i>docker</i> .
WORKDIR /app	Comando para configurar como espacio de trabajo a la carpeta actual.

Continuación de la tabla III.

Comando	Descripción
COPY <i>package.json</i> ./	Copia el archivo <i>package.json</i> el cual servirá para instalar todas las dependencias del proyecto dentro del contenedor de <i>docker</i> .
RUN <i>npm install</i>	Instala todas las dependencias que contiene el archivo <i>package.json</i> .
RUN <i>npm install -g nodemon</i>	Instala un paquete llamado <i>nodemon</i> que servirá para iniciar la aplicación dentro del contenedor.
EXPOSE 3000	Indica que el puerto por el cual la aplicación saldrá del contenedor será el puerto 3 000.
CMD ["nodemon","start"]	Inicia la aplicación desde el contenedor.

Fuente: elaboración propia.

4.1.3.2. Descripción *docker-compose.yml*

Se explican todas las líneas escritas en el archivo el cual se encarga de automatizar lo antes descrito en el archivo *dockerfile* y el enlace posterior ejecución en conjunto con la base de datos.

Figura 15. Contenido de *docker-compose.yml* para la aplicación

A screenshot of a terminal window with a dark background and light-colored text. The terminal shows the configuration for a Docker Compose service named 'aplicacion'. The configuration includes the Docker Compose version '3.7', the service name 'aplicacion', the container name 'diccionario_web', the build context '.', and two bind volumes: one for the application source code and another for the Node.js modules. The service is configured to listen on port 80 and map it to port 3000 on the container, and it is set to restart automatically.

```
version: '3.7'

services:
  aplicacion:
    container_name: diccionario_web
    build: .
    volumes:
      - type: bind
        source: ./
        target: /app
      - /app/node_modules
    ports:
      - "80:3000"
    restart: always
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Tabla IV. Descripción de comandos en *docker-compose.yml*

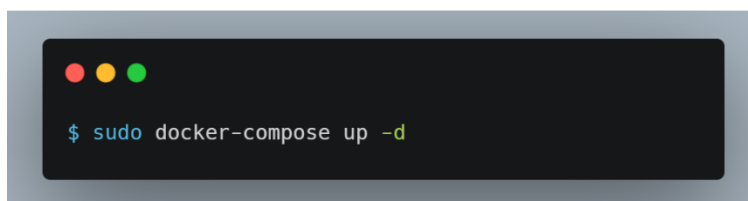
Comando	Descripción
version:'3.7'	Versión del lenguaje propio de <i>docker</i> en el cual se construirá con base en sus estándares las reglas del contenedor.
container_name:diccionario_web	Nombre del contenedor de <i>docker</i> .
build: .	Indica la dirección del archivo <i>dockerfile</i> , en el cual construirá su contenedor.
volumes:	Indica en donde estará el código fuente de la aplicación y además en donde ira a almacenarse dentro del contenedor.
ports: "80:3000"	Indica que el contenedor traducirá el puerto 3 000 por el puerto 80 del servidor.
restart:'always'	Indica que ante cualquier fallo que ocurra en el sistema o en la aplicación, la aplicación se reiniciara.

Fuente: elaboración propia.

4.1.3.3. Iniciar ambiente de desarrollo

Para iniciar con el ambiente de pruebas para el período en el que se desarrolló la aplicación solo basta con colocar en funcionamiento los archivos de configuración de *docker* antes mencionados en este capítulo. Para empezar su funcionamiento se debe de abrir una consola del sistema operativo y ubicarse en la carpeta del proyecto de la aplicación, luego escribir el siguiente comando:

Figura 16. Iniciar contenedores de *docker* para aplicación



Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Luego de correr dicho comando, se tiene que abrir un navegador y escribir la dirección del sitio en este caso para ambiente de pruebas locales, se utiliza por defecto el sitio localhost.

4.1.4. Base de datos

Utilizada para el manejo de la información en cuestiones de lectura y escritura de la información entre un gestor de bases de datos y la aplicación web en donde se asegura que la dicha información se almacenara de manera segura e integra.

4.1.4.1. Descripción de base de datos

Se utilizó la base de datos MySQL dado a todas las ventajas que esta brinda a la aplicación como lo es su facilidad de conexión con el lenguaje de programación en el cual está hecho el diccionario. Se pensó en la manera más eficiente para el despliegue de la base de datos tanto en el ambiente de pruebas como en el de producción, por lo consiguiente se optó por la dockerización de la misma.

4.1.4.2. Descripción de *docker-compose.yml*

La aplicación necesita del un archivo como *docker-compose.yml* para ejecutar todo lo necesario para mantener en funcionamiento la aplicación de manera constante y sin interrupciones.

Figura 17. Contenido de *docker-compose.yml* para la base de datos

```
version: "2"
services:
  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: 'db'
      MYSQL_USER: 'user'
      MYSQL_PASSWORD: 'password'
      MYSQL_ROOT_PASSWORD: 'password'
    ports:
      - '32000:3306'
    expose:
      - '32000'
    volumes:
      - my-db:/var/lib/mysql
volumes:
  my-db:
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Tabla V. Descripción de *docker-compose.yml* para base de datos

Comando	Descripción
version:'2'	Versión del lenguaje propio de <i>docker</i> en el cual se construirá con base a sus estándares las reglas del contenedor.
db:	Nombre del contenedor de <i>docker</i> .
image:mysql:5.7	Indica que la imagen a instalar en el contenedor la cual será de una base de datos MySQL en su versión 5.7.
restart:'always'	Indica que ante cualquier fallo que ocurra en el sistema o en la aplicación, la aplicación se reiniciará.
environment:	Espacio utilizado para la declaración de variables de entorno para la base de datos.
MYSQL_DATABASE: 'db'	Nombre de la base de datos a utilizar para el desarrollo de la aplicación.
MYSQL_USER: 'user'	Usuario para autenticarse en la base de datos.
MYSQL_PASSWORD: 'password'	Contraseña para el usuario que tendrá la base de datos.
MYSQL_ROOT_PASSWORD: 'password'	Contraseña del usuario root (usuario con todos los permisos en la base de datos).
ports: - '32000:3306'	Indica puerto de salida para conectar a la base de datos

Continuación de la tabla V.

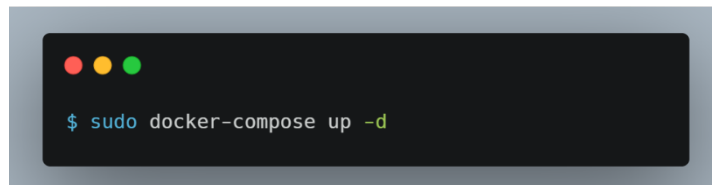
Comando	Descripción
volumes: - my-db:/var/lib/mysql	Indica la ruta virtual y la ruta de la carpeta en la cual se estará almacenando los datos de la base de datos de este contenedor.
volumes: my-db:	Indica cómo se llama la ruta virtual de almacenamiento de los datos de la base de datos.

Fuente: elaboración propia.

4.1.4.3. Iniciar de base de datos de pruebas

Para empezar su funcionamiento se debe abrir una consola del sistema operativo y ubicarse en la carpeta del proyecto de la aplicación, luego escribir el siguiente comando:

Figura 18. Iniciar contenedor de *docker* de base de datos de pruebas

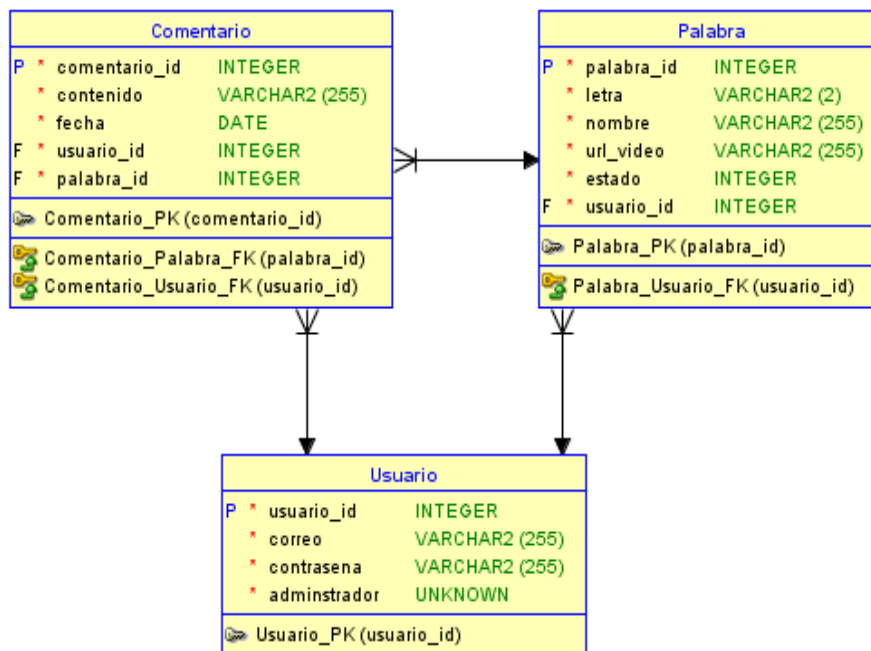
A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The text '\$ sudo docker-compose up -d' is displayed in a light blue font, indicating a command being executed in a shell.

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

4.1.4.4. Modelo entidad relación

Es una representación gráfica de cómo se almacena la información en la base de datos, como están los datos relacionados entre sí y que tipo de datos conforman una entidad.

Figura 19. Modelo entidad relación para la aplicación



Fuente: elaboración propia, empleando la aplicación Data Modeler.

4.1.4.5. Script sql con esquema de la base de datos

La base de datos al momento de crearse no tiene ningún esquema relacional, por lo que se necesita ingresar las siguientes instrucciones para brindar las entidades necesarias para su uso.

Figura 20. Descripción de esquema de base de datos

```
use db;

CREATE TABLE Usuario (
  usuario_id INT NOT NULL AUTO_INCREMENT,
  correo VARCHAR(255) NOT NULL,
  contraseña VARCHAR(255) NOT NULL,
  administrador BOOLEAN NOT NULL,
  PRIMARY KEY(usuario_id)
);

CREATE TABLE Palabra(
  palabra_id INT NOT NULL AUTO_INCREMENT,
  letra VARCHAR(2) NOT NULL,
  nombre VARCHAR(255) NOT NULL,
  url_video VARCHAR(255) NOT NULL,
  estado INT NOT NULL,
  usuario_id INT NOT NULL,
  FOREIGN KEY (usuario_id) REFERENCES Usuario(usuario_id),
  PRIMARY KEY(palabra_id)
);

CREATE TABLE Comentario(
  comentario_id INT NOT NULL AUTO_INCREMENT,
  contenido VARCHAR(255) NOT NULL,
  fecha DATE NOT NULL,
  usuario_id INT NOT NULL,
  palabra_id INT NOT NULL,
  FOREIGN KEY (usuario_id) REFERENCES Usuario(usuario_id),
  FOREIGN KEY (palabra_id) REFERENCES Palabra(palabra_id),
  PRIMARY KEY(comentario_id)
);

show tables;
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

4.1.4.6. Descripción de tablas

En las siguientes tablas se especifica de cada una de las entidades un desglose de sus atributos.

Tabla VI. Descripción de entidades en modelo

Nombre entidad	Descripción/Función
Usuario	Entidad encargada de almacenar todos los datos relevantes tanto de los usuarios normales como de los que son administradores de la página.
Palabra	Entidad encargada de almacenar todas las palabras y videos asociados a estas de tal manera que se identifique qué usuario agrego la nueva propuesta de agregación al diccionario.
Comentario	Entidad encargada de almacenar los datos relevantes de cada comentario asociados a cada uno de los videos de las palabras subidas al diccionario.

Fuente: elaboración propia.

Tabla VII. Entidad usuario

Nombre atributo	Descripción	Tipo	Llave primaria	Llave foránea	Entidad referencia
usuario_id	Identificador único de cada usuario	int	Sí	No	No aplica
correo	Correo de usuario	varchar	No	No	No aplica
contraseña	Contraseña de usuario	varchar	No	No	No aplica
administrador	Indica si el usuario es un administrador del diccionario web.	boolean	No	No	No aplica

Fuente: elaboración propia.

Tabla VIII. **Entidad palabra**

Nombre Atributo	Descripción	Tipo	Llave primaria	Llave foránea	Entidad referencia
palabra_id	Identificador único de cada palabra del diccionario	Int	Sí	No	No aplica
letra	Letra inicial de cada la palabra en mayúscula	varchar	No	No	No aplica
nombre	Nombre de la palabra	varchar	No	No	No aplica
url_video	Dirección que indica donde se encuentra almacenado el video asociado a la palabra	varchar	No	No	No aplica
estado	Estado del video en donde: 0 significa: en espera de revisión, 1 significa video aceptado y 2 significa video rechazado	int	No	No	No aplica

Continuación de la tabla VIII.

Nombre Atributo	Descripción	Tipo	Llave primaria	Llave foránea	Entidad referencia
usuario_id	Identificador asociado al usuario quien propuso la nueva palabra para formar parte del diccionario.	int	No	Si	Usuario

Fuente: elaboración propia.

Tabla IX. **Entidad comentario**

Nombre Atributo	Descripción	Tipo	Llave primaria	Llave foránea	Entidad Referencia
comentario_id	Identificador único de cada palabra del diccionario	Int	Sí	No	No aplica
contenido	Letra inicial de cada la palabra en mayúscula	varchar	No	No	No aplica
fecha	Nombre de la palabra	varchar	No	No	No aplica

Continuación de la tabla IX.

Nombre Atributo	Descripción	Tipo	Llave primaria	Llave foránea	Entidad Referencia
palabra_id	Identificador de la palabra a la cual se quiere asociar un comentario sobre el video.	int	No	Sí	Palabra
usuario_id	Identificador asociado al usuario propietario de la nueva palabra.	int	No	Sí	Usuario

Fuente: elaboración propia.

4.2. Descripción de módulos principales en la aplicación

Descripción general de las implementaciones de código en la aplicación por módulo.

4.2.1. Conexión con base de datos

Configuraciones hechas entre la base de datos y la aplicación en sí, de tal manera que se mantenga estable en todo momento, no interrumpiendo el funcionamiento del *software* y proporcionando una manera de recuperación en caso la conexión llegase a fallar.

Figura 21. Configuración para conexión con base de datos

```
const mysql = require('mysql');

var pool = mysql.createPool({
  host      : '35.194.71.234',
  user      : 'user',
  password  : 'password',
  database  : 'db',
  port      : 32000
});

module.exports = {
  query: function(){
    var sql_args = [];
    var args = [];
    for(var i=0; i<arguments.length; i++){
      args.push(arguments[i]);
    }
    var callback = args[args.length-1];
    pool.getConnection(function(err, connection) {
      if(err) {
        console.log(err);
        return callback(err);
      }
      if(args.length > 2){
        sql_args = args[1];
      }
      connection.query(args[0], sql_args, function(err, results) {
        connection.release();
        if(err){
          console.log(err);
          return callback(err);
        }
        callback(null, results);
      });
    });
  }
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

4.2.2. Manejo de las rutas de la aplicación

Se debe de tener una estructura para acceder a los recursos a los que se dispone por eso se necesita una organización de las rutas lo mas fácil y claro posible.

Figura 22. Archivo de rutas para el manejo de usuarios

```
var express = require('express');
var router = express.Router();

// objeto con los metodos a asociar con las rutas
const usuarioControlller = require('../controllers/usuario.controller')

// creacion de un usuario
router.post('/', usuarioControlller.create);
// busqueda de un usuario en especifico
router.get('/:id', usuarioControlller.findById);
// verificacion de usuario para ingreso de sesion en la aplicacion
router.post('/verify', usuarioControlller.verifyUser);

// incluir este objeto en las variables globales
module.exports = router;
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 23. Archivo de rutas para el manejo de palabras

```
var express = require('express');
var router = express.Router();
//objeto creado para asociar los metodos de las palabras con sus rutas
const palabraControlller = require('../controllers/palabra.controller');
// crear una nueva palabra
router.post('/', palabraControlller.create);
// obtener todas la palabras en en diccionario
router.get('/', palabraControlller.findAll);
// obtener todas la palabras pendientes de revision
router.get('/pending', palabraControlller.findAllPending);
// obtener una palabra mediante su identificador
router.get('/:id', palabraControlller.findById);
// obtener todas las palabras creadas por un usuario en especifico
router.get('/user/:id', palabraControlller.findById);
// actualizar estado de una palabra en especifico
router.put('/', palabraControlller.update);

module.exports = router;
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 24. Archivo de rutas para el manejo de comentarios

```
var express = require('express');
var router = express.Router();
// creacion de objeto que enlaza los comentarios con las rutas
const comentarioControlller = require('../controllers/comentario.controller');
// creacion de un comentario nuevo
router.post('/', comentarioControlller.create);
// obtencion de los comentarios segun una palabra en especifico
router.get('/:id', comentarioControlller.findById);

module.exports = router;
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

4.2.3. Ingreso de nuevos usuarios

Se muestra los fragmentos más importantes del código relacionado con la funcionalidad antes descrita, desde el momento que entra al servidor y se extrae la información más importante para posterior se almacene en la base de datos.

Figura 25. Manejo de un nuevo usuario

```
exports.create = function(req, res) {
  const new_user = new Usuario(req.body);
  if(req.body.constructor === Object && Object.keys(req.body).length !== 0){
    res.status(400).send({
      error:true,
      message: 'Please provide all required field'
    });
  }else{
    Usuario.create(new_user, function(err, user) {
      if (err) {
        res.send(err);
      }
      res.json({
        find:1,
        message:'Usuario added successfully!',
        result:user
      });
    });
  }
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 26. Manejo de nuevo usuario en bases de datos

```
Usuario.create = function (newUser, result) {
  let new_insert = "INSERT INTO Usuario set ?"
  conexion.query(new_insert, newUser, function (err, res) {
    if(err) {
      result(err, null);
    }
    else{
      result(null, res.insertId);
    }
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 27. **Uso de recurso para un nuevo usuario**

```
let body = {
  "correo": variable_correo,
  "contrasena": variable_password,
  "adminstrador": variable_admin
}
await fetch('/api/v1/users/', {
  method: 'POST',
  body: JSON.stringify(body),
  headers: {
    'Content-Type': 'application/json'
  }
})
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 28. **Vista del registro usuarios**

The image shows a web registration form. At the top, there is a navigation bar with links for 'Diccionario', 'Home', 'Ver Diccionario', 'Ingreso', and 'Regístrate'. The main content area is a white box with a light gray border. Inside, the title 'Regístrate' is centered. Below the title are three input fields: 'E-mail', 'Password', and 'Confirm Password'. A teal 'Registrar' button is positioned below the fields. At the bottom of the form, there is a link: '¿Ya tienes una cuenta? [Ingresa](#)'.

Fuente: elaboración propia, empleando la aplicación recortes.

4.2.4. Inicio de sesión

Código necesario para el manejo del inicio de sesión en la aplicación desde que se consulta si el usuario existe en base de datos hasta que se valida que la información coincida con el usuario al que se quiere acceder.

Figura 29. Manejo de inicio de sesión de un usuario

```
exports.verifyUser = function(req, res) {
  Usuario.verifyUser(req.body, function(err, user) {
    if (err)
      res.send(err);
    let finduser = user.length>0?1:0;
    let resultbody = user.length>0?user[0]:{};
    res.json({find:finduser,result:resultbody});
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 30. Manejo de inicio de sesión en base de datos

```
Usuario.verifyUser = function (body, result) {
  let consult_login="Select * from Usuario where correo = ? and contrasena = ? "
  conexion.query(consult_login, [body.correo ,body.pass], function (err, res) {
    if(err) {
      result(err, null);
    }
    else{
      result(null, res);
    }
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 31. **Uso de recurso para inicio de sesión**

```
let body = {
  "correo": variable_correo,
  "pass": variable_password
}

await fetch('/api/v1/users/verify', {
  method: 'POST',
  body: JSON.stringify(body),
  headers: {
    'Content-Type': 'application/json'
  }
})
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 32. **Vista de inicio de sesión**

The image shows a web page with a navigation bar at the top containing 'Diccionario', 'Home', and 'Ver Diccionario' on the left, and 'Ingreso' and 'Registrate' on the right. The main content area features a login form titled 'Ingreso'. The form includes two input fields: 'E-mail' and 'Password'. Below these fields is a teal button labeled 'Ingresar'. At the bottom of the form, there is a link that reads '¿Eres Nuevo? Registrate'.

Fuente: elaboración propia, empleando la aplicación recortes.

4.2.5. Creación de palabras nuevas en el diccionario

Módulo principal de la aplicación el cual se encarga de ingresar nuevas palabras para revisión en el diccionario.

Figura 33. Manejo de una palabra nueva

```
exports.create = function(req, res) {
  const new_word = new Palabra(req.body);
  if(req.body.constructor === Object && Object.keys(req.body).length === 0){
    res.status(400).send({
      error:true,
      message: 'Please provide all required field'
    });
  }else{
    Palabra.create(new_word, function(err, palabra) {
      if (err)
        res.send(err);
      res.json({
        message:"Palabra added successfully!",
        result:palabra
      });
    });
  }
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 34. Manejo de una palabra nueva con base de datos

```
Palabra.create = function (newWord, result) {
  let new_insert = "INSERT INTO Palabra set ?"
  conexion.query(new_insert, newWord, function (err, res) {
    if(err) {
      console.log("error: ", err);
      result(err, null);
    }
    else{
      console.log(res.insertId);
      result(null, res.insertId);
    }
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 35. **Uso de recurso para creación de palabra nueva**

```
let body = {
  nombre: variable_nombre_palabra,
  letra: variable_letra_inicial,
  url_video: url_video,
  estado: estado_palabra,
  usuario_id: usuario_id
}

await fetch('/api/v1/palabras', {
  method: 'POST',
  body: JSON.stringify(body),
  headers: {
    'Content-Type': 'application/json'
  }
})
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 36. **Vista de creación de palabra nueva**

Fuente: elaboración propia, empleando la aplicación recortes.

4.2.6. **Ver palabras de usuario**

Módulo de la aplicación que servirá como diccionario en donde los usuarios puedan buscar sus palabras ingresadas en dado caso fuesen aceptadas.

Figura 37. Manejo de listar palabras subidas por usuario

```
exports.findByUserId = function(req, res) {
  Palabra.findByUserId(req.params.id, function(err, word) {
    if (err)
      res.send(err);
    let findWord = word.length>0?1:0;
    let resultbody = word.length>0?word:[];
    res.json({find:findWord,result:resultbody});
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 38. Manejo de listar palabras por usuario en base de datos

```
Palabra.findByUserId = function (id, result) {
  let list_query = "Select * from Palabra where usuario_id = ? "
  conexion.query(list_query, id, function (err, res) {
    if(err) {
      console.log("error: ", err);
      result(err, null);
    }
    else{
      result(null, res);
    }
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 39. **Uso de recurso para listar palabras de usuario**

```
let usuario_id = identificador_de_usuario
await fetch('/api/v1/palabras/user/'+usuario_id, {
  method: 'GET',
})
```

Fuente: elaboración propia, empleando la aplicación carbón.now.sh.

Figura 40. **Vista de listado de palabras de un usuario**

Diccionario	Home	Ver Diccionario	Ingresar Palabra	Dashboard	Cerrar Sesión
Mis aportes al diccionario					
#	Nombre	Url Video	Estado	Acciones	
1	a mimir	1602583477633video0 (1).mp4	Video Aceptado, ahora forma parte del diccionario	Ver Video ▶	
2	abeja	rutadelvideo.mp4	Video Aceptado, ahora forma parte del diccionario	Ver Video ▶	
3	artista	rutadelvideo.mp4	Video rechazado, porfavor verifica el contenido de tu video	Ver Video ▶	

Fuente: elaboración propia, empleando la aplicación recortes.

4.2.7. **Ver palabras en revisión como administrador**

Módulo manejado por el administrador encargado de aceptar o rechazar los videos entrantes en el diccionario para su revisión y posterior inclusión o rechazo.

Figura 41. Manejo de listar palabras a revisión

```
exports.findAllPending = function(req, res) {
  Palabra.findAllPending(function(err, palabras) {
    console.log('controller')
    if (err)
      res.send(err);
    console.log('res', palabras);
    let findWord = palabras.length>0?1:0;
    let resultbody = palabras.length>0?palabras:[];
    res.send({find:findWord,result:resultbody});
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 42. Manejo de listar palabras a revisión en base de datos

```
Palabra.findAllPending = function (result) {
  let pending_select_query= "Select * from Palabra where estado = 0"
  conexion.query(pending_select_query, function (err, res) {
    if(err) {
      console.log("error: ", err);
      result(null, err);
    }
    else{
      console.log('Palabras : ', res);
      result(null, res);
    }
  });
};
```

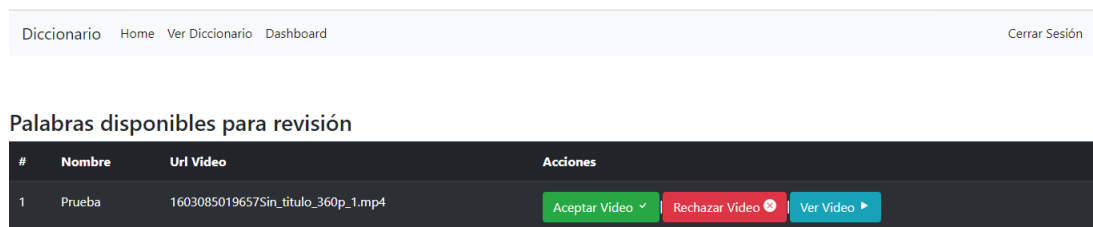
Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 43. **Uso de recurso para listar palabras a revisión**

```
await fetch('/api/v1/palabras/pending', {
  method: 'GET',
})
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 44. **Vista de listado de palabras a revisión**



Fuente: elaboración propia, empleando la aplicación recortes.

Figura 45. **Manejo de listado de diccionario**

```
exports.findAll = function(req, res) {
  Palabra.findAll(function(err, palabras) {
    console.log('controller')
    if (err)
      res.send(err);
    console.log('res', palabras);
    let findWord = palabras.length>0?1:0;
    let resultbody = palabras.length>0?palabras:[];
    res.send({find:findWord,result:resultbody});
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 46. Manejo de listado de diccionario desde la base de datos

```
Palabra.findAll = function (result) {
  conexion.query("Select * from Palabra", function (err, res) {
    if(err) {
      console.log("error: ", err);
      result(null, err);
    }
    else{
      console.log('Palabras : ', res);
      result(null, res);
    }
  });
};
```

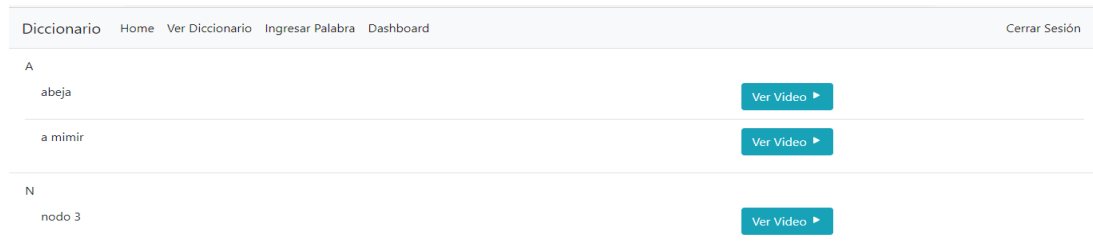
Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 47. Uso de recurso para listar el diccionario

```
await fetch('/api/v1/palabras/', {
  method: 'GET',
})
```

Fuente: elaboración propia, empleando la aplicación carbón.now.sh.

Figura 48. Vista de diccionario



Fuente: elaboración propia, empleando la aplicación recortes.

4.2.8. Ver una palabra en el diccionario

Módulo en la aplicación se tiene que mostrar un video de una palabra seleccionada del diccionario y dependiendo el estado del usuario pueda comentar el video.

Figura 49. Manejo de una palabra

```
exports.findById = function(req, res) {
  Palabra.findById(req.params.id, function(err, word) {
    if (err)
      res.send(err);
    let findWord = word.length>0?1:0;
    let resultbody = word.length>0?word[0]:{};
    res.json({find:findWord,result:resultbody});
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 50. Manejo de una palabra en base de datos

```
Palabra.findById = function (id, result) {
  let query_select= "Select * from Palabra where palabra_id = ? "
  conexion.query(, id, function (err, res) {
    if(err) {
      console.log("error: ", err);
      result(err, null);
    }
    else{
      result(null, res);
    }
  });
};
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 51. Uso de recurso para obtener una palabra

```
await fetch('http://localhost:3000/api/v1/palabras/'+variable_id_palabra+"/",
  {
    method: 'GET',
  }
)
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Figura 52. Vista de una palabra del diccionario



Fuente: elaboración propia, empleando la aplicación recortes.

4.3. Consideraciones de despliegue de aplicación

Se debe tener en cuenta que cuando la aplicación será ejecutada en un servidor en el que se tienen que configurar los puertos de red pertinentes para el correcto funcionamiento, además que las herramientas necesarias instaladas en el servidor en la nube.

4.3.1. Palabras iniciales para el diccionario

Como parte del lanzamiento de la aplicación se elaboraron videos los cuales serán el contenido inicial del diccionario, también con la función de ser un formato de ejemplo para próximos videos subidos al diccionario. Se muestra una lista de capturas de pantalla de los videos de las palabras:

4.3.1.1. Palabra número 1

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta buenos días.

Figura 53. Video corto para buenos días

La palabra que estamos viendo es: **Buenos Días**



Fuente: elaboración propia.

4.3.1.2. Palabra número 2

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta buenas noches.

Figura 54. **Video corto para buenas noches**



Fuente: elaboración propia.

4.3.1.3. **Palabra número 3**

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta bien.

Figura 55. **Video corto para bien**



Fuente: elaboración propia.

4.3.1.4. Palabra número 4

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta cómo estas.

Figura 56. Video corto para cómo estas



Fuente: elaboración propia.

4.3.1.5. Palabra número 5

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar las palabras computadora portátil.

Figura 57. **Video corto para computadora portátil**



Fuente: elaboración propia.

4.3.1.6. **Palabra número 6**

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta excelente.

Figura 58. **Video corto para excelente**



Fuente: elaboración propia.

4.3.1.7. Palabra número 7

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta hola.

Figura 59. Video corto para hola



Fuente: elaboración propia.

4.3.1.8. Palabra número 8

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta mucho gusto.

Figura 60. **Video corto para mucho gusto**



Fuente: elaboración propia.

4.3.1.9. **Palabra número 9**

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta permiso.

Figura 61. **Video corto para permiso**



Fuente: elaboración propia.

4.3.1.10. Palabra número 10

Este video muestra un ejemplo de cómo se deben hacer los movimientos de manos para recrear y practicar la frase corta regular.

Figura 62. Video corto para regular



Fuente: elaboración propia.

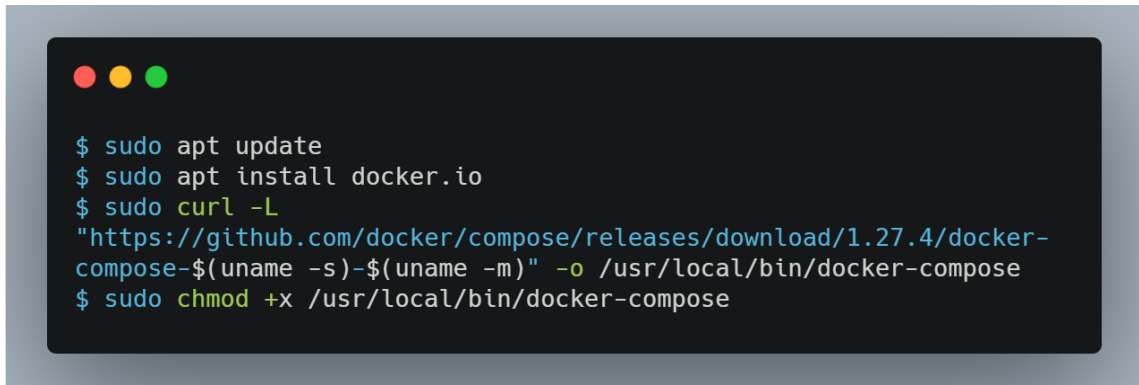
4.3.2. Configuración para despliegue en producción

Los siguientes pasos la forma mas optima y rápida de desplegar la aplicación en un servidor remoto.

4.3.2.1. Instalación de *docker*

En la consola del sistema operativo del servidor en donde se desplegará la aplicación, se tiene que colocar en orden los siguientes comandos uno a uno:

Figura 63. **Instalación de *docker* y *docker-compose***

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal displays the following commands and their output:

```
$ sudo apt update
$ sudo apt install docker.io
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

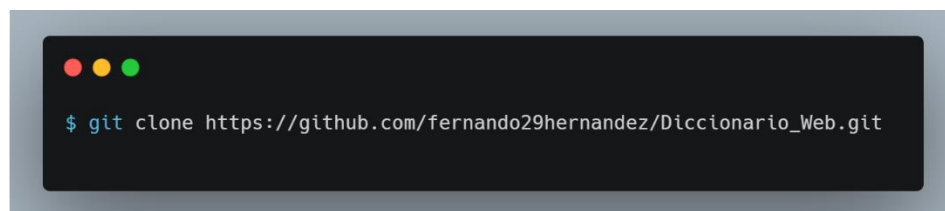
Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Luego de ejecutar todos los comandos antes descritos se tendrá instalado *docker* y *docker-compose* para el ambiente de producción de la base de datos y de aplicación del diccionario.

4.3.2.2. **Descarga de código en servidor**

Se tiene que colocar el siguiente comando en la consola del sistema operativo del servidor y posterior se tiene que ejecutar.

Figura 64. **Clonación de repositorio de aplicación**

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal displays the following command:

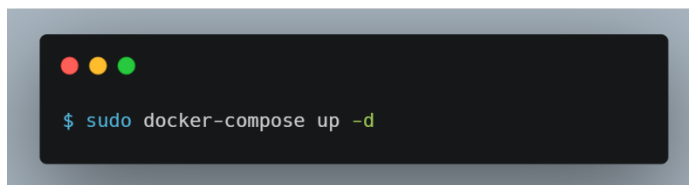
```
$ git clone https://github.com/fernando29hernandez/Diccionario_Web.git
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

4.3.2.3. Iniciar de base de datos en producción

Una vez se tenga el código en el servidor, se tiene que dirigir hacia la carpeta que contiene dicho archivo de configuración de la base de datos, esta es una carpeta con nombre db, luego para crear la instancia de MySQL, se tiene que ingresar el siguiente comando en la consola del sistema operativo:

Figura 65. Inicio de base de datos en ambiente de producción

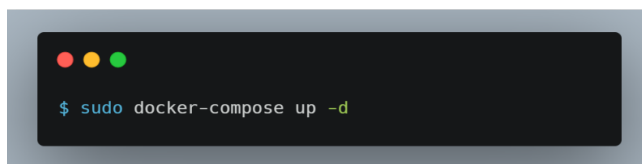
A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The text '\$ sudo docker-compose up -d' is displayed in a light blue monospace font.

Fuente: elaboración propia, Realizado con la aplicación carbon.now.sh.

4.3.2.4. Iniciar aplicación en producción

Con el contenedor de la instancia de MySQL, iniciado toca como siguiente paso el despliegue de la aplicación por lo que se tiene que en primer lugar en la consola del sistema operativo del servidor dirigirse a la carpeta de la aplicación e introducir el siguiente comando:

Figura 66. Inicio de aplicación en ambiente de producción

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The text '\$ sudo docker-compose up -d' is displayed in a light blue monospace font.

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Luego de esto se descargarán todos los complementos necesarios para el despliegue de la aplicación, esperado que todos los procesos terminen se verá al final la siguiente información:

Figura 67. **Salida de resultado de iniciar aplicación**

A terminal window with a dark background and light text. The output shows the following lines:

```
$(nodemon) 2.0.5
$(nodemon) to restart at any time, enter `rs`
$(nodemon) watching path(s): *.*
$(nodemon) watching extensions: js,mjs,json
$(nodemon) starting `node ./bin/www start`
$Database Connected!
```

Fuente: elaboración propia, empleando la aplicación carbon.now.sh.

Indicando así que la aplicación esta lista para usarse y se puede ingresar a la misma por medio de la dirección IP del servidor de producción.

CONCLUSIONES

1. Se completó el desarrollo de la aplicación web propuesta con el enfoque a de ayuda hacia el aprendizaje de lenguaje de señas.
2. Se desarrolló una versión inicial de la aplicación de tal forma que se pueda visualizar cada video de las palabras existentes en el diccionario.
3. Se documento todo el proceso de creación del diccionario para posibles mejoras que puedan sugir y se implementen de una manera sencilla.
4. Se habilitó una caja de comentarios donde es posible dar retroalimentación a los usuarios los cuales suben su video relacionado a una palabra y así en próximos videos mejorar la calidad tanto de su manejo del lenguaje como la diversidad de palabras en el diccionario.

RECOMENDACIONES

1. Practicar para dominar el lenguaje de señas, se espera que el uso de la aplicación logre en gran medida ayudar las personas a practicar dicho lenguaje.
2. Considerar poseer una cuenta en la aplicación para gozar de los beneficios como la posibilidad de contribuir al diccionario y poder comentar los videos propios y de otros usuarios en el diccionario.
3. Respetar toda opinión expresada a través de los comentarios en la aplicación para poder adquirir retroalimentación sobre el progreso en la práctica de la lengua de señas.

BIBLIOGRAFÍA

1. Benemérito Comité Pro-ciegos y sordos de Guatemala. *¿Quiénes Somos?* Disponible en: <https://prociegosysordos.org.gt/QuienesSomos.php>. Consultado septiembre 2020.
2. Canales Creciendo en Señas *Información para La Familia, Niñas y niños sordos de familia sorda*. s.f. Consultado 15-08-2020. Disponible en: https://www.canales.org.ar/situacion_sordos_informacion_familia.php.
3. Centros para el Control y la Prevención de Enfermedades. *Tipos de pérdida auditiva (sordera)*. s.f. Consultado 01-07-2020. Disponible en: <https://www.cdc.gov/ncbddd/spanish/hearingloss/types.html>.
4. CÉSAR HERNÁNDEZ, José L. Pulido. 9 de octubre de 2014. *Las tecnologías de la información en el aprendizaje de la lengua de señas*, Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Consultado 29-09-2020. Disponible en: <https://revistas.unal.edu.co/index.php/revsaludpublica/article/download/36935/62013>.
5. Clínica Barcelona. *¿Qué es la Sordera (Hipoacusia)?*. 18 de mayo del 2018. Consultado 29-08-2020. Disponible en: <https://www.clinicbarcelona.org/asistencia/enfermedades/sordera/definicion>.

6. Clínica Barcelona. *Vivir con Sordera*. 18 de mayo del 2018. Consultado 29-08-2020, Disponible en: <https://www.clinicbarcelona.org/asistencia/enfermedades/sordera/vivir-con-la-enfermedad>.

7. El granero integral. *La importancia de la lengua de señas*. s.f. Consultado 29-08-2020. Disponible en: <https://www.elgranero.com/descubrir/la-importancia-de-la-lengua-de-senas/#:~:text=Ense%C3%B1ar%2C%20conocer%20y%20entenderse%20con,con%20amigos%20o%20seres%20queridos>.

8. Fundacion iSYS. *Kitsord - Lenguaje de señas*. 15 de diciembre de 2015. Consultado 15-08-2020. Disponible en: <https://www.fundacionisys.org/es/apps2015/product/view/48/53#:~:text=%2D%20Kitsord%20es%20una%20aplicaci%C3%B3n%20para,lecciones%20empezando%20por%20el%20abecedario>.

9. GARCÍA, Lesli. GUATEMALA.COM. *La nueva Ley que reconoce la Lengua de Señas en Guatemala*, 29 enero 2020. Consultado 30-08-2020. Disponible en: <https://www.guatemala.com/noticias/sociedad/nueva-ley-que-reconoce-lengua-de-senas-guatemala-enero-2020.html#:~:text=Ley%20de%20Lengua%20de%20Se%C3%B1as%20en%20Guatemala&text=Toda%20persona%20sorda%20tendr%C3%A1%20derecho,Lengua%20de%20Se%C3%B1as%20de%20Guatemala>.

10. Hear-it. *Los niños con discapacidad auditiva tienen dificultades en la escuela.* s.f. Consultado 29-08-2020. Disponible en: <https://www.hear-it.org/es/Los-ninos-con-discapacidad-auditiva-tienen-dificultades-en-la-escuela>.
11. Human Rights Watch. *El lenguaje de señas, un componente clave para los derechos de las personas sordas.* 23 septiembre de 2018. Consultado 15-08-2020. Disponible en: <https://www.hrw.org/es/news/2018/09/23/el-lenguaje-de-senas-un-componente-clave-para-los-derechos-de-las-personas-sordas>.
12. MARTÍNEZ, Sayra, gestiopolis. *Lengua de Señas como perspectiva en la Educación Básica,* s.f. Consultado 14-08-2020. Disponible en: <https://www.gestiopolis.com/lengua-de-senas-como-perspectiva-en-la-educacion-basica/>.
13. MAZARIEGOS, Nestor. CONADI. *Comunidad de personas sordas busca que la lengua de señas sea reconocida como un idioma más de Guatemala.* 25 septiembre 2019. Consultado 28-07-2020, Disponible en: <http://conadi.gob.gt/web/2019/09/25/la-lengua-de-senas-sea-reconocida-como-un-idioma-mas-de-guatemala/>.
14. MAZARIEGOS, Nestor. CONADI. *Promueven aprendizaje de Lengua de Señas.* 12 enero 2021. Disponible en: <https://conadi.gob.gt/web/2021/01/12/promueven-aprendizaje-de-lengua-de-senas/> . Consultado: enero 2021.

15. MIJANGOS CRUZ, Carla Carolina. *Propuesta de comunicación para la implementación de la lengua de señas guatemalteca (LENSEGUA) en la comunicación externa del Congreso de la República*. Trabajo de graduación. Facultad de ciencias de comunicación, Universidad de San Carlos de Guatemala, 2018. 19 p.
16. Organización Mundial de la Salud. *Sordera y pérdida de la audición*. 15 de marzo de 2019. Consultado 25-09-2020. Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss>.
17. PAREDES, Luisa. ELPERIÓDICO. *“App” para alfabetizar a niños con sordera*. Consultado: setiembre 2020. Disponible: <https://elperiodico.com.gt/nacionales/2019/04/23/app-para-alfabetizar-a-ninos-con-sordera/>.
18. PÉREZ DE ARADO, Belén. *¿Lengua de señas?* 2011. Consultado 31-09-2020. Disponible en: <https://cultura-sorda.org/lengua-de-senas/>.
19. PÉREZ ESTRADA, Luis Alberto. *Curso Básico de Lenguaje de Señas Guatemalteco Asistido por Computadora*. Trabajo de graduación. Facultad de Ingeniería, Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, Universidad de San Carlos de Guatemala, 2006. 13 p.
20. Puntodis. *Discapacidad Auditiva*. s.f. Consultado 29-08-2020. Disponible en: https://puntodis.com/featured_item/discapacidad-auditiva/.

21. Redacción Educativa. Hago mi tarea. *Inclusión escolar: Lenguaje de señas en Guatemala*. s.f. Consultado 13-08-2020. Disponible en: <https://www.hagomitarea.com/recursos/docentes/inclusion-escolar-lenguaje-de-senas-en-guatemala/>.
22. Revista de mis antojos. *LA LENGUA DE SEÑAS: ORIGEN Y CARACTERÍSTICAS BÁSICAS*. 26 de marzo de 2015. Consultado 29-09-2020. Disponible en: <http://revistademisantojos.blogspot.com/2015/03/la-lengua-de-senas-origen-y.html>.
23. ROACH, John, News Center Corp. Tecnología IA ayuda a estudiantes con sordera a aprender. Disponible en: <https://news.microsoft.com/es-xl/features/tecnologia-ia-ayuda-a-estudiantes-con-sordera-a-aprender/>. Consultado en septiembre 2020.
24. RODRÍGUEZ, Patrick Willeth, Multemas. *Las tecnologías para las personas con discapacidad auditiva*. 18 septiembre 2010. Consultado 20-08-2020. Disponible en: <https://multemas.blogspot.com/2010/09/las-tecnologias-para-las-personas-con.html>.
25. SAMAYOA URIZAR, Rubí María. "CULTURA Y PSICOLOGÍA DE UNA COMUNIDAD SORDA". p.17. Disponible en: <http://www.repositorio.usac.edu.gt/6547/1/T%2013%20%282972%29.pdf>. Consultado: septiembre 2020.

26. SÁNCHEZ, Paulina. *harmonia.la*. *3 razones para aprender lenguaje de señas*. 14 de agosto de 2017. Consultado 29-08-2020. Disponible en: https://harmonia.la/mi-gente/paternidad-y-ninos/3_razones_para_aprender_el_lenguaje_de_senas.

27. TAVITAS, Deny. *Mis razones del porqué la lengua de señas debe ser un derecho*. 2015. Consultado 30-09-2020. Disponible en: <https://denissetavitas.wordpress.com/2015/09/26/mis-razones-del-porque-la-lengua-de-senas-debe-ser-un-derecho/>.

28. VERGARA, Alida. *Fácil Cloud*. *5 ventajas de usar Node.js*. 2015. Consultado 10-08-2020. Disponible en: <https://www.facilcloud.com/noticias/5-ventajas-de-usar-node-js/>.

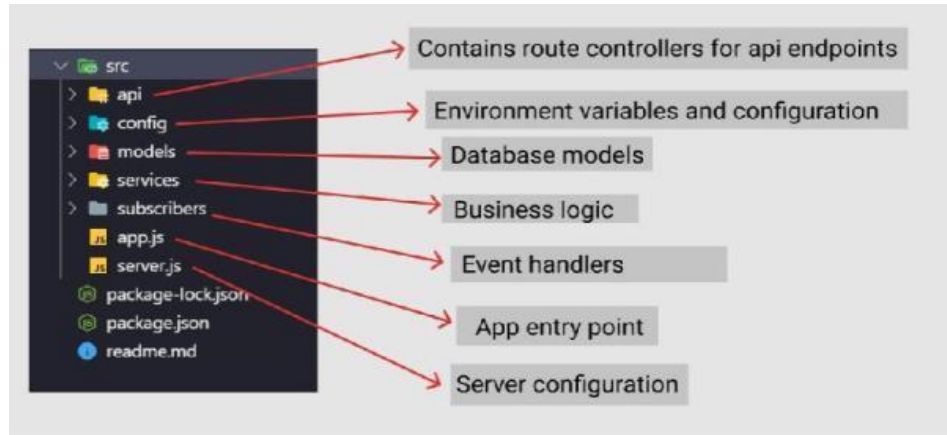
APÉNDICES

Listado de palabras/frases incluidas en el diccionario, de manera inicial:

- Bien
- Buenos días
- Buenas noches
- ¿Cómo estás?
- Computadora portátil
- Excelente
- Hola
- Mucho gusto
- Permiso
- Regular

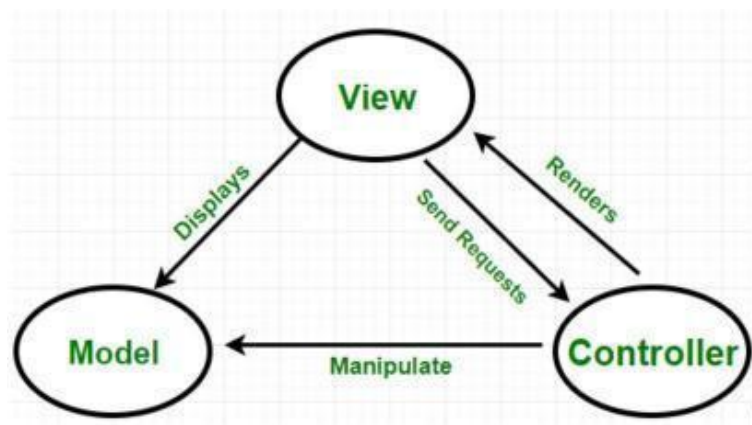
ANEXOS

Anexo 1. Estructura estándar de proyecto



Fuente: How to Architect a Node.Js Project from Ground Up? 2019. <https://soshace.com/how-to-architect-a-node-js-project-from-ground-up/>.

Anexo 2. Diagrama Modelo vista controlador



Fuente: How to Architect a Node.Js Project from Ground Up? 2019. <https://soshace.com/how-to-architect-a-node-js-project-from-ground-up/>.

